Automatic Generation of Lexical Recognition Tests using Natural Language Processing

Von der Fakultät für Ingenieurwissenschaften,
Abteilung Informatik und Angewandte Kognitionswissenschaft
der Universität Duisburg-Essen
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften (Dr.- Ing.)

genehmigte Dissertation

von

Osama Amin Hamed, M.Sc.
aus
Anabta, Palästina

1. Gutachter: Prof. Dr.-Ing. Torsten Zesch
2. Gutachter: Prof. Dr. Karim Bouzoubaa

Tag der mündlichen Prüfung: 27. Juni 2019

Abstract

Lexical recognition tests (LRTs) are widely used to assess the vocabulary size of language learners based on word recognition. In such tests, the learners need to differentiate between words and crafted nonwords that look much like real words. A review of the literature showed that: (i) LRTs are generally human-crafted, which is a very time consuming process and (ii) compared to English and other European languages, Arabic is under-resourced and has received little attention in modern language learning research. In this context, Natural Language Processing (NLP) has been successfully used for a number of tasks related to language learning research. In this thesis, we shed light on the utilization of NLP techniques for the automatic generation of LRTs for the Arabic language in particular.

The main contribution of this thesis is the exploration of the automatic generation of quality lexical recognition tests under the following two aspects: (a) nonwords generation, and (b) test adaption to Arabic. Regarding (a), we find that character n-gram language models can be used to distinguish low from high-quality nonwords. More precisely, high-order models incorporating position-specific information work best for the automatic generation of nonwords for English LRTs. Furthermore, we investigate the validity of the automatically generated LRTs. We conduct 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. Regarding (b), we pave the road for test adaption to Arabic. We address some of the NLP challenges inherited from the Modern Standard Arabic (in particular, the Arabic script). These challenges can be further split into (i) resource creation, (ii) role of diacritical marks (diacritics are the second class of symbols in Arabic script) in designing Arabic LRTs, (iii) obtaining reliable frequency counts in Arabic, and (iv) the role of diacritics in adapting the difficulty of Arabic LRTs. Regarding (i), instead of acquiring costly corpora, we consider automatic diacritization as an alternative step towards the creation of Arabic annotated (diacritized) resources. Thus, we conduct a comparative study of available tools for the automatic diacritization of Arabic text (vowels restoration). We find that Farasa is outperforming all other tools. As a result, we utilize Farasa to create diacritized Arabic resources. Regarding (ii), we noticed that existing tests are neglecting diacritics, a very important feature of the Arabic language that ambiguates the Arabic words and causes many challenges for automatic processing. We enhanced the Arabic LRTs by adding a new parameter. We are the first who added the lexical diacritics parameter to Arabic LRTs. We find that diacritics have the potential to better control the difficulty of the tests. Regarding (iii), we find that diacritics have a significant influence on obtaining reliable frequency counts in Arabic. We also showed that a quite good approximation can be obtained by applying automatic diacritization to non-diacritized corpora. Thus, the automatic diacritization is effective for obtaining reliable frequency counts for Arabic words. Regarding (iv), we conduct a user study to compare diacritized (using the most frequent diacritized form of a word) and non-diacritized lexical recognition tests and find that they are largely comparable. Then, we conduct a large-scale user study and compare the test under three conditions: No Diacritics, Frequent Diacritics, and Infrequent-Diacritics. We find that diacritics can be used to construct more appropriate Arabic LRTs by using the less frequent diacritized form of a word.

Furthermore, we present *lugha*, a Maven-based tool that covers various Arabic text preprocessing and normalization steps. Lugha can be easily integrated into Java-based NLP pipelines. We also enrich the set of Arabic annotated resources and create some diacritized corpora for MSA.

Zusammenfassung

Lexical-Recognition-Tests (LRTs) werden verbreitet eingesetzt um die Wortschatzgröße von Sprachlernern mittel Worterkennung zu ermitteln.[1] In solchen Tests müssen Lerner zwischen Wörtern und künstlichen Nichtwörter unterscheiden, die echten Wörtern sehr ähnlich sehen. Die Literatur zeigt, dass (a) LRTs grundsätzlich manuell erstellt werden, was eine sehr zeitintensive Aufgabe ist, (b) für das Arabische, im Vergleich zu Englisch oder anderen europäischen Sprachen, nur wenige sprachliche Ressourcen zur Verfügung stehen und das Arabische in der modernen Sprachlernforschung nur wenig Aufmerksamkeit erhalten hat. Methoden aus der Verarbeitung natürlicher Sprache (NLP) wurde für verschiedene Aufgaben im Bereich des Sprachenlernen mit Erfolg eingesetzt. In dieser Arbeit beleuchten wir die Anwendbarkeit von NLP-Methoden auf die automatische Generierung von LRTs, insbesondere auch für die arabische Sprache.

Der wesentliche Beitrag der Arbeit ist die Untersuchung der automatischen Generierung von qualitativ hochwertigen LRTs unter Berücksichtigung der folgenden zwei Aspekte: (a) Generierung von Nichtwörtern und (b) die Adaption von Tests für das Arabische. Was (a) angeht, finden wir heraus, dass n-Gram-Modelle auf Buchstabenebene geeignet sind, qualitativ gute von schlechten Nichtwörtern zu unterscheiden. Genauer gesagt ist es so, dass Modelle höherer Ordnung mit positionsspezifischer Information für die automatische Generierung von Nichtwörtern für englischsprachige LRTs am besten funktionieren. Darüber hinaus untersuchen wir die Validität der automatisch generierten LRTs. Wir führen eine Nutzerstudie durch und finden heraus, dass unsere automatisch generierten Tests zu Scores führen, die mit einem manuell erstellten, etablierten LRT in hohem Maße korrelieren.

Was (b) betrifft, so ebnen wir den Weg für eine Testadaption ins Arabische. Wir gehen dabei einige der Herausforderungen an, die sich aus der Verwendung von modernem Hocharabisch (MSA) ergeben, insbesondere aus der arabischen Schrift. Diese Herausforderungen können weiter unterteilt werden in (i) Ressourcenerstellung, (ii) die Rolle der diakritischen Zeichen bei der Erstellung von LRTs (Diakritika sind die zweite Klasse von Symbolen in der arabischen Schrift), (iii) die Gewinnung von zuverlässigen Häufigkeitswerten im Arabischen, und (iv) die Rolle von diakritischen Zeichen bei der Anpassung des Schwierigkeitsgrads von arabischen LRTs.

Hinsichtlich (i) betrachten wir automatische Diakritisierung, anstelle vom Erwerb teurer Corpora, als einen alternativen Schritt auf dem Weg zur Erstellung von annotierten (diakritisierten) arabischen Ressourcen. Wir führen eine Vergleichsstudie zu verfügbaren automatischen Text-Diakritisierungswerkzeugen (Werkzeugen zur Vokalrekonstruktion) fürs Arabische durch. Wir finden, dass Farasa besser als alle andere Werkzeuge abschneidet. Daher benutzen wir Farasa um diakritisierte arabische Ressourcen zu generieren.

Hinsichtlich (ii) haben wir bemerkt, dass existierende Tests Diakritika bisher nicht benutzt haben, die ein sehr wichtiges Merkmal der arabischen Sprache sind, das arabische Wörter desambiguiert und die automatische Verarbeitung vor verschiedene Herausforderungen stellt. Wir reichern arabische LRTs um diesen neuen Parameter an. Wir sind die ersten, die lexikalische Diakritika als Parameter zu arabischen LRTs hinzugefügt haben. Wir finden heraus, dass Diakritika das Potenzial haben die Testschwierigkeit besser zu kontrollieren.

Hinsichtlich (iii) finden wir heraus, dass Diakritika einen signifikanten Einfluss auf die Er-

[1] This German transcript translated by Andrea Horbach.

hebung von zuverlässigen Worthäufigkeiten im Arabischen haben. Wir haben darüber hinaus gezeigt, dass eine relativ gute Approximierung erreicht werden kann, wenn man automatische Diakritisierung auf nicht-diakritisierte Corpora anwendet. Daher ist die automatische Diakritisierung effektiv für die Erhebung von zuverlässigen Worthäufigkeiten im Arabischen.

Hinsichtlich (iv) führen wir eine Nutzerstudie durch um diakritisierte LRTS (bei der die häufigsten diakritisierte Form benutzt wird) und nicht-diakritisierte LRTs verglichen werden und stellen fest, dass sie weitestgehend vergleichbar sind. Dann führen wir eine Nutzerstudie im großen Stil durch und vergleichen den Test in drei Bedingungen: Keine Diakritika, häufige Diakritika und seltene Diakritika. Wir finden heraus, dass Diakritika benutzt werden können um angemessenere arabische LRTs zu erstellen indem man die seltene diakritisierte Form eines Worts benutzt.

Darüber hinaus stellen wir lugha vor, ein Maven-basiertes Tool, dass verschiedene Textverarbeitungs und Normalisierungsschritte fürs Arabische bereitstellt. Lugha kann einfach in Java-basierte NLP-Pipelines integriert werden. Wir reichern außerdem die Menge an Arabischen annotierten Ressourcen weiter an und erstellen diakritisierte Corpora für MSA.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> “While without grammar very little can be conveyed, without vocabulary nothing can be conveyed."
>
> — Wilkins (1972)

Measuring language proficiency is an extremely important aspect of educational research (Izura et al. 2014). According to Robinson (2012), “language proficiency is measured in terms of receptive and expressive language skills, syntax, vocabulary, semantics, and other areas that demonstrate language abilities”. One of the critical aspects of language proficiency is vocabulary size, a variable that has been tackled by researchers interested in word recognition (Izura et al. 2014). The task of having students recognize words for vocabulary proficiency testing goes back quite a long time, cf. (Schmitt 2000). Many existing language tests focus on vocabulary size (Izura et al. 2014). A well-known format for vocabulary testing is known as the lexical recognition test (Meara and Jones 1987; Lemhöfer and Broersma 2012).

Typically, a lexical recognition test (LRT) contains a list of stimuli items consisting of words and nonwords. The task is to give a binary decision for all test items presented/shown in the list. A student is required to respond yes when the item is found in the lexicon (i.e., a word), and no otherwise – when the item is not found in the lexicon (i.e., a nonword). Figure 1.1 shows an example of one item of a lexical recognition test, where a learner has to check wether or not *platery* is a correct lexical item. Meara and Jones (1987) created the Eurocentres Vocabulary Size Test. EVST is an early example of using nonwords for testing, comprising 150 items – where two thirds are real words and one third are nonwords. Lemhöfer and Broersma (2012) created the Lexical Test for Advanced Learners of English (LexTALE), an adapted version of EVST that can be finished faster, as it only uses 60 items (40 words and 20 nonwords). It has been shown that such a small number of items is sufficient to consistently measure the vocabulary size (Huibregtse et al. 2002). Nonwords in a lexical recognition test are typically used as distractors. Thus, they should be close to existing words. 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). For many under-resourced languages, like Arabic, a lot of challenges still remain.

In the past, LexTALE has been manually generated – see Section 5.1. As a consequence, it is a very time consuming process and non-challenging nonwords. However, for the repetitive testing as used in formative assessment (Wang, 2007), it is very desirable to automate this process. LRT’s test stimuli (words and nonwords) can be created automatically with the help of Natural Language Processing (NLP) algorithms. A process that can be further divided into two NLP tasks: (i) word selection from a corpus, and (ii) nonword generation.

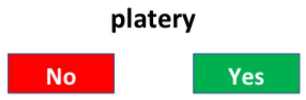


Figure 1.1: Example of an item from a lexical recognition test.

NLP techniques have been successfully used for a number of tasks related to language learning research, such as predicting and manipulating the difficulty of C-tests, X-tests and cloze tests in different languages (Beinborn 2016). In this thesis, we shed light on the utilization of NLP techniques to support second-language assessments. More precisely, we want to predict the level of a second-language (L2) learner by estimating her/his vocabulary size. We have a look at existing LRTs in Section 3.1.

As a general research objective for this thesis, we are focusing on language proficiency testing. We keep our exploration limited to vocabulary size tests based on word recognition as they can be finished fast. We are mainly targeting the lexical recognition tests, but not C-tests (Klein-Braley and Raatz 1982) nor X-tests or cloze tests (Sachs et al. 1997). Our ultimate goal is to create lexical recognition tests automatically in order to enable repetitive automated testing in different languages. As a first step towards achieving this goal, we start by tackling the English language. Then, we try to generalize and extend our approach to one of under-resourced languages. In particular, Arabic, which has received little attention in the recent language learning research, and modern computational linguistics[1]. According to Habash et al. (2016), “the automatic processing of the Arabic language is challenging for a number of reasons”. Many Arabic NLP challenges arise from the problematic Arabic script due to specific properties of Arabic (Farghaly and Shaalan 2009). The two inter-related reasons are: the optionality of its diacritical marks (the second class of symbols in Arabic script (Diab et al. 2007a)) on the one side, and Arabic’s complex morphology on the other side (Habash 2010). In nowadays’ writing, using Modern Standard Arabic (MSA), the Arabic diacritical marks (diacritics) are typically unwritten (Habash 2010). The absence of diacritics acts as a major source of complexity for Arabic NLP systems because they cannot easily determine the meaning of a word (Chennoufi and Mazroui 2017; Said et al. 2013). In analogy to English, the situation would be similar to presenting someone the string *str* and expecting her/him to be able to predict whether the intended English word is *star*, *stir*, *suitor*, *sitar*, or *store* (Saigh and Schmitt 2012). As a result, Arabic has minimal annotated resources. We investigate how to apply our approach for test generation to Arabic. We are taking a closer look on the design process of Arabic tests by addressing some of the NLP challenges and especially resource creation, role of diacritics, reliable frequency counts, and adapting the difficulty of the test. Benefiting from NLP techniques to generate lexical recognition tests automatically falls under the utilization Computer-Assisted Language Learning (CALL). Next, we provide a brief glimpse on CALL.

Computer-Assisted Language Learning Computers are becoming very popular than any time before. They have invaded all parts of our daily lives and permeated many areas of education. Many language learning portals and mobile apps have been developed around the world in the last 10 years for different languages and some of them are astonishing. As a conse-

[1] https://nlp.stanford.edu/projects/arabic.shtml

quence, they become very popular and have reached millions of users worldwide (Beinborn 2016). CALL focuses on how to use computers most effectively to support language learning. Levy (1997) defined CALL as “the search for and study of applications of the computer in language teaching and learning". Ken (2003) offers a definition of CALL that accommodates its changing nature “CALL is any process in which a learner uses a computer or hand-held devices to improve her or his language”[2].

1.1 Main Contributions

The two main contributions of this thesis are: (i) nonwords generation, and (ii) test adaptation to Arabic. The second contribution can be further split into: resource creation, role of diacritics, reliable frequency counts, and adapting test difficulty. In addition, this work is accompanied by two additional contributions in terms of tools and applications. All of these contributions are summarized as follows:

Nonwords Generation To what extent do high-order character n-gram language models have reliability in generating nonwords stimuli for lexical recognition tests? We propose a new approach to generate nonwords automatically. This approach requires a corpus and a set of gold standard nonwords. We use the average precision from information retrieval to rank the generated nonwords. Instead of using human-crafted nonwords, we are enabling the process of test generation automatically.

Resource Creation To what extent do the Arabic diacritization (automatic restoration of diacritical marks) tools are helpful for creating Arabic annotated (diacritized) resources? We investigate the performance of available off-the-shelf Arabic diacritization tools. We are using the best performing tool to diacritize (restore the vowels automatically) for some of the freely available Arabic corpora. This way, we are producing diacritized Arabic corpora and make them available for research.

Role of Diacritics Can we construct Arabic LRTs with diacritics? We investigate the role that diacritics play in designing Arabic LRTs. We find that Arabic LRTs can be constructed with diacritics. This way are extending Arabic LRTs and we are the first who add the lexical diacritics as a parameter to Arabic LRTs. By adding this new parameter, it would be possible to ask very specific questions in a LRT, do the respondents know this specific word with diacritics?

Reliable Frequency Counts To what extent do the frequency counts for Arabic words have reliability without diacritics? We cannot obtain reliable frequency counts for Arabic words in isolation from diacritics because they are misleading. We address the inherited Arabic NLP challenges (e.g. segmentation, discarding extra clitics, etc) and propose reliable solutions. We introduce an NLP pipeline and find that a quite good approximation for frequency counts can be obtained by applying automatic diacritization to non-diacritized corpora.

[2] `https://web.stanford.edu/~efs/callcc/callcc-intro.pdf`

Adapting Test Difficulty To what extent do the Arabic diacritics are helpful for adapting the difficulty of Arabic LRTs? We propose an NLP pipeline to rank the Arabic words based on their occurrences in a corpus. This pipeline enable us to have different views of the same Arabic words based on the attached diacritics. For example, we have been able to identify the most and least frequent diacritized form of a word. As a result, we found that the diacritics can be used to adapt the difficulty (it can be increased) of Arabic LRTs.

Contributed Tools We provide a new resource and extension to existing Arabic NLP tools and make them available for research purposes.

Lugha: It is a set of Arabic NLP Java-based APIs[3] which are useful for Arabic text preprocessing and normalization. This includes Arabic bi-directional encoding (e.g. transliteration from Arabic to Buckwalter encoding (Buckwalter 2004) and vice versa), diacritics removal, punctuations removal, etc.

Contributed Applications We introduce a browser-safe and mobile-friendly web-based LRT application for both English and Arabic. The web interface is implemented using PHP, and the responses are stored in a MySQL database. The interested researchers are able to repeat the described user studies or conduct their own studies, simply by downloading our LRT application[4].

1.2 Organization of Thesis

This thesis falls into 10 chapters that describe the utilization of NLP techniques to support language vocabulary testing, more precisely the automatic generation of LRTs. Apart from the *Introduction* in Chapter 1 and the *Conclusion* in Chapter 10, the remaining chapters can be logically divided into four parts as depicted in Figure 1.2.

The first part, comprising Chapters 2 through 4, provides a theoretical background on vocabulary assessment and analysis of vocabulary knowledge, assessment formats with focus on lexical recognition tests as well as Arabic NLP. The second part contains Chapter 5 and focuses on the automatic generation of LRTs for English, mainly nonwords generation. The third part, comprising Chapters 6 through 7, includes some comparative and benchmarking studies to facilitate resource creation for Arabic. Arabic is an under-resourced language that has minimal annotated resources, compared to that we have for English and some other European languages. Arabic automatic text processing entails many NLP challenges due to its optional diacritical marks. The fourth part, comprising Chapters 8 through 9, answers the question how easy is it to transfer our knowledge from English LRTs to Arabic LRTs. Finally, we outline the results and draw a conclusion in Chapter 10.

Chapter 2 In this chapter, we provide background material and a review of literature on the different aspects of vocabulary knowledge and vocabulary assessment formats.

[3] https://github.com/zesch/lugha-tk

[4] https://github.com/ohamed/ar-lrts

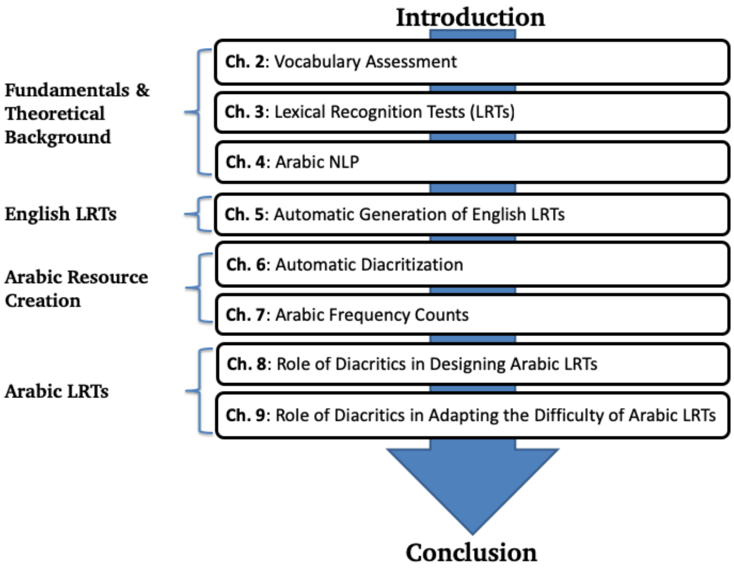

Introduction
Fundamentals & Theoretical Background
Ch. 2: Vocabulary Assessment
Ch. 3: Lexical Recognition Tests (LRTs)
Ch. 4: Arabic NLP
English LRTs
Ch. 5: Automatic Generation of English LRTs
Arabic Resource Creation
Ch. 6: Automatic Diacritization
Ch. 7: Arabic Frequency Counts
Arabic LRTs
Ch. 8: Role of Diacritics in Designing Arabic LRTs
Ch. 9: Role of Diacritics in Adapting the Difficulty of Arabic LRTs
Conclusion

Figure 1.2: Organization of this Thesis.

Chapter 3 In this chapter, we shed light on the different aspects of lexical recognition tests, such as presentation format and scoring criteria.

Chapter 4 In this chapter, we provide an introduction, and a review of literature on the characteristics of the Arabic language and its natural language processing.

Chapter 5 In this chapter, we tackle the automatic generation of nonwords for English lexical recognition tests. The generated nonwords are composed from character sequences based on position-specific character language models. We evaluated our generated nonwords in a user study. The evaluation shows that we are able to automatically generate word-like nonwords, which enables repeated automated testing.

Chapter 6 Arabic diacritized resources can be built with the help of automatic diacritization tools. In this chapter, we fill the gap of minimal Arabic annotated resources by addressing the task of diacritical marks restoration (diacritization). We approach this in a comparative study by benchmarking the available tools for Arabic automatic diacritization. This way, we have solved one of the main challenges encountered during the development of LRTs for Arabic.

Chapter 7 Being done with benchmarking, we want to obtain a reliable frequency counts for Arabic words based on their appearance in a diacritized corpus. In this chapter, we explore the effects that diacritics play in obtaining a reliable frequency counts for Arabic words. To achieve this, we start by handling some preprocessing steps, tokenization and segmentation due to rich morphological features of Arabic, such as affixes and clitics within Arabic words.

Chapter 8 In this chapter, we investigate the role that diacritics play in creating an Arabic lexical recognition test. We conducted a user study that includes heritage learners (Ricks 2015) of Arabic from several German schools. The pupils were asked to respond to diacritized and non-diacritized tests randomly. Arabic LRTs can be constructed with diacritics, which act as a new parameter that has the potential for adapting the test difficulty.

Chapter 9 In this chapter, we address the automatic creation of a difficulty-controlled Arabic LRTs. We use a non-common or a less frequent diacritized form of a word. We evaluate the generated Arabic LRT in a user study to investigate if the automatically generated test is more challenging than the non-diacritized test. We find that the difficulty can be adapted, the test becomes more difficult by using the least frequent diacritized form. As a show case, we support this work with a proof-of-concept, i.e., a web-based interface for lexical recognition tests.

1.3 Publication Record

Parts of this thesis have been previously published at peer-reviewed conferences and internationally recognized journals in the fields of NLP and computational linguistics.

- HAMED, Osama ; ZESCH, Torsten: The Automatic Generation of Nonwords for Lexical Recognition Tests. In: Language and Technology Conference Springer, 2015, pages 321– 331.
- HAMED, Osama ; ZESCH, Torsten: 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. Poznan', Poland, 2015, 473–477.
- HAMED, Osama ; ZESCH, Torsten: A Survey and Comparative Study of Arabic Diacritization Tools. In: JLCL: Special Issue - NLP for Perso-Arabic Alphabets. 32 (2017), Nr. 1, pages 27–47.
- HAMED, Osama ; ZESCH, Torsten: The Role of Diacritics in Designing Lexical Recognition Tests for Arabic. In: 3rd International Conference on Arabic Computational Linguistics (ACLing 2017). Dubai, UAE : ELSEVIER, 2017.
- HAMED, Osama ; ZESCH, Torsten: Exploring the Effects of Diacritization on Arabic Frequency Counts. In: Proceeding of the 2nd International Conference on Natural Language and Speech Processing (ICNLSP 2018). Algiers, Algeria, 2018.
- HAMED, Osama ; ZESCH, Torsten: The Role of Diacritics in Adapting the Difficulty of Arabic Lexical Recognition Tests. In: NEALT Proceedings Series Vol. 36 (2018), pages 23–31.

Chapter 2

Vocabulary Assessment

> “Learning another language is not only learning different words for the same things, but learning another way to think about things."
>
> — Flora Lewis

Meara (1995) reported that second language research was mainly focused on grammars till the 1980s. Linguistic researchers have been targeting vocabulary for more than 40 years (Pignot-Shahov 2012; Alqahtani et al. 2015). This chapter provides background material and a review of literature on the different aspects of vocabulary knowledge and assessment formats. For the scope of this thesis, we tend to make our exploration more focused on vocabulary size as: (i) it can be easily measured, and (ii) valuable information can be provided to language teachers, among others, this includes periodic learning progress, the optimal level to begin learning a language course with etc (Izura et al. 2014). According to Milton (2006) “Vocabulary size ought to be a useful tool as a placement or levels indicator”.

This chapter is intended to be as a preliminary chapter before introducing lexical recognition tests in the next chapter. The contents of this chapter are organized as follows: Section 2.1 provides an overview of vocabulary, introduces the word construct and it’s related terms, vocabulary acquisition as well as second-language acquisition. In section 2.2, we provide an overview of language proficiency as described by the European Union standards. In section 2.3, we provide some analysis on the different aspects of vocabulary knowledge and the descriptive frameworks. This is followed by section 2.4 describing some vocabulary assessments formats. Finally, we summarize the chapter in section 2.5.

2.1 Vocabulary

2.1.1 Defining Vocabulary

The *Cambridge Advanced Learner’s Dictionary*[1] lists two definitions for vocabulary: (i) “all the words known and used by a particular person", and (ii) “all the words which exist in a particular language or subject”. Lessard-Clouston (2013) provides an extended definition, vocabulary is defined as “the words of a language, including single items and phrases or chunks of several words which covey a particular meaning”. The term *word* is included in all definitions.

[1] `https://dictionary.cambridge.org/dictionary/english/vocabulary`

As first step towards understanding vocabulary, we must therefore understand the word construct and what constitutes a word. Nation (2001) lists four terms that are central for defining the word construct in applied linguistics. These are tokens, types, lemmas, and word families. The terms are "ordered from the most specific to the most general" (Šišková 2012).

In the following, we provide an answer to the question: "What is a word?" based on the classification introduced by Nation (2001). Each of the following terms can be used as a measure to describe the number of words known by a particular person.

Tokens and Types For a given text or corpus (spoken or written), the token refers to raw space-delimited strings (words), whereas type refers to unique space-delimited strings (words). It important to notice that the terms "running words" and "tokens" are used interchangeably (Nation 2001).

For example, the sentence "a good football player is a player who scores fifty goals or more in the season". If we split the sentence based on space delimiter, we get the list of words {a, good, football, player, is, a, player, who, scores, fifty, goals, or, more, in, the, season}. Thus, the sentence has sixteeen tokens, but only fourteen types because "a" and "player" are repeated.

In this context, we remind that we are addressing the lexical recognition tests. Therefore, we refer to four important statistical terms to address the lexical richness aspects for corpus data that can be computed using tokens and types. The aspects are (i) lexical richness, (ii) lexical diversity, (iii) lexical sophistication, and (iv) lexical density (Laufer and Nation 1995). The terms can be further used to address language learners' progress (Daller et al. 2003) (as cited in Gregori-Signes and Clavel-Arroitia 2015).

As defined by Šišková (2012), lexical richness is concerned with measuring the number of unique words in a corpus (the types). Matthews and Wijeyewardene (2018) mentioned type-to-token ratio (TTR) as the main lexical diversity index. They defined TTR as "the total number of types divided by the total number of tokens". Lexical sophistication refers to the number of advanced words (low-frequency words) in a corpus (Šišková 2012). According to Malvern and Richards (2012), lexical sophistication is mainly characterized by a greater total number of types, in particular, the low-frequency words and a higher TTR. Lexical density refers to the number of of lexical (content) words in the text (Šišková 2012).

Lemmas The lemma (citation form (Habash 2010)) is the uninflected dictionary form of a word (Schmitt 2000). According to Nation (2001), the lemma "consists of a headword (dictionary form) and some of its inflected and reduced forms". For example, teach, teaches, taught and teaching are forms of the same lexeme, with run as the lemma. As it was stated by the psycholinguistic researchers, the lemma gets its importance because of the brain memorize the base form of a word (Pignot-Shahov 2012).

We can use types and lemmas to measure the number of units in a corpus (e.g. text length). However, lemmas work better because this number can be decreased significantly. Let's clarify this using an example with the help of Brown Corpus (Francis and Kuçera 1982), which contains 61,805 types. This number can be further decreased by almost 40% to 37,617 lemmas (Nation 2013).

Word families According to Nation (2001), a word family "consists of a headword (dictionary form), its inflected forms and its closely related derived forms". Pignot-Shahov (2012) wrote that "word families are similar to lemmas but include all words related to the headword regardless of their word class". Thus, word family is typically broader than lemma and includes different parts of speech (PoS) (Šišková 2012). For example, teach, teaches, taught, teacher, teaching and teachable belong to the same word family. This is an extended set that also includes the "teacher" as a noun and "teachable" as adjective.

Schmitt (2010) wrote that lemmas have been successfully used as unit of abstraction because they are quite straightforward and better than word families, types or tokens. For the languages with minimal inflection, like English, types can provide a good level of abstraction. On the other side, for French and Czech, which are highly inflected, lemmas were found to work better (Šišková 2012).

For the reminder of this thesis, we adopt types as counting unit for English and lemmas as counting unit for Arabic. In our adoption, we are inspired by Lemhöfer and Broersma (2012) and Ricks (2015) respectively. Ricks (2015) used the lemma for Arabic and wrote that "most vocabulary assessments and corpus-based frequency lists rely on the lemma because it is the default unit of analysis for lexical statistics".

2.1.2 Vocabulary Acquisition

Before we talk about the phenomena of vocabulary acquisition, we found it is important to shed light on the types and importance of vocabulary.

Types of Vocabulary The researchers in language-learning divided vocabulary into two types: active and passive vocabulary (Harmer 2001) (as cited in Hernawati 2015). Hernawati (2015) defined the active vocabulary as "the set of words that the students have been taught or learnt and which they are expected to be able to use". On the other side, he defined passive vocabulary as "the set of words that the students will recognize when they meet them but which they will probably not able to produce". For the scope of this thesis, we are tackling lexical recognition tests (LRTs) where the students are required to claim knowing a word, even if they do not know its exact meaning (Lemhöfer and Broersma 2012). Therefore, we are interested in receptive vocabulary.

Importance of Vocabulary Vocabulary serves as the foundation for mastering the different language skills (Maskor et al. 2016b). Wilkins (1972) stated that "while without grammar very little can be conveyed, without vocabulary nothing can be conveyed". Vocabulary knowledge plays a vital role in teaching English and other languages (Al-Fak et al. 2015). Lessard-Clouston (2013) generalized this to include all human languages. He states that "even without grammar but with some useful words and expressions of a language, we can often manage to communicate". This means that a conversation between a teacher and one of her/his students cannot take place fluently without the right amount of vocabulary (quantitative). Furthermore, there is a complementary relationship between vocabulary knowledge and language practice. Knowledge of vocabulary enables learners to practice languages. Consequently, language practice leads to an increased amount of vocabulary knowledge (Nation 2001) (as cited in Alqahtani et al. 2015).

Vocabulary acquisition is a benchmark for language proficiency, as it improves the skills for writing, reading, listening and speaking (Maskor et al. 2016a). Typically, acquiring an extensive vocabulary is one of the largest obstacles in learning a foreign or a second language (L2) (Baharudin et al. 2014). Undoubtedly, vocabulary acquisition has been quickly identified as: (i) an important prerequisite for successful test performance, and (ii) an interesting topic for researchers and designers of language proficiency tests (Ricks 2015).

We remind that the terms language acquisition and language learning are used interchangeably (Skoglund 2006). Language acquisition, including Second-Language Acquisition (SLA), returns back quite a very long time, and it might be as old as a human being (Ricks 2015). It is important to notice that the term second language (L2) is typically used to refers to any language learned by an individual in addition to her/his first language (L1) (Nation 2001). As defined in Wikipedia, "SLA is the process by which people learn a second or a foreign language". SLA and vocabulary acquisition are highly related because the later can be considered as a critical tool that boosts SLA (Alqahtani et al. 2015). Therefore, a limited vocabulary in a second language will impede successful communication. The concept second-language acquisition is only used conventionally. In reality, it can be also used to incorporate the learning of third, fourth, or subsequent languages.[2] Next, we talk about language proficiency testing with focus on the European standards.

2.2 Language Proficiency

As per Wikipedia definition, language proficiency or linguistic proficiency is "the ability of an individual to speak or perform in an acquired language". Measuring language proficiency is a very important aspect of language learning research (Izura et al. 2014).

2.2.1 Language Proficiency Testing

Foreign language proficiency testing industry goes back to 1877. It has been identified as a movement that started in the United States and spread worldwide after that. The idea of creating the"Test of English as a Foreign Language (TOEFL)", for example, was emerged as a case study during the World War II time (Spolsky 1995). Later on, the TOEFL becomes as a perquisite test for foreign students who are looking forward to study at the American universities. Nowadays, the test is accredited world-wide.

According to Meara and Buxton (1987), "without some knowledge of what the words appearing in a test mean, it is difficult to perform at all". All language proficiency tests are in a sense tests of vocabulary. Language proficiency for individuals is highly related to quantitative vocabulary knowledge (vocabulary size). Thus, the more words you know, the more you will be able to understand and communicate (Milton 2013). Nation (2006) tried to answer the question: How much vocabulary a person need to know to master reading (e.g. read a newspaper or novel) and listening (e.g. watch a film, or take part in a conversation)?. He also reported the number of vocabulary acquired by a well-educated native speakers as 20,000 word-families (without proper names and transparently derived forms). As proposed by Nation (2006), language learners are highly recommended to set a learning goal for a better performance in the

[2] https://en.wikipedia.org/wiki/Second-language_acquisition

Skill	Estimated Size	Reference
Reading	8k - 9k word families	Nation (2006)
Listening	6k - 7k word families	Nation (2006)
Native speaker	20k word families	Zechmeister et al. (1995), Goulden et al. (1990)

Table 2.1: Vocabulary size needed to master different language skills (Nation 2006).

four language skill. Table 2.1 summarizes the different sizes as suggested by Nation [3].

The four language skills: reading, listening, speaking and writing are typically tested in a language proficiency test, such as TOEFL or IELTS. However, the early vocabulary research projects have focused on the distinction between passive and receptive (e.g. recognition or reading) vocabulary knowledge (Webb 2008). The utilization of vocabulary-focused assessment formats has not been limited to SLA or vocabulary researchers. It goes behind that and spreads to include common and high-stakes L1 tests such as the SAT[4] test (Ricks 2015).

2.2.2 Common European Framework of Reference for Languages

The Common European Framework of Reference for Languages Learning, Teaching, and Assessment (CEFR) was published by the European Union in 2001. Typically, validated lexical recognition tests have a CEFR equivalence (e.g. LexTALE). The CEFR is intended to provide standards for the comparability of language skills (Little 2011). According to the creator of this scale (the Council of Europe), the CEFR provides a “common basis” / “coherent and comprehensive basis” that describes what knowledge and skills learners have to develop inside of a cultural context on the one side, and a measure of proficiency to classify the learners of new languages on the other side [5]. The CEFR is available for 40 languages, including English, German and Arabic etc. The CEFR has three broad bands: A, B and C, each of those bands is divided into two, giving us six main levels: A1, A2, B1, B2, C1 and C2. As shown in Figure 2.1, the A level (A1 - A2) indicates basic knowledge whereas C level (C1 - C2) corresponds to language proficiency. In other words, learning a new language begins with the starter A1 level and ends with the C2 mastery level.

The proficiency levels are expressed as a combination of the four basic skills: speaking, reading, writing and listening. Listening and reading involve language understanding, while writing and speaking involve language generation. Therefore listening and reading are considered as receptive language activities, whereas writing and speaking are productive language activities (Schmitt 2014; Maskor et al. 2016a). Next we provide more details on vocabulary analysis and word knowledge models.

2.3 Vocabulary Analysis

This section covers the important aspects of vocabulary knowledge, breadth and depth of vocabulary as well as corpus-based frequency.

[3] `https://www.victoria.ac.nz/lals/about/staff/paul-nation`

[4] `https://en.wikipedia.org/wiki/SAT`, accessed: Jun 18, 2018

[5] Council of Europe: `https://rm.coe.int/16802fc0b1`

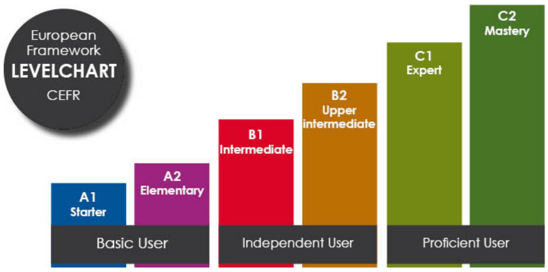


Figure 2.1: The CEFR levels for language testings (source: `http://www.euenglish.hu`).

2.3.1 Words and Word Knowledge

Typically, each vocabulary test is intended to measure one, or more aspects of vocabulary knowledge (Webb and Sasao 2013). Before we talk about vocabulary assessment formats, it is very important to answer the question: "What does it mean to know a word?" We want to follow the descriptive framework of word knowledge as described in the earlier work by Nation (2001). It is important to notice that this work was first introduced by 2001 and remains unchanged in the recent versions of the book by Nation (2013).

The aspects of knowing a word can be defined in different ways. We can think of this as a high-level and detailed classifications. We can claim that someone is knowing a word if and only if, she/he has: (i) receptive knowledge by being able to recognize it, and (ii) productive knowledge by being able use it correctly (Pignot-Shahov 2012). This is just a high-level classification.

An earlier detailed classification has been provided by Nation (2001), he mentioned three significant aspects (categories) that need to be tackled by language teachers. These are form, meaning, and use. Nation's framework "what is involved in knowing a word" is presented in Table 2.2. As described by Šišková (2012), this is a frequently cited framework. In this framework, word knowledge is divided into three categories. Each category can split further into three components. Each component is depicted as having receptive (R) and productive (P) facets. For example, the lexical recognition tests (see next chapter) falls under the "Form" category and classified as test of written/receptive vocabulary. The corresponding raw is highlighted with green in Table 2.2. Next, we present the breadth and depth of vocabulary, and the frequency of words.

2.3.2 Breadth vs. Depth of Vocabulary

Many descriptive frameworks have been developed so far to predict language proficiency. As described by Schmitt (2014), a well-known framework is based on the widely used distinction between size (breadth) and depth (quality) of vocabulary knowledge. A large body of research is referencing the work by Nation (2001) to differentiate between both constructs. However, as cited in Schmitt 2014, Anderson and Freebody (1981) preceded Nation and described vocabulary size and vocabulary depth. He wrote that vocabulary size concerns the number of known words, whereas vocabulary depth concerns the degree or level of knowledge. For example, the depth vocabulary knowledge for the word *bank* includes knowing its other variants or meanings, such as the "commercial institution", "students desk", "the land alongside to a river" or

Category	Component	#	Receptive/Productive
	spoken	1	R: What does the word sound like?
		2	P: How is the word pronounced?
Form	written	3	R: What does the word look like?
		4	P: How is the word written and spelled?
	word parts	5	R: What parts are recognizable in the word?
		6	P: What word parts are needed to express the meaning?
	form and meaning	7	R: What meaning does this word form signal?
		8	P: What word form can be used to express this meaning?
Meaning	concept and referents	9	R: What is included in the concept?
		10	P: What items can the concept refer to?
	associations	11	R: What other words does this make us think of?
		12	P: What other words could we use instead of this one?
	grammatical functions	13	R: In what patterns does the word occur?
		14	P: In what patterns must we use this word?
Use	collocations	15	R: What words or types of words occur with this one?
		16	P: What words or types of words must we use with this one?
	constraints on use	17	R: Where, when, and how often would we expect to meet this word?
		18	P: Where, when, and how often can we use this word?

Table 2.2: Description of "what is involved in knowing a word" (Nation 2001).

bank of something such as"blood bank". It also include collocations such as"central bank", "investment bank" etc.[6]

Vocabulary size is a quantitative aspect and refers to the number of words known by a particular person and at a certain level of proficiency (Nation 2001). Generally, there is no minimum vocabulary size because our vocabulary size grows with age (Skoglund 2006). Vocabulary size has been considered as one of the main factors that help in determining how students learn second language vocabulary (Baharudin and Ismail 2014). Milton (2006) classified vocabulary size as a good indicator of overall language knowledge and ability. However, it is not meant to replace other forms of proficiency testing.

Gyllstad (2013) describes the critical views of breadth and depth to differentiate between the two constructs. As cited in Gyllstad 2013, Meara and Wolter (2004) have argued that "vocabulary size is a measure of a learner's entire vocabulary, since scores on a particular number of words are extrapolated to give an indication of an overall size score, given that the selection of test items is valid. As a consequence, vocabulary size is not a characteristic of individual words. On the other side, vocabulary depth, is typically seen as a characteristic of individual words, where extrapolation is not possible, or at least very difficult".

Table 2.2 depicts Nation's framework (Nation 2001, 2013) that contains a total of 18 receptive and productive aspects. Basically, we have to notice that the table has three main columns (we added the # column for the ease of readability) and three main multiple-row rows. Based on this framework, the components called 'spoken' and 'written' under the multiple-row heading or 'Form' category, combined with 'form and meaning' under the 'Meaning' multiple-row heading – i.e. the rows (1-4 and 7-8) are classified as size (breadth) aspects. On the other side, the remaining rows are usually classified as depth aspects. Thus, the knowledge of word parts (5-6), word associations (11-12), grammatical functions (13-14) as well as collocations (15-16)

[6] Examples from Cambridge English Dictionary: `https://dictionary.cambridge.org/dictionary/english/vocabulary`

fall under the depth of word knowledge (Gyllstad 2013). For further explanations on Nation's framework, we recommend the readers with the detailed review provided by Barouni Ebrahimi (2017).

To fit with the scope of this thesis, our discussion will be more focused on vocabulary size tests. Many existing language tests focus on this variable, for example, the Vocabulary Levels Test (Nation 1983), Vocabulary Size Test, Lexical Recognition Tests (Meara and Jones 1987) and the LexTALE-like tests (Lemhöfer and Broersma 2012; Brysbaert 2013; Izura et al. 2014). We provide more details about LexTALE and other similar tests in the next chapter.

2.3.3 Frequency

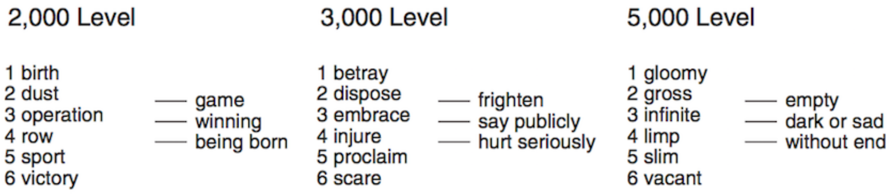
Kartal and Sarigul (2017) presented a detailed survey and a review of the literature on the studies that considered the relationship between word frequency and language acquisition. Typically, the more frequent words are easier than the less frequent ones. We shed light on word frequency because (i) it plays a vital role that affects the acquisition of language vocabulary (Kartal and Sarigul 2017), and (ii) the language tests are typically based on words that belong to different frequency bands, such as the Vocabulary Levels Test (Nation 1983).

Extracting words using frequency-based approach goes back quite a long time. Palmer et al. (1968) wrote that the more frequently used words will be the more easily learned. Similarly, according to Ellis (2002) "Humans are sensitive to the frequencies of events in their experience". Thus, the words that are frequent in a language should be learned first because they are heard or read more often (Pignot-Shahov 2012). The following classification is based on the work by Pignot-Shahov (2012):

High-frequency words Typically, those words are shorter in length and comprised from a few syllables. They are characterized by their usability in a variety of contexts because they are not semantically restricted like lower frequency words. A very distinguishing feature of high-frequency words they have no connotation (positive or negative) or collocation parts attached to them. For instance, the set of words "the, we, they, girl, boy, it, how, and, ... because" are some examples of high-frequency words (Skoglund 2006).

Low-frequency words Here, we refer to the set of words that do not repeat often in a given human language due to their minimal usage. For instance, the words "Ohio, approximately, eponymous, ... scalpel" are some examples of low-frequency words (Skoglund 2006).

Word frequency has been also considered as the main criterion that helps in designing vocabulary tests (Izura et al. 2014; Brysbaert 2013) as it controls the degree of test items difficulty. In this thesis, we are tackling the development of lexical recognition tests (LRTs) for English and Arabic. Diependaele et al. (2013) reported that word recognition is being affected by word frequency. LRTs can be considered as frequency-based vocabulary assessments because the real words need to be selected from different ranges of frequency (frequency bands) in a large corpus (Lemhöfer and Broersma 2012).



2,000 Level

1 birth
2 dust
3 operation
4 row
5 sport
6 victory

—— game
—— winning
—— being born

3,000 Level

1 betray
2 dispose
3 embrace
4 injure
5 proclaim
6 scare

—— frighten
—— say publicly
—— hurt seriously

5,000 Level

1 gloomy
2 gross
3 infinite
4 limp
5 slim
6 vacant

—— empty
—— dark or sad
—— without end

Figure 2.2: Examples from the Vocabulary Levels Test (Schmitt et al. 2001).

2.4 Vocabulary Assessments Format

Vocabulary tests fall into two categories that can be effectively used to measure receptive and productive vocabulary knowledge (Milton 2009). In other words, receptive tests (size tests) concern with measuring the total number of known words, whereas productive tests (levels tests) concern with measuring the mastery of vocabulary at certain frequency bands within a given corpus (McLean and Kramer 2015). Many studies were carried out to measure the receptive vocabulary of learners, such as the Vocabulary Levels Test (Nation 1983) and Vocabulary Size Test (Nation and Beglar 2007). The productive vocabulary tests include, among others, the Productive Vocabulary Levels Test (Schmitt 2014). PVLTs are beyond the scope of this thesis.

2.4.1 Vocabulary Levels Test

The Vocabulary Levels Test (VLT) has been developed over the years and has different variants (Nation 1983, 1990a; Schmitt et al. 2001; McLean and Kramer 2015; Webb et al. 2017; Kremmel and Schmitt 2018). The test was made by Paul Nation during the 1980s. The VLT has been utilized as “tool to measure the written receptive vocabulary knowledge that is mainly required for reading” (Kremmel and Schmitt 2018).

The Vocabulary Levels Test (VLT) is a frequency-based test because it help the learners to learn words that belong to a given frequency level. It was given the name “Levels Test” because it assesses the knowledge of learners at different frequency levels of English word families. Initially, the test used to have four levels: 2000, 3000, 5000 and 10,000. In the old setting, each level is comprising 60 words. It was then updated by Webb et al. (2017) to include five levels: 1000, 2000, 3000, 5000 and 10,000. In this updated setting, each level is comprising 30 words. In both cases, the words are presented as a list of items with three stems as shown in Figure 2.2. The learners are required to pair these three items with appropriate match from six choices.

VLT employs the assumption of matching the words from different frequency with their meanings has the potential to provide information about respondents vocabulary knowledge. Its format incorporates a matching task, where the respondents are required to match words that belong to different frequency bands with definitions or meanings (Pignot-Shahov 2012).

The VLT has been created to act as a diagnostic test, but not a vocabulary size test (Schmitt 2010). A diagnostic test is a test that determines areas of student weakness which require further remedial (Ricks 2015). However, VLT has been wrongly adopted to measure the size of vocabulary by many researchers (Kamimoto 2008). As a consequence, Nation and Beglar (2007) were motivated by the wrong-use of VLT and they designed a suitable format for vocabulary size, which is known as the Vocabulary Size Test.

2.4.2 Measuring Breadth of Vocabulary

Here we provide a brief overview about the choice of the test for measuring breadth/size of vocabulary.

Nation (2012a) identified two main approaches to measure vocabulary breadth: (i) the dictionary-based approach, and (ii) the corpus-based approach. In the first approach, the learners are being tested using a set of words extracted from a dictionary (Pignot-Shahov 2012) or a book (Baharudin et al. 2014). The corpus-based approach, on the other side, the learners are being tested using high frequency words which are drawn on language use. Therefore, it is more suitable for assessing language learners who have limited proficiency (Pignot-Shahov 2012).

Vocabulary Size Test Vocabulary size tests are typically used as quantitative measurement to assess learning progress by reporting the total number of words acquired by a language-learner (McLean and Kramer 2015). The Vocabulary Size Test (VST) [7] was created in 2007 by Paul Nation from Victoria University of Wellington. The test is intended to measure the written receptive vocabulary size for first and second-language learners of English. Originally, the VST used to be presented in English only. The test has been developed over the years, beyond English, we are aware of many bilingual VSTs which are listed in Paul Nation web page[8]. A bilingual test, in which, the item still English, but the multiple-choice options of the items are in the native language of the test takers (Janebi Enayat et al. 2018). Among others, these language include: Arabic, Korean, Japanese, Vietnamese (Nguyen and Nation 2011), etc.

The VST[9] (Nation and Beglar 2007) provides a measure of written receptive word knowledge based on word families. The word family frequency estimates are extracted from the spoken part of the British National Corpus (Nation 2006). Instead of utilizing the VLT's matching format, the VST is incorporating a multiple-choice format to conform with the guidelines provided by Read and Chapelle (2001). Originally, the test has two main variants (i) 20,000 word families version, which can be used with native speakers and non-native speakers, and (ii) 14,000 word families version, which is best used with only English non-native speakers. Nation (2012a) put the test items into three categories: high-frequency vocabulary, mid-frequency vocabulary and low-frequency vocabulary. The VST contains around 2,000 words high-frequency words, 7,000 words mid-frequency words (making the total as 9,000) and more than 11,000 low-frequency words (making the total as 20,000).

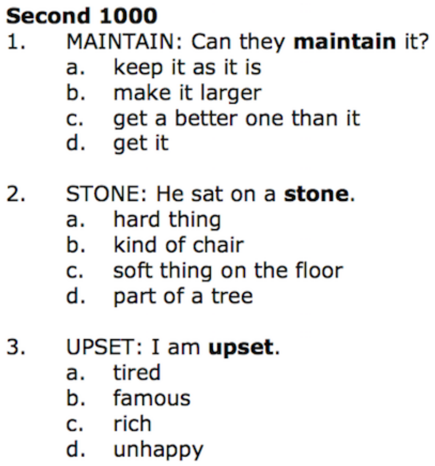
The VST has different number of items, it comes with 70, 100, 140 or 200 items (typically a lemma). Figure 2.3 shows an example from the second 1000 word levels. Each item is followed with 4 possible answers (1 correct, the remaining are distractors) (Nation 2012b).

Usually, the distractors and the correct answer belong to same part of speech (PoS), verbal sentence etc. For example, (tired, famous, rich and unhappy) are all adjectives. In average, the 100 items test takes about 30 minutes, whereas the 140 items test takes about 40 minutes to

[7] `http://www.lextutor.ca/`

[8] `https://www.victoria.ac.nz/lals/about/staff/paul-nation`

[9] `https://www.victoria.ac.nz/lals/about/staff/publications/paul-nation/Vocabulary-Size-Test-14000.pdf`



Second 1000

1. MAINTAIN: Can they **maintain** it?
   a. keep it as it is
   b. make it larger
   c. get a better one than it
   d. get it

2. STONE: He sat on a **stone**.
   a. hard thing
   b. kind of chair
   c. soft thing on the floor
   d. part of a tree

3. UPSET: I am **upset**.
   a. tired
   b. famous
   c. rich
   d. unhappy

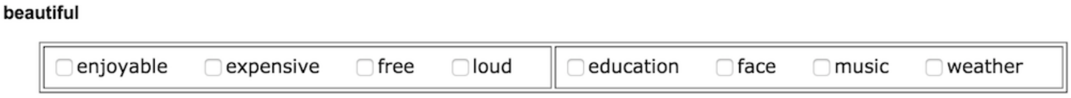
Figure 2.3: Examples from the Vocabulary Size Test (Nation and Beglar 2007).

complete (Nation 2012b). Roughly, the 70 and 200 items tests take about 20 and 60 minutes respectively.

2.4.3 Measuring Depth of Vocabulary

Here we consider other aspects of word knowledge and provide a brief overview about one of the choices for measuring depth of vocabulary.

Word Associates As indicted in its name, the test is based on the word associations task. The terms Word Associates Test Read (1993) and Word Associates Format (Schmitt et al. 2011) are used interchangeably. Read (1993) created the WAF as an attempt to measure the quality (depth) of English word knowledge. Figure 2.4 shows an example item that is excerpted from the Word Associates Test[10]. The test contains 40 items, the test-takers are being shown stimulus words that followed with a set of other words. For each stimulus word, there are eight options (four act as actual associates, the remaining four are distractors). Respondents are expected to select the related words (associates). They have to check four per stimulus from both boxes. The four correct answers for this question are “enjoyable”, “face”, “music” and “weather”. Notice that the last three associates are (adjective + noun) collocations: “beautiful face”, “beautiful music” and “beautiful weather”.



beautiful

□ enjoyable □ expensive □ free □ loud | □ education □ face □ music □ weather

Figure 2.4: Example of an item from the Word Associates Test (source: `https://www.lextutor.ca/`).

So far, we have introduced the Vocabulary Levels Test, the Vocabulary Size Test and the Word Associates Test. In the next chapter, our discussion will be more focused on the recognition/receptive vocabulary size tests that are based on the *Lexical Decision Task*.

[10] `https://www.lextutor.ca/tests/`

2.5 Chapter Summary

In this chapter, we shed light on the different aspects of vocabulary knowledge and vocabulary assessments formats. We introduced the term vocabulary acquisition and talked about its importance for enriching the language four basic skills (language practice), and hence language proficiency. We provide an overview of language proficiency testing based on the European standard, or the Common European Framework of Reference (CEFR). The learner's vocabulary knowledge is affected by the word construct, which can be further analyzed into: tokens, types, lemmas, and word families; which all serve as indicators about the number of words in a corpus. The earliest framework to predict language proficiency was proposed in 1980s and makes a distinction between size and depth of vocabulary knowledge. This framework remains function and developed over the years, for example, it was extended by Nation who provided an extended view with a total of 18 receptive and productive aspects. He views the vocabulary tests either as receptive or productive vocabulary tests. Finally, we provide relevant examples of vocabulary assessments formats, for instance, the VLT, VST, LRTs and word associates.

Chapter 3

Lexical Recognition Tests

> “The Yes/No format, in a striking analogy to the C-test format, has also been reported as being suitable, reliable and valid for placement and screening purposes."
>
> — Harsch and Hartig (2016)

English is the world’s dominant language. It is the language for global communication, scientific research and teaching. A large number of universities around the world require their new applying students to show a particular level of proficiency in English language before they offer them a place for studying. For example, TOEFL[1] and IELTS[2] are the most common commercial tests for English language proficiency world wide. Such tests, however, take couple of hours and they are somehow expensive. To overcome this problem, the universities’ language centres have the potential to establish/offer their own versions of language testing.

Similarly, the language learners are normally required to prove their level of second-language proficiency before they are offered a place at a certain level or/and after completing a certain level. Otherwise, to assess learners’ levels, the language institutions ask the learners to conduct their own placement tests, such as lexical recognition tests (LRTs) (Meara and Jones 1987) or C-tests (Klein-Braley and Raatz 1982; Chapelle 1994). The *Institut für Optionale Studien* at the University of Duisburg-Essen, for example, utilizes C-tests to assess German level for students. As per the scope of this thesis, we are not considering C-tests.

It is important to notice that the two terms “Lexical Recognition Test” and “Yes/No Test” are used interchangeably. This chapter sheds light on the LRTs, which are frequently used for measuring a learner’s vocabulary size based on word recognition (Meara and Buxton 1987). A LRT comprises words and nonwords in a ratio of 2 to 1. LRTs are based on the assumption that recognizing a word is sufficient for ‘knowing’ the word (Cameron 2002). Meara and Buxton (1987) argued that “without some knowledge of what the words appearing in a test mean, it is difficult to perform at all”. On the other side, nonwords play a vital role in the test because they are used as distractors to correct for guessing and response bias because they enforce the learners to respond carefully for the test. If they are too easy, one can do the test without knowing what the words mean and even without mistakes (Brysbaert 2013).

Lemhöfer and Broersma (2012) created the Lexical Test for Advanced Learners of English (LexTALE). LexTALE is recent and a well-established version of LRTs that can be finished

[1] `https://www.ets.org/toefl`

[2] `https://www.ielts.org/`

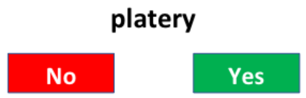


Figure 3.1: Example of a lexical recognition test item as Yes/No question.

fast, as it only uses 60 items. The two main advantages of LRTs, such as LexTALE, are often summarized by its economy (finish fast) and usability (simplicity). It only takes five minutes, only “Yes/No” questions are asked – see Figure 3.1, and scoring can easily be automated.

The chapter is organized as follows: In section 3.1, we provide a detailed overview of the LRTs, and more specifically LexTALE and LexTALE-like tests. In section 3.2, we introduce the quality criteria for language testing with focus on LRTs. This is followed by section 3.3 describing the scoring criteria for language testing, precisely the LRTs. Finally, we summarize the chapter in section 3.4.

3.1 Lexical Recognition Tests

3.1.1 Overview

The LRTs are based on word recognition task (Cameron 2002). Diependaele et al. (2012) identified the lexical decision as one of the most popular tasks used by word recognition researchers. Thus, we start by introducing the concept of word recognition and the Lexical Decision Task (LDT).

As per Wikipedia definition, “word recognition is the ability of a reader to recognize written words correctly”. Sometimes, it is known as the “isolated word recognition” as it entails a learner’s ability to recognize the words in a list out of context.[3]

Word recognition researchers usually make use of LDTs (Diependaele et al. 2012). The LDT is one of the most widely used tasks in psychology and psycholinguistics experiments.[4] The LDT task involves either of two paradigms: (i) decide whether a string (or sequence of letters form) is an existing word, or (ii) measure how quickly (in terms of response time) the people can classify a given string into words or nonwords (Balota and Chumbley 1984). The second paradigm is beyond the scope of this thesis. LDT is a subcategory of the experimental protocol Signal Detection Theory (Nevin 1969), which measures the individuals’ ability to distinguish between the presence and absence of signals (Macmillan 2002).

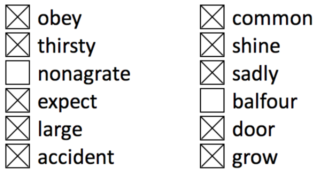
Typically, the signal can have multiple forms that vary based on the experiment. In the LRTs, for example, the signal is present if the checked item is a real or an existing word e.g. BRAIN; whereas it is absent when the item is a nonword e.g. BRANK.[5]

Presentation Formats In LRTs, participants are being shown a list of stimulus items with words and nonwords, presented either as: (i) a consequent “Yes/No” questions similar to that

[3] https://en.wikipedia.org/wiki/Word_recognition

[4] https://en.wikipedia.org/wiki/Lexical_decision_task

[5] Examples are taken from (Diependaele et al. 2012).



☒ obey
☒ thirsty
☐ nonagrate
☒ expect
☒ large
☒ accident

☒ common
☒ shine
☒ sadly
☐ balfour
☒ door
☒ grow

Figure 3.2: Example of a lexical recognition test in checklist format.

Test	# of Words	# of Nonwords	Reference
EVST	100	50	Meara (1990)
LexTALE	40	20	Lemhöfer and Broersma (2012)
LexTALE (Dutch & German)	40	20	Lemhöfer and Broersma (2012)
LexTALE_Fr	56	28	Brysbaert (2013)
LexTALE-Esp	60	30	Izura et al. (2014)
LEXTALE_CH	60	30	Chan and Chang (2018)

Table 3.1: Ratio of words-to-nowords in different LRTs.

in Figure 3.1, or (ii) a checklist format as shown in Figure 3.2. The task is to give a binary decision for all stimuli items by recognizing words from nonwords. Thus, respondents need to differentiate between words and artificial nonwords that look much like real words. In other words, she/he is supposed to respond with 'Yes' when the item is a word that exists in the lexicon/dictionary, and 'No' otherwise.

Next, we briefly describe the different versions of LRTs. Table 3.1 lists the LRTs developed over the years along with the ratio of words-to-nowords for each test. It is important to notice that all tests have same ratio of word to nonwords (2:1). Usually, the test takers have no information about this ratio. Next, we start by presenting the 'Eurocentres Vocabulary Size Test', an early version of LRTs.

3.1.2 Eurocentres Vocabulary Size Test

The EVST is an early example of using nonwords for testing (Meara and Jones 1987). The test is named after Eurocentres schools, a group of language schools based in Switzerland (Meara 1990).

EVST comprises 150 items, two-thirds are real words and the remaining one-third are nonwords. Based on the lexical decision paradigm, the participants are required to indicate which words they know. The nonwords are used as distractors to avoid response bias. The EVST is a computerized test based on ten frequency bands (1000 words each, and obtained from (Thorndike 1944)). EVST is computer adaptive, where it starts with the easiest (most frequent) words by presenting a sample of 10 words and 5 nonwords. According to Macmillan (2002), "the EVST's final score is automatically generated by the computer program". More details about EVST can be found in the work by Izura et al. (2014). Next, we present recent LRTs for English and other languages.

CEFR Level	CEFR Description	LexTALE Score
C1 & C2	Upper & lower advanced or proficient user	80-100%
B2	Upper intermediate	60-80%
B1 & lower	Lower intermediate & lower	below 59%

Table 3.2: The relationship between CEFR levels and LexTALE scores (Lemhöfer and Broersma 2012).

3.1.3 LexTALE: The Lexical Test for Advanced Learners of English

Lemhöfer and Broersma (2012) create LexTALE, an adapted version of the EVST that only uses 60 items – 40 words and 20 nonwords. As a result, the test can be finished faster. Typically, LexTALE uses a computerized form, individual items are usually presented in isolation (see Figure 3.1) in order to minimize context effects. It also uses the paper-based format and fits on a single sheet of paper. This is the so called *checklist* format (see Figure 3.2). Figure 3.3 shows the LexTALE stimuli items presented in a checklist format, along with their correct responses (Y/N), where words are checked, nonwords are not.

Lemhöfer and Broersma (2012) defined LexTALE as "a practically feasible test of vocabulary knowledge for the medium to highly proficient speakers of English as a second language (L2)". Overall, it is quick, easy to administer, free, and a valid and standardized test of vocabulary knowledge. It has also been shown to give a fair indication of general English proficiency. In the following, we briefly describe the main features of LexTALE based on the work by Lemhöfer and Broersma (2012).

Quick The LexTALE takes between 3 - 5 minutes to complete. It contains only 60 items, which makes it an economic and feasible test.

Easy and Usable The LexTALE can either be administered as a paper or computer-based. Furthermore, it can be implemented in any experimental software such as SurveyMonkey[6] or moodle[7]. For an easy replication by interested researchers, it is possible to download the list of stimulus items, instructions, and implementation details from LexTALE[8] official website.

Valid Lemhöfer and Broersma (2012) validated the resulting LexTALE scores by correlating them with other proficiency scores based on a word translation task and the commercial 'Quick Placement Test' (Syndicate 2001). In other words, they find that the LexTALE scores comply with the Common European Framework of Reference for Languages (CERF) levels.

In a large-scale study on advanced learners of English from Netherlands and Korea, the LexTALE was evaluated both as: (i) a measure of English vocabulary size, and (ii) an indicator of general English proficiency. As a result, they found that LexTALE scores correlate well for both tasks and can be partially mapped to one of the six-levels on CERF scale as shown in Table 3.2.

Free The LexTALE is free of charge, researchers and research institution can reuse LexTALE to conduct their own research.

[6] `https://www.surveymonkey.com/`

[7] `https://moodle.org/`

[8] `http://www.lextale.com/downloadthetest.html`

□ mensible
☒ scornful
☒ stoutly
☒ ablaze
□ kermshaw
☒ moonlit
☒ lofty
☒ hurricane
☒ flaw
□ alberation
☒ unkempt
☒ breeding
☒ festivity
☒ screech
☒ savoury
□ plaudate
☒ shin
☒ fluid
□ spaunch
☒ allied
☒ slain
☒ recipient
□ exprate
☒ eloquence
☒ cleanliness
☒ dispatch
□ rebondicate
☒ ingenious
☒ bewitch
□ skave

☒ plaintively
□ kilp
□ interfate
☒ hasty
☒ lengthy
☒ fray
□ crumper
☒ upkeep
☒ majestic
□ magrity
☒ nourishment
□ abergy
□ proom
☒ turmoil
☒ carbohydrate
☒ scholar
☒ turtle
□ fellick
□ destription
☒ cylinder
☒ censorship
☒ celestial
☒ rascal
□ purrage
□ pulsh
☒ muddy
□ quirty
□ pudour
☒ listless
☒ wrought

Figure 3.3: LexTALE items with correct responses, words are checked, nonwords are not.

Better than self-ratings Lemhöfer and Broersma (2012) compared the predictive power of LexTALE with that of self-ratings. They assessed the self-ratings separately in four tasks: writing, reading, listening and speaking proficiency. As a result, self-ratings found to be not as strong as the LexTALE.

3.1.4 LexTALE-like Tests

According to Ricks (2015), “the lion’s share of recent vocabulary-focused research has been devoted to English and a few other European languages”. LexTALE has been adapted to other languages beyond English, such as Dutch and German (Lemhöfer and Broersma 2012), French (Brysbaert 2013), and Spanish (Izura et al. 2014). Recently the test has been adapted to Mandarin Chinese (Chan and Chang 2018).

In the following, we summarize the LexTALE-like tests based on the overviews provided in the published work by Lemhöfer and Broersma (2012), Brysbaert (2013), Izura et al. (2014) and Chan and Chang (2018) respectively.

Dutch and German versions Besides the standardized and validated English version of the LexTALE. There are also German and Dutch versions of LexTALE that were developed in parallel by Lemhöfer and Broersma (2012). For an easy access, both can be done online[9] and/or downloaded[10] for replication.

French version Brysbaert (2013) constructs a French equivalent of LexTALE (LexTALE_Fr), which is a fast, free, and efficient test to measure language proficiency in French. LexTALE_Fr keeps the (2:1) words/nonwords ratio, but the participants get a relatively bigger set of random sequence of 56 French words of varying difficulty and 28 French-looking nonwords. For the ease of experiments, LexTALE_Fr is accompanied by a set of instructions in three languages: French, Dutch, and English. Further information can be found online, under their official website[11].

Spanish version Izura et al. (2014) developed a Spanish version of the LexTALE test (Lextale-Esp). The stimuli are tested by presenting them to a group of L1 speakers and a group of L2 speakers. Brysbaert (2013) noticed that, in particular, constructing suitable nonwords is a challenge. In order to be able to make a good selection of stimuli, Izura et al. (2014) started off with 90 words and 90 nonwords, to end up with 60 good words and 30 good nonwords. In this way, not only the test discriminated well at the high and the low end of Spanish proficiency, but also returned a big difference between the vocabulary size of Spanish native and non-native speakers.

Chinese version Chan and Chang (2018) create the Lexical Test for Advanced Learners of Chinese (LEXTALE_CH). LEXTALE_CH is a free, fast, and effective method for roughly estimating the vocabulary size of Mandarin Chinese. Chan and Chang (2018) started with a pilot study that comprises 180 items (90 are lexical characters, 90 are nonce characters). The test items were tested by a group of L1 Mandarin speakers and a group of L2 Mandarin learners. They end up with 90 items (60 are lexical characters, the remaining 30 are nonce characters). LEXTALE_CH correlates well and can be reliably used to assess L1 and/or L2 Mandarin proficiency in a quick manner.

Using nonwords constitutes an improvement over other forms of vocabulary proficiency testing, as it simplified the setup. For example, the Vocabulary Levels Test (Nation 1990b) is based on matching words with definitions, which is much harder to administer and automate. In the past, nonwords have been manually created. However, for repeated testing, as used in formative assessment (Wang 2007) we need to be able to generate them automatically. Therefore,

[9] `http://www.lextale.com/takethetest.html`

[10] The same url as in 8.

[11] `http://crr.ugent.be/programs-data/`

in the next chapter, we explore the methods for the automatic generation of good nonwords.

3.2 Quality Criteria

LRTs need to fulfill quality criteria. Objectivity, reliability, and validity are the three most important quality criteria for language testing (Hughes 2007) (as cited in Beinborn 2016). Furthermore, another important aspect of quality is usability or economy of the test. Elder and von Randow (2008) described vocabulary size measures as being valid, reliable and suitable for placement testing purposes (as cited in Harsch and Hartig 2016). Our following explanations are inspired by Beinborn (2016) work, who reviews these criteria in the light of C-test.

3.2.1 Objectivity

A test that is objective measures without reference to outside influences, thus the evaluation of a test is independent of the test organizers or raters (Beinborn 2016). An LRT is highly objective because each stimulus has a binary classification: true or false. The score is the same, regardless of it is manually or automatically scored. This is in contrast to essay scoring or short-answer scoring, where individual raters might disagree about the score given/granted for a certain answer.

3.2.2 Reliability

Kluitmann (2008) defined Reliability as “the correlation between the results of one test administration T and another test administration T' under the same circumstances”. A test is reliable if it returns the same result each and every time for the same input. The simplest approach for measuring reliability is a ‘test and retest’ approach (Baba 2002). If the same group completes two comparable versions of an LRT within a short time interval, the results should be highly correlated and hence highly reliable.

3.2.3 Validity

Validity is the extent to which the test measures what it claims to measure (Brown 1989) (as cited in Beinborn 2016). For an LRT this means that it should actually measure the vocabulary size and not general intelligence or knowledge of some topic. Usually, validity is further split into criterion validity, content validity, construct validity, and face validity.

Criterion Validity According to Beinborn (2016), “criterion validity indicates that the test correlates with an expected outcome”. For example, an LRT score should predict the learning success of a participant in a specific language level. In our experiments, e.g. user studies, we evaluate that by comparing the test results with independent assessments provided by external rater, such as a language teacher.

Content Validity According to Beinborn (2016), “content validity captures the representativeness of the test for the measured construct”. The content validity can be improved if the test items are of appropriate difficulty. In the LRTs, this is typically achieved by selecting words from different ranges of relative frequency in a large corpus (Lemhöfer and Broersma 2012). This conforms to the well-known fact: there is a high correlation between the frequency of a word and its difficulty (Greenberg 1965).

Construct Validity "What does the test exactly measure?" is the main question posed about LRTs (Ricks 2015). This indicates that the LRT measures the intended vocabulary size construct through word recognition. The LRT format relies on learner self-ratings, and the Yes/No questions actually indicate word knowledge (Ricks 2015). Construct validity can be approximated by measuring the correlation of the test with a well-established measures of the construct (Beinborn 2016).

Face Validity According to Beinborn (2016), 'face validity refers to the transparency of the test and considers the perspective of the test participants". With words only, the LRTs will apparently lack face validity. Actually, we will have no direct evidence that the claim to know a word (via the Yes button or checkbox) indicates knowledge of any of the various facets of word knowledge (Ricks 2015). The presence of nonwords items among the LRTs functions as a necessary corrective to the format's lack of face validity (Harrington et al. 2006).

3.2.4 Usability or Economy

A test that takes many hours to complete and requires complex equipment is considered to be less useful (Beinborn 2016). LRT is a short, quick, and free test that roughly takes five minutes to complete and can be easily scored automatically.

3.3 Scoring LRTs

Keep in mind that LDT is a subcategory of SDT. The LRTs scoring is based on SDT (Eyckmans 2004).

Based on this LDT categorization, the LRTs have four possible stimulus-response combinations: (i) true positive: the knowledge of a real word (correct responses) is known as TP or a hit. (ii) false positive: claiming to know a nonword (wrongly selected as word) is known as FP or a false alarm. (iii) false negative: not claiming to know a real word is labeled as FN or a miss. and (iv) true negative: rejecting a nonword is considered as TN or a correct rejection.

Table 3.3 shows the corresponding matrix for stimulus and response alternatives. Notice that the stimulus alternatives are (w) for a word and (nw) for a nonword, whereas the response alternatives are yes (Y) and no (N). Among the four responses, there are only two types of correct responses: 'yes' in the case of a real word (TP) and 'no' if the item is a nonword (TN).

Stimulus	Response Alternative	
Alternative	**Y**	**N**
w	TP	FN
nw	FP	TN

Table 3.3: The four LRT's stimulus, response combinations (Abdi 2007).

3.3.1 Existing Scoring Criteria

The issue of LRTs scoring is connected to both the nonwords and to respondent testing behaviour (Huibregtse et al. 2002). There are several possible methods to score LRTs. Here we review the scoring schemes as being used in for the LRTs for: English (LexTALE) (Lemhöfer and Broersma 2012), French (LEXTALE_FR) (Brysbaert 2013), and Spanish (Lextale-Esp) (Izura et al. 2014).

Traditional method The traditional method of calculating a testee's score in an LRT was simply to count up hits and subtract false alarms multiplied by two. This scoring scheme was adopted to find the scores for LEXTALE_FR and Lextale-Esp.

Percentage correct method As as shown in Equation 3.1, it is a simple percentage correct measure which is corrected for the unequal proportion of words and nonwords by averaging the percentages correct for these two item types. Lemhöfer and Broersma (2012) call this measure as % correct$_{\text{av}}$ (averaged % correct).

$$LexTALE_{\text{score}} = \frac{(\frac{\#ofwordscorrect}{40} + \frac{\#ofnonwordscorrect}{20}) * 100}{2} \tag{3.1}$$

SDT method As indicated from the name, this method is based on signal detection theory (SDT). This method has been developed by Huibregtse et al. (2002) and it is called I$_{\text{SDT}}$. This scoring formula corrects for both guessing and personal response style (e.g., bias toward yes or no responses).

Lemhöfer and Broersma (2012) employed the three aforementioned methods, and argue that the second method produces the best results. For further details about the three methods, readers are suggested to refer to the work by Huibregtse et al. (2002) or look at *Appendix B* in the work by Lemhöfer and Broersma (2012).

3.3.2 Used Scoring Criteria

In the remaining chapters, we follow (Lemhöfer and Broersma 2012) and use the second method to calculate the respondents' scores. We have our own thought and the score can be better viewed as a function of recall. Basically, the percentage of correct responses for words represent the recall for words. Whereas, the percentage of correct responses for nonwords represent the recall for nonwords. In other words, it averages the corresponding recalls for words and nonwords as shown in Equation 3.2.

$$score(R) = \frac{(R_w + R_{nw}) \cdot 100}{2} \tag{3.2}$$

where R_w is the recall on words and R_{nw} is the recall on nonwords. They can be computed using the following equations respectively:

$$R_w = \frac{TP}{TP+FN} \tag{3.3}$$

$$R_{nw} = \frac{TN}{TN+FP} \tag{3.4}$$

Some of the respondents might approach answering the test using guessing, a phenomena that involves yes or no bias. A yes bias, in which, a respondent identifies (checks) all items as words. A no bias, in which, a respondent identifies all items as nonwords (checks none of the items). By using Equation 3.2, in spite the different numbers of words versus nonwords, the yes and no biases will be handled appropriately. For instance, a yes bias creates high error rates in the nonwords and would be *penalized* in the same way as a no bias that causes high error rates for words (Lemhöfer and Broersma 2012).

It is important to notice that we are not aiming at increasing the recall for words and nonwords nor the respondents' scores. A respondent's score is not valid until it correlates well with her/his proficiency level.

3.4 Chapter Summary

In this chapter, we provided a detailed overview of lexical recognition tests. Apparently, more efforts have been devoted to English and other European languages in the modern language learning research. The chapter was started by shedding light on the LRTs landscape and the existing LRTs along with their presentation formats, either as Yes/No questions or checklist format. We gave more attention to a recent LRT, namely the LexTALE, because it has been validated on the CERF scale. The two main advantages of LexTALE, are often summarized by its economy and usability. We presented the most important quality criteria for language testing: objectivity, reliability, validity, and economy. We ended this chapter by presenting the different schemes for scoring the LRTs. Our adopted scoring criterion is similar to that used by Lemhöfer and Broersma (2012), except, we are presenting the score as a function of recalls for words and nonwords.

Chapter 4

Arabic NLP

Arabic has its own morphological and syntactic specificities.

To bridge the gap for non-Arabic speakers, this chapter presents background material and a review of literature on the characteristics of the Arabic language and its natural language processing. The chapter is organized as follows: In section 4.1, we provide an overview of the different varieties of Arabic language. In section 4.2, we briefly describe the Arabic script, the dominant writing system used to write Arabic and many other languages around the world. This is followed by section 4.3 by briefly describing some linguistic characteristics of the Arabic language. Section 4.4 presents the Arabic natural language processing (NLP) tools benchmarked in this thesis. We specifically focus on the challenges inherent in Arabic NLP in section 4.5. Section 4.6 is dedicated to present the Arabic corpora used among this thesis. Finally, we summarize the chapter in section 4.7.

4.1 What is Arabic?

We start by defining the Arabic language, which is spoken by more than 422 million people, including 290 million[1] as first language (L1). It is the most widely spoken member of the Semitic languages (Abdelgadir and Ramana 2017), and one of the six official United Nations (UN) languages[2]. The Arabic language is often characterized as being interesting and challenging (Farghaly and Shaalan 2009; Azmi and Almajed 2015). It has also historical, cultural, religious, and political significance (Farghaly and Shaalan 2009; Farghaly 2010). However, we consider Arabic as an under-resourced language because it has rather received a little attention in the recent computational linguistics (Green et al. 2010; Green and Manning 2014). In comparison to English and other European languages, such as German and French that have a large number of resources and natural language processing tools. Arabic has a minimal resources. This is due to a long history of research and investment conducted on such European languages (El-Haj et al. 2015).

It has been noted that a human-language originates and develops in a "natural continuum"

[1] `https://en.wikipedia.org/wiki/List_of_languages_by_total_number_of_speakers`, accessed: Nov 21, 2018

[2] `https://en.wikipedia.org/wiki/Official_languages_of_the_United_Nations`, accessed: Nov 21, 2018

(Habash 2010). The Arabic language is an old language that originates in *Arabian Peninsula* (شبه الجزيرة العربية) in the period that predates Islam. Geographically, the term Arabic has been used rather loosely (Zaidan and Callison-Burch 2014). With the rise of Islam, Arabic has been used as the media of communication (spoken) not only among Arabs, but also among non-Arab Muslims (Albalooshi et al. 2011). Further details about the ethnic background of Arabic speakers can be found in the work by Farghaly (2010). Spoken Arabic is usually known as colloquial Arabic, dialects, or vernaculars (Sadat et al. 2014). In spite of their substantial mutual differences, Arabic can be typically used to refer to a set of language varieties which are culturally, and linguistically related (Samih and Kallmeyer 2017).

Forms of Arabic Habash (2010) defined the Arabic language as "a collection of multiple varieties among which one particular variety has a special status as the formal written standard of the media, culture and education across the Arab World". In the literature, the Arabic varieties are categorized into three general forms (Farghaly and Shaalan 2009). In the following summaries, we are inspired by the overviews published by Farghaly and Shaalan (2009), Habash (2010) and Samih and Kallmeyer (2017).

- **Classical Arabic (CA)** is the primary form of Arabic. It is the language of the Jahillyah literature (Arabic period before the arrival of Islam), the Holy Quran and the Hadith[3]. It's golden time dates back to the 7th and 9th century from Umayyad and Abbasid times where it is used in literary texts, scientific research, and translation. Historically, it has been noted by Farghaly and Shaalan (2009) that the CA (فصحى التراث) has remained unchanged, intelligible and functional for more than 1500 years.
- **Modern Standard Arabic (MSA)** is the second form of Arabic. It is the official language of the 22 countries comprising the Arab World[4]. Official indicates that MSA is typically written not spoken. Although the MSA (فصحى العصر) is the lingua franca (Qwaider et al. 2018) amongst the literary, media (written (newspaper) and spoken (radio and TV)) and education. For example, the daily news at Aljazeera channel and BBC Arabic radio are broadcasted in MSA. However, it is rarely used as native language nowadays. According to linguists researchers, MSA is not only syntactically based on CA, but also morphologically and phonologically (Habash 2010).
- **Dialectal Arabic (DA)** At the regional level, we have the Arabic dialects or dialectal Arabic (لهجات عربية), known as colloquial Arabic and refer to the spoken varieties of Arabic (Sadat et al. 2014; Al-Sobh et al. 2015). There are many varieties of Arabic dialects, which are true native language forms. There are Arab dialects more than the members of the Arab World (Samih and Kallmeyer 2017). Albirini (2016) has noticed a considerable variation amongst the Arabic dialects. This is not limited to the variation from one country to another, it can be further extended to include many other forms of variation. As in the variation amongst one city to another, or one state/region to another

[3] The way of life prescribed as normative for Muslims on the basis of the *Sunnah*, i.e. the teachings and practices of the prophet Mohammad (Peace be upon him), including the interpretations of the Quran (Samih and Kallmeyer 2017).

[4] The Arab world consists of 22 countries (12 in Asia and 10 in Africa), namely Algeria, Bahrain, Comoros, Djibouti, Egypt, Iraq, Jordan, Kuwait, Lebanon, Libya, Mauritania, Morocco, Oman, Palestine, Qatar, Saudi Arabia, Somalia, Sudan, Syria, Tunisia, the United Arab Emirates, and Yemen (Habash 2010).

inside the same county. In Palestine, for example, the people in Hebron city have a dialect that differs from the one used in Tulkarm city. Similarly, the people in Nablus rural district speak dialects that differ from the one used by the urban in Nablus city. Except in few cases, the dialects are typically spoken not officially written. For instance, they are heavily used for informal communication in social media (Habash et al. 2012).

Farghaly and Shaalan (2009) has reported that the three varieties are almost practiced by every Arabian in her/his daily life. For example, "on any given day an Arabic native speaker will use CA while reciting his daily prayers; MSA when listening to or reading the news, and his particular dialect at home with family or friends".

According to Habash (2010), "the relationship between MSA and the dialect in a specific region is rather complex, because the Arabs do not think of these two as separate languages". This leads to a phenomenon or a situation, where two or more varieties of the same language coexist amongst the same speech community (*diglossia* (Ferguson 1959)). The diglossic nature of the Arabic language will be discussed in more details in subsection 4.5.2.

4.2 Arabic Script

The Arabic script has been used for many centuries to write the Arabic language. Arabic has been remarkably known because of its orthography style (spelling conventions) that remains unchanged for about fifteen centuries (Azami 2011). It has been noted by Azmi and Almajed (2015) that "a person with a slight knowledge of the language can easily read 1,400 year-old parchments". The Arabic script has been considered as one of the key linguistic characteristics of the Arabic language that imposes significant challenges to Arabic natural language processing (Farghaly and Shaalan 2009).

The Arabic script (الخط العربي) is the dominant writing system used to write Arabic from the period that predates Islam till nowadays. We describe it as being dominant, because there are some tries to write Arabic in Latin-based orthography[5] during 1960 (Plonka 2006) and Roman script "Arabizi" (Yaghan 2008). As a writing system, Arabic script was also used to write many human-languages around the world, including ones that are not related to Arabic or ones out side the Arab world [6]. The Arabic script is written from right to left (Farghaly and Shaalan 2009) in a cursive style (Habash 2010), even in computer-printed form (Khorsheed and Clocksin 1999). As per Wikipedia definition, a cursive or longhand script (among other names, aka looped writing or joint writing) "is any style of penmanship (calligraphy) in which some characters are written joined together in a flowing manner"[7]. To write words, the script contains two classes of symbols: letters and diacritics (Habash and Rambow 2007). Besides

[5] Dedicated for Lebanese Arabic (as cited in Habash et al. 2012).

[6] For example, Persian, Kurdish, Urdu, Pashto and Turkish in the past (before 1932).

[7] `https://en.wikipedia.org/wiki/Cursive`

these two symbols, we briefly discuss digits and punctuation. As a general paradigm for this chapter, we are sometimes inspired by Habash (2010) work, which is a book about Arabic and its NLP. However, we support our discussion with our own or excerpted figures. In the following, we provide descriptions for letters and diacritics.

4.2.1 Letters

The Arabic alphabet has 29 letters, those include three long vowels (Alif (ا), Waw (و) and Yeh (ي)), 25 consonants like the "د" /d/[8] letter, and the Hamza or glottal stop (ء). The following overviews are based on the work by Habash (2010).

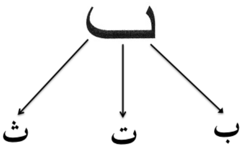


Figure 4.1: One of the letter forms with its three variants: (ث) /v/, (ت) /t/, and (ب) /b/.

Each Arabic letter is comprised from two parts: letter form and letter mark. The form is an essential component for every letter. Arabic alphabet has a total of 19 letter forms. Figure 4.1 shows one of the letter forms as well as all of its variants. The marks (consonantal diacritics) were made during Umayyad eras. Marks are pretty helpful to distinguish different letters. The marks fall into three types: the dots, short Kaf and the Hamza.

- Dots: The dots (points) are mandatory written with some letters form. The dots are five: one, two or three to fall above the letter form and one or two to fall below the letter form. Figure 4.2 shows the categorization of the dots alongside with relevant example letters and their use in some words.
- Short Kaf: The short Kaf is used to mark specific letter shapes of the letter Kaf (the isolated Kaf ك and Kaf at the final position of the word). As in the words "wldk" (ولدك meaning: your son) and "fmk" (فمك meaning: your mouth).
- Hamza: The Hamza letter mark can appear above or below specific letter forms. Hamza is used to indicate both the letter form (ء) at word final position as in the Arabic word سماء /smA'/[9] (meaning: sky), and letter mark that appears with other letter forms like أ, إ, ؤ, or ئ.

Letter Shapes The shapes are used in both print and handwriting, without any distinction. In Arabic, the letters have different shapes depending on their position in a word: initial, medial, final or stand-alone (independent or isolated) (Habash and Rambow 2007; Habash 2010). Because of their cursiveness characteristics (Khorsheed and Clocksin 1999), Arabic letters are jointly written in different ways. An Arabic letters can be classified either as a (i) connector, or (ii) non-connector. A connector letter is "a letter that connects from both sides", whereas non-connector letter "is a letter that does not connect to the subsequent letter" (Awada and Dobell

[8] International Phonetic Alphabet (IPA): `https://en.wikipedia.org/wiki/Help:IPA/Arabic`

[9] Bcukwalter transliteration: `http://www.qamus.org/transliteration.htm`

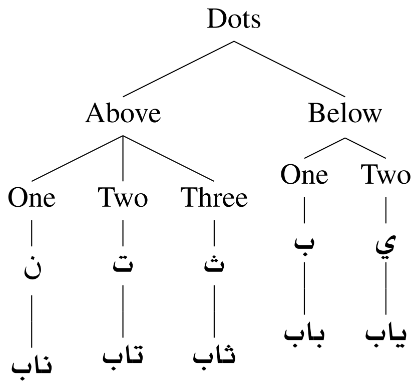
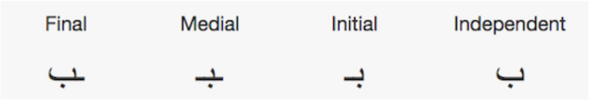


Figure 4.2: The dots either fall above or below the letter.



Final	Medial	Initial	Independent
ـب	ـبـ	بـ	ب

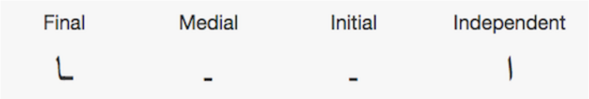
Figure 4.3: An example, the four shapes of the Arabic letter ب /b/ (Awada and Dobell 2016).

2016). Although most of the letters are connectors e.g. (ب) /b/ – see Figure 4.3, there are only six non-connectors letters e.g. (ا) /A/ – see Figure 4.4. It can be clearly seen that initial and medial shapes are almost identical, similarly are the final and independent shapes.

4.2.2 Diacritical Marks

The diacritical marks, or simply the diacritics (also know as tashkeel (Aboelezz 2010)) are distinguishing features of the Arabic language, they represent the second class of symbols in the Arabic script (Farghaly and Shaalan 2009; Habash 2010). Typically, the diacritics are inscribed as atop or below regular letters (Jarrar et al. 2016). Figure 4.5 shows the non-diacritized and the diacritized versions of the nominal sentence "*The Arabic script*". Although the letters are always written, diacritics are optional in almost Arabic MSA writing.

The diacritics were invented in the past to play two main roles: (i) provide a phonetic guide i.e. help readers recite the Quran text correctly, and (ii) clarify the intended meaning of ambiguous words (Jarrar et al. 2016). Arabic diacritics fall into two forms: the basic form (basic diacritics, see Table 4.1), and the extended form (Ahmed and Elaraby 2000). The extended form is a combination that contains the same set of basic diacritics as well as additional diacritics (Quranic diacritics), such as Dagger Alef, Alef Leyna and Alef Wasla. The Quranic diacritics are generally beyond the scope of this thesis, but we introduced them because the Dagger Alef (see Table 4.2) is being introduced by some diacritization tools. In such a case, we normalize the introduced Dagger Alef with a Fatha.



Final	Medial	Initial	Independent
ـا	-	-	ا

Figure 4.4: An example, the two shapes of the Arabic letter ا /A/ (Awada and Dobell 2016).

without diacritics الخط العربي

with diacritics اَلْخَطُّ اَلْعَرَبِيُّ

Figure 4.5: Example of an Arabic sentence without and with diacritics.

Based on its tashkeel state, the Arabic text can be fully diacritized, partially diacritized, or entirely undiacritized. Habash (2010) noted that “the most problematic aspect of diacritics is their optionality”. The NLP task of restoring diacritics (automatic diacritization) or simply diacritization is completely introduced in Chapter 6.

Type	Diacritic Mark	Name	Transl.	IPA	Word Position
Short vowels	دَ	Fatha	a	/a/	Any
	دُ	Damma	u	/u/	Any
	دِ	Kasra	i	/i/	Any
	دْ	Sukun	o	Ø	Any
Nunation	داً	Tanween Fath	F	/an/	End
	دٌ	Tanween Damm	N	/un/	End
	دٍ	Tanween Kasr	K	/in/	End
Gemination	دّ	Shadda	~	:	Any

Table 4.1: Types of Arabic diacritics

Form	Arabic Name	Unicode Name	Example
Quranic Diacritics	أَلِف خَنْجَرِيَّة	Dagger Alef	هٰذَا

Table 4.2: One example of Quranic diacritics.

Changing some of the attached diacritics may change both the syntax and semantics of a word. Thus, an Arabic word can be turned into another one that refers to a different lexical sense. According to Diab et al. (2007a), the diacritics are classified into lexical and syntactic diacritics:

- Lexical: A difference in lexical diacritics typically leads to different lexemes. Thus, a single Arabic word (bare form) might have several meanings given the different diacritized forms (Darwish et al. 2017). For example, the Arabic string عقد /Eqd/ has several lexemes as shown in Table 4.4.
- Syntactic: The final vowel or case-ending diacritics, which typically appear on the last letter of the word, indicate its grammatical role like nominative subject or accusative object (Darwish et al. 2017). The syntactic case of the word within a given sentence determines the syntax-dependent diacritic of the word (Metwally et al. 2016). In the following, we show two examples of the Arabic word علم /Elm/ diacritized with: (i)

"Damma" to indicate a subject, and (ii) "Fatha" to an indicate an object.

subject – يفيدُ عِلْمُ الحاسوب جميع العلوم

object – درسْتُ عِلْمَ الحاسوب

4.2.3 Digits and Punctuations

Typically, the Arabic text contains numbers and sentence boundaries, which are represented by digits and punctuation marks respectively.

Digits As in other languages, the Arabic numbers can be expressed in words or in a decimal system (digits). The digits are typically faster and fall into two categories: (i) Arabic numerals, and (ii) Hindi (Indian) numerals (Attia 2007). The Arabic numerals are similar to those used for English and comprise the ten digits: 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9. On the other side, the Hindi numerals comprise the ten digits: ٠ ١ ٢ ٣ ٤ ٥ ٦ ٧ ٨ and ٩. The majority of Arab league countries are using Hindi numerals (Attia 2007).

Punctuation Marks The Arabic punctuations marks are typically used to identify sentence boundaries (Abandah et al. 2015). Their shape look very similar to those used in English and other European languages. For instance, comma, dot, exclamation mark, colon, quotation mark, angle brackets, and brackets. However, because Arabic is written from right-to-left, some punctuations marks look slightly different as in the case of comma, question mark, and semi colon (Habash 2010). Table 4.3 shows the punctuation marks that different between English and Arabic. Arabic punctuations marks are pretty helpful for sentence boundary detection (Althobaiti et al. 2014), which is an NLP task that split a paragraph of text into multiple independent sentences.

Marks	English	Arabic
Comma	,	،
Question Mark	?	؟
Semi Colon	;	؛

Table 4.3: Punctuations marks that look slightly different in English and Arabic.

4.2.4 Phonology and Orthography

Here, we present a brief description of MSA phonology, followed by a description of how Arabic orthography (spelling standard) is used to map phonology to/from the Arabic script. The following descriptions are based on overviews by Habash (2010).

Phonology In natural languages, phonology concerns the study of how sounds, or phones, are organized. The basic building block in phonology is the phoneme, which is the smallest contrastive unit in the sound system of a specific language. Arabic is pretty rich in

Surface form	**Diacritized form**	**Translation**
عقد	عُقْد /Euqod/	necklace
	عِقْد /Eiqod/	decade
	عَقْد /Eaqod/	contract
	عَقَدَ /Eaqada/	held
	عَقَّدَ /Eaq∼ada/	complicated
	عُقَد /Euqad/	knots

Table 4.4: Lexical ambiguity of the non-diacritized Arabic token عقد /Eqd/.

contrastive phonemes, that result a pair of words with *ONE* phonemic difference only (minimal pairs). One good example, the MSA words (قلب meaning: heart) /qalb/ and (كلب meaning: dog) /kalb/ constitute a minimal pair for the phonemes /q/ and /k/. Phonology is important for the context of this thesis because we can employ the minimal pairs to create Arabic nonwords – see Chapter 8.

Orthography Orthography concerns how the sounds of a language are mapped to/from a particular written text. In other words, it concerns the mapping of Arabic letters to/from sounds to produce correct words. For example, the Arabic word طفل (/Tfl/, meaning: child) begins with ط /T/ but not ت /t/.

The majority of Arabic letters are characterized by having a one-to-one mapping to an MSA phoneme. Table 4.5 shows the mapping of Arabic letters to/from sounds using the International Phonetic Alphabet (IPA), Unicode letter name[10], Buckwalter transliteration, and Pronunciation. Note that the Pronunciation column is taken from the Lebanese Arabic Institute website, a language school located in Beirut and founded by Awada and Dobell (2016).

Transliteration It refers to the task of encoding words in the source language with their approximate phonetic or spelling equivalents in another language, such as encoding the Arabic words into English (forward-transliteration). The reverse process (getting the original Arabic word from the transliterated English string) known as backward-transliteration (Al-Onaizan and Knight 2002). Transliteration might be very helpful for researchers without Arabic knowledge (Farghaly and Shaalan 2009).

In addition to the standard IPA discussed above, many researchers in Arabic NLP use an orthographic transliteration, specifically a Latin-based script, such as Buckwalter (Buckwalter 2004) and Habash-Soudi (Habash et al. 2007b). Following the majority of Arabic NLP researchers, Buckwalter encoding (the 4th column in Table 4.5) is used exclusively for transliterating the Arabic words in this thesis, namely Chapter 6 to 9.

[10] `https://en.wikipedia.org/wiki/Arabic_script_in_Unicode`

Arabic Letter	Unicode Letter Name	IPA	Buckwalter Transl.	Pronunciation like
ء	Hamza	[ʔ]	’	glottal stop as “-” in “uh-oh”
ا	Alef	[a:]	A	sound preceding a vowel, as in “at”, “in” or “out”
ب	Beh	[b]	b	“b” in “band”
ت	Teh	[t]	t	“t” in “tan”
ث	Theh	[θ]	v	“th” in “thin”
ج	Jeem	[ʤ]	j	“j” in “jam”
ح	Hah	[ħ]	H	no English equivalent; by saying “ha”
خ	Khah	[x]	x	“ch” in German “nacht” or “loch”
د	Dal	[d]	d	“d” in “dance”
ذ	Thal	[ð]	*	“th” in “that”
ر	Reh	[r]	r	“r” in Spanish “carro”
ز	Zain	[z]	z	“z” in “zoo”
س	Seen	[s]	s	“s” in “sat”
ش	Sheen	[ʃ]	$	“sh” in “shine”
ص	Sad	[Sˤ]	S	emphatic counterpart of س, “s” in “sauce”
ض	Dad	[dˤ]	D	initial “d” in “dawdle”
ط	Tah	[tˤ]	T	emphatic counterpart of ت, “t” in “taught”
ظ	Zah	[ðˤ]	Z	emphatic counterpart of ذ, “th” in “though”
ع	Ain	[ʕ]	E	no English equivalent
غ	Ghain	[ɣ]	g	French “r” in “Paris”
ف	Feh	[f]	f	“f” in “fish”
ق	Qaf	[q]	q	“c” in “caught”
ك	Kaf	[k]	k	“k” in “kite”
ل	Lam	[l]	l	“l” in “land”
م	Meem	[m]	m	“m” in “man”
ن	Noon	[n]	n	“n” in “now”
ه	Heh	[h]	h	“h” in “hat”
و	Waw	[w], [u:]	w	“w” in “win”
ي	Yeh	[y], [i:]	y	“y” in “yes”

Table 4.5: Mapping of Arabic letters to/from sounds, Pronunciation is excerpted from Lebanese Arabic Institute (Awada and Dobell 2016).

4.3 Linguistic Characteristics

This section describes some linguistic characteristics of the Arabic language, namely morphology and syntax. Morphology describes how the words are structured, whereas the syntax describes how words are combined to form phrases and sentences (Payne 2006). We start this section by presenting clitics due to their importance for morphology. In the context of this thesis, we must handle the clitics in order to obtain reliable frequency counts for Arabic words – see Chapter 7.

4.3.1 Clitics

Alotaiby et al. (2010) defined a clitic as “a linguistic unit that is pronounced and written like an affix but is grammatically independent”. In Arabic morphology, clitics posses a heavy presence because they can be attached to a stem or to each other without any orthographical marks, like an apostrophe (Alotaiby et al. 2014). In analogy to English, it is an unstressed part of a word that occurs only in combination with another word, E.g. *’ll* in the sentence “you’ll play football”.

Regarding the clitics position in a given word, they can be either a predecessor or successor. A predecessor clitic precede the word (stem) like a prefix (proclitics, such as *Y* in Y’all), whereas a successor clitic follow the word (stem) like a suffix (enclitics, such as *’ll* in “you’ll play football”). In the worst case, there could be four concatenated proclitics (e.g. the definite article) and three enclitics (e.g. the object pronouns) attached to a stem (Alotaiby et al. 2010).

The problem in Arabic clitic is severe than in English because the clitics can limit our endeavours to find basic statistics about the Arabic stems or lemmas. For example, the space delimited word (وطلباتنا) /wTlbAtnA/ (meaning: *and our requests*) consists of four clitics /w+Tlb+At+nA/: (i) the conjunction (و) /w/ as prefix, (ii) the lemma (طلب) /Tlb/, (iii) (ات) /At/ as suffix to indicate feminine plural, and (iv) the possessive pronoun (نا) /nA/ as postfix. In order to obtain a reliable frequency count for the stem /Tlb/, we have to use segmenters and/or lemmatizers to discard such extra clitics.

4.3.2 Morphology

The term morphology refers to the study of word structure or form (Ritchey 1998). As in other Semitic languages[11], Arabic has a systematic ‘root-and-pattern’ morphology (El-Haj et al. 2015). Arabic is morphologically rich (Attia 2008) because the morphology (i) plays a very important role in words structuring and derivation (Althobaiti et al. 2014), and (ii) provides a base layer for other linguistic layers, e.g. it interacts with both orthography and syntax (Ahmed 2000).

Morphology concerns with the “morpheme” smallest expressiveness unit of a language (Fromkin et al. 2018). The morpheme can be either a stem, an affix (affixes) or a clitic (Althobaiti et al. 2014). Typically, the stem acts as the main morpheme of a word. Two main types of morphology are used to construct Arabic words, namely form-based and functional morphology (Habash 2010). In this thesis, we are only covering form-based morphology, in particular concatenative morphology where the morphemes are concatenated together. For example the word (معلمون) /mElmwn/ (meaning: teachers) consists of two morphemes: (i) معلم /mElm/, and (ii) ون /wn/. The stem is (معلم) /mElm/ (meaning: teacher), and (ون) /wn/ (equivalence: English plural s) as an affix (suffix) to indicate masculine plural nouns – a type of plural nouns in Arabic. Notice that Arabic has other types of plural nouns, among others, broken plural, summit plural and dual nouns to refer to a group of two. As shown in this example, affixes are attached to a stem (معلم + ون) – they come after the stem in this example. Typically, affixes fall into four categories[12] based on their position (Soori et al. 2013). Namely, an affix can be either: (i) a prefix that precedes the stem, (ii) an infix that is inserted within a stem or a root, (iii) a suffix that succeeds the stem, or (iv) a circumfix, precedes or succeeds the stem (Zerrouki and Balla 2009; Althobaiti et al. 2014). As it was reported by Zerrouki and

[11] Typically, Semitic languages provide no distinction between the lowercase/uppercase letters.

[12] Other researchers exclude infixes and group Arabic affixes into three categories, such as Habash (2010).

Balla (2009), Arabic used the letter the ت /t/ as a common infix to indicate past participle (VIII verb form) from verb form I. For example, the Arabic word اجتهد /AjthD/ (meaning: he worked hard) is taken from جهد /jhd/ (meaning: he strives).

We can use the morphological analyzers to indicate whether a morpheme is an affix or a clitic (Attia 2007). Suppose that we provide the word (وسيكتبونها) /wasayaktubunhA/ (meaning: and they will write it) as input to a morphological analyzer. Probably, it will produce (i) "wasaya+ktub+uwnhA", or (ii) "wa+ sa+ y+ aktub+ uwna+ hA". This first analysis indicates that it consists of two proclitics, one circumfix and an enclitic. Whereas the second one is an extended analysis (Habash 2010) – see Table 4.6 . Notice that row # 2 is the same as # 1 but without diacritics. Furthermore, M.plural (in row # 3) is an abbreviation for masculine plural.

#	Proclitic	Proclitic	Circum1	Stem	Circum2	Enclitic
1	wa+	sa+	y+	aktub+	uwna+	hA
2	w+	s+	y+	ktb+	wn+	hA
3	and	will	3person	write	M.plural	it

Table 4.6: Morphological analysis of the Arabic word /wasayaktubunhA/.

4.3.3 Syntax

Morphology deals with the structure of words. However, the syntax describes the construction of sentences and phrases (Habash 2010). It also determines the syntactic role of words in the context of the sentence. The Arabic computer linguistics account of two main properties of the Arabic language which makes its syntax complex and ambiguous: (i) free word order, and (ii) pronoun dropping (Farghaly and Shaalan 2009). The following descriptions are based on the work presented by Farghaly and Shaalan (2009) and Azmi and Almajed (2015).

Free Word Order Relatively, Arabic has a free word order. The Arabic sentence could be either started with a noun (in this case it is a nominal sentence: جملة إسمية), or a verb (in this case it is a verbal sentence: (جملة فعلية) – see the example below. The free word order is a crucial challenge for many Arabic NLP tasks, as the structure of the sentence can only be clarified through the context (Azmi and Almajed 2015). Although the primary order of Arabic words in CA and MSA sentences is verb-subject-object (VSO), both also allow subject-verb-object (SVO) and object-verb-subject (OVS) orders. The SVO is quite common in newspapers and news-based corpora (Farghaly and Shaalan 2009).

Nominal – جدّي يعمل في التجارة

Verbal – يعمل جدّي في التجارة

Pronoun Dropping The pro-drop property means that the subject may not be explicitly present in the sentence. The subject, for example, can be totally dropped, or substituted with the letter (ت) /t/ as in the sentence "أكلتُ تفاحةً" (Eng: I ate an apple). The (ت) /t/ was attached to the verb to indicate the subject (*I*). This leads to a high degree of complexity for many Arabic NLP tasks. In the case of automatic diacritization, for example, the tools might interpret the noun after the verb as a nominative subject or as an accusative object (Farghaly and Shaalan 2009).

4.4 Arabic NLP Tools

This section presents the Arabic NLP tools that are in use in other parts of the thesis.

MADAMIRA Madamira (Pasha et al. 2014) improves upon its two ancestors: (i) MADA: the Morphological Analysis and Disambiguation for Arabic (Habash et al. 2009), and (ii) AMIRA: the second generation of tools that process Arabic (Diab et al. 2007b) with a Java implementation that is more robust, portable, extensible, and faster. Arabic processing with Madamira includes automatic diacritization, lemmatization, morphological analysis and disambiguation, part-of-speech tagging, stemming, glossing, tokenization, base-phrase chunking, and named-entity recognition. Madamira makes use of fast, linear SVMs implemented using *Liblinear* (Fan et al. 2008).

There are two varieties of Madamira. The first integrates the public version of Arabic morphological analyzer (AraMorph).[13] The second integrates the Standard Arabic Morphological Analyzer (SAMA) and its recommended database (Graff et al. 2009).[14]

FARASA Farasa (Darwish and Mubarak 2016) is an open-source tool, written entirely in native Java. Farasa consists of a segmentation/tokenization module, POS-tagger, Arabic text diacritizer, and dependency parser. Its approach is based on SVM-ranking using linear kernels. Farasa matches or outperforms state-of-the-art Arabic segmenters and diacritizers (Darwish and Mubarak 2016).

[13] `http://www.nongnu.org/aramorph/`

[14] Catalog number LDC2009E73

4.5 Challenges of Arabic NLP

4.5.1 Ambiguity

Although ambiguity is primarily caused by the absence of diacritics (see Chapter 6). Arabic NLP tools are expected to face many challenges due to the high ambiguous nature of the Arabic language. Some aspects of Arabic language contribute to ambiguity, such as the pro-drop structure, complex word structure, lack of capitalization, and minimal punctuation (Farghaly and Shaalan 2009). In the following, we list some relevant examples of ambiguity in Arabic as opposed to different NLP tasks. We only focus on the tasks that are investigated in the context of this thesis, or on the ones that are utilized by the tools to reach or introduce a diacritization. The following overviews are based on the published work by Freihat et al. (2018).

- Segmentation (Tokenization) We prefer to use segmentation as term rather than tokenization. In the field of computational morphology, the term segmentation refers to the process of splitting a word into a list of consecutive morphemes. Typically, one of those corresponds to the word stem, the others are inflectional morphemes (Habash 2010).

  The task of determining whether a morpheme is an affix or a clitic has been considered as confusing (Attia 2007). The ambiguity examples at the segmentation level are many. Among others, the nouns can be disambiguated as: conjunction + pronoun. For instance, the Arabic word وهن /whn/ can be "weakenees", or (هن /hn/ + و /w/) "and they (feminine)". Such cases need to be treated carefully to get a reliable frequency counts for the Arabic words (see Chapter 7).

- Part-of-speech tagging POS tagging is the task of assigning each word in a text (sentence/ corpus) to an appropriate part of speech class or grammatical category (e.g. noun, verb, adjective, adverb, ..., etc) based on both its definition and its context (Elhadj et al. 2014). Arabic makes no distinction between capital/lowercase letters, which are used in specific ways in different Roman script languages (Habash 2010). Therefore, building PoS taggers tools for Arabic is more challenging than in English.

  The ambiguity decreases relatively by correct segmentation, but we still have to solve the ambiguity at the POS tagging level. The nouns can be disambiguated as verbs and vice versa. The Arabic word حمل /Hml/ can be "carrying", "carried", or "pregnancy". A diacritization tool might operates a POS tagger in the underline to disambiguate such challenging cases.

- Lemmatization An important preprocessing step for many text mining applications (Plisson et al. 2004). It concerns with the process of finding the base form of a word (or lemma) from its inflected forms. Lemma is also known as dictionary form, or citation form, and it refers to all words having the same meaning (Mubarak 2017).

  Here, we list some ambiguity examples inherited by lemmatization, grouped as verb and noun ambiguities. (i) Verb ambiguities: In Arabic, the "weak verbs" have the same form in the active or passive voice cases. Thus, a verbs like /sjl/ (سجل meaning: it was reported) cannot be disambiguated without context. (ii) Noun ambiguities: In Arabic, there are several word forms that denote (different) singular/ plural nouns. The word سحب /sHb/, for example, denotes the singular noun /saHob/ (سَحْب meaning: dragging) as well as the plural noun /suHub/ (سُحُب meaning: clouds).

We have shown these examples because diacritics is the only way to distinguish between the items that share the same bare form. Additional relevant examples will be listed with the help of diacritics in Chapter 6. For more information about the aforementioned types of ambiguity, the readers are suggested to see the work by Freihat et al. (2018).

4.5.2 Diglossia

As described by Al-Sobh et al. (2015), the term diglossia was introduced for the first time in 1930 by a French Arabis scholar (*diglossie* in French). Ferguson (1959) transferred the term to English and defined it as:

> "a relatively stable language situation in which, in addition to the primary dialects of the language (which may include a standard or regional standards), there is a very divergent, highly codified (often grammatically more complex) superposed variety – the vehicle of a large and respected body of written literature either of an earlier period or in another speech community – that is learned largely by means of formal education and used for most written and formal spoken purposes but is not used by any sector of the community for ordinary conversations. The superposed variety is the High (H) variety and the regional dialect is the Low (L) variety".

Linguistically, Arabic is often featured by a complex diglossia situation (Diab and Habash 2007). In this situation, Classical Arabic is considered as a high variety, MSA is considered as a more recent variety of CA (occupying an intermediate position), and the regional dialects are considered as low varieties of MSA (Farghaly and Shaalan 2009). Table 4.7 shows an example of diglossia at word level (CA vs. MSA vs. Arabic dialects). It is important to notice that Egyptian dialect is the closest to CA more than the MSA does. Similarly, diglossia has also existence at character level, as in the case of MSA consonants (ج) or (ق) that have different pronunciation based on the regional background (Aboelezz 2010). For a more detailed discussion of diglossia, see Albirini (2016).

English	CA	MSA	Gulf	Syro-Lebanese	Egyptian	Maghreb
Throat	الحلقوم	حنجرة	حلج	زور	حلق	قرجوطة

Table 4.7: The English word *Throat* along with its variants in CA, MSA and Arabic dialects.

Without any doubt, handling diglossia situation for Arabic requires a special attention because the Arabic NLP researchers are required to develop a new solution for each variety of Arabic. Among others, this include phonology, morphology, automatic diacritization etc. For the context of this thesis – i.e. Arabic lexical recognition test, not only we have to avoid selecting any dialectal word for the test but also we have avoid nonwords that have occurrence in colloquial Arabic.

4.6 Arabic Corpora

The Arabic corpora can be divided into two categories: diacritized and non-diacritized corpora.

4.6.1 Diacritized Corpora

Generally, the currently available diacritized corpora are limited to classical texts (usually religious or Arabic poetry), such as the Holy Quran, RDI corpus, and Tashkeela (Zerrouki and Balla 2017) on the one side, and newswire corpora, such as the Arabic Penn Treebank (ATB) from the Linguistic Data Consortium (LDC) on the other side. Below, we summarize the diacritized Arabic corpora. It is important to notice that the Arabic Penn Tree Bank (ATB) and WikiNews summaries are based on the published work by Darwish et al. (2017).

- **Quran**: The small diacritized Quranic corpus is part of Tanzil[15] project. It contains more than 78 thousand tokens that comes in a UTF-8 encoded text file. The file has no Arabic punctuation marks, and every Quranic verse appears in a separate line.
- **RDI**: The corpus was collected by the RDI[16] company for use in the field of automatic diacritization. It is composed of diacritized texts, which are mainly gathered from classical Arabic books with a small percentage from contemporary Arabic writing (modern books). Overall, it contains 20 million tokens. Later on in chapters 6 through 8, the experiments are based on the subset of modern books, a collection of 12 books from the late 1990's.
- **Tashkeela**: The corpus contains more than 60 million diacritized tokens (Zerrouki and Balla 2017). It is a collection of 84 Islamic religious heritage books. As in our published work (Hamed and Zesch 2017a), the books used to be provided in HTML format, encoded in CP1256 Windows Arabic. Nowadays, Tashkeela can be obtained as plain Arabic text. It can be downloaded under GPL license.[17]
- **ATB**: A large body of the previous work on automatic diacritization relied on using the Arabic Penn Tree Bank. The LDC's ATB consists of distinct newswire stories collected from different news agencies and newspapers, including the Agence France-Presse (AFP), Al-Hayat, and An-Nahar newspapers (Maamouri et al. 2004, 2006, 2009). It contains about 1 million tokens. Although ATB is invaluable for many Arabic NLP tasks, among others, POS tagging and parsing, it is sub-optimal for the task of automatic diacritization (Darwish et al. 2017).
- **WikiNews**: A collection of 70 WikiNews articles (mostly from 2013 and 2014) that cover a variety of themes: politics, economics, health, science and technology, sports, arts, and culture. The articles are evenly distributed among the different themes (10 per theme). In total, the corpus contains 18,3K diacritized words. The corpus was collected by Darwish et al. (2017) to be used as a test set for diacritization experiments. In spite of its relatively small size, but the WikiNews corpus is probably the most balanced with respect to domain in our experimental set of corpora.

Limitations The limitations of available diacritized corpora can be summarized as follows:

[15] `http://tanzil.net/download/`

[16] `http://www.rdi-eg.com/RDI/TrainingData/`, accessed in December 2017.

[17] `https://sourceforge.net/projects/tashkeela/`

- Religious Text: One source of text that contains redundant words due to the excerptions from the Holy Quran or Hadith books.
- Arabic Penn Treebank: ATB is limited in terms of (i) size: less than 570k tokens and (ii) diversity: 87,160 unique surface forms without numerals. The AFP news corpus, on the other hand, has approximately 765,890 unique tokens (Cole et al. 2001). Moreover, ATB is often featured by inconsistent diacritizations (Darwish et al. 2017).

Arabic is one of the languages that suffer from a lack of resources. It has been noted that the costs of acquiring corpora can prevent some researchers from conducting interesting research (Zaghouani 2014), and in general can impede progress on reproducible research. NLP scientists who are working with under-resourced languages like Arabic, often suffer from a "cold-start" problem (El-Haj et al. 2015).

4.6.2 Non-diacritized Corpora

Here we use freely available non-diacritized corpora, which are much easier to obtain than manually diacritized corpora that are costly to create.

- **Al-Jazeera Corpus**: A collection of files crawled from Al-Jazeera[18] news portal. The corpus contains about 1.2M tokens.
- **Al-Khaleej-2004 Corpus**: A collection of files from Akhbar Al Khaleej[19] newspaper. The corpus contains about 3M tokens.
- **Al-Watan-2004 Corpus**: A collection of files from Alwatan[20] newspaper. The corpus contains about 10M tokens.
- **KACST Corpus**: A collection of files from "KACST Arabic"[21] newspaper. The corpus contains about 2M tokens.
- **Tweets**: A collection of about 10,006 Arabic tweets (Nabil et al. 2015). The corpus contains about 138K tokens.

However, the non-diacritized corpora can be annotated (turned into diacritized) using the best performing (superior) off-the-shelf diacritization tool. Therefore, we evaluate off-the-shelf diacritization tools later in chapters 6 and 7 respectively.

4.7 Chapter Summary

This chapter gave an overview of some important linguistic characteristics of the Arabic language and its NLP. Arabic has some features that have slowed down the progress in Arabic NLP compared to the advancements in English and other European languages. These features include the following: the absence of capitalisation, inflectional and derivational morphology, and the omitted diacritics. In any NLP task, the absence of diacritics contributes most significantly to ambiguity. However, other types of ambiguity flow to surface with segmentation,

[18] http://www.aljazeera.net/portal

[19] https://sites.google.com/site/mouradabbas9/corpora

[20] The same url as in 19.

[21] https://sourceforge.net/projects/kacst-acptool/files/

part-of-speech tagging, lemmatization, and named entity recognition. As opposed to the challenging features of Arabic language, the automatically generated Arabic LRT is expected to face several NLP challenges more than in English.

Chapter 5

Automatic Generation of English LRTs

> To control and correct for guessing, nonwords are used in LRTs format.
>
> — Harsch and Hartig (2016)

Lexical recognition tests are frequently used for measuring language proficiency. In such tests, the learners need to differentiate between words and artificial nonwords that look much like real words. Lexical recognition tests achieve a quite good approximation of a learner's vocabulary with a relatively small number of test items (Huibregtse et al. 2002). Thus, lexical recognition tests can be quickly finished and usually fit on a single sheet of paper. This is the so called *checklist* format as shown in Figure 3.2. When used in a computerized form, individual items are usually presented in isolation (e.g. LexTALE like tests) in order to minimize context effects.

Generally, the automatic generation of LRTs involves two NLP tasks: (i) a simple task, which is words selection from a corpus and (ii) a complex task, which is nonwords generation. 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. Our ultimate goal is to enable/support the automatic generation of Arabic lexical recognition tests. We divide the task into two subtasks, and consider the English language as starting point to achieve the first milestone. Our goals, in this chapter, are to explore methods for automatically generating good nonwords and to automatically generate word-like nonwords which enables repeated automated testing. We compare different ranking strategies and find that our best strategy (a specialized higher-order character-based language model) creates word-like nonwords. We evaluate our generated 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.

The chapter is organized as follows: Section 5.1 starts by reviewing the algorithms for nonwords generation, mainly for English. In section 5.2, we present our approach to generate nonwords for vocabulary proficiency testing. This is followed by the experimental setup in section 5.3, where we want to find the best-ranking strategy. The results and the nonword generated using the best-ranking strategy are presented in section 5.4. In order to test how well our generated nonwords work in a lexical recognition test, we conduct a user study in section 5.5. Finally, we summarize the chapter in section 5.6.

5.1 State-of-the-Art Nonwords Generation

There are two main paradigms for creating nonwords (either manually or automatically): (i) to start from a known word and change it using letters or diacritics substitution approach to get a nonword, or (ii) to use smaller units (letters, syllables) to construct a larger nonword string. This section reviews the previous computational methods for generating nonwords. The first paradigm was followed by only one approach, the English Lexicon Project[1] (Balota et al. 2007), whereas the second paradigm was followed by the ARC nonword database (Rastle et al. 2002), WordGen (Duyck et al. 2004), LINGUA (Westbury et al. 2007), and Wuggy (Keuleers and Brysbaert 2010).

Here, we provide a brief description of these different tools and shed light on the advantages and disadvantages. The following summaries are mainly based on the published work by Balota et al. (2007), Rastle et al. (2002), Duyck et al. (2004), Westbury et al. (2007), and Keuleers and Brysbaert (2010) respectively.

5.1.1 English Lexicon Project

The nonwords were created using letters substitution approach, where they constructed a nonword database by manually changing one or more letters starting with known English words. For example, ‘boy’ becomes ‘poy’. The ELP’s goal is to provide a nonwords database for psycholinguistic research.

ELP’s database was created for lexical decision task experiments (Balota et al. 2007). In such experiments, a string of letters (either a word or a nonword, e.g., ‘Flirp’) are being shown to the participants. They were asked to press one button - labeled with ‘Yes’ - if the string is a word and another button - labeled with ‘No’ - if the string is a nonword.

This leaded to a large database containing 40,481 mono- and multisyllabic nonwords (the majority is monosyllabic), along with a search engine that facilities online access to this database[2].

5.1.2 ARC Nonwords Database

ARC Nonwords Database, or simply ARC DB is a web-based psycholinguistic resource[3] for nonwords that complements the existing MRC psycholinguistic database (Coltheart 1981).

The ARC DB contains more than 358,500 nonwords, which follow the phonotactic and orthographic rules of (Australian) English. However, it does not rank them according to their quality.

The items can be selected from the ARC DB, based on a wide variety of properties that have a theoretical importance for reading investigation. The ARC DB provides only information for monosyllabic nonwords.

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

[2] The same url as in 1

[3] http://www.maccs.mq.edu.au/~nwdb/

5.1.3 WordGen

Duyck et al. (2004) introduced WordGen as software for generating stimuli items (words and nonwords). WordGen is complementary to both the MRC and ARC as it is useful for generating English multisyllabic nonwords. The WordGen algorithm relies on lemma databases in estimating bigram frequencies. It makes use of the CELEX lemma databases for English (Baayen et al. 1995) and Lexique for French (New et al. 2004).

Researchers refered to some major drawbacks that affect the quality of nonwords generated by WordGen: (i) on the one side, Loth (2011) mentioned that WordGen does not handle the special characters in languages beyond English properly, which affects the summed bigram frequency . For example, the German umlauts: ä, ö and ü can be ignored or underestimated because the tool ignores two bigrams in a given noun such as “Glück”[4]. (ii) on the other side, the time needed to generate nonwords with WordGen increases rapidly with the length of the nonword (Keuleers and Brysbaert 2010).

Overall, WordGen supports properties that are similar to the ones we use for ranking (see Section 5.2), 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.

5.1.4 LINGUA

Westbury et al. (2007) introduced LINGUA, the language-independent neighbourhood generator of the University of Alberta.

LINGUA’s nonword generation process is based on a Markov chain model. However, Westbury et al. (2007) did not provide any further details about the algorithm used for nonwords generation process nor the computational mechanism was clarified. Also, the actual properties of the generated nonword strings are not accessible to the user.

LINGUA ensures that all nonword n-grams occur in an existing word. Similarly, the distribution of n-gram frequencies in the generated nonwords resembles the features of the language in the source corpus based on either position-specific or unspecific counts of the n-grams.

Hence, the tool provides a good possibility for generating typical and pronounceable nonwords, but the user does not have the possibility to manipulate the output in a controlled way. LINGUA is not capable to produce a language independent measure of word-likeness.

5.1.5 Wuggy

Wuggy is a pseudoword generator particularly geared towards making nonwords for psycholinguistic experiments. It builds on WordGen but introduces syllable template to build nonwords that more closely resemble a certain word. The Wuggy algorithm was intended to resolve the timing problem found in WordGen. Thus, the Wuggy algorithm has the built-in restriction: only elements originating from words with n syllables are used to generate sequences of n syllables.

A limitation of the Wuggy algorithm is that it does not generate the pronunciations for orthographic pseudowords. This means that Wuggy cannot indicate whether a word is a pseu-

[4] It means fortune.

dohomophone[5], such as “keap"[6].

5.1.6 Summary

All the approaches discussed here are more geared towards psycholinguistic research (lexical decision tasks), by 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 automatically conducted.

5.2 Generating Nonwords

In this section, we describe our approach to generate nonwords for vocabulary proficiency testing. 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 ‘word-likeness’.

5.2.1 Candidate Generation

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.

5.2.2 Candidate Ranking

Here, 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). Figure 5.1 shows an example of two nonwords (*milc*, nh-size = 5) and (*mirk*, nh-size = 3). Thus, *milc*, is ranked higher than *mirk*.

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

[5] A nonwords that is pronounced identically to a real word.

[6] Example is taken from (Taft and Russell 1992).

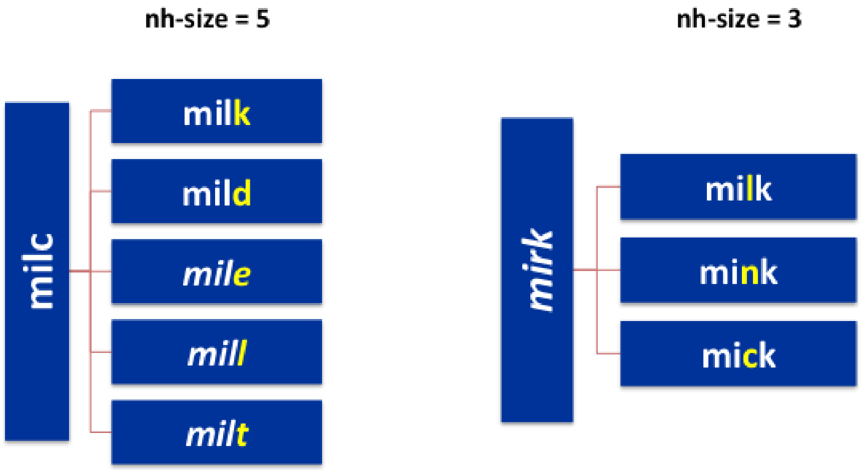


Figure 5.1: Candidate ranking using neighborhood size (nh-size).

German to you even if you don't speak German[7]. 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 assign to each word its probability as returned by the language model. This gives equal probability to all character n-grams as shown in Table 5.1.

LM	Probability (P)
1-gram	$P(golay) = P(g) + P(o) + P(l) + P(a) + P(y)$
2-gram	$P(golay) = P(go) + P(ol) + P(la) + P(ay)$
3-gram	$P(golay) = P(gol) + P(ola) + P(lay)$

Table 5.1: The probability of *golay* in simple character language model (LM).

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 overlapping parts: *start*, *middle*, and *end*. Figure 5.2 shows an example of our split. For each part, we separately train and apply a position-specific character language model. Table 5.2 updates the probability of *golay* as obtained from this model.

[7] It means *generous* in English.

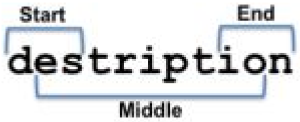


Figure 5.2: Example for position specific splitting. Each part is scored with its own character language model.

LM-PS	Probability (P)
1-gram-PS	$P(golay) = P(g)_s + P(o)_m + P(l)_m + P(a)_m + P(y)_e$
2-gram -PS	$P(golay) = P(go)_s + P(ol)_m + P(la)_m + P(ay)_e$
3-gram -PS	$P(golay) = P(gol)_s + P(ola)_m + P(lay)_e$

Table 5.2: The probability of *golay* in position-specific character language model (LM-PS).

5.3 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.

5.3.1 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.

As evaluation metric, we are utilizing average precision (AP) from information retrieval. Table 5.3 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.

5.3.2 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

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

	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 5.4: Average precision of ranking strategies

the results, we repeat the experiment 100 times (generating new random nonwords every time) and report mean average precision values.

5.4 Results

Table 5.4 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

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 5.5: 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.

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 5.5, 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.

5.5 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.

5.5.1 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

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

12 letters[8] and a corpus frequency 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 5.6.

5.5.2 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[9] (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)[10] 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 Criteria As we have seen in Section 3.3, there are several possible methods to score LRTs. We only want one combined score for word and nonword performance - in order to avoid test-wiseness effects, e.g. students answering that they know all the words. For each participant, we compute the test score using the scoring scheme introduced for LexTALE, as it

[8] 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.

[9] https://moodle.org

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

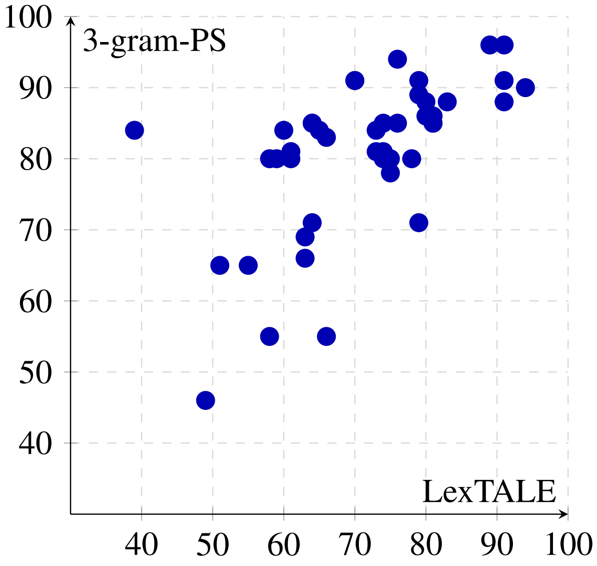


Figure 5.3: Participants' scores on original LexTALE test vs. the test generated by our approach. Original scoring function.

turned out to yield the best results (Lemhöfer and Broersma 2012). In order to yield a single score, the two recall values are averaged as in Equation 3.2.

5.5.3 Study Results

Characteristics of the Population We recruited 80 participants from two German universities (University of Duisburg-Essen and Rhein-Waal University of Applied Sciences), 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 score according to formula 3.2 and then compute Spearman correlation ρ between the resulting score vectors for both tests. We obtain a correlation of 0.68 and Figure 5.3 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 recall for correctly detecting words vs. correctly rejecting nonwords in Table 5.7. We see that the recall for words is almost the same for both tests (.70 vs. .73), while our nonwords are much easier to recognize (.90 recall) compared to LexTALE (.75). This indicates that our nonwords do have lower quality compared to the LexTALE nonwords, as we suspected in Section 5.4. 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.

According to Pellicer-Sánchez and Schmitt (2012), including the nonwords into the scoring might not be necessary at all. Let us see how well participants recognize words, while the nonwords are only distractors. If we drop the nonword part from Equation 3.2, we can directly use the recall on words a_w as the test score. We obtain a correlation is 0.70 (compared to 0.68

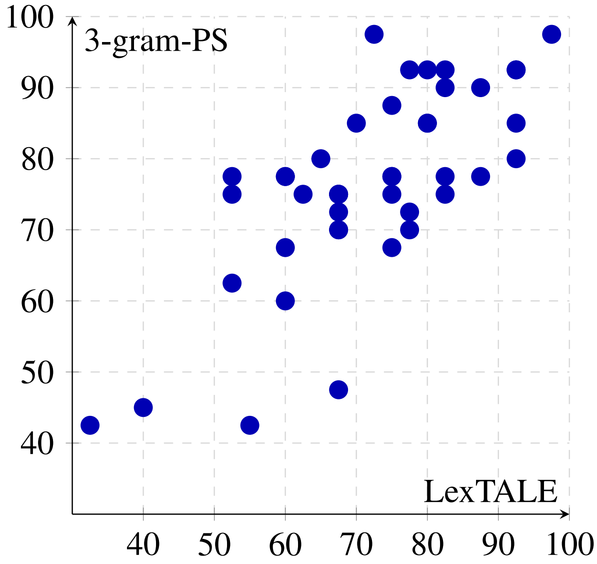


Figure 5.4: Participants' scores on original LexTALE test vs. the test generated by our approach. Adapted scoring function using only words.

LexTALE		3-gram-PS	
R_w	R_{nw}	R_w	R_{nw}
.70	.75	.73	.90

Table 5.7: Recall of student responses for words a_w and nonwords a_{nw}.

from above when taking also nonwords into account). Figure 5.4 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.

5.5.4 Self-Assessment

Lemhöfer and Broersma Lemhöfer and Broersma (2012) provide a partial mapping from the test score to CEFR 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 CEFR levels. In 73% of all cases, both test agree on the same level (with a 33% chance of random agreement).

CEFR 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, compared to 49% for our test showing again that both tests behave quite similar.

5.6 Chapter Summary

In this chapter, we have tackled the task of automatically generating nonwords for lexical recognition tests with focus on the English language. We showed that character language models can be used to distinguish low and high-quality nonwords, and that higher-order models incorporating position-specific information worked the best.

Evaluating the generated nonwords in a user study showed that our approach yields test scores that are highly correlated with the scores obtained from an established lexical recognition test (the LexTALE). The study also showed that the difficulty of the nonwords has little effect on how well words are recognized.

The nonwords are intended to act as distractors forcing students to make mistakes on the words. Therefore, the scoring scheme cannot be simplified as indicated by Pellicer-Sánchez and Schmitt (2012) to ignore nonword performance because the test has no validity after that.

With our experiments, we showed that lexical recognition tests for English can be fully automatically created.

Chapter 6

Automatic Diacritization

> “The most problematic aspect of diacritics is their optionality."
>
> — (Habash 2010)

LRTs are corpus-based assessments that make use of words counts in a huge corpus. To obtain a reliable frequency count for Arabic words, we typically need a large set of diacritized text. It has been noted that the costs of acquiring diacritized MSA corpora can prevent some researchers from conducting interesting research (Zaghouani 2014). Besides to this, the currently available diacritized MSA corpora are limited in size, and domain (Darwish et al. 2017). For these two reasons, we are going to create our own diacritized resources by utilizing off-the-shelf automatic diacritization tools. To get to our target, we conduct a comparative study to benchmark these diacritization tools.

Automatic diacritization is the task of restoring missing diacritics in languages that are usually written without diacritics like Arabic; or in languages that have diacritically marked characters in their orthography like Dutch, German, Hungarian, Lithuanian, or Slovene (Acs and Halmi 2016). The challenge is that in Arabic the same word written without diacritics have different meanings depending on their diacritization, which can only be resolved by the context and proper knowledge of the grammar (Rashwan et al. 2011). For instance the Arabic word علم /Elm/.

Restoring diacritics is an important task, as diacritized texts are crucial for many Natural Language Processing (NLP) applications, including automatic speech recognition (Zitouni et al. 2006; Ananthakrishnan et al. 2005), statistical machine translation (Diab et al. 2007a), text-to-speech (Shaalan et al. 2009), text analysis, information retrieval (Azmi and Almajed 2015), and the normalization and analysis of social media texts (Čibej et al. 2016). Diacritized text is also important at the early stages of language learning and for second language (L2) learners.

Although there is a large body of research on the topic, only very few tools are freely available, and it is still unclear what performance level can be expected in a practical setting.

We aim at a fully reproducible comparison and will thus only include tools that are freely available and can be integrated into our comparison pipeline. To the best of our knowledge, there are currently only two tools that fulfill these requirements: *Madamira* (Pasha et al. 2014) and *Farasa* (Darwish and Mubarak 2016). There exist some additional tools like Mishkal (Afifi and Annabi 2012), which is only available as a web service, and ArabicDiacritizer (Rebai and BenAyed 2015), which only works in a Windows environment. Additionally, both tools limit the size of the input text and cannot be easily integrated in our Java-based comparison framework. For the same reasons of ensuring reproducibility, we only use training and test data that is publicly available without license fees.

In this chapter, we conduct a comparative study between the available tools for diacritization using a reasonable amount and variety of test data in two evaluation modes: strict and relaxed. While the strict mode expects the diacritics to be exactly the same as in gold standard text, the relaxed mode normalizes the texts (output and gold standard) to hold a specific (smaller) ratio of diacritics. Thus, the relaxed mode does not punish a tool that only provides partial diacritization. In order to put the results into perspective, we implement two strong baselines: a dictionary lookup system and one based on character-based sequence labeling. The first baseline labels each word using the diacritized form that appears most often in the training set. The second baseline treats diacritization as a sequence classification problem using conditional random fields (CRF). We report the error rates for the baselines and state-of-the-art systems using diacritized text from Classical Arabic (Quran and Tashkeela corpora) and contemporary writing (RDI corpus) in both evaluation modes.

The chapter is organized as follows: In section 6.1, we provide a linguistic background. In section 6.2, we briefly describe the state-of-the-art Arabic diacritization. This is followed by section 6.3 by describing our experimental setup. In section 6.4, we shed the light on the experimental results. We provide a qualitative analysis in section 6.5. Finally, we summarize the chapter in section 6.6.

6.1 Linguistic Background

Languages based on the Arabic script usually represent only consonants in their writing and do not mark the short vowels (Belinkov and Glass 2015). The Arabic script الخط العربي is written from right to left and contains two classes of symbols for writing words: letters and diacritics (Habash and Rambow 2007; Habash 2010). We start this section with a high-level presentation of diacritics, which is necessary for the automatic diacritization task.

6.1.1 Diacritics

The diacritics are optional. If present, they appear as small strokes that are placed above or below the letter, such as ـَ ـُ ـِ (Fatha دَ /da/, Damma دُ /du/, Kasra دِ /di/).

Diab et al. (2007a) group these diacritical marks into three categories: vowel, Nunation, and Shadda (gemination). The vowel diacritics refer to the three short vowels (Fatha ـَ /a/, Damma ـُ /u/, and Kasra ـِ /i/) and a diacritic indicating the absence of any vowel (Sukun) (Bouamor et al. 2015). The set of Nunation or Tanween diacritics is comprised from three items: Tanween Fath, Tanween Damm, and Tanween Kasr. In a sense, the Nunation diacritics look like a doubled version of their corresponding subset of diacritics (Fatha, Damma and Kasra) from short vowels (Habash 2010). They are named in Arabic as such: Fathatan, Dammatan, Kasratan (dual feminine nouns are used to indicate two Fathas, two Dammas and two Kasras respectively). Phonologically, one can think of Nunation sound as a short vowel followed by a non-written sound of the Arabic consonant letter (ن) /n/. For example, (دٌ) is pronounced /dun/ and transliterated using Buckwalter encoding (Buckwalter 2004) as /duN/. The gemination mark (Shadda) is a consonant-doubling diacritical mark, e.g. (دّ) /d∼/. Shadda can be combined with diacritics from the other two categories, which results in a total of thirteen diacritical marks. For instance, (دُّ) /d∼u/ and (دٌّ) /d∼uN/ are two examples where the Shadda is combined with Damma and Tanween Damm respectively.

There are general rules for diacritizating Arabic text. Shaddah and Sukun can be generally attached to any word letter, only one exception. For example, Shaddah and Sukun cannot follow a word-initial letter, whereas Tanween appears only at word-final position (Elshafei et al. 2006). To be more specific, Shaddah cannot be attached on a word-initial letter alone (it must be combined with another diacritic). Table 4.1 exemplifies the shapes of diacritics in conjunction with the Arabic letter (د) /d/.

Automatic diacritization is challenging because some diacritics vary to indicate semantic differences, and some vary to play syntactic conditions (case-related) (Habash 2010). Functionally, diacritics fall into two types: lexical and inflectional diacritics (Diab et al. 2007a). The lexical diacritics are dedicated to distinguish between two lexemes; for example, /kAtib/ (كاتِب), meaning /writer/, and /kAtab/ (كاتَب), meaning "to correspond". The inflectional diacritics are dedicated to distinguish different inflected forms of the same lexeme. For example, the final (last-letter) diacritic in /kitaAbu/ (كِتَابُ), meaning "book," is *Damma* to indicate the nominative case (nominal subject or verb subject) and the final diacritic in /kitaAba/ (كِتَابَ) is *Fatha* to indicate the accusative case (verb object) of the same word – see the example below.

nominal subject – كِتَابُ الرياضيات صعب

verb object – درسْتُ كِتَابَ الرياضيات

In unicode, the diacritics are presented as additional characters (Abandah et al. 2015). Therefore, the diacritized word is longer than the non-diacritized word. For example, the diacritized word /Eal∼ama/ (عَلَّمَ) has seven unicode characters, whereas the bare form /Elm/ (علم) has only three.

6.1.2 Diacritization Levels

The level of diacritics refers to the number of diacritical marks presented on a word to avoid text ambiguity for human readers. Even in non-diacritized newswire text, 1.6% of all words have at least one diacritic indicated by their author to guide the reader with disambiguation (Habash 2010) (as cited in Habash et al. 2016). In the following, we are based on the classification provided by Ahmed and Elaraby (2000), who grouped the diacritization levels into three levels (full, half, and partial):

Full All the letters are given appropriate diacritics. This applies to classical Arabic (CA), as in religion-related books, and at early stages of language learning, such as in children's books.

Half Only the morphological-independent letters are diacritized. In other words, all the letters of a word, except those that depend on the syntactic analysis of the word, are diacritized. For example, the word /wldh/ (ولده), one of the meanings "his son" consists of two clitics /wld+h/ = (ه) + (ولد), i.e. the stem /wld/ and the possessive pronoun /h/ as suffix. With the half diacritization, it would be written like (وَلَده) instead of (وَلَدُهُ). This means that the diacritic was dropped from the pronoun /h/ (ه) (morphology-dependent) and from the stem last letter /d/ (د) (syntactic).

Partial Any other setting where one letter or a subset of letters is diacritized. While studying the impact of diacritization on statistical machine translation, Diab et al. (2007a) proposed to divide this level into four sub-levels for use with inflectional and lexical diacritics. A special case of partial diacritization is to drop the short vowels and Sukun. For example, the short vowel is dropped from the letter that precedes a long vowel with similar sound like when *Fatha* is dropped from a letter if followed by an *Alef* (ا). Additionally, the Arabic definite article ال has only two diacritization possibilities depending on the preceding letter. The *Alef* is always diacritized with *Fatha*, and the *Lam* (ل) either has Sukun or has no diacritics.

6.1.3 Ambiguity

Writing Arabic without diacritics introduces three types of ambiguity (Azmi and Almajed 2015). The first is part-of-speech (POS) tagging ambiguity (Maamouri et al. 2006). This is

Type	Bare Form	Diacritized	Gloss / Transliteration
POS	علم	عِلْم	Science / Eilom
		عَلَم	Flag / Ealam
		عَلِمَ	He knew / Ealima
		عُلِم	It was known / Eulim
		عَلَّمَ	He taught / Eal∼ama
Syntactic	مدير البنك الجديد	قابَلَتُ مُديرَ البَنكِ الجَديدَ	the new bank manager / mudyra Albanki Aljadyda
		قابَلَتُ مُديرَ البَنكِ الجَديدِ	the manager of the new bank / mudyra Albanki Aljadydi
Structure	ولي	وَلِي	and for me / waliy
		وَلِيّ	a pious person favored by God / waliy∼

Table 6.1: Types of ambiguity caused by missing diacritics

the case with the words that have the same spelling and POS tag but a different lexical sense, or words that have the same spelling but different POS tags and lexical senses (homograph ambiguity) (Farghaly and Shaalan 2009). Second, there is ambiguity on the grammatical level (syntactic ambiguity). Sentences and phrases can be interpreted in more than one way, and diacritics are the only means to resolve ambiguity (Maamouri et al. 2006). The third is internal word structure ambiguity, such as when Arabic words are segmented in different ways. The agglutination property of Arabic might produce a problem that can only be resolved using diacritics. Table 6.1 summarizes the aforementioned types of ambiguity with excerpted examples from (Metwally et al. 2016; Farghaly and Shaalan 2009).

6.2 State-of-the-Art on Arabic Diacritization

In this section, we present the diacritized datasets usually used for evaluation and then give an overview of the results on different corpora that have so far been obtained using the standard evaluation metrics.

6.2.1 Datasets

In this subsection, we are referencing the diacritized datasets presented in Section 4.6. A summary of these datasets is shown in Table 6.2.

6.2.2 Evaluation Metrics

In the literature, two standard evaluation metrics are used almost exclusively to measure systems performance (Rashwan et al. 2011; Said et al. 2013). It can either be expressed in terms of error rates on the character or on the word level. The smaller the error rate, the better the performance.

Corpus	Description	Availability	# of tokens
Quran	Religious	Free	78 000
RDI	Religious/Modern	Free	20 000 000
Tashkeela	Religious	Free	60 000 000
ATB	News	Commercial	1 000 000
WikiNews	News	Free	18 300

Table 6.2: Overview of diacritized corpora.

DER Diacritization Error Rate (DER) is the proportion of letters which are incorrectly labeled with diacritics. The following assumptions are made: (i) each letter or digit in a word is a potential host for a set of diacritics, and (ii) all diacritics on a single letter are counted as a single binary choice in strict mode. The DER can be calculated as follows:

$$DER = (1 - \frac{|T_S|}{|T_G|}) \cdot 100 \tag{6.1}$$

where $|T_S|$ is the number of letters assigned correctly by the system, and T_G is the number of diacritized letters in the gold standard text.

WER Word Error Rate (WER) is the percentage of incorrectly diacritized white-space delimited words. In order to be counted as incorrect in strict mode, at least one letter in a word must have a diacritization error. All words are counted, including numbers and punctuation.

While the diacritization techniques work relatively well on lexical diacritics (located on word stems), they are much less effective for inflectional diacritics (typically at word-final position) (Habash et al. 2007a). In most cases, the last letter indicates the case ending. However, in some cases as with plural masculine nouns جمع المذكر السالم and dual masculine and feminine nouns المثنى the suffixes substitute the diacritics. The suffixes are added to the word to indicate case and number. For example, the suffixes (ون) or (ان) are added to the word to indicate plural masculines and dual masculine or feminine nouns in accusative case respectively. However, the suffix (ين) is added to the word to indicate plural masculines and dual masculine or feminine nouns in nominative and dative cases. Assigning the correct case can often only be decided using a wider context, thus diacritization tools usually perform worse on the last letter compared to the other positions in the word (Habash et al. 2007a). It is thus usual to also report a variant of the above two mentioned metrics that ignore the last letter (assumed to have no syntactic diacritics), denoted as **DER-1** and **WER-1**.

6.2.3 Evaluation Modes

When comparing multiple tools, we distinguish two different evaluation modes:

Strict Mode Whenever a letter has a set of diacritics in the gold standard text, a diacritization tool is expected to predict this set exactly. This evaluation mode is most often used and gives an advantage to tools providing full diacritization.

Relaxed Mode This evaluation mode gives an advantage to tools that only output diacritics when being confident about the results. This might be useful for half or partial diacritization settings, e.g. the tools that drop the default diacritics. This is not so useful for other settings, e.g. full diacritization in children books.

Diacritization	**Word Letters**					
Tool	**A**	**l**	**E**	**r**	**b**	**y**
Gold	a	o	a	a	i	~u
Tool 1	-	o	a	a	i	~u
Tool 2	-	o	a	a	-	~u
Tool 3	-	-	-	a	-	~u
Tool 4	-	-	a	a	-	~u
in relaxed evaluation?	No	No	No	Yes	No	Yes

Table 6.3: The normalization of diacritics for comparison in relaxed evaluation mode.

In order to provide a fair comparison between multiple tools, the relaxed evaluation mode only takes into account cases where all tools under consideration return a diacritic for a given letter. Table 6.3 shows an example using the word العربي /AlErby/.

6.2.4 Overview of Diacritization Results

The work on Arabic diacritization goes back quite a long time (El-Sadany and Hashish 1989) and many different approaches have been proposed including hidden Markov model (Elshafei et al. 2006), n-gram language models (Hifny 2012; Alghamdi et al. 2010), statistical machine translation (Schlippe et al. 2008), finite state transducers (Nelken and Shieber 2005), maximum entropy (Zitouni et al. 2006), and deep learning (Rashwan et al. 2015; Abandah et al. 2015; Belinkov and Glass 2015).

Additionally, many researchers have proposed to improve classification with morphological analysis (Habash and Rambow 2005; Rashwan et al. 2011; Bebah et al. 2014; Metwally

			All Diacritics		Ignore Last	
Test Corpus	**Size** (10^3)	**Approach**	**DER**	**WER**	**DER-1**	**WER-1**
ATB (Parts 1–3)	144	(Nelken and Shieber 2005)	12.8	23.6	6.5	7.3
	52	(Zitouni et al. 2006)	5.5	18.0	2.5	7.9
	52	(Habash and Rambow 2007)	4.8	14.9	2.2	5.5
	613	(Schlippe et al. 2008)	4.3	19.9	1.7	6.8
	116	(Schlippe et al. 2008)	4.7	21.9	1.9	8.4
	16	(Alghamdi et al. 2010)	13.8	46.8	9.3	26.0
	52	(Rashwan et al. 2011)	3.8	12.5	1.2	3.1
	37	(Abandah et al. 2015)	2.7	9.1	1.4	4.3
	52	(Metwally et al. 2016)	-	13.7	-	-
Quran	1	(Elshafei et al. 2006)	4.1	-	-	-
	76	(Abandah et al. 2015)	3.0	8.7	2.0	5.8
Tashkeela	1902	(Hifny 2012)	-	8.9	-	3.4
	272	(Abandah et al. 2015)	2.1	5.8	1.3	3.5
Tashkeela+RDI	199	(Bebah et al. 2014)	7.4	21.1	3.8	7.4
WikiNews	18	(Pasha et al. 2014)	5.4	19.0	1.9	6.7
	18	(Rashwan et al. 2015)	4.3	16.0	1.0	3.0
	18	(Belinkov and Glass 2015)	7.9	30.5	3.9	14.9
	18	(Darwish et al. 2017)	3.5	12.8	1.1	3.3

Table 6.4: Performance of Arabic diacritization systems grouped by test corpus

et al. 2016) and the standard n-gram language model. A recent approach by Darwish et al. (2017) employed a Viterbi decoder and SVM-rank to properly guess words diacritization.

Comparison Table 6.4 gives an overview of the reported results from the literature. The results are grouped by the corpus that was used for testing in order to allow for a fair comparison. There is a major drawback with these reported results: they do not follow a well-established framework for testing. For example, most numbers are still not directly comparable because they were obtained using different test sets. Moreover, some works used a fixed test set without performing any cross-validation, which further limits the weight that should be put on those numbers. The only exception to this is the last block of results, where Darwish et al. (2017) compared their system with other systems using the *WikiNews* test set. Under this controlled setting, their system outperforms all other systems regarding DER and WER. If we ignore the case-endings, the Rashwan et al. (2015) system performs best.

As most of the systems from the literature are not freely available, we have no way of directly comparing them. In this chapter, we establish a comparative study that only includes the systems and corpora that are freely available in a controlled settings.

6.3 Experimental Setup

In this section, we present our experimental setup: used data, baselines, diacritization tools, and evaluation metrics.

The experiments were carried out using DKPro TC, the open-source UIMA-based framework for supervised text classification (Daxenberger et al. 2014). The baseline experiments were conducted as ten-fold cross-validation, reporting the average over the ten folds.

6.3.1 Datasets

Table 6.5 shows the statistics for the experimental sub-datasets (punctuation marks are not counted). All the experiments use a general setup for test sample-size: 78K, 100K, and 100K drawn from the Quran, RDI company, and Tashkeela corpora respectively.

		# words	**∅ chars**	**Words /**
ID	**Corpus**	(10^3)	**per word**	**sentence**
Q	Quran	78	4.25	12.6
T	Tashkeela	100	4.11	14.7
R	RDI	100	4.47	34.1

Table 6.5: Statistics of corpora sub-datasets used in this study.

Data Preprocessing The Quran text requires no special preprocessing. However, the files from Tashkeela and RDI contain Quranic symbols like the Dagger Alif (a small Alif quite common in Quranic Arabic (Dukes and Habash 2010)) or English letters. In order to prepare those corpora for training and testing purposes, the following preprocessing steps are performed:

(i) convert them from HTML to plain text files that have one sentence per line, (ii) clean the files by removing the Quranic symbols and words written in non-Arabic letters, and (iii) normalize the Arabic text by removing extra white spaces, and Tatweel (elongation or Kashida (Habash 2010)) symbols that look like horizontal strokes (—). As indicted by Habash (2010), Tatweel is used to stretch words (force horizontal justification) and to indicate prominence (emphasis) in languages that lack to Capital letters (e.g. Aarabic). Figure 6.1 shows the word بسم without and with Tatweel.

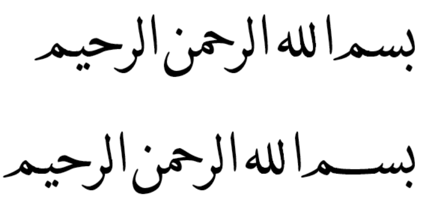


Figure 6.1: The word بسم without and with Tatweel.

6.3.2 Baselines

We implemented two baselines: a simple dictionary lookup approach and a sequence labeling approach.

Dictionary Lookup This baseline labels each word with the diacritized form that appears most often in the training corpus. Words that are not found in the dictionary are not diacritized.

Sequence Labeling We treat diacritization as a sequence labeling problem and propose a baseline solution using conditional random fields (Lafferty et al. 2001). Given a sentence (set of non-diacritized words) separated using white-space delimiters, each word in the sentence is a sequence of characters, and we want to label each letter with its corresponding labels from the diacritics set $D = (d_1, ..., d_N)$. We represent each word as an input sequence $X = (x_1, ..., x_N)$ where we need to label each consonant in X with the diacritics that follow this consonant. Note that an Arabic letter has a maximum of two diacritics, and if it has two, then one of them is always Shadda. The Shadda might accompany all diacritics except Sukun. This means that the machine learning algorithm is required to predict a letter label from a set of 14 labeling possibilities (including the 'no diacritic' option). Thus, in order to diacritize sequence X, we must find its labeling sequence Y (usually of word length) derived from D.
A word might have more than one valid labeling. The word /ktAb/ (كتاب) represented as (k,t,A,b), can be labeled with $Y_1 = (i,a,o,u)$ or $Y_2 = (i,a,o,a)$ resulting in the diacritized words /kitaAobu/ and /kitaAoba/ respectively.

Our features are character n-grams language models (LMs) in sequence labeling approach. The features extractor selects the character-level features relevant to diacritics from annotated corpora. It collects the diacritics on previous, current and following character and up to the 6th character.

Note that the out-of-vocabulary (OoV) rate of this approach is zero as it is able to provide a sequence of diacritics for arbitrary unknown words.

6.3.3 Diacritization Tools

To the best of our knowledge, the only diacritization tools that can be tested on large corpora and are easily integrated with Java frameworks are *Madamira* and *Farasa* that were early introduced in the thesis.

Madamira Madamira diacritizer (Pasha et al. 2014) makes use of fast, linear support-vector machines (SVMs) implemented using *Liblinear*.

Madamira was trained on the training portion of ATB (parts 1, 2 and 3). As it was mentioned in Chapter 2, there are two varieties of Madamira that integrates: (i) the public version of Arabic morphological analyzer (AraMorph), and (ii) the Standard Arabic Morphological Analyzer (SAMA) and its recommended database.

As per our cooperation with other researchers, the diacritization experiments are carried out using the SAMA enabled version of Madamira *v2.1*. Madamira was used to diacritize the test sequences from the three corpora. As the resulting diacritized text is encoded using

Corpus	Approach	OoV	All Diacritics		Ignore Last	
		rate	DER	WER	DER-1	WER-1
Quran	Dict. Lookup	11.8	19.7	27.5	16.1	16.8
	Sequence Labeling	0.0	21.4	28.3	9.0	19.9
	Madamira	3.4	21.1	36.7	15.4	20.9
	Farasa	0.3	**12.2**	**19.0**	**8.9**	**9.5**
RDI	Dict. Lookup	13.3	26.6	31.8	19.7	22.5
	Sequence Labeling	0.0	24.9	37.0	15.4	22.4
	Madamira	2.1	17.8	28.4	13.1	14.2
	Farasa	0.1	**10.5**	**15.7**	**6.7**	**7.6**
Tashkeela	Dict. Lookup	13.4	26.9	32.2	19.9	22.7
	Sequence Labeling	0.0	24.9	37.0	15.7	22.3
	Madamira	2.2	17.9	28.6	13.1	14.2
	Farasa	0.1	**10.6**	**15.9**	**6.8**	**7.7**

Table 6.6: Error rates in strict evaluation mode. The “OoV” rate refers to the ratio of tokens that were not diacritized by the system.

Buckwalter transliteration, it is necessary to decode it into Arabic text. We compare the mapped Arabic text with a gold standard sequence and then calculate the different metrics.

Farasa Farasa diacritizer (Darwish and Mubarak 2016) is an open-source diacritizer, written entirely in native Java. Farasa’s approach is based on SVM-ranking using linear kernels.

6.4 Results

We now report the results of our diacritization experiments using first ‘strict’ and then ‘relaxed’ evaluation.

6.4.1 Strict Evaluation

Table 6.6 gives an overview of our evaluation results in *strict* mode. The results are grouped by the corpus that was used for testing. Note that the OoV column refers to the ratio of tokens that got “No Analysis” and thus no diacritization by the system.

In general, the error rates are rather high. With keeping in mind that the reported results are non-comparable, none of the methods (including the two well-known state-of-the-art systems) comes even close to the numbers in Table 6.4. It is likely that many approaches do not use strict evaluation mode when reporting results, even if it is the most comparable setup. When indirectly competing with other published results, the numbers obtained in that way are just not competitive.

Looking at individual results, Farasa outperforms all other methods under all metrics. For the remaining three approaches, there is no clear trend, but it should be noted that the baselines perform surprisingly well even if they make no real attempt at resolving ambiguity. Sequence labeling doesn’t take context into account and the dictionary lookup makes a majority class decision for each ambiguous token. We suspect that many tokens within a domain are not ambiguous and the repetitious nature of the religious texts increases the effect.

Table 6.6 also shows the out-of-vocabulary rate for each approach. As expected, the dictionary lookup baseline has a rather high rate and sequence labeling has no out-of-vocabulary tokens at all, because it always returns one of the possible diacritization patterns. For all corpora, Farasa has a lower OoV rate than Madamira.

When looking into individual OoV examples, we find that in some cases the tools do not return any analysis. However, in some cases they change the input token instead of just adding diacritics. For example in one case in Madamira, the verb /rawaAhu/ (رَوَاهُ), meaning "narrated by" is changed into /ruwaAp/ (رُوَاة), meaning "narrators". Another example is the passive verb /yusotavonaY/ (يُسْتَثْنَى), meaning "to be excluded" that is changed into the present tense verb /yasotavoniy/ (يَسْتَثْنِي), meaning "excludes". In both examples, the last letter is changed into a very similar, but different form. We see a similar behavior in Farasa, where in some examples a word containing two adjacent *Lam* (ل) letters (with Shadda on the second *Lam*), where the first *Lam* is a preposition. In this case, there is an additional Alif letter introduced between the two Lam letters. For example, the word (لِلَّه) /lil∼ah/ (l + Allah) is transformed into (لِالهِ) /liAlhi/ – i.e. (l + Alh).

	Diacritics per letter			Diacritized letters per word		
Approach	**Quran**	**RDI**	**Tashkeela**	**Quran**	**RDI**	**Tashkeela**
Gold	.84	.83	.83	.78	.77	.77
Dict. Lookup BL	.84	.84	.82	.78	.77	.77
Seq. Labeling	.82	.78	.78	.77	.74	.74
Madamira	.55	.59	.61	.51	.54	.56
Farasa	.58	.58	.61	.55	.54	.58

Table 6.7: Average number of diacritics per letter and average number of diacritized letters per word.

In Table 6.7, we show the average number of diacritics per letter as well for the gold standard and all systems used in our experiments. It shows that Madamira and Farasa both assign about the same amount of diacritics on average, but substantially fewer than the gold standard. This means that both tools are especially punished by the strict evaluation. These findings motivate us to repeat the evaluation using the *relaxed mode*.

6.4.2 Relaxed Evaluation

Table 6.8 shows the results in relaxed mode, where we only take into account cases where all tools under consideration return a diacritic for a given letter. As expected, the error rates drop substantially, but not evenly for all approaches. In order to better show the improvement (decrease in error rates) obtained by switching from strict to relaxed evaluation mode, we report the relative change between both modes in Table 6.9. It can be clearly seen that this switching improves the tools performance in general. Sometimes, a tool is making a dramatical change, such as the dictionary lookup baseline under the DER and DER-1 metrics.

Looking again at the error rates in Table 6.8, relaxed evaluation mode reveals that Farasa

Corpus	Approach	All Diacritics		Ignore Last	
		DER	**WER**	**DER-1**	**WER-1**
Quran	Dict. Lookup	**7.3**	24.0	**3.2**	15.6
	Seq. Labeling	15.1	22.0	7.6	13.5
	Madamira	14.5	26.4	10.2	15.6
	Farasa	7.8	**14.0**	5.0	**6.8**
RDI	Dict. Lookup	10.1	27.9	**3.4**	16.7
	Seq. Labeling	16.7	28.0	12.0	13.6
	Madamira	12.5	20.4	8.6	10.2
	Farasa	**8.3**	**13.8**	5.0	**5.1**
Tashkeela	Dict. Lookup	10.1	28.1	**3.3**	16.7
	Seq. Labeling	24.0	35.4	15.0	22.0
	Madamira	12.4	20.3	8.5	10.1
	Farasa	**8.3**	**13.9**	5.0	**5.1**

Table 6.8: Error rates in relaxed evaluation mode

Corpus	Approach	All Diacritics		Ignore Last	
		DER	**WER**	**DER-1**	**WER-1**
Quran	Dict. Lookup	63	13	80	7
	Seq. Labeling	29	22	16	32
	Madamira	31	28	34	25
	Farasa	36	26	44	28
RDI	Dict. Lookup	62	12	83	26
	Seq. Labeling	33	24	22	39
	Madamira	30	28	34	28
	Farasa	21	12	25	33
Tashkeela	Dict. Lookup	62	13	83	26
	Seq. Labeling	4	4	4	1
	Madamira	31	29	35	29
	Farasa	22	13	26	34

Table 6.9: The relative change (in %) between the strict and relaxed evaluation modes

is still performing better than Madamira in all cases, but for the DER and DER-1 metrics the dictionary lookup baseline is close or even better. The big difference between DER and WER performance for the dictionary lookup approach is most likely explained by errors in the inflectional diacritics that are impossible to resolve without looking at the context. However, that such a simple approach performs so well is surprising and shows that there is still a lot room for improvement in the area of automatic diacritization.

6.5 Qualitative Analysis

As most of the systems from the literature are not freely available, we have no way of directly comparing our results with those approaches unless they have the same settings. There is still

Error	Error	Annotator 1		Annotator 2	
Category	Subcategoy	Madamira	Farasa	Madamira	Farasa
Form/Spelling	Shadda	2	3	2	3
	Tanween	6	6	6	6
Morphology	Partial-Inflection	1	1	0	1
	Full-Inflection	2	0	2	0
Grammar	Active-Passive Voice	2	2	2	2
Diacritization	Missing Short Vowel	6	0	5	0
	Confused Short Vowel	1	5	1	4
Overall		20	17	18	16

Table 6.10: The annotated WERs subcategories.

a gap between our experimental results in relaxed mode and some of the reported published results in Table 6.4. Part of the gap can certainly be attributed to differences in the corpora. To see how the systems are performing, we also conducted a small diacritization experiment that only involves the best baseline (dictionary lookup), Madamira, and Farasa. We conduct a simple experiment using a blind MSA test set, a sample with 94 non-diacritized words (crawled from the internet). It was then diacritized using dictionary lookup (which was trained with RDI), Madamira (SAMA-enabled), and Farasa. We gave the resulting diacritized text to two Arabic teachers with appropriate experience to conduct the evaluation.

To look at the kinds of errors we were getting, the annotators were asked to identify the incorrectly diacritized words using word error rates (WERs) metrics because it is easy to manage for the volunteer teachers. Additionally, they were asked to state the reason if a diacritization produced by Madamira or Farasa was incorrect. For that purpose, we are using an error classification scheme developed for Arabic learner corpora (Abuhakema et al. 2008).

Form/Spelling This category refers to errors caused by Shadda (consonant doubling), or Tanween (nunation).

Morphology This category refers to a correct lexical item, but wrong case ending, e.g. Kasra instead of Fatha.

Grammar This category refers to errors caused by changes in grammatical role, e.g. active or passive voice (المبني للمعلوم و المجهول).

Diacritization This category refers to errors caused by incorrect, missing or redundant short vowels (i.e. lexical diacritics).

Table 6.10 shows the distribution of error categories as reported by the annotators. The inter-evaluator agreement for the annotated WER (using Cohen's kappa) is almost perfect with values of .93 and .96 for Madamira and Farasa respectively. The majority of the mistakes are due to form/spelling and diacritization errors. In the form/spelling category, both tools make a lot of Tanween errors. This is to be expected, as it has been reported that the diacritization tools work relatively well on lexical diacritics, but that they are much less effective for case-ending diacritics (Habash et al. 2007a). In the 'Diacritization' category, we observe a quite different behavior. Madamira has more missing vowels, i.e. it seems to rather not return a diacritic than to get it wrong. Farasa is on the opposite side of the trade-off with no missing short vowels, but almost as many confused short vowels.

6.6 Chapter Summary

In this chapter we established a framework to compare the state-of-the-art publicly available Arabic diacritizers. The test data was drawn from the Quran, Tashkeela, and RDI corpora. Under controlled settings, we compared two strong baselines and two well-known systems: Madamira and Farasa. The error rates are reported in strict and relaxed evaluation modes to ensure fair comparison. We found that Farasa is outperforming Madamira in both evaluation modes, but that in relaxed mode the simple dictionary lookup baseline is surprisingly strong. In general, our error rates are much higher than the ones reported in the literature and we currently have no satisfying explanation for the difference. We are making our evaluation framework publicly available in order to foster additional research in this area and to allow for more approaches to be tested under reproducible conditions.

There is still a large space for improvement in the area automatic diacritization. Instead of using the traditional machine learning algorithms, we recommend the recurrent neural networks (RNNs) as an alternative. As per the accuracy of RNNs achieved so far (Abandah et al. 2015), RNNs are expected to achieve superior results that might be hard to obtain using traditional approaches, such as rule-based or SVMs.

Chapter 7

Arabic Frequency Counts

In Chapter 6, the comparative study showed that Farasa is outperforming Madamira and the baselines in both evaluation modes. Here we investigate if the automatic diacritization can be used to obtaining reliable frequency counts for Arabic words.

Statistical natural language processing relies on corpora as a source of token frequency counts. Obtaining reliable frequency counts is challenging in Arabic due to the specific morphological structure and the omitted diacritics in almost all modern MSA writings. In this chapter, we explore the effects of diacritization on Arabic frequency counts. We want to see how severely this situation affects the resulting language models. For this purpose, we analyze the properties of the few available manually diacritized corpora and use them to evaluate automatic diacritization tools. We then apply the best performing tool on non-diacritized texts and investigate the properties of the resulting language models.

The chapter is organized as follows: Section 7.1 starts by narrating the importance of frequency counts in NLP. In section 7.2, we provide a refreshment by presenting a short Arabic linguistic background. This is followed by corpus analysis in section 7.3, where we analyze some existing corpora with diacritized Arabic text. We investigate the reliability of automatic diacritization tools in section 7.4. Leveraging non-diacritized corpora is discussed in section 7.5. Finally, we summarize the chapter in section 7.6.

7.1 Frequency Counts

Obtaining reliable frequency counts from corpora is in the core of many natural language processing applications, e.g. language identification (Baldwin and Lui 2010), text categorization (Cavnar et al. 1994), language modeling (Hamed and Zesch 2015), and vocabulary assessment (Ricks 2015). According to (Ricks 2015), an accurate frequency counts are a prerequisite for most widely used vocabulary assessment formats.

In the simplest form, we want to be able to answer the question whether a given lemma A is more frequent than some lemma B. In more complex scenarios, we want to represent a surface form with a semantic representation like embeddings (Levy and Goldberg 2014). In morphology-poor languages like English, both scenarios are relatively easy to tackle by relying on surface forms. However, even then we might conflate many possible forms, e.g. due to part-of-speech ambiguity (*the bank* vs. *to bank*) or sense ambiguity (*bat/animal* vs. *bat/club*). To solve this problem, sophisticated solutions have been proposed like distinguishing embeddings by word sense, syntactic category, or semantic role (Li and Jurafsky 2015; Tu et al. 2017).

When moving to languages beyond English, we encounter additional problems. For example in agglutinative languages like Turkish, heavy morphological processing is necessary to obtain reliable counts for base forms (Aksan and Yaldır 2010). Another example is Arabic

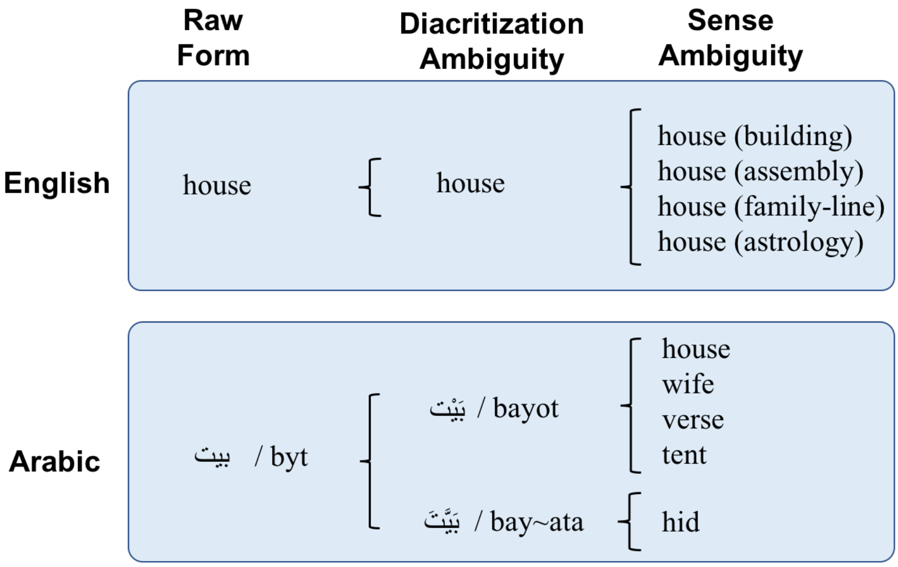


Figure 7.1: Sources of lexical ambiguity in English and Arabic.

where a single surface form can have multiple syntactic categories or meanings (Merhben et al. 2009). This effect is much stronger than in English and is caused by the absence of diacritics in everyday Arabic writing.

Figure 7.1 compares the situation in English and Arabic. English uses relatively few diacritics, so for most tokens there is no diacritization ambiguity. There are some exemptions like *mate/maté* or *expose/exposé*, but they are rather rare. In Arabic, diacritics play a much more central role. For example, the Arabic token بيت */byt/* has diacritizations like بَيْت /bayot/ and بَيَّت /bay∼ata/. As can be seen in the last column in Figure 7.1, this issue is not to be confused with sense ambiguities that exist in English and Arabic on top of the diacritization ambiguity. For example, the diacritized form بَيْت /bayot/ still has different meanings like *house*, *verse*, etc.

The problem of diacritization ambiguity is especially troubling for corpus-based studies, as diacritics are usually only used for specific settings like language teaching or religious texts (Belinkov and Glass 2015). In everyday writing, only letters (roughly corresponding to consonants and long vowels) are written while diacritics are omitted and the corresponding short vowels need to be inferred by the reader (Habash 2010).

In this chapter, we use the few existing manually diacritized Arabic corpora to characterize the extent of the problem, i.e. how much the distribution of diacritized forms for a given surface form actually differ between corpora. We then conduct experiments with existing automatic diacritization tools in order to examine whether we can derive reliable diacritized frequency counts from plain, non-diacritized corpora.

Form	Gloss	Postfix	Suffix	Lemma	Prefix	Antefix
العرب /AlErb/	the Arabs	–	–	عرب	ال	–
الطلبات /AlTlbAt/	the requests	–	ات	طلب	ال	–
وطلباتنا /wTlbAtnA/	and our requests	نا	ات	طلب	و	–
ليحدثونكم /lyHdvwnkm/	to talk with you	كم	ون	حدث	ي	ل

Table 7.1: Types of of clitics in Arabic.

7.2 Linguistic Background

7.2.1 Diacritics

We know from Chapter 2 that the Arabic script is comprised of two classes of symbols: letters and diacritics. Although the letters are always written, the diacritics are optional.

7.2.2 Clitics

We introduced the clitics in Chapter 2. The clitics can precede the stem like a prefix (proclitics) or follow the lemma like a suffix (enclitics) (Dixon 2010). Proclitics are clitics that precede the word (like a prefix), e.g. the conjunction (و) "w", and the definite article (Habash 2010). Enclitics are clitics that follow the word (like a suffix), e.g. the object pronoun (هم) "hm" (eng: them).

Table 7.1 contains among others, the word (وطلباتنا /wTlbAtnA/ *and our requests*) consists of four clitics /w+Tlb+At+nA/: (i) the conjunction /w/ as prefix, (ii) the lemma /Tlb/, (iii) /At/ as suffix to indicate feminine plural, and (iv) the possessive pronoun /nA/ as postfix.

In order to obtain a reliable frequency count for the stem /Tlb/, we have to use segmenters and/or lemmatizers to discard such extra clitics. There is well-established research that compares the available Arabic segmenters and lemmatizers. The results of (Darwish and Mubarak 2016) state that Farasa outperforms or matches state-of-the-art Arabic segmenters like QCRI Advanced Tools For Arabic (QATARA) (Darwish et al. 2014) and Madamira (Pasha et al. 2014).

7.3 Corpus Analysis

In this section, we analyze existing corpora with diacritized Arabic text in order to estimate how severe the ambiguity problem caused by omitted diacritics is for corpus-based language models.

7.3.1 Diacritized Datasets

For our analysis, we need a gold standard in the form of manually diacritized corpora. The most widely used is the Arabic Penn Treebank (ATB) (Darwish et al. 2017), which is quite

Corpus Name	**Corpus Domain**	**# of Tokens**	**# of Types**	**Partially Diacritized**
WikiNews	News	28 371	1 333	36%
Quran	Religious	131 427	3 169	22%
RDI	Contemporary Writing	152 378	2 984	28%
Tashkeela	Religious	151 928	3 019	27%

Table 7.2: Summary of the diacritized experimental data used in the corpus analysis after segmentation.

costly. As it has been noted, the costs of acquiring corpora can prevent some researchers from conducting interesting research (Zaghouani 2014) and are in general an impediment to reproducible research. We thus only rely on a set of freely available corpora: WikiNews, Quran, RDI, and Tashkeela that were presented in Section 4.6.

- WikiNews: It contains 18,3K diacritized words.
- Quran: It contains about (no more than) 78K diacritized words.
- Tashkeela: Our experimental data is drawn from a subset of 11 books, containing 4,926K diacritized words.
- RDI: Our experimental data is drawn from a subset of 12 modern Arabic books containing 297K diacritized words.

The WikiNews and Quran corpora are relatively small. Thus, we experiment with the whole WikiNews (18.3K words) and the whole Quran (78K words). On the other hand, RDI and Tashkeela are relatively big. Thus, a subset of 100K words is drawn from RDI as well as from Tashkeela. Table 7.2 provides a summary of the experimental data used for corpus analysis, it worths noting that types and unique tokens are used interchangeably. Note that the number of tokens is reported after segmentation (see next subsection) and thus it is always larger than 87,341, the number of raw tokens (Sayoud 2012) in the Quran corpus.

7.3.2 Pre-Processing

We carry out all analyses using DKPro Core[1], a collection of software components for natural language processing based on the Apache UIMA framework.[2] We normalize the text by removing extra white spaces, Tatweel, Quranic symbols, words written in non-Arabic letters (e.g. Latin script), numbers, and punctuation (including: dashes, brackets, and curly braces). Additionally, Tashkeela and RDI are in HTML format and need to be converted into plain text first.

We then apply the Farasa segmenter (Darwish and Mubarak 2016) to split affixes and clitics. A very common form of clitic is the definite article ال /Al/ (eng: the) that is attached to the noun. For example, the word /rjAl/ (eng: men) occurs one time, whereas the word /AlrjAl/ (eng: the men) occurs seven times.

[1] https://dkpro.github.io/dkpro-core/

[2] https://uima.apache.org/

Quran		Tashkeela		RDI		WikiNews	
Form	**Transliteration**	**Form**	**Transliteration**	**Form**	**Transliteration**	**Form**	**Transliteration**
رَبِّ	/rab~i/	سَلَّمَ	/sal~ama/	سَلَّمَ	/sal~ama/	أَعْلَنَ	/OaEolana/
قُلْ	/qulo/	قَوْلُ	/qawolu/	قَوْلِ	/qawoli/	مُتَّحِدَ	/mut~aHida/
أَرْضِ	/OaroDi/	قَوْلِ	/qawoli/	قَوْلُ	/qawolu/	سَنَ	/sana/
كَفَرُ	/kafaru/	يَدِ	/yadi/	رَضِيَ	/raDiya/	أَفْضَلِ	/OafoDali/
كُنْتُمْ	/kunotumo/	رَضِيَ	/raDiya/	يَدِ	/yadi/	مُنَظَّمَ	/munaZ~ama/
شَيْء	/$ayo’/	شَرْحُ	/$aroHu/	شَرْحُ	/$aroHu/	حَرَكَ	/Haraka/
يَعْلَمُ	/yaEolamu/	بَيْعِ	/bayoEi/	يَلْزَمُ	/yalozamu/	مِصْرَ	/miSora/
قَوْمِ	/qawomi/	ذَكَرَ	/*akara/	مَرَّ	/mar~a/	صِحَّ	/S~iH~a/
رَحْمَنِ	/raHomani/	حَقِّ	/Haq~i/	يَضْمَنُ	/yaDomanu/	شَرِكَ	/$arika/
رَبَّ	/rab~a/	قُلْ	/qulo/	بَيْعِ	/bayoEi/	أَكَّدَ	/Oak~ada/
رَبِّ	/rab~i/	يَجِبُ	/yajibu/	مَسْأَلَة	/masoOalap/	مَرْكَبَ	/marokaba/
يُؤْمِنُ	/yuWominu/	مَرَّ	/mar~a/	قُلْ	/qulo/	كُرَ	/kura/
بِسْمِ	/bisomi/	يَلْزَمُ	/yalozamu/	ذَكَرَ	/*akara/	عِدَّ	/Eid~a/
مُؤْمِنِ	/muWomini/	مَسْأَلَة	/masoOalap/	يَجِبُ	/yajibu/	مُؤْتَمَرِ	/muWotamari/
رَبُّ	/rab~u/	أُنْظُرْ	/AunoZuro/	دَيْنِ	/dayoni/	عَمَلِيَّ	/Eamaliy~a/
حَقِّ	/Haq~i/	يَضْمَنُ	/yaDomanu/	أُنْظُرْ	/AunoZuro/	سَبَبِ	/sababi/
خَلَقَ	/xalaqa/	يَصِحُّ	/yaSiH~u/	أَرْضِ	/OaroDi/	فَتْرَ	/fatora/
أَرْضَ	/OaroDa/	حُكْمُ	/Hukomu/	حَقِّ	/yajuwzu/	مَرَضِ	/maraDi/
جَنَّ	/jan~a/	أَرْضِ	/OaroDi/	أَصَحِّ	/OaSaH~i/	دَوْلِيَّ	/dawoliy~a/
تَعْمَلُ	/yaEolamu/	مُدَّ	/mud~a/	حُكْمِ	/Hukomu/	أَزْهَرِ	/Oazohari/

Table 7.3: The top-20 most frequent Arabic words in four corpora.

7.3.3 Analysis

A first interesting finding is that a lot of tokens in the four corpora are only partially diacritized. This is mainly due to default diacritics where the long vowels (Alif (ا), Waw (و), and Ya (ي)) are usually non-diacritized. One of the most frequent examples is the definite article ال /Al/. In the WikiNews corpus, for instance, the *Alif* letter (ا /A/) is non-diacritized in all the words (3,892 times) that start with /Al/. Moreover, the *Lam* letter (ل) is non-diacritized (has no *Sukun*[3]) in 34% of all cases. We restrict our analysis to the fully-diacritized forms.

In order to characterize the corpora, in Table 7.3, we look at the 20 most frequent tokens

[3] It is a circle-shaped diacritic placed above a letter, it looks like a tiny o.

in each corpus (excluding stopwords).[4] From our investigation, it is clear that we get a mix of lexical and syntactic variation. For example, the *Quran* top-20 include different cases of the word رب like /rab∼u/, /rab∼a/, or /rab∼i/ that indicate nominative, accusative, genitive respectively. Note that the RDI and Tashkeela columns look very similar because both corpora have many excerptions from the Holy Quran and Hadith books. This shows that we might overestimate the problem when taking all diacritics into account. We thus approximate a setting with only lexical diacritics by ignoring the diacritic on the last letter (that usually only serves a syntactic function).

We then compute for each surface token (i.e. without diacritization) the number of diacritized forms that appear in a corpus. The higher that number, the higher the diacritization ambiguity of the surface form. Figure 7.2 displays the results. All distributions follow Zipf's law (Zipf 1950) as is to be expected for the vocabulary of a language (Sorell 2012). There is a surprising amount of surface forms that only appear with one diacritization, but there is still a considerable amount of surface forms with quite high ambiguity. When comparing between the corpora, the religious texts use much more ambiguous forms than WikiNews.

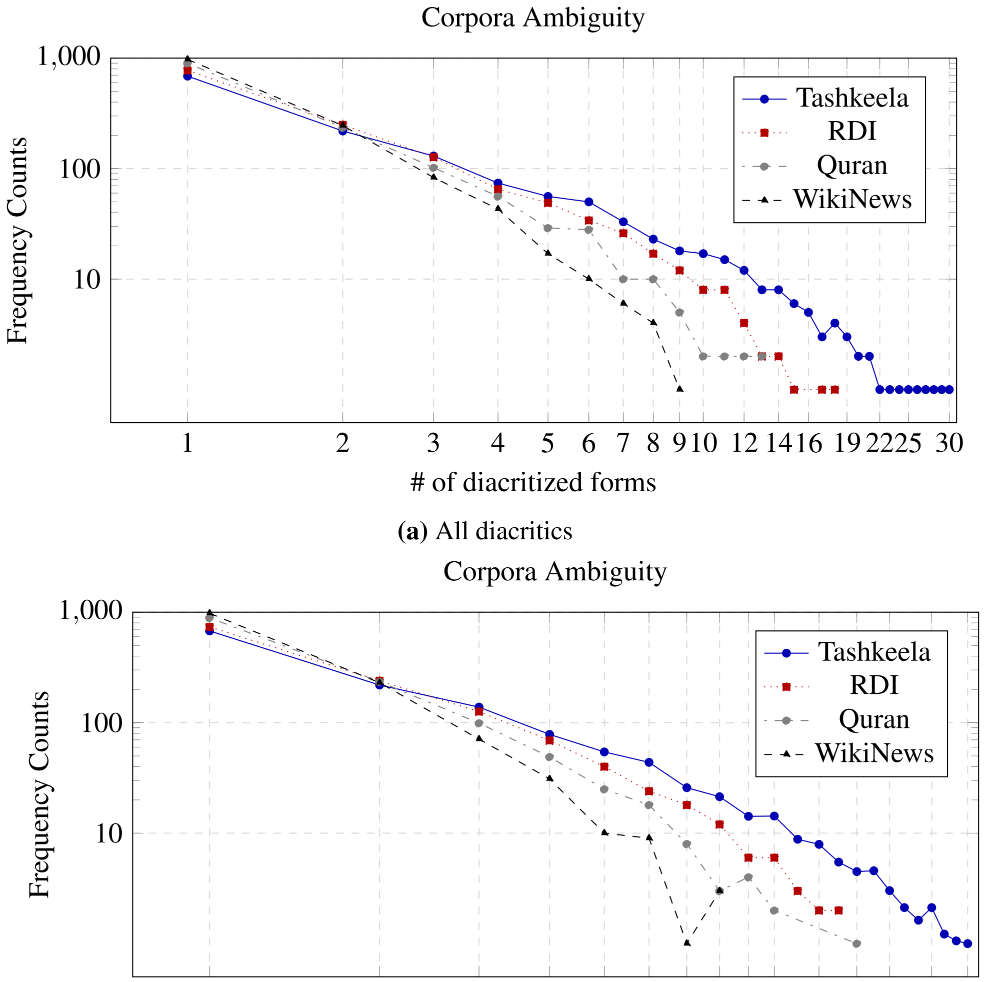
7.4 Reliability of Automatic Diacritization

So far, we have only analyzed the properties of existing, manually annotated corpora and found that there is quite a lot of ambiguity due to diacritization. We now want to determine how reliable this ambiguity is captured by existing automatic diacritization tools. For that purpose, we experiment with two tools (Madamira (Pasha et al. 2014) and Farasa (Darwish and Mubarak 2016)), which were also found to perform well in a comparative study of Arabic diacritization tools (Hamed and Zesch 2017a).

- Madamira: Improves upon its two ancestors MADA (Habash et al. 2009) and AMIRA (Diab et al. 2007b). The Arabic processing with Madamira includes: automatic diacritization, lemmatization, morphological analysis and disambiguation, part-of-speech tagging, stemming, glossing, tokenization, base-phrase chunking, and named-entity recognition. When the text is processed by Madamira, Madamira output file contains among others, diacritization, gloss, cases, num, stem etc. This provided file is well-formed and on a sentence and word level, it can be either in text or xml format. Madamira can provide more than one analysis/disambiguation for the same word. Overall, we noticed that Madamira is pretty slow.
- Farasa: Is an open-source tool for Arabic NLP. Its approach is based on SVM-rank using linear kernels. Farasa consists of a segmentation/tokenization module, POS tagger, Arabic text diacritizer, and dependency parser. The output file provided by Farasa is for the whole input file (it has no indication about sentence nor word number), it is a plain text file (space-delimited) and has no structure as in Madamira, and it contains no gloss. Overall, we noticed that Farasa is pretty fast.

If the tools work well, they will be able to re-construct the same distribution of diacritized forms given a surface form as found in the gold standard. If they do not work well enough, they might just project the learned model on the corpus without taking the context into account.

[4] Stopwords are filtered using a list obtained from: `https://github.com/mohataher/arabic-stop-words`



(a) All diacritics

(b) Without last letter

Figure 7.2: Distribution of statistics of Arabic words ambiguity in the studied corpora.

7.4.1 Experiments & Results

We take the four gold standard corpora and remove all diacritics. We then diacritize the resulting text using Farasa and Madamira. Finally, we segment each token using the Farasa segmenter. We then compare the diacritization results with the gold standard using two highly ambiguous base forms: علم /Elm/ and ذكر /*kr/. Table 7.4 and Table 7.5 show the results, where we use only the fully-diacritized forms. Note that the ‘Form’ column is shown without syntactic diacritics, as we focus on lexical differences. Consequently, counts for forms only differing in case endings are merged regardless of the part-of-speech.

Differences between corpora This analysis allows us another, closer look at the lexical ambiguity in our gold standard corpora. For the religious corpora, the distributions are quite similar. However, the most frequent lexeme is not always the same, e.g. it is /*ikor/ (*prayer*)

		Quran			Tashkeela			RDI			WikiNews		
Form	**Translation**	**Gold**	**Farasa**	**Madamira**	**Gold**	**Farasa**	**Madamira**	**Gold**	**Farasa**	**Madamira**	**Gold**	**Farasa**	**Madamira**
عِلْم / Eilom	knowledge	101	111	65	50	53	30	31	37	17	4	5	6
عَلَم / Ealam	flag	-	-	4	-	-	2	2	-	2	1	1	-
عُلْم / Eulom	flags	-	1	-	-	-	-	-	-	-	-	-	-
عَلِم / Ealim	he knew	26	25	19	16	14	6	12	12	4	-	-	-
عُلِم / Eulim	it was known	-	2	5	10	9	5	10	8	8	-	-	-
عَلَّم / Eal~am	he taught	19	11	2	2	2	-	4	3	-	1	-	-
عَلِّم / Eal~im	teach!	1	1	-	-	-	-	-	-	-	-	-	-
عُلِّم / Eul~im	it was taught	2	1	-	-	-	-	-	-	-	-	-	-

Table 7.4: Frequency counts for the Arabic word علم /Elm/.

		Quran			Tashkeela			RDI			WikiNews		
Form	**Translation**	**Gold**	**Farasa**	**Madamira**	**Gold**	**Farasa**	**Madamira**	**Gold**	**Farasa**	**Madamira**	**Gold**	**Farasa**	**Madamira**
ذِّكْر /*~ikor/	mentioning	16	15	-	8	13	1	2	8	-	3	3	3
ذِكْر /*ikor/	prayer	45	38	5	22	32	3	38	36	2	-	1	-
ذَّكَر /*~akar/	male	5	6	17	9	4	9	7	3	4	-	-	-
ذَكَر /*akar/	he mentioned	11	29	52	79	87	69	70	84	91	4	2	3
ذُكِر /*ukir/	it was mentioned	7	11	-	39	25	-	35	26	8	-	1	-
ذَكَّر /*ak~ar/	he preached	-	-	4	-	-	4	-	-	1	-	-	1
ذُكِّر /*uk~ir/	he was reminded	9	4	-	-	-	-	-	-	-	-	-	-
ذَكِّر /*ak~ir/	to remind	7	-	-	-	-	-	-	-	-	-	-	-
ذْكَر /*okar/	N/A	-	-	-	1	-	-	-	-	-	-	-	-

Table 7.5: Frequency counts for the Arabic word ذكر /*kr/.

for the Quran corpus, but /*akar/ (*he mentioned*) for Tashkeela and RDI. Unfortunately, the WikiNews corpus is rather small and does not allow to draw any such conclusions.

Differences between tools When looking at the tools, Farasa clearly gives better estimates than Madamira. For example, Madamira underestimates the frequency of /*ikor/ (*prayer*) while Farasa comes quite close.

It is worth mentioning that some of the forms have zero counts in all gold corpora, such as the form عُلْم /Eulom/ (eng: flags) located in the 3rd row in Table 7.4.

7.5 Leveraging Non-diacritized Corpora

In this section, we experiment with using non-diacritized plain text corpora as the basis for obtaining frequency counts. We then compare counts between corpora in order to shed light on the level of difference to be expected between corpora from different domains.

7.5.1 Datasets

Here we use freely available non-diacritized corpora that are presented in Section 4.6: Al-Jazeera, Al-Khaleej-2004, Al-Watan-2004, KACST and Tweets. By carefully selecting the type of corpus, we hope to be able to influence the distribution.

We select a random set of 100K tokens from each of the corpora and diacritize the subset using Farasa, as it is the best performing tool in our evaluation in the previous section.

7.5.2 Experiments & Results

Table 7.6 and Table 7.7 compare the frequency counts of the Arabic words علم /Elm/ and ذكر /*kr/ when the sample subsets of 100K words are diacritized using Farasa. We find that some of the surface forms are mostly likely in some domains more than in other domains.

		Religious			**News**					**Social**
Form	**Translation**	**Quran**	**Tashkeela**	**RDI**	**WikiNews**	**Al Jazeera**	**Al-Khaleej**	**Al-Watan**	**KACST**	**Tweets**
عِلْم /Eilom/	knowledge	111	53	37	5	25	8	9	33	33
عَلَم /Ealam/	flag	-	-	-	1	1	1	-	3	2
عُلْم /Eulom/	flags	1	-	-	-	-	-	-	-	-
عَلِم /Ealim/	he knew	25	14	12	-	5	2	2	2	7
عُلِم /Eulim/	it was known	1	9	8	-	-	-	-	1	1
عَلَّم /Eal~am/	he taught	11	2	-	-	1	-	1	1	-
عَلِّم /Eal~im/	teach!	1	-	-	-	-	-	4	-	-
عُلِّم /Eul~im/	it was taught	1	-	-	-	-	-	-	-	-

Table 7.6: Frequency distribution for Arabic word علم /Elm/ as being diacritized by Farasa.

		Religious			News					Social
Form	Translation	Quran	Tashkeela	RDI	WikiNews	Al Jazeera	Al-Khaleej	Al-Watan	KACST	Tweets
ذِّكْر /*~ikor/	mentioning	15	13	8	3	1	-	4	2	3
ذِكْر /*ikor/	prayer	38	32	36	1	4	1	8	5	38
ذَّكَر /*~akar/	male	6	4	3	-	1	-	-	1	1
ذَكَر /*akar/	he mentioned	29	87	84	2	40	11	11	17	17
ذُكِر /*ukir/	it was mentioned	11	25	26	1	5	1	-	2	2
ذَكَّر /*ak~ar/	he preached	-	-	-	-		-	-	-	-
ذُكِّر /*uk~ir/	he was reminded	4	-	-	-	-	-	-	-	-
ذَكِّر /*ak~ir/	to remind	-	-	-	-	-	-	-	-	-
ذْكَر /*okar/	N/A	-	-	-	-	-	1	1	-	-

Table 7.7: Frequency distribution for Arabic word ذكر /*kr/ as being diacritized by Farasa.

Interestingly, some of the surface forms encountered in the automatically diacritized corpus are never found in the dictionary. As per the investigation in the Arabic dictionary, we are unaware of any corresponding meaning for the diacritized Arabic string ذْكَر /*okar/, listed in the last row of Table 7.7. It can be clearly seen that the first letter (ذ) is annotated with Sukun, it looks like ذْ. The appearance of this diacritized-form is due to a diacritized form produced by Farasa, where the ذ letter is annotated with Sukun ذْ. Thus, Sukun is erroneously mapped to the ذْ letter after segmentation.

7.6 Chapter Summary

As per our analysis in this chapter, we showed that diacritics have a significant influence on obtaining reliable frequency counts in Arabic. However, we also showed that a quite good approximation can be obtained by applying automatic diacritization to non-diacritized corpora that are much easier to collect than manually diacritized corpora. By selecting the domain of the source corpus, there is even a bit of control about the resulting distribution of lexemes. Overall, the automatic diacritization can be used to obtaining reliable frequency counts for Arabic words.

The reported results are based on two ambiguous Arabic words. We have to generalize these results by conducting further experiments using additional ambiguous Arabic words and using other full corpora. We recommend to implementing this procedure as a web-interface that enables us to know the number of occurrences (frequency) of a given word (diacritized or non-diacritized form) in a corpus.

Chapter 8

Role of Diacritics in Arabic LRTs

Measuring language proficiency is an important task for educators. Vocabulary size is one part of the overall proficiency that can be used to approximate learning progress or to select a suitable language course (Maskor et al. 2016b). Increasing your vocabulary is an essential component of language learning and is also one of the main conditions in mastering a language (Baharudin and Ismail 2014).

Lexical recognition tests (LRTs) are one of the best-known and most widely used vocabulary assessment formats (Schmitt 2000). The main advantage is their simplicity (Meara and Jones 1987), as participants are just being shown a list of words and asked to say 'Yes' when they know the word or 'No' otherwise. In order to make the task difficult and to avoid cheating, besides real words like *obey*, also nonwords like *nonagrate* are shown. Figure 3.2 shows an example for such a test in checklist format. If all words are checked and all nonwords are not checked correctly, this shows a result for a learner with a good vocabulary knowledge.

So far, we have tackled the Arabic resource creation and the reliable frequency counts for Arabic words. In this chapter, we address some of these challenges by taking a closer look on the design process of Arabic tests and especially the role of diacritical marks that are a defining feature of Arabic. We investigate the role that diacritics play in designing an Arabic lexical recognition test. For that purpose, we conduct a user study that compares an existing non-diacritized test for Arabic (Ricks 2015) with an adapted version including diacritics. We also discuss the NLP-related challenges when aiming to automatically build LRTs with diacritics.

The chapter is organized as follows: Section 8.1 starts by providing some background on Arabic lexical recognition tests. In section 8.2, we describe the role of diacritics in Arabic LRTs. This is followed in section 8.3, where we describe the construction of a diacritized Arabic LRT. We investigate the role of diacritical marks on Arabic LRTs by conducting a user study in section 8.4. The results are discussed in section 8.5. Finally, we summarize the chapter in section 8.6.

8.1 Arabic LRTs

In chapter 4, we looked into the previous work on lexical recognition tests, which has mainly focused on English, and a few other European languages. We find very few studies investigating the lexical recognition tests for Arabic.

Recall that lexical recognition tests are used for measuring the vocabulary size of learners (Cameron 2002). They are based on the assumption that recognizing a word is sufficient for 'knowing' the word, i.e. they only measure the size or breadth of vocabulary knowledge, but not the depth or quality (Anderson and Freebody 1981; Schmitt 2014). However, for many purposes like placement tests or quickly assessing the progress of vocabulary acquisition, lexical recognition tests have been successfully applied.

In order to cover a wide difficulty range, the words to be used in such tests are usually selected based on corpus frequency. The tests additionally use carefully selected nonwords that act as distractors. This is necessary, as otherwise learners could easily game the test by simply pretending to ‘know’ all the words. In a test with a mix of words and nonwords, such a strategy leads to a rather low score.

Lexical recognition tests already achieve a quite good approximation of a learner’s vocabulary with a relatively small number of test items (Huibregtse et al. 2002). Thus, lexical recognition tests can be quickly finished and usually fit on a single sheet of paper. This is the so called *checklist* format as shown in Figure 3.2. When used in a computerized form, individual items are usually shown in isolation in order to minimize context effects as shown in Figure 3.1.

8.1.1 English LRTs

Recall the *Eurocentres Vocabulary Size Test* (Meara and Jones 1990), which is an early example of using nonwords for testing. It uses 150 items – two thirds real words that were selected by frequency, and one third manually-crafted nonwords. Lemhöfer and Broersma (2012) constructed a smaller version of this test called *LexTALE* that can be finished faster. It only uses 40 words (selected by relative frequency in the CELEX corpus (Baayen et al. 1995)) and 20 manually-crafted nonwords. LexTALE scores are validated 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), and Spanish (Izura et al. 2014).

8.1.2 Arabic LRTs

We are only aware of a very limited set of studies on Arabic lexical recognition tests which all use non-diacritized Arabic.

Test of Arabic Vocabulary (TAV) Baharudin et al. (2014) developed the *Test of Arabic Vocabulary* that uses 40 words selected by corpus frequency, but no nonwords. Thus, the test is vulnerable to test-wiseness or overconfidence (just answering ‘yes’ for each item). The following TAV summary is based on the published work by Baharudin et al. (2014).

The Arabic frequency for this word list are take from “Word Count of Modern Arabic Prose” book (Landau 1959). This book contains more than 12,400 Arabic words from a variety of Arabic prose and lists of words collected from the daily newspapers in Egypt. The word items were selected from the most frequent 4000. For every 1000 words, 50 words were randomly selected. The list of items contains a variety of PoS including nouns, verbs and particles. In total, 200 items were selected for the test based on four frequency bands. The items were refined with help of the items difficulty index and items discrimination index (Anastasi and Urbina 1997). Item difficulty is the percentage of people who correctly answer the item. Item discrimination refers to the effectiveness of an item to discriminate between higher-scoring students and lower-scoring ones in a particular test (Aiken 1997). With the help of both indexes, 40 word items were selected to represent 4000 words. The selected words did not include named entities such as names of people, places and others. Overall, it is hard to adopt TAV for repetitive testing as used in formative assessment (Wang 2007).

Arabic Checklist Test Ricks (2015) developed a checklist-format test with 40 words and 20 nonwords (following the format introduced with LexTALE). Words were randomly selected from the most frequent 5000 words in Buckwalter/Parkinson frequency dictionary (Buckwalter and Parkinson 2011), but excluding dialectal words. Nonwords were created using letters substitution approach as inspired by (Stubbe 2012).

8.1.3 Generating LRTs

The automatic generation of LRTs involves two steps: (i) selecting words from a corpus and (ii) generating nonwords. In the past, nonwords have been manually created (Lemhöfer and Broersma 2012). However, for the repetitive testing as used in formative assessment (Wang 2007), nonwords test stimuli need to be generated automatically. As it was shown in chapter 5, we proposed an approach to generate nonwords automatically using character n-gram language models. There, we applied our approach to English, and considered word selection using frequency per million word.

8.2 Effect of Diacritics

We now turn to the role of diacritics in Arabic lexical recognition tests. We argue that they play a crucial role in selecting words as well as designing suitable nonwords.

Usually, learners of Arabic build their vocabulary knowledge from diacritized material. Especially in the early stages, learners might find it difficult or unnatural to recognize words without diacritics. For example, all the textbooks in the series "I Love the Arabic Language" are diacritized[1]. Thus, a non-diacritized lexical recognition test might systematically underestimate the performance of low proficiency learners.

Selecting Words Arabic text is mostly written non-diacritized, i.e. without any diacritical marks, except for religious texts, educational texts, and some poetry (Habash 2010). Figure 8.1 shows the non-diacritized and the diacritized versions of the sentence "*The boy drank milk*".

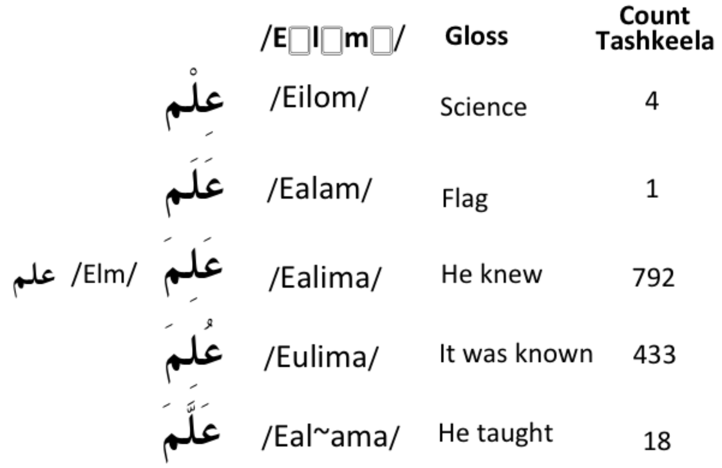
w/o diacritics **شرب الولد حليبا**

w/ diacritics **شَرِبَ الوَلَدُ حَليبَاً**

Figure 8.1: Example sentence with and without diacritics (eng: *The boy drank milk*).

The analogy with English is imperfect, but in a sense the situation would be similar to presenting someone the string *str* and expecting them to be able to determine whether the intended English word is *star*, *stir*, *suitor*, *sitar*, or *store* depending on the context (Saigh and Schmitt 2012). So, in a lexical recognition test, when we ask a learner if she knows the 'word' *str*, we are actually asking whether she knows any of the words from the list above which is quite an imprecise question.

[1] http://www.noorart.com/school_section/i_love_the_arabic_language_arabic_curriculum



		/E□l□m□/	Gloss	Count Tashkeela
	عِلْم	/Eilom/	Science	4
	عَلَم	/Ealam/	Flag	1
علم /Elm/	عَلِمَ	/Ealima/	He knew	792
	عُلِمَ	/Eulima/	It was known	433
	عَلَّمَ	/Eal~ama/	He taught	18

Figure 8.2: Examples of diacritized forms of the Arabic word علم /Elm/. Frequency counts are based on the Tashkeela corpus.

The lack of diacritics usually leads to considerable lexical and morphological ambiguity (Zaghouani et al. 2016). Following an example from Maamouri et al. (Maamouri et al. 2006), we show in Figure 8.2 a non-exhaustive list of diacritizations of the Arabic word علم /Elm/. We hypothesize that the difficulty to recognize a non-diacritized word is actually determined by the relative probability of its most-frequent diacritized form. We also argue that this effect goes way beyond the related issue of word senses for English lexical recognition tests, where showing a word like *tree* also tests whether one knows the most frequent sense (*a tall perennial woody plant*) and not one of the more specialized ones (*data structure in computer science*). As we can see in Figure 8.2, the ambiguity introduced by non-diacritized text includes phonological, morphological, and syntactic cases (Zaghouani et al. 2016).

In order to determine the most likely diacritization, we can check the frequency based on diacritized corpora. We use a subset of 11 books of the Tashkeela corpus (Zerrouki and Balla 2017). Because of the religious nature of the texts in this corpus, the counts for *Science* and *Flag* are very low, while *He knew* and *It was known* are two orders of magnitude higher. Thus, learners of Arabic that mainly read religious texts will be able to recognize the diacritized form of /Elm/ meaning *He knew*, but not *Science*, while no such distinction can be made in the non-diacritized version of the test.

Designing Nonwords So far we have only discussed existing word forms, but diacritics might also play a crucial role in designing better nonwords. Arabic diacritization is an orthographic way to describe Arabic word pronunciation (Zaghouani et al. 2016). We hypothesize that the non-diacritized nonwords are probably easier than the diacritized ones. The diacritized nonwords can distract better with closely related, especially if they are labeled with pronounceable diacritics.

8.3 Constructing a Diacritized Arabic Test

In this section, we discuss the linguistic and technical challenges that occur in the two steps (word selection and nonword generation) of automatically constructing a lexical recognition test.

8.3.1 Selecting Arabic Words

Selecting words for lexical recognition tests is mainly based on frequency counts. However, obtaining reliable frequency counts is more challenging in Arabic than in English due to complex morphology and the issue of diacritization.

Morphology Arabic is a morphology rich language and its words are highly inflected and derived (Aqel et al. 2015). For example, the word (/wktAbnA/, 'وكتابنا', and our book) consists of three clitics w + ktAb + nA: (i) the conjunction article /w/ as prefix, (ii) the stem /ktAb/, and (iii) the possessive pronoun /nA/ as suffix.

So in order to get a reliable frequency count for the lemma *ktAb* (engl. *book*), we have to use segmenters and lemmatizers to discard such extra clitics (Habash 2010). Fortunately, there are tools such as Farasa (Darwish and Mubarak 2016) that are specifically designed for that task.

Another example of morphology standing in the way of frequency counting is the pervasiveness of the definite article /Al/ (ال) that is directly attached to a word. For example, in the arTenTen corpus (Belinkov et al. 2013), which comprises 5.8 billion words, the frequency of the word (/Tfl/, 'طفل', child) is 4,557, whereas the frequency of the same word along with the definite article (/AlTfl/, 'الطفل', the child) is 15,325.

Automatic Diacritization We suggest using the diacritized lemmas for a better frequency count. However, as there is only a limited number of rather small corpora with manually annotated diacritics (Al-Sulaiti and Atwell 2006) one has to fall back to automatic diacritization in order to obtain reliable frequency counts of diacritized word forms.

Although there is a large body of research on the topic (Azmi and Almajed 2015; Metwally et al. 2016), only very few tools are freely available and it is still unclear what performance level can be expected in a practical setting. It is especially unclear whether existing tools will simply project the distribution of diacritics found in the training corpus to new data or if they generalize well enough to be useful for the purposes of designing lexical recognition tests. We are not aware of any research that actively targets this question.

8.3.2 Designing Arabic Nonwords

In a lexical recognition test, a good nonword acts as a distractor, i.e. it is similar enough to real words that it forces test-takers to be careful about their answers. However, nonwords should of course not be a valid word from the vocabulary of a language.

Anderson and Freebody (Anderson and Freebody 1983) discuss two methods for creating nonwords in English: (i) pseudo-derivatives, which entails adding a prefix or suffix to a real

MSA	Dialectal	Nonword
حريق	حريء	هريء

Table 8.1: Nonwords and diglossia.

word, so ‘loyal’ becomes ‘loyalment’; and (ii) letter substitution, where one or two vowels and/or consonants are substituted in a real word, so ‘boy’ becomes ‘poy’. As we saw in Subsection 4.2.4, substituting letters is not too complicated due to the peculiarities of the Arabic alphabet, especially the minimal pairs.

Dealing with Similar Shapes There are two types of symbols in the Arabic script for writing words: letters and diacritics (Habash and Rambow 2007). We introduced the Arabic letters in Chapter 4. We know that Arabic letters typically consist of two parts: letter form and letter mark. In total, the Arabic alphabet has 19 letter forms (Habash 2010). The letter marks fall into three sub-types: the dots, the short Kaf and the Hamza (ء).[2] The Arabic alphabet features a significant number (including minimal pairs) of letters that differ only in the position (e.g. ج ح خ) or number (e.g. ب ت ث) of dots placed around the letter form. Thus, nonwords can be easily created in Arabic. For instance, the nonword /jArx/ (جارخ) and the real word (/xArj/, ‘خارج’, outside) can be only differentiated by the placement of their dots (points). In analogy to English, this is more subtle than *fish* versus *shif* (Ricks 2015).

Dealing with Similar Sounds An orthography is a specification of how the sounds of a language are mapped to/from a particular script (Habash 2010). Some phonemes of Arabic language have emphatic counterparts. The learners in a type of phonemic L1 transfer, have a tendency to conflate these L2 phoneme pairs. Special attention should be paid to ensure that nonwords could not be coined, by means of substituting one or more of these “confusable” pairs of letters with similar sounds like ت /t/ and ط /T/ or د /d/ and ض /D/.[3]

One nonword, for example, that is hard to be rejected is /ItfAl/[4] (إتفال), as it can be easily confused with the real word (/⟩TfAl/ or /OTfAl/, ‘أطفال’, children). Another good example for hard rejection is /DfDE/ ضفضع, which should be easily confused with the real word (/dfDE/, ‘ضفدع’, frog).

Dealing with Diglossia The Arabic language has at least three forms: Classical Arabic, Modern Standard Arabic (MSA), and Dialectal Arabic (Farghaly and Shaalan 2009). This leads to situations where a speaker of Arabic might use two varieties of the language. This kind of situation is what is linguistically known as diglossia (Ferguson 1959).

Consider, for example, the MSA word (*Hryq*, ‘حريق’, fire) and the dialectal word (*Hry’* حريء) which is the Syro-Lebanese counterpart of the MSA word. As nonwords are often generated by means of swapping one letter, *hry’* (هريء) could be generated by swapping the ح /H/ with ه /h/ in Syro-Lebanese instance. The three instances are shown together in

[2] Habash (2010) noted that “the dots should not be confused with Hebrew Niqqud (dots), which are optional diacritics comparable to Arabic diacritics. Arabic dots and other letter marks are all obligatory.”

[3] In almost cases, the capital letters are used to refer to a stressed letter.

[4] Transliterated using Safe Buckwalter.

MSA	Gulf	Egyptian	Maghreb
جبهة	يبهة	قورة	فرنت

Table 8.2: The dialectal varieties for MSA word /jbhp/ (جبهة).

Table 8.1. However, this is problematic, as the dialectal word (*Hry'* حريء) is well known in the Levantine area. Thus such a nonword would be much easier for Syro-Lebanese speakers and much more difficult for others, as it is much closer to an existing word than when looking at MSA only.

The same argument can be extended to real words. For example, the MSA word (/jbhp/, 'جبهة', forehead) has many dialectal counterparts (a subset is shown in Table 8.2). The MSA word is rather similar to the word in the Gulf dialect, while it would be more difficult to recognize for speakers of Egyptian and Maghreb dialects whose dialectal words are quite different.

8.4 User Study

In order to investigate the role of diacritical marks on Arabic lexical recognition tests, we conduct a user study where we compare a non-diacritized and a diacritized test. In order to avoid memorization effects, one student cannot answer the non-diacritized (ND) and diacritized (D) version of the same test. Thus, in our user study, we use two tests of ND/D pairs. In order to avoid sequence effects, one group begins with the diacritized version, while the other group begins with the non-diacritized one. We visualize this setup in the following figure:

Group 1 (G1)	Group 2 (G2)
test A (D)	test A (ND)
test B (ND)	test B (D)

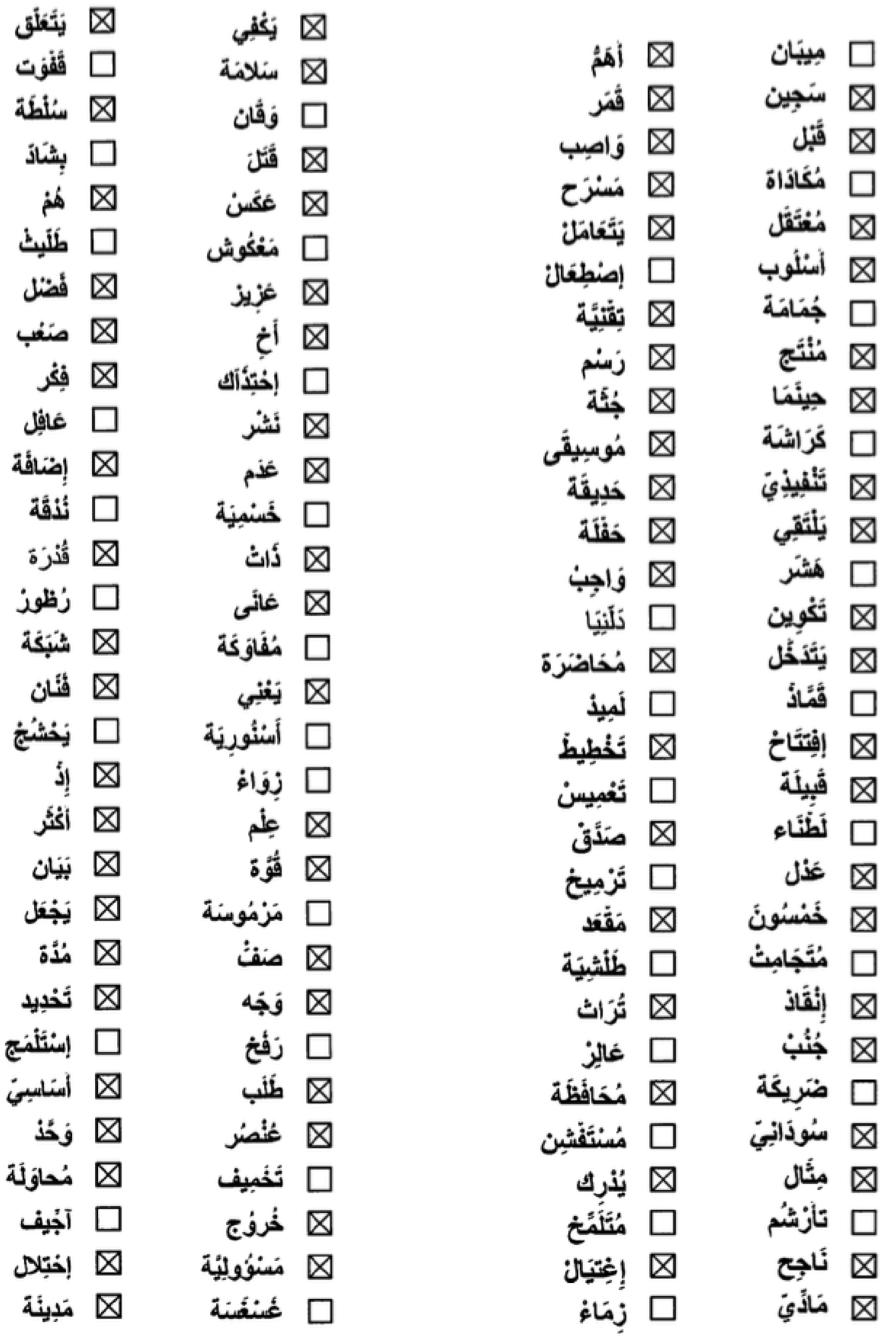
As a starting point, we utilize two non-diacritized Arabic tests prepared by Ricks (Ricks 2015) that both contain the traditional number of 40 words and 20 nonwords. In order to avoid guessing, the items were randomized.[5] We created a diacritized version of both tests by using the most probable form (see discussion in Section 8.2 and especially Figure 8.2). For nonwords, we use a plausible version of diacritics. We also normalized all the initial Alif letters by adding the Hamza (glottal stop) to both versions of the test. Figure 8.3 shows the diacritized test versions.

We provided the participants in our study with a set of instructions including some sample items. The study itself was implemented as a paper-based survey under direct supervision of a teacher. We recruited 40 students (22 female) from 5 German schools (4^{th} to 10^{th} grade), who are studying the Arabic European syllabus "I Love the Arabic Language" that conforms to the Common European Framework of Reference (CEFR)[6]. All students are native German speakers, but with Arabic as a family language.

In order to provide an external gold standard for the proficiency level of each student, we

[5] In the original version, all the words were presented first and the nonwords after that. This is clearly not optimal, as participants can quite easily detect and exploit this setup.

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



☒ يَكْفِي
☒ سَلامَة
☐ وَقُان
☒ قَتَلَ
☒ عَكَسْ
☐ مَغْكُوش
☒ عَزِيز
☒ أَخ
☐ إِخْتِذَأك
☒ نَشْر
☒ عَدَم
☐ خَسْمِيَة
☒ ذَات
☒ عَانَى
☐ مُفَاوَكَة
☒ يَغْنِي
☐ أَسْنُورِيَة
☐ زِوَاغ
☒ عِلْم
☒ قُوَّة
☐ مَزْمُوسَة
☒ صَفْ
☒ وَجْه
☐ رَفْخ
☒ طَلَب
☒ عُنْصُر
☐ تَخْمِيف
☒ خُرُوج
☒ مَسْؤُولِيَة
☐ غُسْغُسَة

☒ يَتَعَلَّق
☐ قَفُوت
☒ سُلْطَة
☐ بِشَادّ
☒ هُمْ
☐ طَلَّيث
☒ فَضْل
☒ صَعْب
☒ فِكْر
☐ عَافِل
☒ إِضَافَة
☐ نُدْقَة
☒ قُدْرَة
☐ رُظُوز
☒ شَبَكَة
☒ فَنَّان
☐ يَخْشُج
☒ إِذْ
☒ أَكْثَر
☒ بَيَان
☒ يَجْعَل
☒ مُدَّة
☒ تَحْدِيد
☐ إِسْتَلْمَج
☒ أَسَاسِيّ
☒ وَخُذْ
☒ مُحَاوَلَة
☐ آجْيف
☒ إِخْتِلال
☒ مَدِينَة

(a) test A

☐ مِيَبَان
☒ سَجِين
☒ قَبْل
☐ مُكَادَاة
☒ مُعْتَقَل
☒ أُسْلُوب
☐ جُمَامَة
☒ مُنْتَج
☒ حِينَمَا
☐ كَرَاشَة
☒ تَنْفِيذِيّ
☒ يَلْتَقِي
☐ هَشَر
☒ تَكْوِين
☒ يَتَدَخَّل
☐ قَمَّاذ
☒ إِفْتِتَاح
☒ قَبِيلَة
☐ لَطْنَاء
☒ عَدْل
☒ خَمْسُونَ
☐ مُتَجَامِث
☒ إِنْقَاذ
☒ جُنُب
☐ ضَرِيكَة
☒ سُودَانِيّ
☒ مِثَال
☐ تَأْرْشُم
☒ نَاجِح
☒ مَادِّيّ

☒ أَهَمّ
☒ قَمَر
☒ وَاصِب
☒ مَسْرَح
☒ يَتَعَامَلْ
☐ إِصْطِغَال
☒ تِقْنِيَّة
☒ رَسْم
☒ جُثَّة
☒ مُوسِيقَى
☒ حَدِيقَة
☒ حَفْلَة
☒ وَاجِب
☐ دَلَنِيَا
☒ مُحَاضَرَة
☐ لَمِيذ
☒ تَخْطِيط
☐ تَغْمِيس
☒ صَدَق
☐ تَرْمِيخ
☒ مَقْعَد
☐ طَلْشِيَة
☒ تُرَاث
☐ غَالِز
☒ مُحَافَظَة
☐ مُسْتَفْشِن
☒ يُدْرِك
☐ مُتَلَمِّخ
☒ إِغْتِيَال
☐ زِمَاغ

(b) test B

Figure 8.3: The two diacritized tests used in our study. Words are checked, nonwords are not.

Students	Words						Nonwords					
	ND			D			ND			D		
Group	P	R	F	P	R	F	P	R	F	P	R	F
G1	.93	.70	.78	.96	.75	.82	. 66	.91	.75	.71	.94	.79
G2	.95	.81	.86	.96	.85	.89	.77	.93	.82	.81	.93	.86

Table 8.3: Results for non-diacritized and diacritized tests comparing groups.

asked the Arabic teacher (*before* conducting the study) to evaluate each student on a three-point proficiency scale: (1) beginner, (2) intermediate, and (3) advanced. This gold standard provides a basis for judging the construct validity of our Arabic lexical recognition tests (Milton 2007).

8.5 Results and Discussion

Our test design that tries to avoid memory effects entails that the non-diacritized (ND) and diacritized (D) variant of a test are solved by different groups. In order to make meaningful comparisons between the ND and D variants, we first have to make sure that the groups are comparable.

8.5.1 Group Comparison

To compare the two groups, we compute other evaluation metrics, namely precision (P), recall (R), and F-measure (F) for words and nonwords. These metrics are computed using the equation presented in subsection 3.3.2 as well as the followings equations:

$$P_w = \frac{TP}{TP+FP} \tag{8.1}$$

$$F_w = 2 * \frac{P_w * R_w}{P_w + R_w} \tag{8.2}$$

$$P_{nw} = \frac{TN}{TN+FN} \tag{8.3}$$

$$F_{nw} = 2 * \frac{P_{nw} * R_{nw}}{P_{nw} + R_{nw}} \tag{8.4}$$

In Table 8.3, we show precision, recall, and F-measure for both groups. We see that while the precision is comparable for both groups, group 2 has higher recall in general which can be explained by a slightly higher number of high proficiency students in this group. In general, we see that for words, the precision is quite high, while for nonwords the recall is quite high. This is related to the strategy applied by most students that they only check the words that they actually know. Leading to high precision for words, and high recall for nonwords which is the fallback.

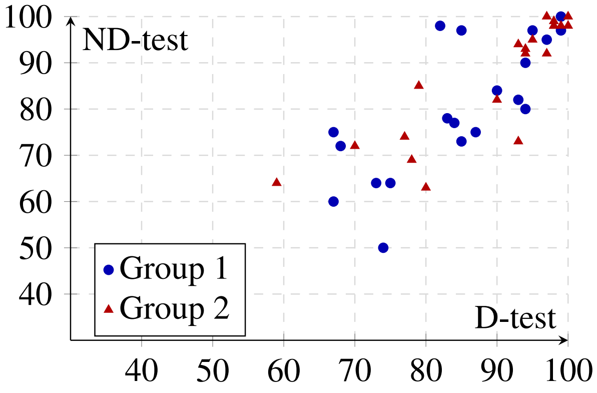


Figure 8.4: Participants' scores on the ND-test & D-test.

Scoring Criteria It was noticed in Chapter 4.7, there are several possible methods to compute LRTs scores. For each participant, we compute the test score using Equation 3.2, the score consists of the ratio of correct responses for words and nonwords – i.e. the recall for each class. We are using Equation 3.2 because a yes bias – by identifying (checking) all items as words – (creating high error rates in the nonwords) would be *penalized* in the same way as a no bias – by identifying all items as nonwords (checking none of the items) – (causing high error rates for words), independently of the different numbers of words versus nonwords.

Figure 8.4 shows a scatterplot of the two groups regarding the assigned test scores. It also confirms our finding that both groups are comparable.

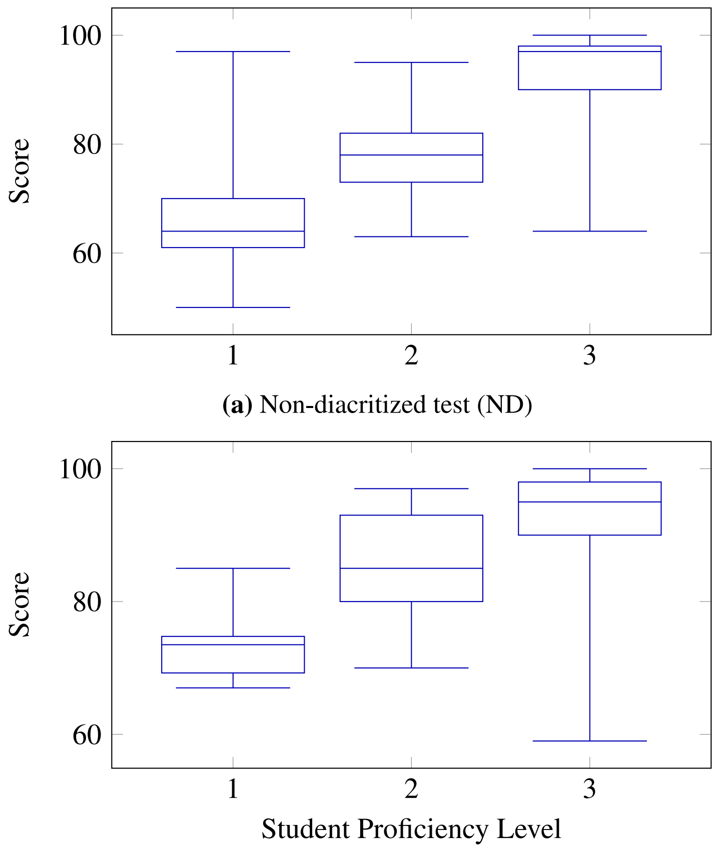
8.5.2 Proficiency Level

Figure 8.5 visualizes the relationship between the evaluation of the teacher and the scores assigned by our two test versions. Both versions assign on average higher scores to more proficient students, i.e. they measure the language proficiency to some extent. The non-diacritized (ND) version of the tests has higher variance for the low proficiency students, while the diacritized (D) version has higher variance for the high proficiency students. However, due to our relatively small sample size, we cannot draw definite conclusions from that observation.

In order to analyze the differences between the three levels, we additionally show a breakdown of precision, recall, and F-measure grouped by proficiency level in Table 8.4. We observe the usual trend of high precision for words, and high recall for non-words related to the test strategy of only checking known words. We also see that the results for the D and ND variants of our test are relatively similar, which means that using precision instead of recall in Equation 3.2 would not make much of a difference.

8.5.3 Qualitative Analysis

Table 8.5 shows the most difficult pair of word and nonword for each group and test condition. The two words with the highest difficulty index are /**aAto*/ (ذَاتْ) and /*IgtyAl*/ (إغتيال). Both are selected by 5 (i.e. missed by 15) students out of 20. This is most likely because **aAto* normally appears as a noun-phrase, whereas *IgtyAl* is most common in countries with conflicts



(a) Non-diacritized test (ND)

(b) Diacritized test (D)

Figure 8.5: Visualization of teacher evaluation in the ND-test and D-test.

Proficiency	**Words**						**Nonwords**					
	ND			**D**			**ND**			**D**		
Level	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
1	.88	.50	.61	.95	.54	.67	.50	.86	.55	.52	.93	.66
2	.93	.68	.77	.96	.78	.85	.62	.90	.68	.71	.93	.80
3	.97	.87	.91	.97	.89	.91	.83	.95	.87	.86	.94	.88

Table 8.4: Results grouped by proficiency level.

like Palestinian territories. The two nonwords with the highest difficulty index are */xsmyp/* (خسمية) and */mukaAdaAp/* (مُكَاذَاة). Both nonwords are very similar to Arabic words: (i) *xsmyp* is similar to (/Hsmyp/, 'حسمية', finality) and (ii) *mukaAdaAp* is similar to (/muqaADaAp/, 'مُقَاضَاة', prosecution). An interesting observation is that non-words have much lower error rate than words. This is mainly due to the above mentioned strategy of only checking words one really knows.

	Words				Nonwords		
Group, Test	**Arabic**	**Transliteration**	**Meaning**	**%Wrong**	**Arabic**	**Transliteration**	**%Wrong**
G2, test A (ND)	وحد	wHd	unify	.55	خسمية	xsmyp	.40
G1, test B (ND)	إغتيال	IgtyAl	assassination	.75	زماء	zmA'	.35
G1, test A (D)	ذَات	*aAto	self	.75	بِشَاد	bi$aAd	.25
G2, test B (D)	إِغتِيَال	IigityaAlo	assassination	.50	مُكَادَاة	mukaAdaAp	.40

Table 8.5: List of most difficult items for each test variant.

8.5.4 Limitations

As it can be clearly seen from the results of our study, the scores for the advanced learners are relatively high, i.e. we are already seeing ceiling effects here. As a consequence, in the current form the tests cannot be used for truly advanced students.

Nonwords As our empirical results show, nonwords generated from non-existing roots are too easy to spot. The same is true for nonwords composed of rare phonemes or rare combinations of phonemes that feature too many of the least common letters, which can cause the stimuli items composed from them to appear unlikely to test respondents including those with beginner or intermediate levels. However, we believe that using diacritics opens the possibility to utilize nonwords that use existing roots, but with a non-existing configuration of diacritics. Exploring this option remains as future work.

Words Generally, the quality of the words is acceptable, but can be further improved as we are seeing some ceiling effects even for medium proficiency students, i.e. some of the words are much too easy. Consequently, we need a better way of controlling the frequency. The Buckwalter/Parkinson frequency dictionary provides a list of the 5,000 most frequently used words in MSA as well as several of the most widely spoken Arabic dialects. A better valid option might be the revised Arabic WordNet, which comes with irregular plurals (Abouenour et al. 2013). Even better would be corpora reflecting the type of reading material students are likely to have seen at a certain proficiency level, but to the best of our knowledge no such corpus is currently available.

8.6 Chapter Summary

We have tackled the task of designing lexical recognition tests in Arabic by first discussing the specific challenges that are imposed by the language. It seems clear that using diacritics has the potential to (i) improve the quality of nonwords and (ii) to better control the difficulty of the tests.

We compared the diacritized and non-diacritized lexical recognition tests in a user study and find that they are largely comparable. This is in line with our hypothesis that students will recognize the most probable diacritized word which we used in our test.

In the next chapter, we want to use less likely diacritized forms and explore how well we can control the difficulty of the tests. We envision to create tests that are better able to discriminate medium and high proficiency learners as we already see ceiling effects in the non-diacritized test versions, mainly due to very easy nonwords. We also want to explore ways to automatically create Arabic lexical recognition tests, a task that entails a lot of NLP challenges regarding automatic diacritization, morphological analysis, and language modeling.

Chapter 9

Adapting the Difficulty of Arabic LRTs

LRT is a corpus-based assessment that makes use of words frequency counts[1] in a huge corpus. In the previous chapter, we have shown that non-diacritized Arabic lexical recognition tests show serious ceiling effects as they are too easy for most learners. It is sufficient for a learner to recognize the root form as they know one of its diacritized forms – probably the most frequent diacritized of a word. Table 9.1 shows the frequency counts of some diacritized forms of the root /*kr/.[2]

Our hypothesis in this chapter is that we can construct a more appropriate Arabic lexical recognition test by using less frequent diacritized forms, such as ذَكَّرَ /*ak∼ara/ or ذُكِّرَ /*uk∼ira/. For that purpose, we first have to find a way to reliably estimate the frequency of diacritized word forms. Then, we conduct a user study, measuring the difficulty of the resulting lexical recognition test under three conditions: (i) No Diacritics: non-diacritized words, (ii) Frequent Diacritics: diacritized using the most frequent diacritized word form, and (iii) Infrequent-Diacritics: diacritized using the least frequent diacritized form of a word.

In this chapter, we investigate the role that diacritics play in adapting the difficulty of Arabic lexical recognition tests. The chapter is organized as follows: In section 9.1, we talk about counting Arabic words and treat the challenges entailed in estimating the frequency counts for Arabic words in a diacritized corpus. In section 9.2, we present the user study setup for the three test versions. In section 9.2, we present the web-interface for our LRTs application. This is followed by section 9.3 containing the user study results, we are comparing the test versions and provide a visualization for item analysis. Finally, we summarize the chapter in section 9.4.

9.1 Counting Arabic Words

Obtaining reliable frequency counts for Arabic words is a task that entails a lot of NLP challenges regarding availability of corpora, automatic diacritization, segmentation, etc.

9.1.1 Availability of Corpora

We typically need a large amount of diacritized Arabic text to estimate the frequency of diacritized word forms, but there is a lack of such resources. Generally, the currently available diacritized corpora are limited to Classical Arabic (usually religious text), such as the Holy

[1] We refer to Arabic diacritized words.

[2] The frequency counts are based on the Tashkeela corpus (Zerrouki and Balla 2017), a corpus of classical Arabic books texts that are provided with diacritics.

Surface form	**Diacritized form**	**Gloss**	**Counts**
ذكر	ذَّكَر /*∼akar/	Male	18
	ذِكْر /*ikor/	Prayer	10
	ذَكَرَ /*akar/	He mentioned	1454
	ذُكِرَ /*ukir/	It was mentioned	2001
	ذَكَّرَ /*∼akar/	He reminded	1
	ذُكِّرَ /*uk∼ir/	He was reminded	4

Table 9.1: Examples of diacritized forms of the Arabic word ذكر /*kr/.

Quran[3], Hadith books, Arabic Language Resources corpus as obtained from RDI company[4] and Tashkeela (Zerrouki and Balla 2017); or Modern Standard Arabic (usually commercial news wires), such as Penn Arabic Treebanks (ATB) and Agence France Presse (AFP) that can be purchased from the Linguistic Data Consortium (LDC).

Source Corpus As the costs of acquiring annotated corpora can prevent researchers from conducting their research Zaghouani (2014), we only want to use freely available corpora. One option is the corpora provided by Mourad Abbas and contains newspaper articles crawled from the internet[5]. However, as we are trying to build an educational application that measures language proficiency, we need text that covers a broader variety of topics. We are thus using the corpus introduced by Freihat et al. (2018), which was assembled from texts and text segments from a varied set of online Arabic language resources such as Wikipedia, news portals, online novels, social media, and medical consultancy web pages. Table 9.2 shows the distribution of sub corpora in the resource as published by Freihat et al. (2018).

9.1.2 Automatic Diacritization

It has been shown that automatic diacritization can be used to obtain reliable frequency counts for Arabic words (Hamed and Zesch 2018) by automatically diacritizing a large non-diacritized source corpus. According to a recent benchmark (Hamed and Zesch 2017a) comparing the available tools for diacritization: Farasa (Darwish and Mubarak 2016), Madamira (Pasha et al. 2014) as well as two strong baselines (a dictionary lookup system and one based on character-based sequence labeling). Farasa is outperforming the other approaches under all conditions. Therefore, we use Farasa to diacritize the crawled source corpus. The diacritized corpus is available upon request.

[3] http://tanzil.net/download/

[4] http://www.rdi-eg.com/

[5] Available at: `https://sites.google.com/site/mouradabbas9/corpora`

Resource	Proportion
Aljazeera online	30%
Arabic Wikipedia	20%
Novels	15%
Alquds newspaper	10%
Altibbi	10%
IslamWeb	5%
Social networks (FB, Twitter)	5%
Other	5%

Table 9.2: Proportion of corpus resource.

9.1.3 Lemmatization

As we want to use lemmas, not surface forms in our Arabic lexical recognition test, we need to perform lemmatization. This step is necessary as Arabic is a morphology-rich language and its words are highly inflected and derived (Aqel et al. 2015). Darwish and Mubarak (2016) reported that Farasa outperforms or matches state-of-the-art Arabic segmenters/lemmatizers like QCRI Advanced Tools For Arabic (QATARA) (Darwish et al. 2014) and Madamira (Pasha et al. 2014).

We (Hamed and Zesch 2018) explore the effects of diacritization on Arabic frequency counts. We have shown that Farasa clearly gives better estimates than Madamira. Therefore, we integrate Farasa segmenter/lemmatizer in our NLP pipeline.

9.1.4 NLP Pipeline

To reliably estimate the frequency counts for the diacritized LRT word items, we run the following NLP pipeline, given the source corpus as input: (i) diacritize the source corpus using the Farasa diacritizer, (ii) segment the space-delimited diacritized words using Farasa, (iii) discard the extra clitics, (iv) label the roots or lemmas with the corresponding diacritics with the help of DKPro Core[6], a collection of software components for natural language processing based on the Apache UIMA framework, and (v) assign the frequency counts for each lemma based on the attached diacritics. We depict these steps as show in Figure 9.1

After carrying out the aforementioned NLP pipeline on this source corpus, we get frequency counts similar to that in Table 9.1. The frequency counts contain, among others, the most and least frequent diacritized form of a word that are corresponding to a given non-diacritized root/lemma. Now we are ready to construct the tests and conduct the user study.

[6] `https://dkpro.github.io/dkpro-core/`

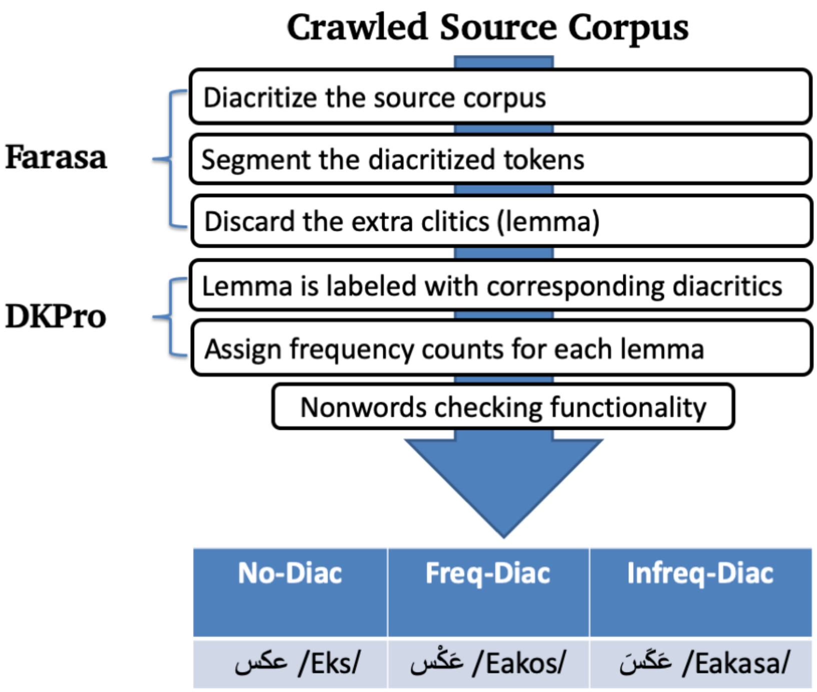

Crawled Source Corpus
Farasa
Diacritize the source corpus
Segment the diacritized tokens
Discard the extra clitics (lemma)
DKPro
Lemma is labeled with corresponding diacritics
Assign frequency counts for each lemma
Nonwords checking functionality
No-Diac
Freq-Diac
Infreq-Diac
عكس /Eks/
عَكْس /Eakos/
عَكَسَ /Eakasa/

Figure 9.1: NLP pipeline steps.

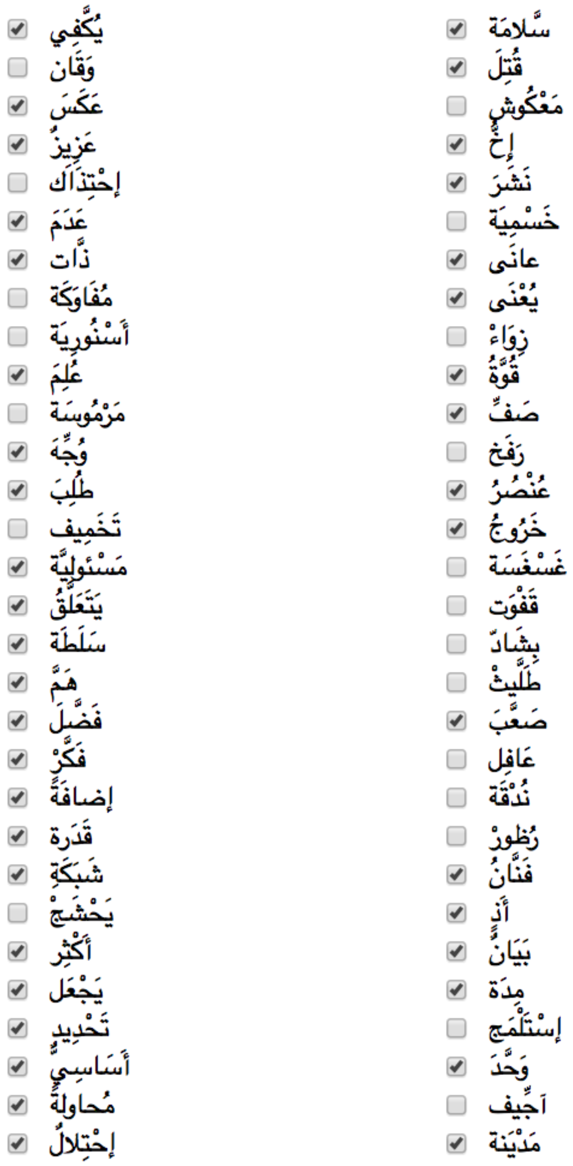
Word	Nonword	Swapped-letter
حسمية	خسمية	ح to خ
عاقل	عافل	ق to ف
رفح	رفخ	ح to خ
معكوس	معكوش	س to ش

Table 9.3: Nonwords created by letter transposition of minimal pairs.

9.2 User Study Setup

In order to investigate the role of diacritical marks on improving the construct validity of Arabic lexical recognition tests, we conduct a user study where we compare three tests that differ in the diacritization settings.

- **No Diacritics (S1)**: We use the non-diacritized version of 'test A' as used by Hamed and Zesch (2017b). The nonwords have been generated using a letter substitution/transposition approach in an existing word. Table 9.3 contains some examples of such nonwords.
- **Frequent-Diacritics (S2)**: We diacritize all words from S1 with the *most* frequent diacritized form. The nonwords are the same as in S1 and diacritized using a pronounceable (plausible) version of diacritics.
- **Infrequent-Diacritics (S3)**: We diacritize all words from S1 with the *least* frequent diacritized form. Figure 9.2 shows the resulting test in checklist format.



☑ يُكَّفِي
☐ وَقَان
☑ عَكَسَ
☑ عَزِيزٌ
☐ إحْتِذَاَك
☑ عَدَمَ
☑ ذَّات
☐ مُفَاوَكَة
☐ أَسْنُورِيَة
☑ عُلِمَ
☐ مَرْمُوسَة
☑ وُجِّهَ
☑ طُلِبَ
☐ تَخَمِيف
☑ مَسْئولِيَّة
☑ يَتَعَلَّقُ
☑ سَلَطَة
☑ هَمَّ
☑ فَضَّلَ
☑ فَكَّرْ
☑ إضافَةً
☑ قَدَرة
☑ شَبَكَةِ
☐ يَحْشَجْ
☑ أَكْثِر
☑ يَجْعَل
☑ تَحْدِيدٍ
☑ أَسَاسِيٌ
☑ مُحاولةً
☑ إحْتِلالٌ

☑ سَّلامَة
☑ قُتِلَ
☐ مَعْكُوشٍ
☑ إخٌ
☑ نَشَرَ
☐ خَسْمِيَة
☑ عانَى
☑ يُعْنَى
☐ زِوَاءْ
☑ قُوَّةُ
☑ صَفٍّ
☐ رَفَخ
☑ عُنْصُرُ
☑ خَرُوجُ
☐ غَسْغَسَة
☐ قَفْوَت
☐ بِشَادّ
☐ طَلَّيثْ
☑ صَعَّبَ
☐ عَافِل
☐ نُدْقَة
☐ رُظورْ
☑ فَنَّانُ
☑ أَذٍ
☑ بَيَانٌ
☑ مِدَة
☐ إسْتَلْمَج
☑ وَحَّدَ
☐ آجِّيف
☑ مَدْيَنة

Figure 9.2: The diacritized tests items for test A in *infrequent-diacritics* setting (S3), words are checked, nonwords are not.

Pilot Study Before conducting the main user study, an Arabic teacher reviewed the three tests. For example, he made sure that no dialectal words are used because they could only be recognized by Arabic speakers who understand that dialect.

A few students (n = 11) were asked to participate in the user study, so that we check the overall format, design, and test instructions. No modifications have been made to overall test format or design. Minor modifications had to be made to test instructions after the pilot study.

Main 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, mother tongue (L1), and the knowledge of Arabic language (number of years they had taken Arabic

	40 Words			20 Nonwords		
Test Setting	**P**	**R**	**F**	**P**	**R**	**F**
S1 – No Diacritics	.95	.95	.95	.93	.89	.91
S2 – Freq. Diac.	.91	.92	.91	.90	.82	.86
S3 – Infreq. Diac.	.92	.80	.86	.71	.85	.77

Table 9.4: Results for the three tests settings.

courses). Then, participants had to finish the actual lexical recognition test. The test version which participants received (non diacritics, frequent diacritics, infrequent diacritics) was assigned randomly to avoid sequence effects.

Web-Interface In order to conduct the study, we created a multi-device web interface using PHP and MySql database. Figure 9.3 shows how it looks like. We make the implementation available to allow for easy replication.[7]

9.3 User Study Results

We advertised our study through different channels, such as mail listings and social media. Overall, 263 people participated in the study, 143 are male, 120 are female. The average age is 28.1 years. Overall, the participants are randomly distributed over the three tests as follows: 96 participants were assigned to S1, 78 participants were assigned to S2, and the remaining 89 participants were assigned to S3.

In Table 9.4, we show precision, recall, and F-measure for the three test settings for both words and nonwords, averaged over all participants. We see that while the precision for words is comparable over all three tests, our test version S3 with infrequent diacritics has lower recall. This is the intended effect or more people not recognizing the words (remember that the non-diacritized tests are too easy and we want people to fail a bit more often).

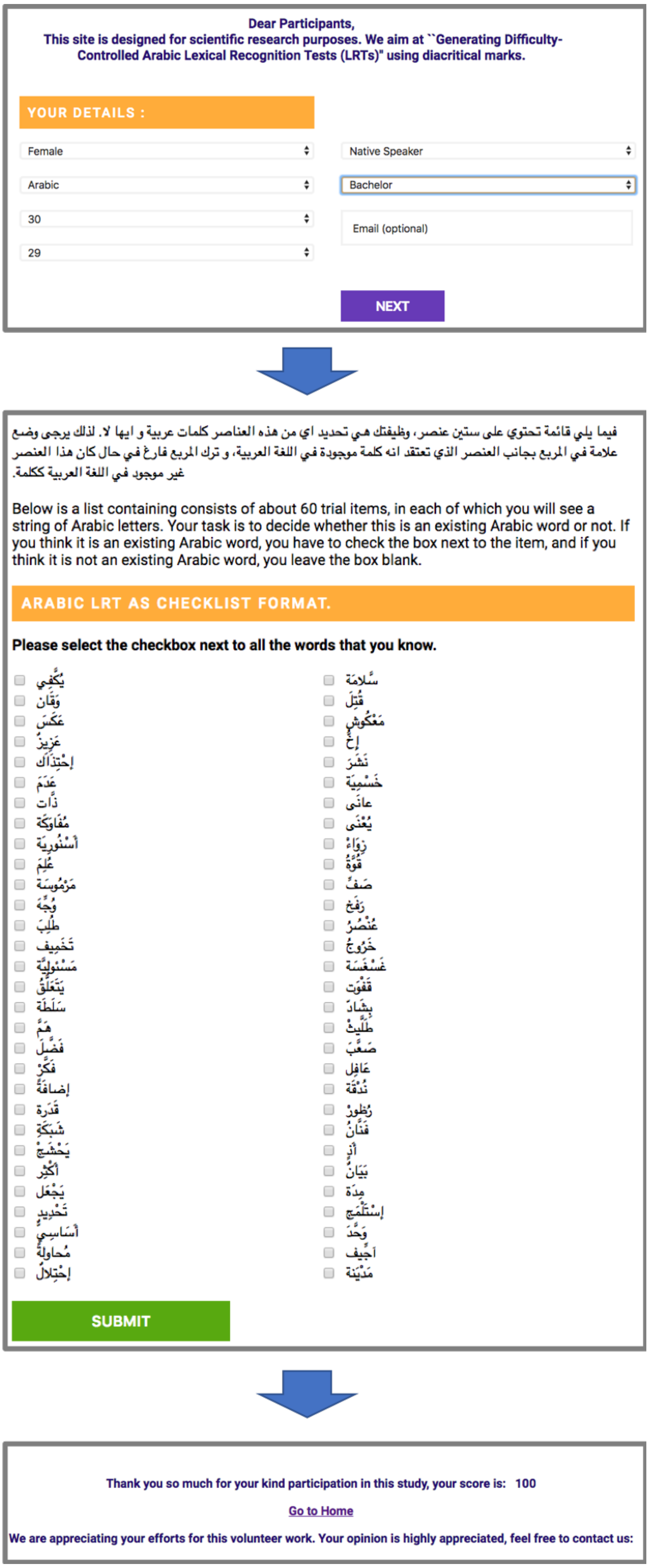
9.3.1 Comparing Test Versions

In order to compare the difficulty of the two diacritized tests S2 and S3 with the original non-diacritized test S1, we compute for each respondent a combined test score using the scoring scheme introduced in Chapter 4 (Equation 3.2), and utilized by Hamed and Zesch (2017b). In order to account for the unequal number of words and nonwords in the test, it averages the corresponding recalls.

We are using Equation 3.2 again because a yes bias – by identifying (checking) all items as words – (creating high error rates in the nonwords) would be *penalized* in the same way as a no bias – by identifying all items as nonwords (checking none of the items) – (causing high error rates for words), independently of the different numbers of words versus nonwords.

Then, we compute the average score (over all participants) for each variant. We obtain average scores of 91.8, 86.8, and 82.3 for the three tests respectively. We compute the statistical

[7] https://github.com/ohamed/ar-lrts



Dear Participants,
This site is designed for scientific research purposes. We aim at ``Generating Difficulty-Controlled Arabic Lexical Recognition Tests (LRTs)" using diacritical marks.

YOUR DETAILS :

Female	Native Speaker
Arabic	Bachelor
30	Email (optional)
29	

NEXT

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
- ☐ إحْتِذَاك
- ☐ عَدَمَ
- ☐ ذَاتٌ
- ☐ مُفَاوَكَة
- ☐ أَسْنُورِيَة
- ☐ عُلِمَ
- ☐ مَرْمُوسَة
- ☐ وُجِّهَ
- ☐ طَلِبَ
- ☐ تَخَمِيف
- ☐ مَسْئُولِيَّة
- ☐ يَتَعَلَّقُ
- ☐ سَلَطَة
- ☐ هَمَّ
- ☐ فَضَّلَ
- ☐ فَكَّرْ
- ☐ إضافَةً
- ☐ قَدَرة
- ☐ شَبَكَةٍ
- ☐ يَحْشَجْ
- ☐ أَكْثِر
- ☐ يَجْعَل
- ☐ تَحْدِيدٍ
- ☐ أَسَاسِيٍ
- ☐ مُحاولَةٌ
- ☐ إحْتِلالٌ
- ☐ سَّلامَة
- ☐ قُتِلَ
- ☐ مَعْكُوشٍ
- ☐ إِخٌ
- ☐ نَشَرَ
- ☐ خَسْمِيَة
- ☐ عانَى
- ☐ يُعْنَى
- ☐ زِوَاءْ
- ☐ قُوَّةٌ
- ☐ صَفٌّ
- ☐ رَفَخ
- ☐ عُنْصُرُ
- ☐ خَرُوجُ
- ☐ غَسْغَسَة
- ☐ قَفْوَت
- ☐ بِشَادَ
- ☐ طَلِّيثْ
- ☐ صَعَّبَ
- ☐ عَافِل
- ☐ نُدْقَة
- ☐ رُظورْ
- ☐ فَنَّانُ
- ☐ أنِ
- ☐ بَيَانٌ
- ☐ مِدَة
- ☐ إسْتَلْمَج
- ☐ وَحَّدَ
- ☐ أَجِّيف
- ☐ مَدْيَنة

SUBMIT

Thank you so much for your kind participation in this study, your score is: 100

Go to Home

We are appreciating your efforts for this volunteer work. Your opinion is highly appreciated, feel free to contact us:

Figure 9.3: Web system.

Compared Tests	p-value
S1 - S2	0.0063
S1 - S3	0.0001
S2 - S3	0.0001

Table 9.5: Statistical significance of the differences between the tests.

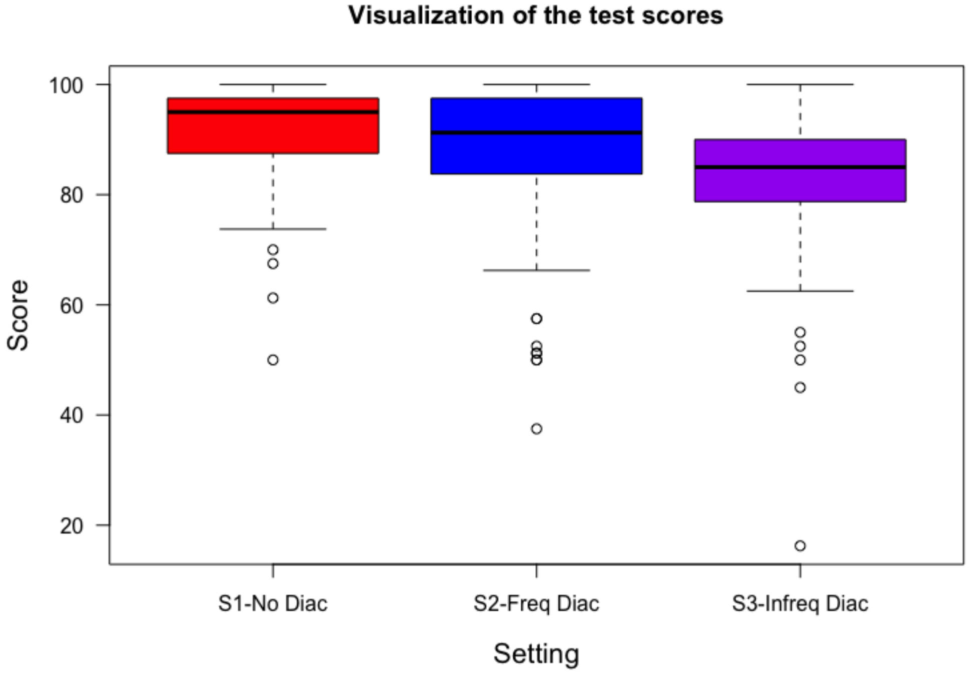


Figure 9.4: Visualization of the test scores under the three settings.

significance of the differences between the three tests using the *t-test* – see Table 9.5. All differences between the scores are statistically significant.

We visualize the relationship between the setting and the scores obtained by the participants in each test as shown in Figure 9.4. The non-diacritized test S1 shows the predicted ceiling effect. The differences to the diacritized version with the most frequent diacritics (S2) are actually larger than we would have predicted (recall that our hypothesis was that even in the non-diacritized version, subjects would fall back to the most frequent diacritized form). However, in line with our predictions the third test version (S3) using infrequent diacritics is much more difficult than both other tests and shows no ceiling effects. It should thus be better suited for accurately measuring the vocabulary size of more advanced learners than the other test versions.

9.3.2 Item Analysis

So far, we have only looked at the test results in general (across all items), but it remains unclear whether all words get more difficult or whether the effect is stronger for some words.

Thus, we visualize the scores for each word in our three experimental settings using a heatmap along with their frequency counts as shown in Table 9.6. As the score corresponds to how many participants of our study recognized a word, light colors mean easy items and darker colors mean difficult items. We find that some words get much harder when using the least frequent diacritization, while there is almost no effect for other words. In order to check whether this effect can be attributed to the frequency of the underlying forms, we also plot the counts as obtained from the source corpus for the majority of the word items.

Overall, there is no obvious relationship between the scores of the word in the three settings and their frequency counts. For example, هم /hm/ from S1 occurs 4,510 times, هُمْ /humo/ (meaning: *they*) from S2 occurs 2,388 times, and هَمّ /ham∼/ (meaning: *worry*) from S3 occurs 57 times. However, we don't observe a big drop in the respective scores that are 94%, 93% and 90% for S1, S2, and S3.

9.4 Chapter Summary

In this chapter, we have shown that using Arabic lexical recognition tests with less frequent diacritized forms is a way to avoid the ceiling effects of previously proposed non-diacritized tests. We also show how the necessary frequency counts can be obtained by automatically diacritizing source corpora.

In future work, we need to further investigate why some infrequent diacritized forms are hard while other (similarly infrequent) diacritized forms are easy. We hypothesize that the corpora used in this study might not reliably reflect the knowledge of learners. Also, even if we tried to minimize the effects of dialects, there might be strong influences from words being frequently used in a dialect or not. Moreover, we recommend other researchers in the field to avoid using non dictionary items like يتعلق /ytElq/, يعني /yEny/.

	Buckwalter	S1	S2	S3		*freq*	
Arabic	Transliteration	No Diac	Freq. Diac	Infreq. Diac	S1	S2	S3
عنصر	EnSr	.99	.91	.83	50	35	15
قتل	qtl	.98	.95	.95	416	184	77
قوة	qwp	.98	.92	.92	181	115	8
صعب	SEb	.98	.92	.84	132	41	1
أكثر	Okvr	.98	.95	.90	1561	1120	122
أساسي	OsAsy	.98	.95	.91	753	195	20
مدينة	mdynp	.98	.95	.84	98	80	2
يكفي	ykfy	.97	.94	.58	139	97	6
عكس	Eks	.97	.88	.90	101	99	2
نشر	n$r	.97	.90	.86	424	181	100
عدم	Edm	.97	.95	.91	931	640	133
طلب	Tlb	.97	.94	.89	399	192	7
خروج	xrwj	.97	.92	.68	481	158	21
فضل	fDl	.97	.92	.86	113	84	8
فكر	fkr	.97	.95	.85	332	305	12
قدرة	qdrp	.97	.95	.51	34	25	6
بيان	byAn	.97	.91	.91	883	370	3
يجعل	yjEl	.97	.94	.90	122	111	11
تحديد	tHdyd	.97	.94	.91	512	310	49
سلامة	slAmp	.96	.96	.66	34	26	6
عزيز	Ezyz	.96	.94	.92	472	304	42
علم	Elm	.96	.92	.92	348	279	4
صف	Sf	.96	.87	.70	131	38	9
وجه	wjh	.96	.92	.80	568	274	12
يتعلق	ytElq	.96	.90	.89	127	110	17
شبكة	$bkp	.96	.91	.81	22	19	1
محاولة	mHAwlp	.96	.94	.92	15	13	2
ذات	*At	.95	.87	.65	1234	205	42
إذ	I*	.95	.91	.31	328	302	11
مسؤولية	msWwlyp	.94	.94	.72	734	540	27
سلطة	slTp	.94	.91	.85	33	27	4
هم	hm	.94	.90	.93	4510	2388	57
إضافة	IDAfp	.94	.94	.91	325	197	5
مدة	mdp	.94	.95	.41	129	92	10
أخ	Ox	.93	.85	.25	38	33	5
يعني	yEny	.93	.91	.86	338	337	1
فنان	fnAn	.93	.87	.89	876	481	12
إحتلال	IHtlAl	.93	.90	.90	316	249	26
عانى	EAnY	.87	.87	.71	21	14	4
وحد	wHd	.65	.78	.86	335	326	5

Table 9.6: Heatmap visualizing the average score per word, along with their frequency counts. Items are sorted by S1 score.

Chapter 10

Conclusion

In this chapter, we summarize the findings of this thesis and give an overview of our main contributions. Furthermore, we discuss limitations and provide pointers for future work.

10.1 Summary

This thesis has shown that it is possible to generate lexical recognition tests automatically by using NLP techniques that applies to the same language family. For example, English NLP techniques can be applied to other European languages, whereas Arabic NLP techniques can be applied to other semitic languages. In the following, the main and additional findings are summarized and their potential implications for future research are pointed out.

Contributions Here, we map our main two objectives with their corresponding thesis chapters. Our first objective was on nonwords generation which we investigated in Chapter 5. On the other side, our second objective was on test adaptation to Arabic which we investigated in the next remaining chapters, namely Chapter 6 through 9.

- **Nonwords Generation** In Chapter 5, we investigated the automatic generation of nonwords. We proposed an algorithm that requires a corpus as input to generate an English LRT. We started with candidate ranking and described the different ranking strategies used to find good (i.e. word-like) nonwords. We found that nonwords can be generated as character sequences based on position-specific character language models, and words can be selected by using relative frequency factor. In a user study, we compared the automatically generated and the human crafted test (LexTALE). We have seen that an automatically generated LRT is highly correlated with the well-established LexTALE.

  This led to additional research questions. Can we use the same algorithm to generate LRTs for Arabic? What are the expected limitations and challenges? The lack of annotated (diacritized) resources is a gap that prevents us from creating Arabic LRTs. Therefore, as a first step towards achieving this goal, we must put our hands on diacritized Arabic resources. Obtaining a reliable frequency counts for Arabic words entails several NLP challenges that prevented us so far from creating Arabic LRTs . We want to fill the gap and get rid-of the inherited Arabic NLP challenges due to the absence of diacritics. Chapter 6 through 9 were dedicated to get rid-of *diacritics-related* limitations and challenges: lack of annotated resources and diacritics restoration, role of diacritics, Arabic reliable-frequency counts, and adapting the items difficulty.

- **Arabic Resource Creation** In Chapter 6, we investigated the creation of Arabic diacritized resources. A preliminary step that is needed to obtain a reliable frequency count for Arabic words. In this chapter, we produced our own diacritized corpora using off-the-shelf diacritization tools. For that purpose, we conducted a comparative study between the available tools for diacritization. We benchmarked the tools using a reasonable amount and variety of test data in two evaluation modes: strict and relaxed. Under controlled settings, we compared two strong baselines and the two well-known systems: Madamira and Farasa. We found that Farasa is outperforming Madamira in both evaluation modes. The home message, Farasa is highly recommended for creating Arabic diacritized resources.

- **Reliable Frequency Counts** In Chapter 7, we investigated how to obtain a reliable frequency count for Arabic words We explored the effects of diacritization on Arabic frequency counts. We implemented an NLP pipeline that integrates the DKPro Core Java framework in order to explore how severely this situation affects the resulting language models. Our analysis shows that diacritics have a significant influence on obtaining reliable frequency counts in Arabic. However, we also show that a quite good approximation can be obtained by applying automatic diacritization to non-diacritized corpora that are much easier to collect than manually diacritized corpora.
- **Role of Diacritics** In Chapter 8, we investigated the role of diacritics on Arabic LRTs. This work is centred around addressing one of the NLP challenges by taking a closer look on the design process of Arabic LRTs. We expanded the Arabic LRTs by considering the *lexical diacritics* that are a defining feature of Arabic as new parameter that improves the test. We compared the diacritized[1] and non-diacritized LRTs in a user study and found that they are largely comparable. It is recommended to enable the diacritics parameter. The latter has the potential to improve the quality of nonwords because the diacritized nonwords can distract better with closely related, especially if they are labeled with pronounceable diacritics.
- **Adapting Test Difficulty** In Chapter 9, we tackled the creation of a difficulty-controlled Arabic LRTs automatically. The difficult and more challenging items could be created by means of using a non-common or the least frequent diacritized form of a word. It seems clear that using diacritics has the potential to better control the difficulty of the tests (it becomes more difficult) using the least frequent diacritized form of a word in a huge diacritized corpus.

10.2 Limitations and Outlook

In this thesis, we tackled the automatic generation of lexical recognition tests for English and Arabic, in particular MSA.

Limitations So far, we conducted some experiments (Chapter 5 and Chapter 7) and three user studies, one on English LRT (Chapter 5) and the remaining two are on Arabic LRTs (Chapter 8 and Chapter 9). In the following, we raise up some of the limitations that we were not able to investigate further due to time restrictions. However, we still believe that our published results are all preliminary, and behave as expected.

In Chapter 5, for example, the nonwords were ranked based on the language models from Brown corpus, which is a relatively small corpus. In Chapter 9, we approached the frequency

[1] Using the most frequent diacritized form of a word as obtained from Madamira/Farasa diacritization tools.

counts based on a crawled corpus. We had no guarantee that the corpus was balanced with respect to domain.

The population size for the three studies was relatively small. We got 45, 40 and 263 participants for each study respectively. Typically, LRTs are meant for L2 learners. In Chapter 9, we got 263 and were aiming at Arabic language learners. Unfortunately, we only had few response from Arabic language learners.

In contrast to English, where we correlate our generated English LRT with LexTALE. There is a lack for a well-established Arabic LRT. As a result, we did not correlate our generated Arabic LRT in Chapter 9 with another existing Arabic LRT. Handling these two limitations is kept as a future work.

In this thesis, we utilized a non-common or the least frequent diacritized form of a word to increase the difficulty of Arabic LRTs. However, we provide no guarantee about the reliability of respondent answers.

Outlook As future outlook, we think of and recommend the interested researchers to take the initiative by conducting some additional studies which relates to Arabic language learners, dialectal LRTs, the reliability of responses, and an extended words selection process that integrates the influence of Arabic linguistic features such as distribution of words in a text or a tradeoff to include words that have different PoS tagging.

We are looking forward to conduct more extended studies in cooperation with some Arabic language institutions. These studies are targeting learners who are learning Arabic as a foreign language. By following this track, our published results would be better generalized.

Tackling the task from a dialects perspective imposes other parameters. Khalifa et al. (2016) wrote that "Dialectal Arabic (DA) poses serious challenges for Natural Language Processing (NLP)". We believe that supporting the automatic generation of dialectal LRTs entails a lot of NLP challenges regarding availability of corpora, corpus linguistics, tools for automatic diacritization or segmentation, etc. Compared to MSA and other European languages, there is a scarcity of tools and corpora available for DA (Khalifa et al. 2016). The performance of Arabic NLP tools depends on the training data (Farghaly and Shaalan 2009). For instance, a tool trained on MSA text achieves a good performance with MSA text but not necessarily with dialectal Arabic, such as Moroccan Arabic. Similarly, a tool that performs well on one Arabic dialect might not work well with other Arabic dialects (Khalifa et al. 2016). We argue this because of the phonological, lexical, and morphological differences between Arabic dialects (Bouamor et al. 2014). Other important factors that can further complicates the issue are standardization, and code switching. In contrast to MSA, DAs lack for standardized published orthographies (Habash et al. 2012). In their daily conversations, DA speakers mix or code switch between their regional dialect and MSA (Elfardy et al. 2014). Therefore, tackling the dialectal LRTs is kept as a future work.

To investigate the reliability of a respondent responses to words, we want to follow Pellicer-Sánchez and Schmitt (2012) approach. This can be reached using a follow-up test that contains multiple-choice questions on all the words contained in the test. For example, if the test contains among others, the diacritized word ذُكِّرَ /*uk~ir/ (eng: He was reminded). Remember that the word ذكر has different meanings based on the attached diacritics. The corresponding multiple-choice question for this item will be followed by 3 or 4 answers (one is correct, the remaining are distractors) like: The Arabic word ذُكِّر /*uk∼ir/ means (a) Male, (b) Prayer, (c) He was reminded) or (d) He mentioned.

Closing Remark Nowadays, there is a trend of newly emerging applications (platforms) for second-language (L2) learning, among others, this includes *Duolingo*[2], *Babbel*[3], *busuu*[4], and *WordBit*[5]. If you decide to learn German using Duolingo, for example, the learning process starts with a placement test, which is based on translations. On the other side, LRT is meant to be a quick placement test for L2 learners based on word recognition. By the end of this research, we provided a strong framework for the researchers in Arabic language research as well as English. In this way, we did support multilingualism because our approach can be easily transferred to other languages. As next steps, we have the belief that our work can be easily integrated in real and running L2 learning platforms for different languages. Furthermore, business-wise, this work is fair enough to establish a startup for Arabic language assessment.

[2] `https://www.duolingo.com/`

[3] `https://www.babbel.com/`

[4] `https://www.busuu.com/`

[5] `https://www.apkmonk.com/app/net.wordbit.deen/`

Bibliography

Abandah et al. 2015
ABANDAH, Gheith ; GRAVES, Alex ; AL-SHAGOOR, Balkees ; ARABIYAT, Alaa ; JAMOUR, Fuad ; AL-TAEE, Majid: Automatic diacritization of Arabic text using recurrent neural networks. In: *International Journal on Document Analysis and Recognition (IJDAR)* 18 (2015), Nr. 2, pages 183–197

Abdelgadir and Ramana 2017
ABDELGADIR, Ehsan M. ; RAMANA, VSV L.: *A Handbook on "Introduction to Phonetics & Phonology": For Arabic students*. Notion Press, 2017

Abdi 2007
ABDI, Hervé: Signal detection theory (SDT). In: *Encyclopedia of measurement and statistics* (2007), pages 886–889

Aboelezz 2010
ABOELEZZ, M: A Latinised Arabic for all? Issues of representation, purpose and audience. In: *Romanization of Arabic Names. Proceedings of the International Symposium on Arabic Transliteration Standard: Challenges and Solutions, Abu Dhabi, UAE*, 2010, pages 100–110

Abouenour et al. 2013
ABOUENOUR, Lahsen ; BOUZOUBAA, Karim ; ROSSO, Paolo: On the evaluation and improvement of Arabic WordNet coverage and usability. In: *Language resources and evaluation* 47 (2013), Nr. 3, pages 891–917

Abuhakema et al. 2008
ABUHAKEMA, Ghazi ; FARAJ, Reem ; FELDMAN, Anna ; FITZPATRICK, Eileen: Annotating an Arabic Learner Corpus for Error. In: *Proceedings of the Language Resources and Evaluation Conference (LREC)*, 2008

Acs and Halmi 2016
ACS, Judit ; HALMI, József: Hunaccent: Small Footprint Diacritic Restoration for Social Media. In: *Normalisation and Analysis of Social Media Texts (NormSoMe) Workshop Programme*, 2016, pages 1

Afifi and Annabi 2012
AFIFI, Sohaib ; ANNABI, Walid: *Mishkal : Arabic Text Vocalization*. `https://sourceforge.net/projects/mishkal/`, 2012. – Accessed: 2019-04-13

Ahmed and Elaraby 2000
AHMED, Attia ; ELARABY, Mohamed: *A large-scale computational processor of the arabic morphology, and applications*, Faculty of Engineering, Cairo University Giza, Egypt, Diss., 2000

Ahmed 2000
AHMED, Mohamed A.: A large-scale computational processor of the Arabic morphology, and applications. In: *A Master's Thesis, Faculty of Engineering, Cairo University, Cairo, Egypt* (2000)

Aiken 1997
AIKEN, Lewis R.: *Psychological testing and assessment*. Allyn & Bacon, 1997

Aksan and Yaldır 2010
AKSAN, Yeşim ; YALDIR, Yılmaz: A corpus-based word frequency list of Turkish: Evidence from the subcorpora of Turkish National Corpus project. In: *Proceedings of the 15th International Conference on Turkish Linguistics*. Szeged, 2010, pages 47–58

Al-Fak et al. 2015
AL-FAK, Ibrahim M. et al.: Vocabulary input in English language teaching: Assessing the vocabulary load in spine five. (2015)

Al-Onaizan and Knight 2002
AL-ONAIZAN, Yaser ; KNIGHT, Kevin: Machine transliteration of names in Arabic text. In: *Proceedings of the ACL-02 workshop on Computational approaches to semitic languages* Association for Computational Linguistics, 2002, pages 1–13

Al-Sobh et al. 2015
AL-SOBH, Mahmoud A. ; ABU-MELHIM, Abdel-Rahman H. ; BANI-HANI, Nedal A.: Diglossia as a result of language variation in Arabic: Possible solutions in light of language planning. In: *Journal of Language Teaching and Research* 6 (2015), Nr. 2, pages 274–279

Al-Sulaiti and Atwell 2006
AL-SULAITI, Latifa ; ATWELL, Eric S.: The design of a corpus of contemporary Arabic. In: *International Journal of Corpus Linguistics* 11 (2006), Nr. 2, pages 135–171

Albalooshi et al. 2011
ALBALOOSHI, Noora ; MOHAMED, Nader ; AL-JAROODI, Jameela: The challenges of Arabic language use on the Internet. In: *2011 International Conference for Internet Technology and Secured Transactions* IEEE, 2011, pages 378–382

Albirini 2016
ALBIRINI, Abdulkafi: *Modern Arabic Sociolinguistics: Diglossia, variation, codeswitching, attitudes and identity*. Routledge, 2016

Alghamdi et al. 2010
ALGHAMDI, Mansour ; MUZAFFAR, Zeeshan ; ALHAKAMI, Hazim: Automatic restoration of arabic diacritics: a simple, purely statistical approach. In: *Arabian Journal for Science and Engineering* 35 (2010), Nr. 2, pages 125

Alotaiby et al. 2010
ALOTAIBY, Fahad ; FODA, Salah ; ALKHARASHI, Ibrahim: Clitics in Arabic Language: A Statistical Study. In: *PACLIC*, 2010, pages 595–601

Alotaiby et al. 2014
ALOTAIBY, Fahad ; FODA, Salah ; ALKHARASHI, Ibrahim: Arabic vs. English: Comparative statistical study. In: *Arabian Journal for Science and Engineering* 39 (2014), Nr. 2, pages 809–820

Alqahtani et al. 2015
ALQAHTANI, Mofareh et al.: The importance of vocabulary in language learning and how to be taught. In: *International journal of teaching and education* 3 (2015), Nr. 3, pages 21–34

Althobaiti et al. 2014
ALTHOBAITI, Maha ; KRUSCHWITZ, Udo ; POESIO, Massimo: AraNLP: a Java-based Library for the Processing of Arabic Text. (2014)

Ananthakrishnan et al. 2005
ANANTHAKRISHNAN, Sankaranarayanan ; NARAYANAN, Shrikanth ; BANGALORE, Srinivas: Automatic diacritization of Arabic transcripts for automatic speech recognition. In: *Proceedings of the 4th International Conference on Natural Language Processing*, 2005, pages 47–54

Anastasi and Urbina 1997
ANASTASI, Anne ; URBINA, Susana: *Psychological tests*. 1997

Anderson and Freebody 1981
ANDERSON, Richard C. ; FREEBODY, Peter: Vocabulary Knowledge. In: *DOCUMENT RESUME CS 006 138 Guthrie, John T., Ed. Comprehension and Teaching; Research Reviews. International Reading Association, Newark, Del.* (1981), pages 77

Anderson and Freebody 1983
ANDERSON, Richard C. ; FREEBODY, Peter: Reading comprehension and the assessment and acquisition of word knowledge. In: *Advances in reading/language research* (1983)

Aqel et al. 2015
AQEL, Afnan ; ALWADEI, Sahar ; DAHAB, Mohammad: Building an Arabic Words Generator. In: *International Journal of Computer Applications* 112 (2015), Nr. 14

Attia 2007
ATTIA, Mohammed A.: Arabic tokenization system. In: *Proceedings of the 2007 workshop on computational approaches to semitic languages: Common issues and resources* Association for Computational Linguistics, 2007, pages 65–72

Attia 2008
ATTIA, Mohammed A.: *Handling Arabic morphological and syntactic ambiguity within the LFG framework with a view to machine translation*, University of Manchester, Diss., 2008

Awada and Dobell 2016
AWADA, Samar ; DOBELL, Brian: *Lebanese Arabic Institute Arabic Alphabet*. `https://www.lebanesearabicinstitute.com/arabic-alphabet/`, 2016. – Accessed: 2019-04-13

Azami 2011
AZAMI, Muhammad M.: *The History of the Quranic Text: From Revelation to Compilation: A Comparative Study with the Old and New Testaments*. UK Islamic Academy, 2011

Azmi and Almajed 2015
AZMI, Aqil ; ALMAJED, Reham: A survey of automatic Arabic diacritization techniques. In: *Natural Language Engineering* 21 (2015), Nr. 03, pages 477–495

Baayen et al. 1995
BAAYEN, R H. ; PIEPENBROCK, Richard ; GULIKERS, Leon: The CELEX lexical database (release 2). In: *Linguistic Data Consortium, Philadelphia* (1995)

Baba 2002
BABA, K: Test review: Lex30. In: *Language Testing Update* 32 (2002), pages 68–71

Baharudin and Ismail 2014
BAHARUDIN, Harun ; ISMAIL, Zawawi: Vocabulary learning strategies and arabic vocabulary size among pre-university students in Malaysia. In: *International Education Studies* 7 (2014), Nr. 13, pages 219

Baharudin et al. 2014
BAHARUDIN, Harun ; ISMAIL, Zawawi ; ASMAWI, Adelina ; BAHARUDDIN, Normala: TAV of arabic language measurement. In: *Mediterranean Journal of Social Sciences* 5 (2014), Nr. 20, pages 2402

Baldwin and Lui 2010
BALDWIN, Timothy ; LUI, Marco: Language identification: The long and the short of the matter. In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* Association for Computational Linguistics, 2010, pages 229–237

Balota and Chumbley 1984
BALOTA, David A. ; CHUMBLEY, James I.: Are lexical decisions a good measure of lexical access? The role of word frequency in the neglected decision stage. In: *Journal of Experimental Psychology: Human perception and performance* 10 (1984), Nr. 3, pages 340

Balota et al. 2007
BALOTA, David A. ; YAP, Melvin J. ; HUTCHISON, Keith A. ; CORTESE, Michael J. ; KESSLER, Brett ; LOFTIS, Bjorn ; NEELY, James H. ; NELSON, Douglas L. ; SIMPSON, Greg B. ; TREIMAN, Rebecca: The English lexicon project. In: *Behavior research methods* 39 (2007), Nr. 3, pages 445–459

Barouni Ebrahimi 2017
BAROUNI EBRAHIMI, Alireza: Measuring Productive Depth of Vocabulary Knowledge of the Most Frequent Words. (2017)

Bebah et al. 2014
BEBAH, Mohamed ; AMINE, Chennoufi ; AZZEDDINE, Mazroui ; ABDELHAK, Lakhouaja: Hybrid approaches for automatic vowelization of Arabic texts. In: *arXiv preprint arXiv:1410.2646* (2014)

Beinborn 2016
BEINBORN, Lisa M.: *Predicting and Manipulating the Difficulty of Text-Completion Exercises for Language Learning*, Technische Universität Darmstadt, Diss., 2016

Belinkov and Glass 2015
BELINKOV, Yonatan ; GLASS, James: Arabic diacritization with recurrent neural networks. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pages 2281–2285

Belinkov et al. 2013
BELINKOV, Yonatan ; HABASH, Nizar ; KILGARRIFF, Adam ; ORDAN, Noam ; ROTH, Ryan ; SUCHOMEL, Vıt: arTen-Ten: a new, vast corpus for Arabic. In: *Proceedings of WACL* (2013)

Bouamor et al. 2014
BOUAMOR, Houda ; HABASH, Nizar ; OFLAZER, Kemal: A Multidialectal Parallel Corpus of Arabic. In: *Proceedings of the Language Resources and Evaluation Conference (LREC)*, 2014, pages 1240–1245

Bouamor et al. 2015
BOUAMOR, Houda ; ZAGHOUANI, Wajdi ; DIAB, Mona ; OBEID, Ossama ; OFLAZER, Kemal ; GHONEIM, Mahmoud ; HAWWARI, Abdelati: A Pilot Study on Arabic Multi-Genre Corpus Diacritization Annotation. In: *ANLP Workshop 2015*, 2015, pages 80

Brown 1989
BROWN, James D.: Cloze item difficulty. In: *JALT journal* 11 (1989), Nr. 1, pages 46–67

Brysbaert 2013
BRYSBAERT, Marc: LEXTALE_FR: A fast, free, and efficient test to measure language proficiency in French. In: *Psychologica Belgica* 53 (2013), Nr. 1, pages 23–37. – ISSN 0033–2879

Buckwalter 2004
BUCKWALTER, Tim: Buckwalter Arabic Morphological Analyzer Version 2.0. Linguistic Data Consortium, University of Pennsylvania, 2002. LDC Cat alog No.: LDC2004L02 / ISBN 1-58563-324-0. 2004. – Forschungsbericht

Buckwalter and Parkinson 2011
BUCKWALTER, Tim ; PARKINSON, Dilworth: *A frequency dictionary of Arabic: Core vocabulary for learners*. Routledge, 2011

Cameron 2002
CAMERON, Lynne: Measuring vocabulary size in English as an additional language. In: *Language Teaching Research* 6 (2002), Nr. 2, pages 145–173

Cavnar et al. 1994
CAVNAR, William B. ; TRENKLE, John M. et al.: N-gram-based text categorization. In: *Ann arbor mi* 48113 (1994), Nr. 2, pages 161–175

Chan and Chang 2018
CHAN, I L. ; CHANG, Charles B.: LEXTALE_CH: A quick, character-based proficiency test for Mandarin Chinese. In: *Proceedings of the Annual Boston University Conference on Language Development* Cascadilla Press, 2018

Chapelle 1994
CHAPELLE, Carol A.: Are C-tests valid measures for L2 vocabulary research? In: *Second language research* 10 (1994), Nr. 2, pages 157–187

Chennoufi and Mazroui 2017
CHENNOUFI, Amine ; MAZROUI, Azzeddine: Morphological, syntactic and diacritics rules for automatic diacritization of Arabic sentences. In: *Journal of King Saud University-Computer and Information Sciences* 29 (2017), Nr. 2, pages 156–163

Čibej et al. 2016
ČIBEJ, Jaka ; FIŠER, Darja ; ERJAVEC, Tomaž: Normalisation, Tokenisation and Sentence Segmentation of Slovene Tweets. In: *Normalisation and Analysis of Social Media Texts (NormSoMe) Workshop Programme*, 2016, pages 5

Cole et al. 2001
COLE, Andy ; GRAFF, David ; WALKER, Kevin: Arabic Newswire Part 1 Corpus (1-58563-190-6). In: *Linguistic Data Consortium (LDC)* (2001)

Coltheart 1981
COLTHEART, Max: The MRC psycholinguistic database. In: *The Quarterly Journal of Experimental Psychology Section A* 33 (1981), Nr. 4, pages 497–505

Daller et al. 2003
DALLER, Helmut ; VAN HOUT, Roeland ; TREFFERS-DALLER, Jeanine: Lexical richness in the spontaneous speech of bilinguals. In: *Applied linguistics* 24 (2003), Nr. 2, pages 197–222

Darwish et al. 2014
DARWISH, Kareem ; ABDELALI, Ahmed ; MUBARAK, Hamdy: Using Stem-Templates to Improve Arabic POS and Gender/Number Tagging. In: *Proceedings of the Language Resources and Evaluation Conference (LREC)*, 2014, pages 2926–2931

Darwish and Mubarak 2016
DARWISH, Kareem ; MUBARAK, Hamdy: Farasa: A New Fast and Accurate Arabic Word Segmenter. In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016).* Paris, France : European Language Resources Association (ELRA), 2016. – ISBN 978–2–9517408–9–1

Darwish et al. 2017
DARWISH, Kareem ; MUBARAK, Hamdy ; ABDELALI, Ahmed: Arabic Diacritization: Stats, Rules, and Hacks. In: *Proceedings of the Third Arabic Natural Language Processing Workshop*, 2017, pages 9–17

Daxenberger et al. 2014
DAXENBERGER, Johannes ; FERSCHKE, Oliver ; GUREVYCH, Iryna ; ZESCH, Torsten et al.: DKPro TC: A Java-based Framework for Supervised Learning Experiments on Textual Data. In: *ACL (System Demonstrations)*, 2014, pages 61–66

Diab et al. 2007a
DIAB, Mona ; GHONEIM, Mahmoud ; HABASH, Nizar: Arabic diacritization in the context of statistical machine translation. In: *Proceedings of MT-Summit*, 2007

Diab and Habash 2007
DIAB, Mona ; HABASH, Nizar: Arabic dialect processing tutorial. In: *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Tutorial Abstracts* Association for Computational Linguistics, 2007, pages 5–6

Diab et al. 2007b
DIAB, Mona ; HACIOGLU, Kadri ; JURAFSKY, Daniel: Automated methods for processing arabic text: From tokenization to base phrase chunking. In: *Arabic Computational Morphology: Knowledge-based and Empirical Methods. Kluwer/Springer* (2007)

Diependaele et al. 2012
DIEPENDAELE, Kevin ; BRYSBAERT, Marc ; NERI, Peter: How noisy is lexical decision? In: *Frontiers in psychology* 3 (2012), pages 348

Diependaele et al. 2013
DIEPENDAELE, Kevin ; LEMHÖFER, Kristin ; BRYSBAERT, Marc: The word frequency effect in first-and second-language word recognition: A lexical entrenchment account. In: *The Quarterly Journal of Experimental Psychology* 66 (2013), Nr. 5, pages 843–863

Dixon 2010
DIXON, Robert M.: *Basic linguistic theory volume 2: Grammatical topics*. Bd. 2. Oxford University Press, 2010

Dukes and Habash 2010
DUKES, Kais ; HABASH, Nizar: Morphological Annotation of Quranic Arabic. In: *Proceedings of the Language Resources and Evaluation Conference (LREC)*, 2010

Duyck et al. 2004
DUYCK, Wouter ; DESMET, Timothy ; VERBEKE, Lieven P. ; BRYSBAERT, Marc: WordGen: A tool for word selection and nonword generation in Dutch, English, German, and French. In: *Behavior Research Methods, Instruments, & Computers* 36 (2004), Nr. 3, pages 488–499

El-Haj et al. 2015
EL-HAJ, Mahmoud ; KRUSCHWITZ, Udo ; FOX, Chris: Creating language resources for under-resourced languages: methodologies, and experiments with Arabic. In: *Language Resources and Evaluation* 49 (2015), Nr. 3, pages 549–580

El-Sadany and Hashish 1989
EL-SADANY, Tarek ; HASHISH, Mohamed: An Arabic morphological system. In: *IBM Systems Journal* 28 (1989), Nr. 4, pages 600–612

Elder and von Randow 2008
ELDER, Cathie ; RANDOW, Janet von: Exploring the utility of a web-based English language screening tool. In: *Language Assessment Quarterly* 5 (2008), Nr. 3, pages 173–194

Elfardy et al. 2014
ELFARDY, Heba ; AL-BADRASHINY, Mohamed ; DIAB, Mona: AIDA: Identifying code switching in informal Arabic text. In: *Proceedings of The First Workshop on Computational Approaches to Code Switching*, 2014, pages 94–101

Elhadj et al. 2014
ELHADJ, Yahya O. ; ABDELALI, Ahmed ; BOUZIANE, Rachid ; AMMAR, Adel H.: Revisiting Arabic part of speech tagsets. In: *2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA)* IEEE, 2014, pages 793–802

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

Elshafei et al. 2006
ELSHAFEI, Moustafa ; AL-MUHTASEB, Husni ; ALGHAMDI, Mansour: Statistical methods for automatic diacritization of Arabic text. In: *The Saudi 18th National Computer Conference. Riyadh* Bd. 18, 2006, pages 301–306

Eyckmans 2004
EYCKMANS, June: Measuring receptive vocabulary size. Reliability and validity of the Yes/No vocabulary test for French-speaking learners of Dutch. In: *Utrecht: LOT* (2004)

Fan et al. 2008
FAN, Rong-En ; CHANG, Kai-Wei ; HSIEH, Cho-Jui ; WANG, Xiang-Rui ; LIN, Chih-Jen: LIBLINEAR: A library for large linear classification. In: *Journal of machine learning research* 9 (2008), Nr. Aug, pages 1871–1874

Farghaly 2010
FARGHALY, Ali: *The Arabic language, Arabic linguistics and Arabic computational linguistics*. 2010

Farghaly and Shaalan 2009
FARGHALY, Ali ; SHAALAN, Khaled: Arabic natural language processing: Challenges and solutions. In: *ACM Transactions on Asian Language Information Processing (TALIP)* 8 (2009), Nr. 4, pages 14

Ferguson 1959
FERGUSON, Charles A.: Diglossia. In: *word* 15 (1959), Nr. 2, pages 325–340

Francis and Kuçera 1964
FRANCIS, W. N. ; KUÇERA, Henry: *Manual of Information to Accompany a Standard Corpus of Present-day Edited American English, for use with Digital Computers*. 1964

Francis and Kuçera 1982
FRANCIS, W. N. ; KUÇERA, Henry: Frequency analysis of English usage. (1982)

Freihat et al. 2018
FREIHAT, Abed Alhakim Ali K. ; BELLA, Gabor ; HAMDY, Mubarak ; GIUNCHIGLIA, Fausto et al.: A Single-Model Approach for Arabic Segmentation, POS-Tagging and Named Entity Recognition. In: *International Conference on Natural Language and Speech Processing ICNLSP 2018*. Algiers, Algeria, 2018

Fromkin et al. 2018
FROMKIN, Victoria ; RODMAN, Robert ; HYAMS, Nina: *An introduction to language*. Cengage Learning, 2018

Goulden et al. 1990
GOULDEN, Robin ; NATION, Paul ; READ, John: How large can a receptive vocabulary be? In: *Applied linguistics* 11 (1990), Nr. 4, pages 341–363

Graff et al. 2009
GRAFF, David ; MAAMOURI, Mohamed ; BOUZIRI, Basma ; KROUNA, Sondos ; KULICK, Seth ; BUCKWALTER, Tim: Standard arabic morphological analyzer (sama) version 3.1. In: *Linguistic Data Consortium LDC2009E73* (2009)

Green et al. 2010
GREEN, Spence ; GALLEY, Michel ; MANNING, Christopher D.: Improved models of distortion cost for statistical machine translation. In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* Association for Computational Linguistics, 2010, pages 867–875

Green and Manning 2014
GREEN, Spence ; MANNING, Christopher D.: *Arabic Natural Language Processing.* `https://nlp.stanford.edu/projects/arabic.shtml`, 2014

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

Gregori-Signes and Clavel-Arroitia 2015
GREGORI-SIGNES, Carmen ; CLAVEL-ARROITIA, Begoña: Analysing lexical density and lexical diversity in university students' written discourse. In: *Procedia-Social and Behavioral Sciences* 198 (2015), pages 546–556

Gyllstad 2013
GYLLSTAD, Henrik: Looking at L2 vocabulary knowledge dimensions from an assessment perspective—challenges and potential solutions. In: *Bardel, C., Lindqvist, C., & Laufer, B.(Eds.) L* 2 (2013), pages 11–28

Habash 2010
HABASH, Nizar: Introduction to Arabic natural language processing. In: *Synthesis Lectures on Human Language Technologies* 3 (2010), Nr. 1, pages 1–187

Habash et al. 2012
HABASH, Nizar ; DIAB, Mona T. ; RAMBOW, Owen: Conventional Orthography for Dialectal Arabic. In: *Proceedings of the Language Resources and Evaluation Conference (LREC)*, 2012, pages 711–718

Habash et al. 2007a
HABASH, Nizar ; GABBARD, Ryan ; RAMBOW, Owen ; KULICK, Seth ; MARCUS, Mitchell P.: Determining Case in Arabic: Learning Complex Linguistic Behavior Requires Complex Linguistic Features. In: *EMNLP-CoNLL*, 2007, pages 1084–1092

Habash and Rambow 2005
HABASH, Nizar ; RAMBOW, Owen: Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In: *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics* Association for Computational Linguistics, 2005, pages 573–580

Habash and Rambow 2007
HABASH, Nizar ; RAMBOW, Owen: Arabic diacritization through full morphological

tagging. In: *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers* Association for Computational Linguistics, 2007, pages 53–56

Habash et al. 2009
HABASH, Nizar ; RAMBOW, Owen ; ROTH, Ryan: MADA + TOKAN: A toolkit for Arabic tokenization, diacritization, morphological disambiguation, POS tagging, stemming and lemmatization. In: *Proceedings of the 2nd international conference on Arabic language resources and tools (MEDAR), Cairo, Egypt*, 2009, pages 102–109

Habash et al. 2016
HABASH, Nizar ; SHAHROUR, Anas ; AL-KHALIL, Muhamed: Exploiting Arabic Diacritization for High Quality Automatic Annotation. In: *Language Resources and Evaluation Conference*, 2016

Habash et al. 2007b
HABASH, Nizar ; SOUDI, Abdelhadi ; BUCKWALTER, Timothy: On arabic transliteration. In: *Arabic computational morphology*. Springer, 2007, pages 15–22

Hamed and Zesch 2015
HAMED, Osama ; ZESCH, Torsten: 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*. Poznań, Poland, 2015, pages 473–477

Hamed and Zesch 2017a
HAMED, Osama ; ZESCH, Torsten: A Survey and Comparative Study of Arabic Diacritization Tools. In: *JLCL: Special Issue - NLP for Perso-Arabic Alphabets.* 32 (2017), Nr. 1, pages 27–47

Hamed and Zesch 2017b
HAMED, Osama ; ZESCH, Torsten: The Role of Diacritics in Designing Lexical Recognition Tests for Arabic. In: *3rd International Conference on Arabic Computational Linguistics (ACLing 2017).* Dubai, UAE : Elsevier, 2017

Hamed and Zesch 2018
HAMED, Osama ; ZESCH, Torsten: Exploring the Effects of Diacritization on Arabic Frequency Counts. In: *Proceeding of the 2nd International Conference on Natural Language and Speech Processing (ICNLSP 2018).* Algiers, Algeria, 2018

Harmer 2001
HARMER, Jeremy: The Practice of English Language Teaching. Essex: Longman Group UK Limited, 2001. – Forschungsbericht. – 296 p S. – ISBN 0–582–04656–4

Harrington et al. 2006
HARRINGTON, Michael et al.: The Yes/No test as a measure of receptive vocabulary knowledge. In: *Language Testing* 23 (2006), Nr. 1, pages 73–98

Harsch and Hartig 2016
HARSCH, Claudia ; HARTIG, Johannes: Comparing C-tests and Yes/No vocabulary size tests as predictors of receptive language skills. In: *Language testing* 33 (2016), Nr. 4, pages 555–575

Hernawati 2015
HERNAWATI, M: BUILDING UP THE STUDETS'ENGLISH VOCABULARY TROUGH FANNY STORIES AT SMP NEGERI 2 DUAMPANUA KAB. PINRANG. In: *ETERNAL (English, Teaching, Learning, and Research Journal)* 1 (2015), Nr. 2, pages 201–215

Hifny 2012
HIFNY, Yasser: Smoothing techniques for Arabic diacritics restoration. In: *12th Conference on Language Engineering*, 2012, pages 6–12

Hughes 2007
HUGHES, Arthur: *Testing for language teachers*. Ernst Klett Sprachen, 2007

Huibregtse et al. 2002
HUIBREGTSE, Ineke ; ADMIRAAL, Wilfried ; MEARA, Paul: Scores on a yes-no vocabulary test: Correction for guessing and response style. In: *Language testing* 19 (2002), Nr. 3, pages 227–245

Izura et al. 2014
IZURA, Cristina ; CUETOS, Fernando ; BRYSBAERT, Marc: Lextale-Esp: A test to rapidly and efficiently assess the Spanish vocabulary size. In: *Psicológica* 35 (2014), Nr. 1, pages 49–66

Janebi Enayat et al. 2018
JANEBI ENAYAT, Mostafa ; AMIRIAN, Seyed Mohammad R. ; ZAREIAN, Gholamreza ; GHANIABADI, Saeed: Reliable Measure of Written Receptive Vocabulary Size: Using the L2 Depth of Vocabulary Knowledge as a Yardstick. In: *SAGE Open* 8 (2018), Nr. 1, pages 2158244017752221

Jarrar et al. 2016
JARRAR, Mustafa ; ZARAKET, Fadi ; ASIA, Rami ; AMAYREH, Hamzeh: Diacritic-Based Matching of Arabic Words. In: *ACM Transactions on Asian Language Information Processing.(Forthcoming)* (2016)

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

Kamimoto 2008
KAMIMOTO, Tadamitsu: *Nation's vocabulary levels test and its successors: a re-appraisal*, Swansea University, Diss., 2008

Kartal and Sarigul 2017
KARTAL, Galip ; SARIGUL, Ece: Frequency effects in second language acquisition: An annotated survey. In: *Journal of Education and Training Studies* 5 (2017), Nr. 6, pages 1–8

Ken 2003
KEN, Beatty: Teaching and researching computer-assisted language learning. In: *London and New York: Pearson Education* (2003)

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

Khalifa et al. 2016
KHALIFA, Salam ; BOUAMOR, Houda ; HABASH, Nizar: DALILA: The Dialectal Arabic Linguistic Learning Assistant. In: *Proceedings of the Language Resources and Evaluation Conference (LREC).* Portoroz, Slovenia, 2016

Khorsheed and Clocksin 1999
KHORSHEED, Mohammad S. ; CLOCKSIN, William F.: Structural Features of Cursive Arabic Script. In: *BMVC* Citeseer, 1999, pages 1–10

Klein-Braley and Raatz 1982
KLEIN-BRALEY, Christine ; RAATZ, Ulrich: Der C-Test: ein neuer Ansatz zur Messung allgemeiner Sprachbeherrschung. In: *AKS-Rundbrief* 4 (1982), pages 23–37

Kluitmann 2008
KLUITMANN, S: Testing English as a Foreign Language. Two EFL-Tests used in Germany. In: *Unpublished Doctoral Thesis (ELT), University of Albert-Ludwigs* (2008)

Kremmel and Schmitt 2018
KREMMEL, Benjamin ; SCHMITT, Norbert: Vocabulary levels test. In: *The TESOL Encyclopedia of English Language Teaching* (2018)

Lafferty et al. 2001
LAFFERTY, John ; MCCALLUM, Andrew ; PEREIRA, Fernando: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. (2001)

Landau 1959
LANDAU, Jacob M.: A Word Count of Modern Arabic Prose. (1959)

Laufer and Nation 1995
LAUFER, Batia ; NATION, Paul: Vocabulary size and use: Lexical richness in L2 written production. In: *Applied linguistics* 16 (1995), Nr. 3, pages 307–322

Lemhöfer and Broersma 2012
LEMHÖFER, Kristin ; BROERSMA, Mirjam: Introducing LexTALE: a quick and valid Lexical Test for Advanced Learners of English. In: *Behavior research methods* (2012), Juni, Nr. 2, pages 325–43. `http://dx.doi.org/10.3758/s13428-011-0146-0`. – DOI 10.3758/s13428–011–0146–0. – ISSN 1554–3528

Lessard-Clouston 2013
LESSARD-CLOUSTON, Michael: *Teaching Vocabulary.* ERIC, 2013

Levy 1997
LEVY, Michael: *CALL: context and conceptualisation.* 1997

Levy and Goldberg 2014
LEVY, Omer ; GOLDBERG, Yoav: Dependency-Based Word Embeddings. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL), Baltimore, Maryland, USA,*, 2014, pages 302–308

Li and Jurafsky 2015
LI, Jiwei ; JURAFSKY, Dan: Do Multi-Sense Embeddings Improve Natural Language Understanding? In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2015, pages 1722–1732

Little 2011
LITTLE, David: The common European framework of reference for languages: A research agenda. In: *Language Teaching* 44 (2011), Nr. 03, pages 381–393

Loth 2011
LOTH, Sebastian: *Congruency and typicality effects in lexical decision*, Royal Holloway, University of London, Diss., 2011

Maamouri et al. 2004
MAAMOURI, Mohamed ; BIES, Ann ; BUCKWALTER, Tim ; MEKKI, Wigdan: The penn arabic treebank: Building a large-scale annotated arabic corpus. In: *NEMLAR conference on Arabic language resources and tools* Bd. 27, 2004, pages 466–467

Maamouri et al. 2006
MAAMOURI, Mohamed ; BIES, Ann ; KULICK, Seth: Diacritization: A challenge to arabic treebank annotation and parsing. In: *Proceedings of the Conference of the Machine Translation SIG of the British Computer Society* Citeseer, 2006

Maamouri et al. 2009
MAAMOURI, Mohamed ; BIES, Ann ; KULICK, Seth: Creating a methodology for large-scale correction of treebank annotation: The case of the Arabic Treebank. In: *Proceedings of MEDAR International Conference on Arabic Language Resources and Tools, Cairo, Egypt*, 2009

Macmillan 2002
MACMILLAN, Neil A.: Signal detection theory. In: *Stevens' handbook of experimental psychology* (2002)

Malvern and Richards 2012
MALVERN, David ; RICHARDS, Brian: Measures of lexical richness. In: *The Encyclopedia of Applied Linguistics* (2012)

Maskor et al. 2016a
MASKOR, Zunita M. ; BAHARUDIN, Harun et al.: Receptive Vocabulary Knowledge or Productive Vocabulary Knowledge in Writing Skill, Which One Important? In: *International Journal of Academic Research in Business and Social Sciences* 6 (2016), Nr. 11, pages 261–271

Maskor et al. 2016b
MASKOR, Zunita M. ; BAHARUDIN, Harun ; LUBIS, Maimun A. ; YUSUF, Nurul K.: Teaching and Learning Arabic Vocabulary: From a Teacher's Experiences. In: *Creative Education* 7 (2016), Nr. 03, pages 482

Matthews and Wijeyewardene 2018
MATTHEWS, Joshua ; WIJEYEWARDENE, Ingrid: Exploring relationships between automated and human evaluations of L2 texts. In: *Language Learning & Technology* 22 (2018), Nr. 3, pages 143–158

McLean and Kramer 2015
McLEAN, Stuart ; KRAMER, Brandon: The creation of a new vocabulary levels test. In: *Shiken* 19 (2015), Nr. 2, pages 1–11

Meara 1990
MEARA, Paul: Some notes on the Eurocentres vocabulary tests. In: *Foreign language comprehension and production* (1990), pages 103–113

Meara 1995
MEARA, Paul: *Single-subject studies of lexical acquisition*. 1995

Meara and Buxton 1987
MEARA, Paul ; BUXTON, Barbara: An alternative to multiple choice vocabulary tests. In: *Language testing* 4 (1987), Nr. 2, pages 142–154

Meara and Jones 1990
MEARA, Paul ; JONES, G: Eurocentres vocabulary size test 10KA. In: *Zurich: Eurocentres* (1990)

Meara and Jones 1987
MEARA, Paul ; JONES, Glyn: Tests of vocabulary size in English as a foreign language. In: *Polyglot* 8 (1987), Nr. 1, pages 1–40

Meara and Wolter 2004
MEARA, PAUL ; WOLTER, BRENT: V_Links: Beyond vocabulary depth. In: *Angles on the English speaking world* 4 (2004), pages 85–96

Merhben et al. 2009
MERHBEN, Laroussi ; ZOUAGHI, Anis ; ZRIGUI, Mounir: Ambiguous Arabic Words Disambiguation: The results. In: *Recent Advances in Natural Language Processing (RANLP), Student Research Workshop, Borovets, Bulgaria*, 2009, pages 45–52

Metwally et al. 2016
METWALLY, Aya S. ; RASHWAN, Mohsen A. ; ATIYA, Amir F.: A multi-layered approach for Arabic text diacritization. In: *IEEE International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), 2016* IEEE, 2016, pages 389–393

Milton 2006
MILTON, James: Language lite? Learning French vocabulary in school. In: *Journal of French Language Studies* 16 (2006), Nr. 2, pages 187–205

Milton 2007
MILTON, James: Lexical profiles, learning styles and the construct validity of lexical size tests. In: *Modelling and assessing vocabulary knowledge* (2007), pages 47–58

Milton 2009
MILTON, James: *Measuring second language vocabulary acquisition*. Bd. 45. Multilingual Matters, 2009

Milton 2013
MILTON, James: Measuring the contribution of vocabulary knowledge to proficiency in the four skills. In: *C. Bardel, C. Lindqvist, & B. Laufer (Eds.) L* 2 (2013), pages 57–78

Mubarak 2017
MUBARAK, Hamdy: Build Fast and Accurate Lemmatization for Arabic. In: *arXiv preprint arXiv:1710.06700* (2017)

Nabil et al. 2015
NABIL, Mahmoud ; ALY, Mohamed A. ; ATIYA, Amir F.: ASTD: Arabic Sentiment Tweets Dataset. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP), Lisbon, Portugal*, 2015, pages 2515–2519

Nation 2006
NATION, I: How large a vocabulary is needed for reading and listening? In: *Canadian Modern Language Review* 63 (2006), Nr. 1, pages 59–82

Nation 2001
NATION, Ian S.: *Learning Vocabulary in another Language*. Cambridge University Press, 2001

Nation 2013
NATION, Ian S.: *Learning Vocabulary in Another Language Google eBook*. Cambridge University Press, 2013

Nation 1990a
NATION, Ian Stephen P.: Teaching and Learning Vocabulary. Teaching Methods. In: *United States: Cengage Learning, Inc.* 9 (1990)

Nation 1983
NATION, IS P.: *Testing and teaching vocabulary*. 1983

Nation 2012a
NATION, ISP: Measuring vocabulary size in an uncommonly taught language. In: *International Conference on Language Proficiency Testing in the Less Commonly Taught Languages*, 2012, 17–18

Nation and Beglar 2007
NATION, ISP ; BEGLAR, David: A vocabulary size test. In: *The language teacher* 31 (2007), Nr. 7, pages 9–13

Nation 1990b
NATION, Paul: Teaching and learning vocabulary. Rowley, MA: Newbury House. (1990)

Nation 2012b
NATION, Paul: The Vocabulary Size Test. (2012)

Nelken and Shieber 2005
NELKEN, Rani ; SHIEBER, Stuart: Arabic diacritization using weighted finite-state transducers. In: *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages* Association for Computational Linguistics, 2005, pages 79–86

Nevin 1969
NEVIN, John A.: Signal detection theory and operant behavior: A review of david m. green and john a. swets' signal detection theory and psychophysics. 1. In: *Journal of the Experimental Analysis of Behavior* 12 (1969), Nr. 3, pages 475–480

New et al. 2004
NEW, Boris ; PALLIER, Christophe ; BRYSBAERT, Marc ; FERRAND, Ludovic: Lexique 2: A new French lexical database. In: *Behavior Research Methods, Instruments, & Computers* 36 (2004), Nr. 3, pages 516–524

Nguyen and Nation 2011
NGUYEN, Le Thi C. ; NATION, Paul: A bilingual vocabulary size test of English for Vietnamese learners. In: *RELC journal* 42 (2011), Nr. 1, pages 86–99

Palmer et al. 1968
PALMER, Harold E. ; HARPER, David ; HARPER, David: *The scientific study and teaching of languages*. JSTOR, 1968

Pasha et al. 2014
PASHA, Arfath ; AL-BADRASHINY, Mohamed ; DIAB, Mona ; EL KHOLY, Ahmed ; ESKANDER, Ramy ; HABASH, Nizar ; POOLEERY, Manoj ; RAMBOW, Owen ; ROTH, Ryan: MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic. In: *Proceedings of the Language Resources and Evaluation Conference (LREC)*, 2014, pages 1094–1101

Payne 2006
PAYNE, Thomas: *Exploring language structure: A student's guide*. Cambridge University Press, 2006

Pellicer-Sánchez and Schmitt 2012
PELLICER-SÁNCHEZ, Ana ; SCHMITT, Norbert: Scoring Yes–No vocabulary tests: Reaction time vs. nonword approaches. In: *Language Testing* 29 (2012), Nr. 4, pages 489–509

Pignot-Shahov 2012
PIGNOT-SHAHOV, Virginie: Measuring L2 receptive and productive vocabulary knowledge. In: *Language Studies Working Papers* 4 (2012), Nr. 1, pages 37–45

Plisson et al. 2004
PLISSON, Joël ; LAVRAC, Nada ; MLADENIĆ, Dr et al.: A rule based approach to word lemmatization. (2004)

Plonka 2006
PLONKA, Arkadiusz: Le nationalisme linguistique au liban autour de sa'id'aql et l'idée de langue libanaise dans la revue Lebnaan en nouvel alphabet'. In: *Arabica* (2006), pages 423–471

Qwaider et al. 2018
QWAIDER, Chatrine ; SAAD, Motaz ; CHATZIKYRIAKIDIS, Stergios ; DOBNIK, Simon: Shami: A Corpus of Levantine Arabic Dialects. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, 2018

Rashwan et al. 2011
RASHWAN, Mohsen ; AL-BADRASHINY, Mohamed ; ATTIA, Mohamed ; ABDOU, Sherif ; RAFEA, Ahmed: A stochastic Arabic diacritizer based on a hybrid of factorized and unfactorized textual features. In: *Audio, Speech, and Language Processing, IEEE Transactions on* 19 (2011), Nr. 1, pages 166–175

Rashwan et al. 2015
RASHWAN, Mohsen A. ; AL SALLAB, Ahmad A. ; RAAFAT, Hazem M. ; RAFEA, Ahmed: Deep learning framework with confused sub-set resolution architecture for automatic Arabic diacritization. In: *IEEE Transactions on Audio, Speech, and Language Processing* 23 (2015), Nr. 3, pages 505–516

Rastle et al. 2002
RASTLE, Kathleen ; HARRINGTON, Jonathan ; COLTHEART, Max: 358,534 nonwords: The ARC nonword database. In: *The Quarterly Journal of Experimental Psychology: Section A* 55 (2002), Nr. 4, pages 1339–1362

Read 1993
READ, John: The development of a new measure of L2 vocabulary knowledge. In: *Language testing* 10 (1993), Nr. 3, pages 355–371

Read and Chapelle 2001
READ, John ; CHAPELLE, Carol A.: A framework for second language vocabulary assessment. In: *Language testing* 18 (2001), Nr. 1, pages 1–32

Rebai and BenAyed 2015
REBAI, Ilyes ; BENAYED, Yassine: Text-to-speech synthesis system with Arabic diacritic recognition system. In: *Computer Speech & Language* 34 (2015), Nr. 1, pages 43–60

Ricks 2015
RICKS, Robert: The Development of Frequency-Based Assessments of Vocabulary Breadth and Depth for L2 Arabic. (2015)

Ritchey 1998
RITCHEY, Tom: General morphological analysis. In: *16th euro conference on operational analysis*, 1998

Robinson 2012
ROBINSON, Peter: *The Routledge encyclopedia of second language acquisition*. Routledge, 2012

Sachs et al. 1997
SACHS, J ; TUNG, P ; LAM, RYH: How to construct a cloze test: Lessons from testing measurement theory models. In: *Perspectives* (1997)

Sadat et al. 2014
SADAT, Fatiha ; KAZEMI, Farnazeh ; FARZINDAR, Atefeh: Automatic identification of arabic dialects in social media. In: *Proceedings of the first international workshop on Social media retrieval and analysis* ACM, 2014, pages 35–40

Said et al. 2013
SAID, Ahmed ; EL-SHARQWI, Mohamed ; CHALABI, Achraf ; KAMAL, Eslam: A hybrid approach for Arabic diacritization. In: *International Conference on Application of Natural Language to Information Systems* Springer, 2013, pages 53–64

Saigh and Schmitt 2012
SAIGH, Kholood ; SCHMITT, Norbert: Difficulties with vocabulary word form: The case of Arabic ESL learners. In: *System* 40 (2012), Nr. 1, pages 24–36

Samih and Kallmeyer 2017

SAMIH, Younes ; KALLMEYER, Laura: *Dialectal Arabic Processing Using Deep Learning*, Ph. D. thesis, Heinrich-Heine-Universität Düsseldorf, Düsseldorf, Germany, Diss., 2017

Sayoud 2012

SAYOUD, Halim: Author discrimination between the Holy Quran and Prophet's statements. In: *Literary and Linguistic Computing* 27 (2012), Nr. 4, pages 427–444

Schlippe et al. 2008

SCHLIPPE, Tim ; NGUYEN, ThuyLinh ; VOGEL, Stephan: Diacritization as a machine translation problem and as a sequence labeling problem. In: *8th AMTA conference, Hawaii*, 2008, pages 21–25

Schmitt 2000

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

Schmitt 2010

SCHMITT, Norbert: *Researching vocabulary: A vocabulary research manual*. Springer, 2010

Schmitt 2014

SCHMITT, Norbert: Size and depth of vocabulary knowledge: What the research shows. In: *Language Learning* 64 (2014), Nr. 4, pages 913–951

Schmitt et al. 2011

SCHMITT, Norbert ; NG, Janice Wun C. ; GARRAS, John: The word associates format: Validation evidence. In: *Language Testing* 28 (2011), Nr. 1, pages 105–126

Schmitt et al. 2001

SCHMITT, Norbert ; SCHMITT, Diane ; CLAPHAM, Caroline: Developing and exploring the behaviour of two new versions of the Vocabulary Levels Test. In: *Language testing* 18 (2001), Nr. 1, pages 55–88

Shaalan et al. 2009

SHAALAN, Khaled ; ABO BAKR, Hitham ; ZIEDAN, Ibrahim: A hybrid approach for building Arabic diacritizer. In: *Proceedings of the EACL 2009 workshop on computational approaches to semitic languages* Association for Computational Linguistics, 2009, pages 27–35

Šišková 2012

ŠIŠKOVÁ, Zdislava: Lexical Richness in EFL Students' Narratives. In: *Language Studies Working Papers* 4 (2012), pages 26–36

Skoglund 2006

SKOGLUND, Dallas E.: *A comparison of Norwegian and American pupils' English vocabulary usage in upper secondary schools*, Diplomarbeit, 2006

Soori et al. 2013

SOORI, Hussein ; PLATOS, Jan ; SNASEL, Vaclav: Simple Stemming Rules for Arabic Language, 2013. – ISBN 978–3–642–31602–9, pages 99–108

Sorell 2012
SORELL, C J.: Zipf's law and vocabulary. In: *The Encyclopedia of Applied Linguistics* (2012)

Spolsky 1995
SPOLSKY, Bernard: *Measured words: The development of objetive language testing*. Oxford University Press, 1995

Stubbe 2012
STUBBE, Raymond: Do pseudoword false alarm rates and overestimation rates in Yes/No vocabulary tests change with Japanese university students' English ability levels? In: *Language Testing* 29 (2012), Nr. 4, pages 471–488

Syndicate 2001
SYNDICATE, UCLE: *Quick placement test*. 2001

Taft and Russell 1992
TAFT, Marcus ; RUSSELL, Bruce: Pseudohomophone naming and the word frequency effect. In: *The Quarterly Journal of Experimental Psychology* 45 (1992), Nr. 1, pages 51–71

Thorndike 1944
THORNDIKE, EL: The teacher's word book of 30,000 words. In: *Teacher's College, Columbia University, New York* (1944)

Tu et al. 2017
TU, Lifu ; GIMPEL, Kevin ; LIVESCU, Karen: Learning to Embed Words in Context for Syntactic Tasks. In: *2nd Workshop on Representation Learning for NLP* (2017)

Vatanen et al. 2010
VATANEN, Tommi ; VÄYRYNEN, Jaakko J. ; VIRPIOJA, Sami: Language Identification of Short Text Segments with N-gram Models. In: *Proceedings of the Language Resources and Evaluation Conference (LREC)* Citeseer, 2010

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

Webb 2008
WEBB, Stuart: Receptive and productive vocabulary sizes of L2 learners. In: *Studies in Second language acquisition* 30 (2008), Nr. 1, pages 79–95

Webb et al. 2017
WEBB, Stuart ; SASAO, Yosuke ; BALLANCE, Oliver: The updated vocabulary levels test. In: *ITL-International Journal of Applied Linguistics* 168 (2017), Nr. 1, pages 33–69

Webb and Sasao 2013
WEBB, Stuart A. ; SASAO, Yosuke: New directions in vocabulary testing. In: *RELC Journal* 44 (2013), Nr. 3, pages 263–277

Westbury et al. 2007
WESTBURY, Chris ; HOLLIS, Geoff ; SHAOUL, Cyrus: LINGUA: the language-independent neighbourhood generator of the University of Alberta. In: *The Mental Lexicon* 2 (2007), Nr. 2, pages 271–284

Wilkins 1972
WILKINS, David A.: *Linguistics in language teaching*. E. Arnold, 1973, 1972

Yaghan 2008
YAGHAN, Mohammad A.: "Arabizi": A contemporary style of Arabic Slang. In: *Design issues* 24 (2008), Nr. 2, pages 39–52

Zaghouani 2014
ZAGHOUANI, Wajdi: Critical survey of the freely available Arabic corpora. In: *In Proceedings of the International Conference on Language Resources and Evaluation (LREC'2014), OSACT Workshop. Rejkavik, Iceland* (2014)

Zaghouani et al. 2016
ZAGHOUANI, Wajdi ; BOUAMOR, Houda ; HAWWARI, Abdelati ; DIAB, Mona ; OBEID, Ossama ; GHONEIM, Mahmoud ; ALQAHTANI, Sawsan ; OFLAZER, Kemal: Guidelines and framework for a large scale Arabic diacritized corpus. In: *The Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, 2016, pages 3637–3643

Zaidan and Callison-Burch 2014
ZAIDAN, Omar F. ; CALLISON-BURCH, Chris: Arabic dialect identification. In: *Computational Linguistics* 40 (2014), Nr. 1, pages 171–202

Zechmeister et al. 1995
ZECHMEISTER, Eugene B. ; CHRONIS, Andrea M. ; CULL, William L. ; D'ANNA, Catherine A. ; HEALY, Noreen A.: Growth of a functionally important lexicon. In: *Journal of Literacy Research* 27 (1995), Nr. 2, pages 201–212

Zerrouki and Balla 2009
ZERROUKI, Taha ; BALLA, Amar: Implementation of infixes and circumfixes in the spellcheckers. In: *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, 2009

Zerrouki and Balla 2017
ZERROUKI, Taha ; BALLA, Amar: Tashkeela: Novel corpus of Arabic vocalized texts, data for auto-diacritization systems. In: *Data in Brief* 11 (2017), pages 147–151

Zipf 1950
ZIPF, George K.: *Human behavior and the principle of least effort*. 1950

Zitouni et al. 2006
ZITOUNI, Imed ; SORENSEN, Jeffrey ; SARIKAYA, Ruhi: Maximum entropy based restoration of Arabic diacritics. In: *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics* Association for Computational Linguistics, 2006, pages 577–584

DOI: 10.17185/duepublico/70628
URN: urn:nbn:de:hbz:464-20191017-091538-9