UNIVERSITY OF LEEDS

An Automatic Modern Standard Arabic Text Simplification System: A Corpus-Based Approach

Nouran Khallaf

Submitted in accordance with the requirements for the degree of Doctor of Philosophy

The University of Leeds
School of Languages, Cultures, and Societies

<March 2023>

Publications

The candidate confirms that the work submitted is her own, except where work which has formed part of jointly authored publications has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated below. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others.

Chapters 4 and 5 of this thesis are based on jointly authored publications. The candidate is the principal author of all original contributions presented in these papers, and the co-authors acted in an advisory capacity, providing feedback, general guidance, and comments.

(Chapter 4- Section B):

Khallaf, N., and Sharoff, S., (2021). **Automatic Difficulty Classification of Arabic Sentences.** In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 105–114, Kyiv, Ukraine (Virtual). Association for Computational Linguistics. EACL 2021

Available at: https://aclanthology.org/2021.wanlp-1.11/.

Chapter 5:

Khallaf, N., Sharoff, S., Soliman, R., (2022). **Towards Arabic Sentence Simplification via Classification and Generative Approaches.** In *Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 43–52, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics. EMNLP 2022.

Available at: https://aclanthology.org/2022.wanlp-1.5/.

Declaration

The candidate confirms that the work submitted is her own and that appropriate credit has been given where reference has been made to the work of others.

Acknowledgement

In the name of God most gracious most merciful. All praise to the almighty, all-knowing, and most wise. I am grateful to God for giving me the courage and patience to endure this journey of seeking knowledge.

I am extremely grateful to my primary supervisor, Prof. Serge Sharoff, for his exceptional guidance and support throughout my research journey. His insightful comments, constructive criticism, and tireless dedication to improving the quality of my work were instrumental in shaping the direction of this thesis.

I would also like to express my deepest appreciation to my co-supervisors, Dr.Rasha Soliman and Prof. Michael Ingelbly, for their invaluable feedback and support. Their expert guidance and encouragement were pivotal in helping me navigate the complexities of my research and in shaping the final outcome of this work.

I am also indebted to my mentor, Prof. Emad Khalil, for his valuable guidance and unwavering support throughout my academic career. His profound insights, exceptional knowledge, and willingness to share his expertise have been instrumental in shaping my intellectual growth and personal development.

My sincere thank goes to the Egyptian Ministry of Higher Education and Scientific Research and Newton-Mosharafa Fund - British Council for sponsoring this research.

I would also like to extend my gratitude to my late father, whose unwavering love, guidance, and support have been a source of inspiration and motivation for me throughout my life. Although he is no longer with us, his presence and memory have been a constant reminder of what can be achieved with hard work, dedication, and perseverance.

My heartfelt thanks also go to my dear friends, Huda Alghatani, Nouran Mohamed and Souad Boumachaal, for their unwavering support, encouragement, and understanding throughout this journey. Their constant presence, thoughtful advice, and unwavering support were a source of strength and motivation that kept me going through the most challenging times.

Moreover, I would also like to express my gratitude to my friends, Sara Bondoka, Asmaa Shehata, and Rachael. Asmaa, a childhood friend from my hometown, has been a constant source of inspiration and encouragement. Sara has been a steadfast supporter and an endless source of positive energy. Her unwavering belief in me and my abilities has been a source of motivation and inspiration that kept me going when things got tough. Rachael has been a constant source of joy and laughter. Her infectious enthusiasm and unwavering support have been a bright spot in my life, and I feel blessed to have her as a friend.

Lastly, I would like to express my sincere gratitude to my mother and my siblings - Shaza, Mayada, Jana, and Ahmed - for their unwavering love, encouragement, and support. My mother's sacrifices, dedication, and selflessness have always been a guiding light in my life, and I am immensely grateful for her unwavering love and support. My siblings, with their unconditional love and unwavering support, have always been there for me, offering encouragement and advice when I needed it the most. Their unwavering belief in my abilities has been a constant reminder of what truly matters in life and what can be achieved with hard work, dedication, and perseverance. I could not have accomplished any of this without them, and I am forever grateful for their love and support.

Abstract

This thesis brings together an overview of Text Readability (TR) about Text Simplification (TS) with an application of both to Modern Standard Arabic (MSA). It will present our findings on using automatic TR and TS tools to teach MSA, along with challenges, limitations, and recommendations about enhancing the TR and TS models.

Reading is one of the most vital tasks that provide language input for communication and comprehension skills. It is proved that the use of long sentences, connected sentences, embedded phrases, passive voices, non-standard word orders, and infrequent words can increase the text difficulty for people with low literacy levels, as well as second language learners. The thesis compares the use of sentence embeddings of different types (fastText, mBERT, XLM-R and Arabic-BERT), as well as traditional language features such as POS tags, dependency trees, readability scores and frequency lists for language learners. The accuracy of the 3-way CEFR (The Common European Framework of Reference for Languages Proficiency Levels) classification is F-1 of 0.80 and 0.75 for Arabic-Bert and XLM-R classification, respectively and 0.71 Spearman correlation for the regression task. At the same time, the binary difficulty classifier reaches F-1 0.94 and F-1 0.98 for the sentence-pair semantic similarity classifier.

TS is an NLP task aiming to reduce the linguistic complexity of the text while maintaining its meaning and original information (Siddharthan, 2002; Camacho Collados, 2013; Saggion, 2017). The simplification study experimented using two approaches: (i) a classification approach and (ii) a generative approach. It then evaluated the effectiveness of these methods using the BERTScore (Zhang et al., 2020) evaluation metric. The simple sentences produced by the mT5 model achieved P 0.72, R 0.68 and F-1 0.70 via BERTScore while combining Arabic-BERT and fastText achieved P 0.97, R 0.97 and F-1 0.97.

To reiterate, this research demonstrated the effectiveness of the implementation of a corpus-based method combined with extracting extensive linguistic features via the latest NLP techniques. It provided insights which can be of use in various Arabic corpus studies and NLP tasks such as translation for educational purposes.

Table of Contents

List of Tables

List of Figures

Transliteration Scheme

The transliteration system used in this thesis is mainly the DIN31635 (Intellibe Intellaren)[1] the romanisation scheme for Arabic. It is presented in the following table showing the Arabic letters, their equivalents in the DIN31635 system and the nearest equivalents in the IPA system.

1. Consonants

Transliteration symbols	Arabic	IPA	Transliteration symbols	Arabic	IPA
ʔ	ء	ʔ	ḍ	ض	ɮˤ
b	ب	b	ṭ	ط	tˤ
t	ت	t	ḍ̭	ظ	ðˤ
ṯ	ث	θ	ʕ	ع	ʕ
dj / j	ج	dʒ / ʒ	ġ	غ	ɣ
ḥ	ح	ħ	f	ف	f
ḳ	خ	x	q	ق	q
d	د	d	k	ك	k
ḏ	ذ	ð	l	ل	l
r	ر	r	m	م	m
z	ز	z	n	ن	n
s	س	s	h	ه	h
š	ش	ʃ	w	و	w
ṣ	ص	sˤ	y	ي	j

2. Vowels

Transliteration symbols	Arabic long vowels	IPA	Transliteration symbols	Arabic short vowels	IPA
ā	ا	ɑː / aː	a	ـَ	a / ɑ
ā	ا	æː	a	ـَ	æ
ī	ي	iː	i	ـ◌ِـ	i
ē	ي	eː	e	ـِ	e
ū	و	uː	u	ـُ	u
ō	و	oː	o	ـُ	o

[1] http://www.intellaren.com/intellibe

List of Abbreviations

ACTFL	American Council on the Teaching of Foreign Language Proficiency Levels.
ALC	Arabic Learner Corpus
APE	Automatic Post-Editing
ATS	Automatic Text Simplification
ARA	Automatic Readability Assessment
I-AR	Arabic Internet Corpus
BERT	Pre-training of Deep Bidirectional Transformers
CEFR / CEFRL	The Common European Framework of Reference For Languages Proficiency Levels
CWI	Complex Word Identification
GLOSS	Global Language Online Support System Corpus
L1	First Language
L2	Second Language
LS	Lexical Simplification
LM	Language Model
MT	Machine Translation
MSA	Modern Standard Arabic
NLP	Natural Language Processing
NLG	Natural Language Generation
NMT	Neural Machine Translation
POS	Part Of Speech
SG	Substitution Generation
SMT	Statistical Machine Translation
SR	Substitution Ranking
SS	Substitution Selection
TESOL	Teaching English as A Second Language
TR	Text Readability
TS	Text Simplification
T5	Text-to-Text Transfer Transformer
WE	Word Embeddings
WSD	Word Sense Disambiguation

Chapter one: Introduction

Reading is one of the most vital tasks that provides language input, communication, and comprehension skills; it is essential for any human to interact with others and the world. In addition, it directly influences speaking and writing production skills (Al-Ajlan et al., 2008). In fact, these words maybe unnecessary we process reading skills in everyday life for different purposes, ranging from reading a train timetable to various types of articles and books (e.g., news articles and academic books). Text can often be complex and challenging to read; each time, the reader faces several difficulties, as a person's language literacy level is variable. It has been proved that the use of long sentences, connected sentences, embedded phrases, passive voice, non-standard word orders, and infrequent words can positively affect sentence readability and increase the text difficulty for people with low literacy levels, as well as second language learners (Siddharthan, 2004; Beigman Klebanov et al., 2004; Devlin and Unthank, 2006; Gasperin et al., 2009).

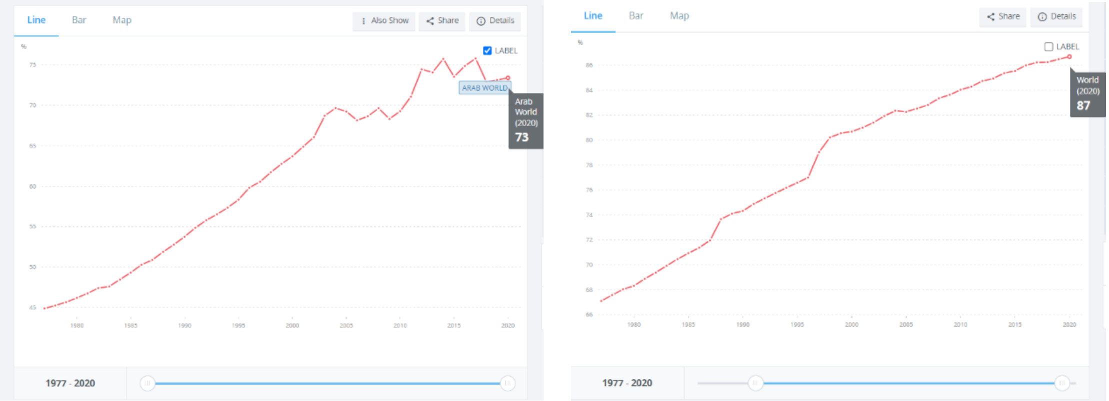


Figure 1.1 Literacy rate, adult total (% of people ages 15 and above) – in Arab versus World

In 2020 as shown in Figure 1.1, the average literacy rate of adults aged 15 and above is 73% in Arab countries compared with a rate of 87% in the rest of the world[2]. This indicates that the percentage of illiteracy is approximately double the average in the rest of the world. However, there is not a defined

[2] According to UNESCO: https://data.worldbank.org/indicator/SE.ADT.LITR.ZS?locations=1A

percentage of people with low literacy as this is embedded in the literacy percentage. Also, there is a correlation between the percentages of illiteracy and low literacy levels.

In the last century, measuring the complexity, difficulty, and readability of text has gained interest from various perspectives, including education, psychology, and linguistics. Thus, several definitions of text readability outline distinct perspective and disciplines (Cavalli-Sforza et al., 2018). Since the definition of Text Readability (TR) varies among researchers, it is essential to clarify how the term has varied over time. One of the earliest definitions (Dale & Chall in (Collins-Thompson, 2014)) was "the sum of all elements in textual material that affect a reader's understanding, reading speed, and level of interest in the material". It was suggested that several interlaced readability elements are present in a text. They have been grouped into three main categories: the linguistic properties of the text, text characteristics, and the reader's aspects. The linguistic properties of the text included sentence structure, sentence length, and semantic features (e.g., the number of ideas and the fluency in explaining them). The text characteristics involved text format, writing style, and graphical and illustration adjuncts. The characteristics of a text can vary depending on the intended audience. From a psychological standpoint, the reader characteristics factor plays a critical role in understanding the text that relies on the reader's background knowledge, experiences, interests, age, and literacy level. All the categories affect the interaction between the reader and a given text, which consequently influences the overall comprehension of a text (Tamimi et al., 2014; Collins-Thompson, 2014; Cavalli-Sforza et al., 2018). By measuring text readability, writers can identify areas where they can simplify their writing to make it more accessible to readers. This is particularly important when writing for a diverse audience, including those with limited literacy skills, non-native speakers of the language, or people with disabilities

Although differences of opinion still exist, there appears to be some agreement that TR is the property of a given text to be readable and easy to comprehend by its readers, investing reasonable time and reasonable effort (Cavalli-Sforza et al., 2018). Measuring text readability is an essential step in simplifying a text.

Hence, Automatic Text Simplification (ATS) has attracted various Natural Language Processing (NLP) researchers (Gasperin et al., 2009). Text Simplification (TS) is an NLP task aiming to reduce the linguistic complexity of the text while maintaining its meaning and original information (Siddharthan, 2002; Camacho Collados, 2013; Saggion, 2017). In other words, it reformulates the text to make it more explicit, readable, and understandable for human users and NLP tools.

Building on previous research (presented in Chapters 2 and 3), the present study aims to shed light on both Modern Standard Arabic (MSA/Arabic[3]) sentence simplification and readability classification methods.

Shardlow (2014) states that the TS task might include lexical and/or syntactic simplification to produce a new equivalent text which conveys the same meaning and message with simpler words and structure. As defined, TS involves text transformation with new lexical items and/or rewriting sentences to ensure both their readability and understandability for the target audience (Bott et al., 2012). This definition also suggests that TS could be classified as a type of Text Style Transfer (TST), where the target style of the generated text is “simple” (Jin et al., 2021).

Some scholars believe that the automation of the TS task is challenging since the concept of easy-to-read is not universal (Petersen and Ostendorf, 2007; Vickrey and Koller, 2008). However, Camacho Collados (2013) approaches TS differently by considering that a slightly simplified text for a specific target user is generally more straightforward for other users. However, a profound simplification for particular user may lead to a more complex text for another. However, most research has been providing promising attempts to reach this goal.

Accordingly, the TS task varies depending on the final application or the target audience. Hence, there are various types of simplification systems based on the purpose and who is the end-user of the system. A reasonable approach to tackle this issue could be to follow a general simplification strategy. There are three critical aspects of the simple text:

[3] "Arabic or MSA" is used in the rest of this thesis, referring to Modern standard Arabic language variety.

(i) It is made up of common simple words, simple sentences, and direct language.

(ii) Unnecessary information is omitted.

(iii) It can be shorter by the number of words but with a large number of sentences (Bott et al., 2012; Camacho Collados, 2013).

1.1 Motivations and significance of the study

The primary motivation for this research lies in the potential of Natural Language Processing (NLP) techniques to enhance and aid speech pathology, particularly for individuals with special needs. This realization, along with the lack of inclusion methods for dyslexic children in Arabic schools, initiated a journey towards exploring text simplification (TS) techniques utilized in other languages, and identifying a noticeable gap in the Arabic language.

Evidence suggests the importance of TS involves: (i) its usage in designing and simplifying the language curriculum for both second and first-language learners, in making text easy to read for first-language early learners; in assisting first-language users with cognitive impairments and low literacy language level; and (ii) being a fundamental pre-process in NLP applications such as text retrieval, extraction, summarisation, categorisation and translation (Saggion, 2017);

The actual simplification system is also language-dependent, given that rules are usually defined based on linguistic features. Arabic is a highly morphologically rich language with a flexible word order, making it difficult to identify the correct word boundaries and tokenization. Additionally, Arabic has more than average of multifunctional nouns, which can take on different grammatical roles depending on their context. This ambiguity makes it challenging to identify the correct syntactic structure of a sentence.

Furthermore, Arabic lacks vocalization diacritics in most text, which makes it challenging to disambiguate homographs and identify the correct meaning of a word. These factors make automatic Arabic text simplification a challenging task that requires a deep understanding of the language's morphology, syntax, and semantics.

While there has been relatively less research on Arabic TS compared to English, there have been some efforts to develop Arabic ATS. These systems often rely on rule-based approaches, machine learning techniques, or a combination of both to simplify the language. However, the current state-of-the-art systems for Arabic text simplification still face significant challenges and limitations.

There is an unreleased prototype system by Al-Subaihin and Al-Khalifa (2011) at King Saud University which is inaccessible, and another starting project by Al Khalil et al. (2017) at New York University in Abu-Dhabi, both systems will be discussed later in Chapter 3. In addition to these limitations, there is a general shortage of Arabic resources, namely, datasets and Arabic NLP tools.

Therefore, developing the first published Arabic text simplification system would be a significant achievement in the field, and it could pave the way for further research and development in this area.

1.2 Aims and objectives

The primary aim of this study is to build an Automatic Arabic TS system using robust NLP techniques. The other essential aims of the thesis are as follows:

- Provide a measuring algorithm to classify the linguistic complexity and readability of the Arabic text
- Provide a set of Arabic NLP resources essential for TR and TS
- Provide a system for Arabic TS, which generates easy-to-read Arabic text.

The objectives of the thesis are as follows:

1- To investigate how text complexity/readability can be measured
2- To explore possible approaches to simplify the Arabic text on lexical and syntactic levels
3- To investigate why some texts are more challenging to simplify than others

In contrast to previous research in the field of TR and TS, which has often focused on analysing the overall readability of a text, this research took a more focused approach. Specifically, the research centred on analysing individual sentences

1.3 Research questions

As mentioned earlier, there is a general lack of Arabic resources, including datasets and Arabic NLP tools. So, this research would deliver the first published Arabic TS system. Therefore, the research questions that need to be investigated are:

1. How can text complexity/readability be measured?
2. What are the text components that lead to lexical and syntactic complexity?
3. What are the principles of Arabic TS?
4. Why are some texts difficult to simplify?
5. What are the representations and methods for successful Arabic TS models?

This PhD research project delivers a readability measuring tool and a text simplification tool that a wide range of users can use; particularly, learners of Arabic as a foreign language since this tool will assist them in understanding complex Arabic texts leading them to master the Arabic language. It can also help other groups of people, including children, the functionally illiterate, and people with cognitive disabilities, and in such cases, the tool will make their lives easier, helping them to simplify complex Arabic text and make it easily read and understood. This tool would also be a precursor application to simplify Arabic text before translating it.

1.4 Research approach

The research aims to improve the Arabic TS methodology by adopting a hybrid approach that combines machine learning and rule-based techniques. This approach will take advantage of the extensive Arabic corpora that are available and freely accessible, including written Modern Standard Arabic (MSA) and Arabic vocabulary lists, as shown in Table 1.1. By using machine learning techniques, the proposed model will be able to learn from large amounts of data and identify patterns and relationships that can be used to predict the readability and complexity of Arabic texts. Rule-based techniques will also be incorporated to ensure that the model adheres to linguistic rules and guidelines that are specific to Arabic language.

Overall, the proposed hybrid approach has the potential to significantly improve the Arabic TS methodology by providing a more accurate and reliable way to assess the readability and complexity of Arabic texts. This could have important implications for a range of applications, from education and language learning to content creation and information dissemination.

Table 1.1 Summary of the data structure that will be used in the research

Resource	**Number of Tokens**	**Number of Files**	**Number of Sentences**
Vocabulary lists			
Buckwalter list (Buckwalter and Parkinson, 2014)	5000	1 excel sheet	
KELLY's list (Kilgarriff et al., 2014a)	9000	1 excel sheet	
Al-Kitaab fii TaAallum al-Arabiyya (Al-Kitaab)(Brustad et al., 2013)	4024	1 excel sheet	
Corpora			
GLOSS [4]		274 files	7832
Arabic learner corpus (ALC) (Alfaifi and Atwell, 2013)	282,732	1585 Both text and XML files	The average length of a text is 178 words
A random snapshot of Arabic Internet Corpus (I-AR)(Sharoff, 2006)		241,659 text files	Selected 8627 sentences
Arabic parallel corpus (Al-Raisi et al., 2018)	3,991,928	1 text file	100,000

[4] https://gloss.dliflc.edu/

1.5 Thesis contributions

The following subsections provide a precise summary of the main dimensions in which the current research aims to make original and innovative linguistic and computational contributions.

One of the primary objectives is to develop a framework for Arabic TS using hybrid techniques derived from various methodologies. It uses state-of-the-art corpora along with the new simplification of Arabic resources. Prior to that, it will provide a readability method to measure Arabic sentence complexity.

The study also aims to apply extensive evaluation methods to validate the proposed readability and simplification approaches. These evaluation methods will measure the efficiency and usefulness of the proposed techniques in real language learning and NLP applications.

1.5.1 Language resources

In the course of this PhD research, I have developed three significant resources that contribute to the field of Arabic Language studies. These resources were developed to address gaps in current knowledge and offer accessible tools for researchers, educators, and learners alike.

1. **Arabic Vocabulary List:** One of the major contributions of my research is the development of an expansive Arabic vocabulary list. This list encompasses 8,834 unique words, each classified according to the Common European Framework of Reference for Languages (CEFR) proficiency levels. This provides an accessible, open-source online language resource that can be used to guide vocabulary acquisition and proficiency assessment in Arabic as a second language.

2. **Arabic Sentence Corpus:** Building on the vocabulary list, I have also created an Arabic sentence corpus that aligns with the CEFR guidelines. This corpus consists of 16,045 sentences, which have been automatically classified for readability using a novel system proposed as part of this research[5]. The creation of this resource advances the field's ability to

[5] https://github.com/Nouran-Khallaf/Arabic-Readability-Corpus

assess readability and difficulty in Arabic texts and offers a valuable tool for language instruction and curriculum development.

3. **Saqq Al-Bambu parallel Corpus:** The final resource created as part of this research is the Saqq Al-Bambu corpus, a unique compilation of 2,980 parallel complex/simple Arabic sentences. While this resource is subject to copyright restrictions and will not be publicly available, it provides an innovative approach to language study, offering parallel sentence structures to facilitate comprehension and learning.

Together, these resources form a substantial contribution to Arabic language research and teaching methodologies. They serve as a testament to the potential of rigorous, focused academic research to create tangible resources that aid in the understanding and acquisition of the Arabic language. Through the continued use and development of these resources, I believe we can continue to advance our understanding of Arabic language proficiency and teaching.

1.5.2 Arabic TR and TS models

This research provides two different models for the TS system, which is the ultimate aim of this project. The simplification study experimented using two approaches: (i) a classification approach leading to Lexical Simplification (LS) pipelines which use Arabic-BERT (Safaya et al., 2020), a pre-trained contextualised model, as well as a model of fastText word embeddings (Grave et al., 2018); and (ii) a generative approach, a Seq2Seq technique by applying a multilingual Text-to-Text Transfer Transformer mT5 (Xue et al., 2021) focus more on syntactic simplification. The simple sentences produced by the mT5 model achieved P 0.72, R 0.68 and F-1 0.70 via BERTScore while combining Arabic-BERT and fastText model achieved P 0.97, R 0.97 and F-1 0.97.

In addition, the research provides an Arabic sentence difficulty classification system, which predicts the difficulty of sentences for language learners using either the CEFR proficiency levels or the binary classification. The accuracy of our 3-way CEFR classification is F-1 of 0.80 and 0.75 for Arabic-Bert and XLM-R classification, respectively, and 0.71 Spearman correlation for regression. Our binary difficulty classifier reaches F-1 0.94 and F-1 0.98 for the sentence-pair semantic similarity classifier. This classifier would be another resource that could be used by researchers or Arabic second-language tutors to select the appropriate text for their purposes. These applications will pave the way for the extension and consistent improvement of the current research project and future work.

1.6 Overview of this research project

After this current chapter, the introduction, this thesis consists of five more chapters:

- **Chapter Two: Literature review (Text Readability)**

The second chapter will focus on the literature review related to Text Readability (TR) and how it can be measured. It will begin by providing a general background on TR and different approaches to measuring it. The chapter will then explore the history of measuring TR, from traditional formulae to the automation of TR assessment.

The focus of the chapter will then shift to Automatic Text Readability (Automatic TR) applications, methods, and evaluations specifically related to the Arabic language. It will review the available resources for Arabic ARA, such as wordlists and corpora, and explore the challenges and opportunities associated with developing Automatic TR applications for Arabic.

The chapter will also provide an overview of different Machine Learning (ML) algorithms that have been used to develop Automatic TR models targeting either first or second learners of the Arabic language. The strengths and weaknesses of these different approaches will be discussed, and their potential applications will be explored. Overall, the literature review presented in this chapter will provide a comprehensive understanding of the current state of the art in Arabic Automatic TR and the different approaches that have been used to develop Automatic TR models targeting either first or second learners of the Arabic language.

- **Chapter Three: literature review (Text Simplification)**

The chapter will begin by reviewing the state of the art in TS and exploring different approaches to TS. The chapter will then present a comprehensive literature review of significant studies related to TS in various languages, with a particular focus on Arabic. It will provide an overview of the techniques and methods used in TS processes, including sentence splitting, lexical simplification, and paraphrasing.

Additionally, the chapter will describe the manual and automatic evaluation techniques used to evaluate the effectiveness of automated TS systems. It will explore the different metrics used to evaluate the quality of simplified texts, including grammaticality, fluency, and readability.

- **Chapter Four: Arabic Sentence Readability**

This chapter will present the process of understanding which methods improve an Arabic Sentence Readability classification by describing the resources and techniques used. So, this chapter is divided into two sections as follows:

  - **Section A: Datasets and tools**

This section will describe the building of Arabic resources that are used in this research in performing a series of experiments. First, it will provide a complete description of a new Arabic vocabulary list classified against CEFR levels. Second, it will provide a sentence-level complexity annotated corpus, built using a combination of available Arabic readability classified corpora. Finally, it will present an Arabic parallel simple/complex sentence corpus compiled from a novel.

  - **Section B: Arabic sentence difficulty classifier**

In this section, I present a new MSA Sentence difficulty classifier, which predicts the difficulty of sentences using either the CEFR proficiency levels or the binary classification as simple or complex. First, it will compare the use of sentence embeddings of different kinds (fast- Text, mBERT, XLM-R and Arabic-BERT), as well as traditional language features such as Paer of Speech (POS) tags, dependency trees, readability scores and frequency lists for language learners. Then, it will provide an error analysis that results in improving the sentence complexity annotated corpus. Additionally, this chapter will evaluate these different methods and select the best-performing classifier to be used later in the following chapter.

- **Chapter Five: Using neural methods to detect and simplify difficult sentences.**

This chapter will present an attempt to investigate various methods to understand how to reach a reliable MSA sentence-level simplification model. The main objective of this chapter is to investigate different methods for developing a reliable MSA sentence-level simplification model. The chapter is divided into three sections.

First, it explains the framework of sentence simplification using two approaches, namely a classification approach and a generative approach. The classification approach involves LS (Lexical Simplification) pipelines that use Arabic-BERT, a pre-trained contextualized model, as well as a model of fastText word embeddings. The generative approach uses a Seq2Seq technique by applying a multilingual Text-to-Text Transfer Transformer (mT5) and OpenNMT approach.

Second, the chapter describes the attempt to compile a simple/complex parallel Arabic corpus, which can be used to train and evaluate the simplification model. Finally, it discusses the evaluation results of the developed models, including manual and automatic evaluations. The aim is to identify the best-performing model that can be used in the next chapter for developing a TS system.

- **Chapter Six: Summary and conclusion**

This chapter reflects on the significant contributions made through this research. This chapter aims to summarise the key findings and insights gained from this thesis, which focused on the critical areas of text simplification and text readability. The thesis has explored and analysed several existing methods and techniques and proposed new models to enhance the readability and clarity of complex texts.

Moving forward, I have identified potential areas of future work, which will build on the current research and further improve the performance of text simplification and readability models. I plan to investigate the impact of incorporating new linguistic features and domain-specific knowledge into the models, along with exploring new approaches to address the challenges of multi-level simplification.

Despite the significant progress made, there are still limitations and challenges that need to be addressed in future research. I have discussed these in detail in this chapter, which include issues related to the quality of simplification outputs, the lack of resources for evaluation and the need for more extensive testing and analysis. I believe that addressing these challenges will be crucial for enhancing the performance of text simplification and readability models. In conclusion, this thesis has contributed significantly to the field of text simplification and readability. The proposed models have demonstrated promising results, and presented several areas of future work to further improve TS and TR performance. I hope that the findings and recommendations will inspire further research and development in this critical area, ultimately leading to better accessibility and understanding of Arabic complex texts for all.

Chapter Two: Literature review (Text Readability)

> "*Research problem a human being pondering the nature of language is not unlike a snowman attempting to comprehend the nature of snow, for the snowman's instruments of cognition are no less snowy than the human beings are wordy*". Seamus Heaney, speaking at the 1982 IRA World Congress on Reading, Dublin, Ireland(Dreyer, 1984).

It was initiated from the previous quote trying to understand the natural language, the nature of the text, and the contextual understandability of any given text. This chapter will provide a general background to ***Text Readability*** (TR). First, it will provide different definitions of TR across various perspectives. Then provide an overview of measuring text readability presented in traditional formulae and Machine Learning (ML) approaches for automatic text readability. Then it explores the resources and applications for measuring text readability in different languages. Then shed light on Arabic resources and tools, especially word lists and corpora, that will be used in the proposed Arabic TR model.

2.1. Text readability

Text Readability is the degree to which a text can be understood (Klare, 2000). The readability studies focus on the relationship between a given text and the cognitive burden of a reader. Many elements influence this intricate relationship, including lexical and syntactic complexity, discourse cohesiveness, and previous knowledge (Crossley et al., 2017). Thus, the primary purpose of readability studies is to measure the level of the comprehensibility of a text in connection with reader understandability (Zamanian and Heydari, 2012, p.45). In addition, measuring text readability aims to grade the text's difficulty or ease. In the last century, measuring text *complexity/difficulty/readability* gained interest from various perspectives, including education, psychology, and linguistics. Thus, several definitions of text readability depend on the perspective and discipline (Cavalli-Sforza et al., 2018).

One of the earliest and possibly the most comprehensive definitions is by Dale and Chall (1948, p.5) "*The total (including all the interactions) of all elements within a given piece of printed material that affect the success a group of readers* *has with it. Success is the extent to which they understand it, read it at an optimal speed, and find it interesting*." The following is another definition that conveys a similar meaning, was *"the sum of all elements in textual material that affect a reader's understanding, reading speed, and level of interest in the material*" (Collins-Thompson, 2014). According to Richards et al. (1992, p.306), as Cited in Zamanian and Heydari (2012, p.45), readability means: *"how easily written materials can be read and understood. TR depends on several factors, including the average length of sentences, the number of new words contained, and the grammatical complexity of the language used in a passage".* While Mc Laughlin (1969), in compiling the SMOG "'Simple Measure of Gobbledygook" readability formula, defined readability as, "*the degree to which a given class of people find certain reading matter compelling and comprehensible*."

It is suggested that there are several interlaced readability elements present in a text. They have been grouped into three main categories: the linguistic properties of the text, text characteristics, and the reader's characteristics. The linguistic properties of the text included sentence structure, sentence length, and semantic features (e.g., a comma is better than a full stop for identifying the number of ideas and fluency in explaining these ideas). The text characteristics involved format, writing style, and graphical and illustration adjuncts. Text characteristics vary according to the targeted reader. From a psychological standpoint, the reader characteristics factor plays a critical role in understanding the text that relies on the reader's background knowledge, experiences, interests, age, and literacy level. All the categories affect the interaction between the reader and a given text, which consequently influences the overall comprehension of a text (Tamimi et al., 2014; Collins-Thompson, 2014; Cavalli-Sforza et al., 2018). Shardlow (2014) identified the difference between readability and understandability, treating them independently using linguistic factors that affect their measuring score. He noted that *"Readability defines how easy to read a text maybe" and "Understandability is the amount of information a user may gain from a piece of text."* (Shardlow, 2014). Shardlow indicated that the main factors for text readability are linguistic properties, including sentence structure and the language used. In contrast, factors for text understandability involve background knowledge, the ability and experience of the reader, and similar characteristics. This differentiation agrees broadly with the earliest text readability definition.

More recently, the Longman Dictionary of Language Teaching and Applied Linguistics defined Readability as "*how easily written materials can be read and understood, depending on many factors, including the average length of sentences in a passage, the number of new words a passage contains, and the grammatical complexity of the language used*."(Richards and Schmidt, 2002, p.453). Although differences in TR views still exist, there is some agreement that TR is the attribute of a given text to be readable and easily comprehended by its readers, investing reasonable time and reasonable effort (Cavalli-Sforza et al., 2018).

In that sense, the TR score is a combination of its sentences' readability measurements. However, in principle, not all sentences presented in a complex text are complex or equal in complexity. Therefore, the best way to identify the complex factors in any text is to investigate the complexity of its components separately, which suggests measuring individual sentences' complexity rather than the overall text complexity. In this way, it is easier to identify the complex components and give an actual representation of text complexity.

The importance of TR lies in establishing well-defined standards for readability measurements based on the diversity of readers' intellectual abilities*. In Arabic, TR* has been generally utilised in education (Al-Ajlan et al., 2008; Tamimi et al., 2014; Collins-Thompson, 2014) to select the appropriate text for a student's level from primary education until high-level training. The writers also use it to ensure that their writing matches the reading proficiency of the target reader (Tamimi et al., 2014). Moreover, TR was not restricted to the field of education but was also considered in writing medical prescriptions and instructions, mainly when they targeted a diverse population with different literacy levels to ensure clear health awareness. In addition, it is applied in industry and business when writing manuals, system documentation, and guides–targeting specific consumers. Furthermore, governmental agencies, to assure clarity and accessibility, apply text readability to their texts addressed to citizens with different language proficiency levels (Collins-Thompson, 2014; Saddiki et al., 2015).

Schriver (1990) work provides a comprehensive exploration of the theoretical research on readability, which aims to identify the key factors that impact a reader's understanding of a text and assess its level of difficulty based on cognitive levels. In particular, Schriver's focus is on the psychological aspect of text readability. While defining eight primary cognitive levels, the brain would follow to understand the text as a reader and a writer evaluate the text to refine the errors affecting text coherence. Schriver (1990) also presented a refined hypothesis of a reader comprehending a text in Figure 2.2, also explaining the cognitive process of text evaluation (done by a writer in this case) to make the text easier to comprehend, as shown in Figure 2.1. These hypotheses were derived and modified from fundamental research by Thibadeau et al. (1982) and Hayes et al.(1989). Reader comprehension and writer evaluation cognitive levels are almost the same, while in the writer evaluation, another step is added to consider the reader's needs. Reading to evaluate involves understanding and criticising the text's effectiveness for the target audience. Following four steps to judge a text: 1) Detecting the error; 2) Diagnosing or characterising or explaining text problems; 3) electing strategies among various methods; 4) Fixing problems by taking action to solve them. Therefore, when reading for evaluation, the author consciously looks for problematic text features and seeks alternative solutions. For any piece of writing, we must ensure the author's message is well received by the target audience. Theoretically, the optimal readability measure would consider the psychological factor that affects understandability. At the same time, the desired text simplifier would be able to evaluate the text according to these criteria to define the errors and find the appropriate solutions. However, till now, the gap between the TR psychological facts and the computational methods measuring text readability still exists. According to Tamimi et al. (2014), "*readability level is an important indication to determine the possible audiences of a written text and to evaluate the desired impact on its readers*". Therefore, there is a need to determine the lexical and syntactic features of the Arabic language that affect the readability of the Arabic text.

The rest of this chapter is an overview of research on text readability and developing automatic computational models for readability assessment and offering a comprehensive analysis of different formulae (Section 2.3), neural approaches to readability classification, reviewing readability research for English (Section 2.4), and some European languages while focusing on Arabic and identifying their performance, achievements, and limitations towards the current state-of-the-art of readability assessment architecture. It is based on three central comprehensive automatic readability assessment surveys Collins-Thompson (2014) for English, Cavalli-Sforza et al. (2018) for Arabic, and a recent survey by Vajjala (2021). It sketchily provides ways to improve the current state of Arabic readability research. Finally, identifying some challenges for future research.

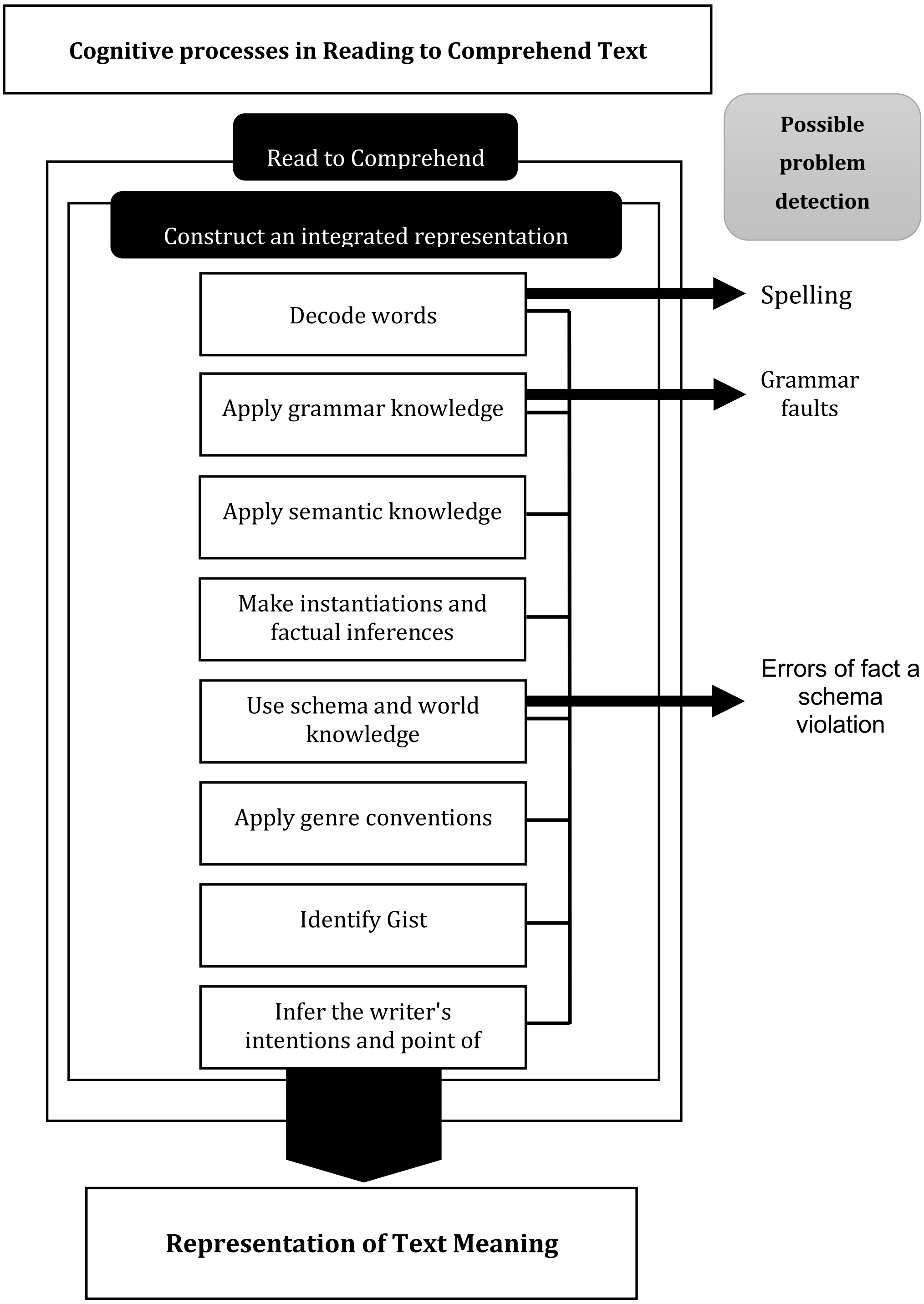

Cognitive processes in Reading to Comprehend Text
Possible problem detection
Read to Comprehend
Construct an integrated representation
Decode words
Apply grammar knowledge
Apply semantic knowledge
Make instantiations and factual inferences
Use schema and world knowledge
Apply genre conventions
Identify Gist
Infer the writer's intentions and point of
Spelling
Grammar faults
Errors of fact a schema violation
Representation of Text Meaning

Figure 2.1 The process of reading to comprehension (Schriver, 1989)

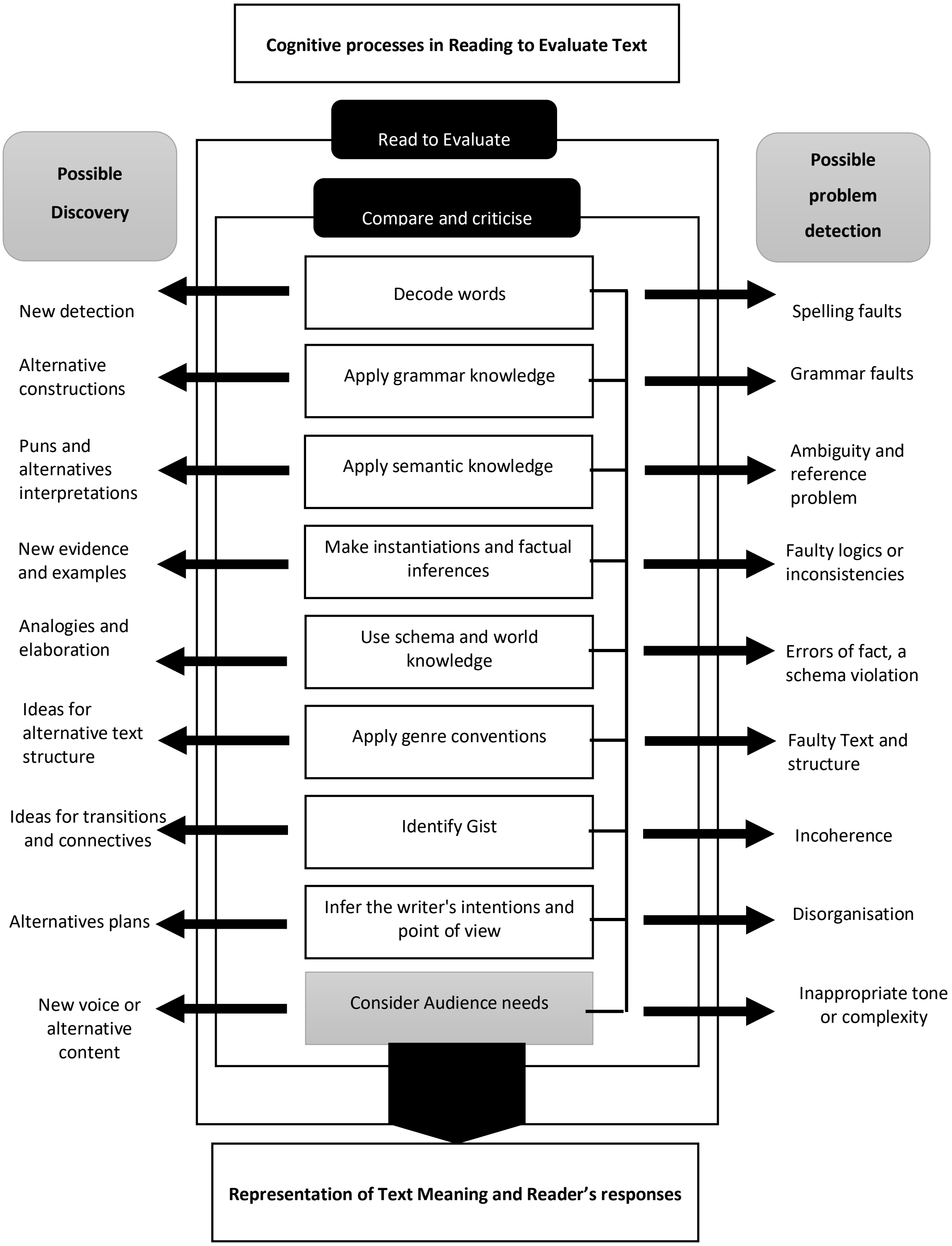

Cognitive processes in Reading to Evaluate Text
Read to Evaluate
Compare and criticise
Possible Discovery
Possible problem detection
Decode words
Apply grammar knowledge
Apply semantic knowledge
Make instantiations and factual inferences
Use schema and world knowledge
Apply genre conventions
Identify Gist
Infer the writer's intentions and point of view
Consider Audience needs
New detection
Alternative constructions
Puns and alternatives interpretations
New evidence and examples
Analogies and elaboration
Ideas for alternative text structure
Ideas for transitions and connectives
Alternatives plans
New voice or alternative content
Spelling faults
Grammar faults
Ambiguity and reference problem
Faulty logics or inconsistencies
Errors of fact, a schema violation
Faulty Text and structure
Incoherence
Disorganisation
Inappropriate tone or complexity
Representation of Text Meaning and Reader's responses

Figure 2.2 The process of reading to evaluate the text quality(Schriver, 1989)

2.2. Measuring text readability

Early attempts in measuring how difficult a text is focused on creating lists of complex words followed by developing a "traditional formula" for readability which is a simple weighted linear function of easy-to-calculate variables such as number/length of syllables/words/sentences in a text, percentage of complex words, etc. Thorndike (1921) presented one of the first English frequency lists containing 10,000 words based on various resources and classified against complexity levels to be used in teaching. Vogel and Washburne (1928) classified the children's textbooks based on the children's different reading abilities.

They initiated measuring the difficulty of sentence structure, not only the difficulty of words. The last two decades have seen many efforts to develop readability formulae, especially for the English language. These formulae attempt to decode the text's complexity elements to measure the readability and allocate the text against a pre-defined readability scale. Since the early beginnings across different disciplines considering the readability factor, the linguistic indicators of word and sentence length have remained the main factors of modern readability formulas. Psychologist Kitson (1921), in his psycholinguistic study in *The Mind of the Buyer*, in which he demonstrated how and why readers of various magazines and newspapers differed. He confirmed that the number of syllables in a word and the average sentence length were strong predictors of readability.

Historically, readability in texts has been assessed using statistical readability formulae, which attempt to determine the correlation with the level of readability. The readability formulae are defined by Kondru (2006), *"A readability formula is an equation that gives an estimate of the readability of a text. The estimate is generally in terms of the number of years of education one needs to have to comprehend that text"* (Kondru, 2006, p.7). Many traditional readability metrics are linear models with a few (often two or three) predictor variables based on superficial properties of words, sentences, and documents. These shallow features include the average number of syllables per word, average sentence length (ASL), or binned word frequency, but they also include other statistical parameters such as word complexity. As cited in Zamanian and

Heydari (2012), the first attempt to develop a method for measuring vocabulary in textbooks and other reading materials used for school was by Lively and Pressey in 1923. They aimed to establish a mechanism for quantifying vocabulary in textbooks and other school-related reading materials by relating the difficulty of a word to its frequency. However, because they did not give a scale to interpret the readability levels, their technique was not used for assessing readability; instead, their study signalled the beginning of work on readability formulae.

2.3. Readability formulae

The last two decades have seen several efforts to develop readability formulae, especially for the English language. These formulae attempt to decode the text's complexity elements to measure the readability and allocate the text to a specific readability scale (Cavalli-Sforza et al., 2018). English language researchers have introduced more than 200 readability formulae (DuBay, 2004), such as Flesch Reading Ease (Flesch, 1948), SMOG (Mc Laughlin, 1969), Dale-Chall readability formula (Dale and Chall, 1948), etc. These formulae are explained in detail as follows.

2.3.1. Universal readability formulae

<u>The Flesch Reading Ease formula</u> (Rudolf Flesch, 1948) is the most popular readability formula that is still in use until the present time. It gives texts a score from 0 to 100 or higher inversely related to understanding, as shown in Table 2.1, with 0 being the most difficult to read while 100 representing the easiest. For example, a text score of 40 means that it is difficult to read and corresponds to a college-level text. The table shows the Flesch Reading Ease score and the Flesch–Kincaid Grade Level *FKGL*, both used to measure English text's difficulty level. The Flesh Reading Ease score (*in equation (1)*) is based on Average Sentence Length (ASL) and Average of Syllables per Word (ASW), as expressed in the equation:

$$Flesch\ reading\ ease = 206.835 - 1.015\left(\frac{total\ words}{total\ sentences}\right) - 84.6\left(\frac{total\ syllables}{total\ words}\right) \quad (1)$$

As the Flesch formula was initially developed based on schoolbooks, it has flaws compared to assessing readability with authentic texts and readers. As with any other formula, it ignores reader differences and the influence of content, layout, and retrieval aids.

The constants in the equation devised based on a series of empirical tests designed to correlate with human assessments of readability. Each constarians refere to specific indicator as following:

- **206.835**: This is the maximum possible score, indicating the simplest and easiest text to read.
- **1.015**: This constant is used to scale the average sentence length (total words / total sentences) contribution to the final readability score.
- **84.6**: This constant is used to scale the average number of syllables per word (total syllables / total words) contribution to the final readability score.

Table 2.1 Flesch Score interpertation (Flesch, 1979)

Score	**School-level**	**Difficulty level**
100.00-90.00	5th grade	Very easy to read.
90.0–80.0	6th grade	Easy to read.
80.0–70.0	7th grade	Fairly easy to read.
70.0–60.0	8th & 9th grade	Plain English/ standard English.
60.0–50.0	10th to 12th grade	Fairly difficult to read.
50.0–30.0	College	Difficult to read.
30.0–0.0	College graduate	Very difficult to read.

Dale-Chall readability formula (Dale and Chall, 1948) (DCRF) is originally used as an indicator of vocabulary complexity. Based on an obtained list of 3,000 easy words from fourth-grade US children. Based on this formula, any word that exists outside of this list is considered difficult. It was compiled to overcome the shortcoming in the Flesch Reading Ease formula. Hence, adding a new variable, the average of difficult words in the whole piece of writing. The following formula is used in the calculation (*equation (2)*):

$$DCRF = 0.1579\left(\frac{difficult\ Words}{total\ Words} * 100\right) + 0.0496\left(\frac{total\ Words}{total\ Sentence}\right) \quad (2)$$

This formula measures two main parameters, PDW=Percentage of Difficult Words (words not on the Dale-Chall word list), and (2) ASL=Average Sentence Length in Words. The calculated score, referred to as *Raw Score*, is converted to school grade intervals using the conversion scheme shown in Table 2.2. In the equation **0.1579** scales the percentage of words that are considered difficult (i.e., not on a specific list of 3,000 familiar words) in the text. The percentage of difficult words is calculated as (difficult words / total words) * 100. Whereas **0.0496** scales the average sentence length in words, which is calculated as total words / total sentences.

Table 2.2 Dale-Chall Raw Score to Grade Interval

Raw Score	Grade Interval
4.9 and below	4th grade and below
5.0 - 5.9	5th –6th grade
6.0 - 6.9	7th – 8th grade
7.0 - 7.9	9th –10th grade
8.0 - 8.9	11th – 12th grade
9.0 - 9.9	Grade 13 through 15 (college)
10 and above	Grade 16 and above (college graduate)

The Gunning Fog-Index (GFI)(equation (3)), in *"The Technique of Clear Writing"* (Gunning, 1968), estimates two variables, average sentence length and the number of words with more than two syllables for every 100 words. The Fog-Index gained popularity because of its ease of usage. If the list of easy words is unavailable, it is possible to use the GFI approach and consider all the words consisting of two syllables or more as brutal. Gunning's Fog-Index, shown in Table 2.3, consists of 12 levels speeded across the educational levels, where higher index values indicate lower readability level. It is calculated with the following expression:

$$GFI = 0.4\left(\frac{total\ words}{total\ sentences} + 100\ (Hard\ Words)\right) \quad (3)$$

$$Hard\ Words — \rightarrow words\ with\ more\ than\ two\ syllables$$

Table 2.3 Gunning's Fog-Index, as presented in (Zamanian and Heydari, 2012)

Estimated Reading Grades		Fog-Index
Easy reading range	Sixth grade	6
	Seventh grade	7
	Eighth grade	8
	High school freshman	9
	High school sophomore	10
	High school junior	11
	High school senior	12
Danger line	College freshman	13
	College sophomore	14
	College junior	15
	College senior	16
	College graduate	17

Automated Readability Index (ARI) Another readability formula that returns scores related to the years of education required to understand the text is the ARI (Senter and Smith, 1967). At this point, they used another shallow feature: the average word length calculated from the number of characters per word as in equation (4).

$$ARI = 4.71\left(\frac{total\ Characters}{total\ Words}\right) + 0.5\left(\frac{total\ words}{total\ Sentence}\right) - 21.43 \tag{4}$$

DuBay(2004, p.25) remarked on those previous formulae as the foundation of measuring text readability, and the creators of previous formulae shed light on the demand for readability calculation. Moreover, they sparked additional research not just on how to enhance the formulae but also on the other elements influencing reading success.

Since the 1960s, there has been an acceleration in the readability studies to investigate how these formulae work and to develop other ways for a more profound representation of text's readability factors. Fry (1968) recreated a readability test using a reading graph, one of the earliest studies of that era. He proved its reliability in measuring text difficulty compared to other formulae. The Fry Graph in Figure 2.3 shows how text difficulty is calculated based on where the text's score is located in this graph. The estimation of text score is derived from plotting the average sentence length on Y-axes and the average number of syllables per word on the X-axes of a random sample of 100 words selected from the text under investigation. The Fry grade is derived from averaging these scores to get the grade level associated with the entire text allocated in the graph.

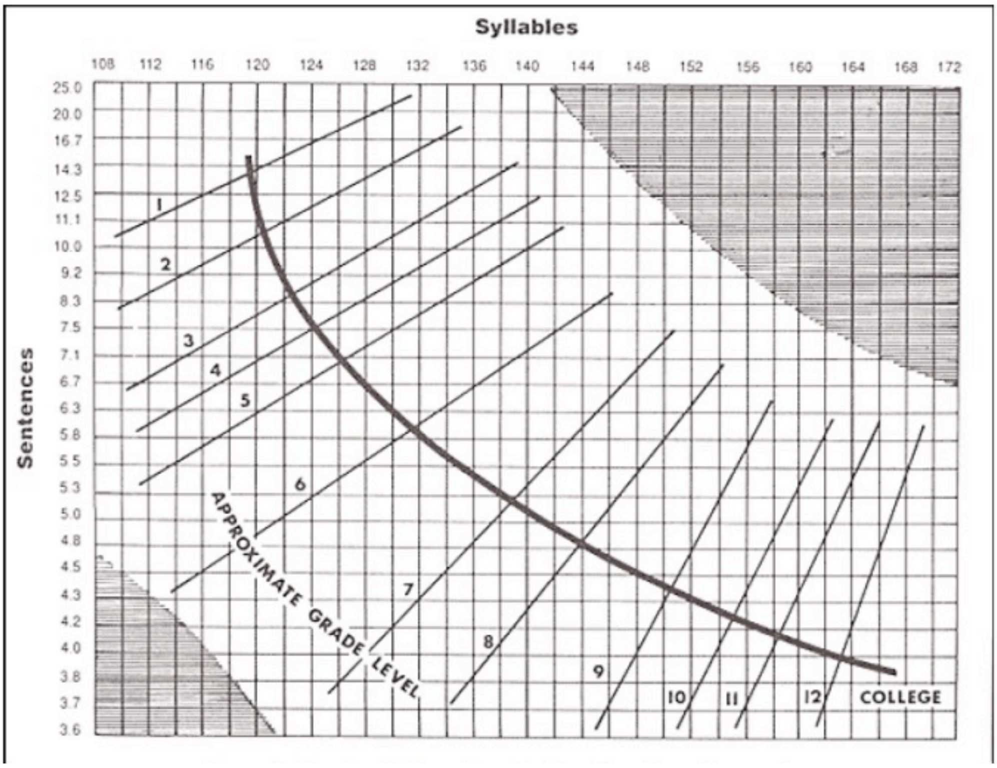


Figure 2.3 Fry Graph for estimating Reading Ages (in years), depending on locating score of word length average and sentence length, it estimates the school level grade. (Fry, 1968, p.577)

SMOG (Mc Laughlin, 1969) (Simple Measure of Gobbledygook grade) is a readability formula originally used for checking health messages. The main difference in this calculation from others is that the averaging sentences and words are multiplied rather than added. The SMOG score is determined by applying the following formula (equation (5)), which involves tallying the number of words containing three or more syllables (referred to as polysyllables) across 30 sentences:

$$SMOG = 1.0430\sqrt{number\ of\ polysyllables\,\frac{30}{total\ Sentences}} + 3.1291 \qquad (5)$$

$$number\ of\ Polysyllables \longrightarrow number\ of\ words\ with\ three\ or\ more\ syllables$$

The Flesch-Kincaid Formula (Kincaid et al., 1975) is a recalibration of the original Flesch Formula to include a better understanding of the text corresponding to the number of years of school education. This formula is also used in Microsoft Office Word to calculate the difficulty of a written piece. Rather than the Reading Ease Score, it rates text on a standard U.S. grade-school level. As such, it assigns values corresponding to the grade level that can easily read this text. For example, a document score of seven means that a 7th-grade student can understand this document. The formula is defined as follows (*equation (6)*):

$$Flesch - \ Kincaid = 0.39\left(\frac{total\ words}{total\ sentences}\right) + 11.8\left(\frac{total\ syllables}{total\ words}\right) - 15.59 \qquad (6)$$

To reach an accurat$Flesch - \ Kincaid\ score$, text must include more than 200 words before the Flesch Reading Ease and Flesch-Kincaid Grade Level can be used properly (Graesser et al., 2004).

2.3.2. Arabic readability formulae

Readability as mentioned above measures was designed for specific use in English texts. There are a few attempts to adapt these formulas to other languages. However, in Arabic, there were quite a few researchers who addressed the creation of new formulae inspired by the previous work for English. According to Cavalli-Sforza et al.(2018), in 1977, the first Arabic readability formula for the last three grades of elementary education, named ***Dawood*** in equation (7), was designed to consider the former English formulae. In addition to the average word length and sentence length, three new parameters were introduced: the average word's highest frequency, the percentage of nominal clauses, and the percentage of definite nouns (Daud et al., 2013; Saddiki et al., 2015). The Dawood formula is calculated as following

$$Dawood = -(0.0533 \times W) - (0.2066 \times S) + (5.5543 \times P) - 1.0801 \qquad (7)$$

W = Average word length in characters
S = Average sentence length in words
P = Average word frequency

A second attempt was the ***Al-Heeti*** formula in equation (8). It includes only one factor: the average word length. The simplicity of this formula gives it a high tendency to be used by researchers: it is easy to automate and apply to any language (Tamimi et al., 2014; Fouad and Atyah, 2016). However, Fouad and Atyah (2016) argued that this formula is too simple for a highly morphological language like Arabic

$$Al - Heeti = (AWL \times 4.414) - 13.468 \qquad (8)$$

AWL = Average word length in characters

Mat Daud et al. (2013) claimed that the average word length could not be considered an influential factor for Arabic readability because, unlike English, most Arabic words consist of three syllables and are easy or hard, depending on frequency. They produced their formula (9) based on a KACSTAC[6] The Corpus of Al-Thubaity (2015) indicates their simplicity by using the frequency of words in the corpus to rank in reverse the more frequent words at the end of the list and calculating the average word frequency ranking per sentence rather than the average number of words.

$$Average\ of\ Word\ word\ frequency\ for\ sentence = \frac{total\ reversed\ ranking\ of\ each\ word}{number\ of\ words\ per\ sentence} \qquad (9)$$

Recently, a third formula was proposed by *Al Tamimi et al.* (2014) entitled ***AARI Base***, the Automatic Arabic Readability Index Base, as set out in *equation (10)*. They applied the factor analysis technique to a group of readability factors to rank their impact. They then used principal component analysis to remove redundant and weak factors to determine better classification factors. These factors included word length (number of characters), word frequency and the occurrence of difficult words, average sentence length, sentence complexity, the clarity of the text's idea, the use of topology or metaphors, and grammatical complexity. Finally, they applied this formula to the ten grades of the Jordanian

[6] KASTAC is a general corpus of Arabic of more than 700 million words, approximately 7.5 million unique words, including texts from academic and non-academic sources covering several fields

school curriculum. They reduced the clustering to only three grouping levels rather than the original ten by the principal component analysis$A$

$$ARIBase = (3.28 \times NOC) + (1.43 \times ACW) + (1.24 \times AWS) \qquad (10)$$

NOC = Number of Characters
ACW = Average Character per Word
AWS = Average Words per Sentence

Equation (10) was reformulated as a predictor for grade level:

$$Gradelevel = (AARI + 472.42)/1046.3$$

El-Haj and Rayson (2016) introduced a readability metric for Arabic OSMAN given in equation (11) below. The metric depends on five parameters: average sentence length, average syllables per word: average word length, the ratio of long words, and the ratio of syllabically complex words. Their formula assumes that the average Arabic word length is five characters and replace with a comma the average syllable count is four syllables. They reformulate the Flesh-Kincaid and Gunning Fog formulas through experiments on a sample from the parallel Arabic-English corpus from the United Nations (UN) Corpus. To ensure the accuracy of syllabification counts, they used a diacritisation tool Mishkal[7]. Their metric is

$$OSMAN = 200.791 - \left(1.015 \times \frac{A}{B}\right) - 24.181 \times \left(\frac{C}{A} + \frac{D}{A} + \frac{G}{A} + \frac{H}{A}\right) \qquad (11)$$

A = total number of words
B= total number of sentences
C= number of hard words (words surface form > 5 characters)
D=number of syllables in the word
G= total number of characters
H= total number of complex words (word's syllable >4)

It should be noted that Arabic readability formulae were mainly dedicated to measuring readability for Arabic first language learners (L1)[8].

[7] https://sourceforge.net/projects/mishkal/

[8] First Language learners L1 is used in this thesis, referring to native speakers

2.3.3. Critiques and Shortcomings of Readability Formulae

The advantages of traditional methods such as statistical readability formulae are straightforward to complied with and implement in software to determine the readability level of written materials. They can provide a certain level of accuracy in grading the text by a numerical score easily located on a school grading scale. Even though readability formulae are widely used to measure text complexity, they have yet to be proven to fail in measuring the actual text readability level. Readability formulae are inaccurate as they need to provide a sufficient foundation for determining reading difficulty. There were many shortcomings raised across different studies:

- Readability formulae ignore many factors that affect text readability beyond the frequency, word difficulty, and average sentence length (Kirkwood and Wolfe, 1980; Bruce et al., 1981). Therefore, readability formulae do not provide a sufficient foundation for determining reading difficulty.

- Readability formulae cannot measure the context, difficulty of concept, complexity of ideas, or text coherence. Therefore, they are not consistent with the psycholinguistic theory of reading (Kirkwood and Wolfe, 1980). Bailin and Grafstein (2001) highlighted that the readability formulae developers treated the readability as if it is controlled by one main factor. However, there is no single straightforward measure of readability.

- The ignorance of the readers' unique variables, such as prior knowledge and interest level (Bruce et al., 1981).

- The absence of theoretical statistical grounds or evidence of their value justification.

- Dreyer (1984, p.336) asserted in this regard: "*Formulas do not measure textual factors such as word frequency, concept density, level of abstraction, nor whether there is an appropriate organisation, coherence, logical presentation of ideas. Consequently, formulas cannot distinguish scrambled text from well-ordered prose*."

- While studying the correlation between the linguistic formulae and the linguistic factors, formulae ignore the whole-text aspects that consider the arrangement and structure of sentences and paragraphs in texts and how information flows through the text (Schriver, 1989).
- Another critique raised by Carrell (1987) is that the shallow-based readability formulae were widely applied to first language learners (L1). Although they proved their functionality to a certain level in measuring text readability for the L1 readers, they failed to work for (L2) learners' needs.
- There is one last additional issue with the formulae, which is the inconsistencies between their scores. This discrepancy has been studied and proved in Chen's (1986) study of "comparing seven computerised readability formulae over the same textbooks", as cited in (Zamanian and Heydari, 2012). Chen's (1986) findings revealed that (1) there was no general agreement among the formulae on how to evaluate a textbook difficulty, and (2) there were significant disparities across formulae, resulting in the same textbook being scored at different grade levels. As a result, the wide range of scores generated by various algorithms proves that they are not ideal difficulty indicators.

In contrast to the previously mentioned criticisms regarding the applicability and the proficiency of the readability formulae, McClure (1987, p.12), in his interview with Dr J. Peter Kincaid (who developed the Kincaid Readability Formula), stated that "*a readability formula is an evaluation tool, not a reading or writing tool*". Hence, the readability formula is used to measure/evaluate written material but cannot be considered guidance for writing/rewriting pieces of text.

2.4. Readability levels classifications

The readability graded levels are essential as the readers and documents are always different for any given situation (Forsyth, 2014). Unfortunately, no available readability levels are specified to annotate text readability. However, the language proficiency levels such as (ILR, CEFR, and ACTFL) could be used as a readability scale. Hence, text readability and complexity are one aspect of various aspects of proficiency in a language. These levels are explained in the following sections.

2.4.1. The ILR proficiency levels

The Inter-Agency Language Roundtable (ILR) scale is a language proficiency scale developed in the 1950s by U.S. government agencies. The ILR scale provides a standardized way to measure language proficiency across different languages and language curricula. The ILR scale assesses four language proficiency skills: reading, writing, speaking, and listening, applicable to all languages and unrelated to any particular language curriculum. For example, in reading proficiency comprehension levels, there are six primary levels with two sub-levels for each: "base levels", which indicate the ability to perform the level's function, and other "plus levels", when the performance is higher than the former level but cannot reach the different main base level. The latest ILR proficiency levels are illustrated in Table 2.4.

Table 2.4 Interagency roundtable level of proficiency

Reading Grade	Proficiency Level
0	No proficiency
0+	Memorised proficiency
1	Elementary proficiency
1+	Elementary proficiency Plus
2	Limited working proficiency
2+	Limited working proficiency Plus
3	General professional proficiency
3+	General professional proficiency Plus
4	Advanced professional proficiency
4+	Advanced professional proficiency
5	Functionally native proficiency

2.4.2. CEFR levels

The Common European Framework of Reference for Languages CEFR or CEFRL[9] is a framework developed by the Council of Europe to describe language proficiency levels in a consistent and transparent manner. The CEFR includes six

[9] *https://www.fluentin3months.com/cefr-levels/* It might be better to refer here to the Council of Europe website instead.

proficiency levels, from A1 for beginners to C2 for advanced learners (see Table 2.5). The CEFR also includes detailed descriptors for each proficiency level, which can be used to assess an individual's language ability in different contexts and for different skills, such as reading, writing, listening, and speaking. The interpretation of these levels in the reading testing proficiency level is shown in Table 2.6.

Table 2.5 CEFR language ability levels

A1	Breakthrough	Basic user
A2	Waystage	
B1	Threshold	Independent user
B2	Vantage	
C1	Effective Operational Proficiency	Proficient user
C2	Mastery	

Table 2.6 Interpretation of the CEFR reading testing proficiency to the content of the actual text

Basic simple Text	A1	Texts contain familiar everyday expressions and fundamental phrases aimed at the satisfaction of the needs of a concrete type, with clear short sentences.
	A2	Texts with frequently used expressions related to areas of most immediate relevance (e.g. very basic personal and family information, shopping, local geography, employment), the surrounding environment.
Moderate Text	B1	Here text will contain familiar matters regularly encountered in the wider environment, such as work, school, leisure, etc., with a simple sentence structure using the coordination and given reason clauses.
	B2	Starting here with specialised texts with unfamiliar terms and terminologies, detailed text on a wide range of subjects, and explaining a viewpoint on a topical issue giving the advantages and disadvantages of assorted options.

Complex Text	C1	Longer texts with complex structures with organisational patterns, connectors, cohesive devices, and implicit meanings.
	C2	More complex, more prolonged, specialised texts with ambiguous structures

Overall, the CEFR is a widely recognized and useful framework for assessing language proficiency levels, including reading proficiency. The CEFR descriptors can provide learners, teachers, and organizations with a clear understanding of what language skills and knowledge are expected at each proficiency level and can help individuals set goals and track progress in their language learning journey.

2.4.3. ACTFL levels

The American Council on the Teaching of Foreign Languages (ACTFL) Proficiency Guidelines were first published in 1986 as an adaptation for the academic community of the U.S. Government's Interagency Language Roundtable (ILR) Skill Level Descriptions (Tschirner et al., 2015).

ACTFL is a framework that assesses an individual's proficiency in a foreign language. The framework includes five primary proficiency levels, which are Novice, Intermediate, Advanced, Superior, and Distinguished. Each level is further divided into sub-levels, such as Novice Low, Novice Mid, Novice High, etc.

To link the ACTFL proficiency levels with the CEFR levels, a linking and validation study was conducted. The results of the study are shown in Table 2.7. The table shows the one-directional alignment of ACTFL proficiency levels with the CEFR levels.

Based on Table 2.7, Novice Low and Novice Mid align with CEFR level 0 and CEFR level 0+, respectively. Novice High aligns with CEFR level A1.1, and Intermediate Low aligns with CEFR level A1.2. Intermediate Mid aligns with CEFR level A2, and Intermediate High aligns with CEFR level B1.1. Advanced Low aligns with CEFR level B1.2, and Advanced Mid aligns with CEFR level B2. Advanced High aligns with CEFR level C1.1, and Superior aligns with CEFR level C1.2. Finally, distinguished aligns with CEFR level C2.

Overall, the ACTFL proficiency levels and the CEFR levels are complementary frameworks that can be used to assess an individual's language proficiency. The linking and validation study has provided a way to compare language proficiency across different frameworks.

Table 2.7 One direction alignment for ACTFL-CERF levels (Anon, 2019, p.4)

ACTFL	CERF	LRI
Novice Low	0	0
Novice Mid	0	0+
Novice High	A1.1	1
Intermediate Low	A1.2	1+
Intermediate Mid	A2	2
Intermediate high	B1.1	2+
Advanced Low	B1.2	3
Advanced Mid	B2	3+
Advanced High	C1.1	4
Superior	C1.2	4+
Distinguished	C2	5

2.5. Automatic Text Readability

Automatic TR presents the automatic method of assessing the target text complexity to select the appropriate readers. It is a way to determine how hard/easy a text is. It is the representation of the sum of all elements of the textual material that affect the reader's comprehension. The Automatic TR resolved to apply supervised machine learning approaches following the pipeline outlined in Vajjala (2021). Automatic TR pipeline involves four main steps, as shown in Figure 2.4 Automatic TR pipeline representation (Vajjala, 2021, p.3)

- **Step one:** constructing gold standard training corpus classified on text/sentence level with readability levels/labels
- **Step Two:** defining a set of features to be computed from text
- **Step Three:** machine-learning model learns how to predict the gold standard label from the extracted feature

- **<u>Step four:</u>** optimised model is applied to the unseen subset of the corpus (test set)

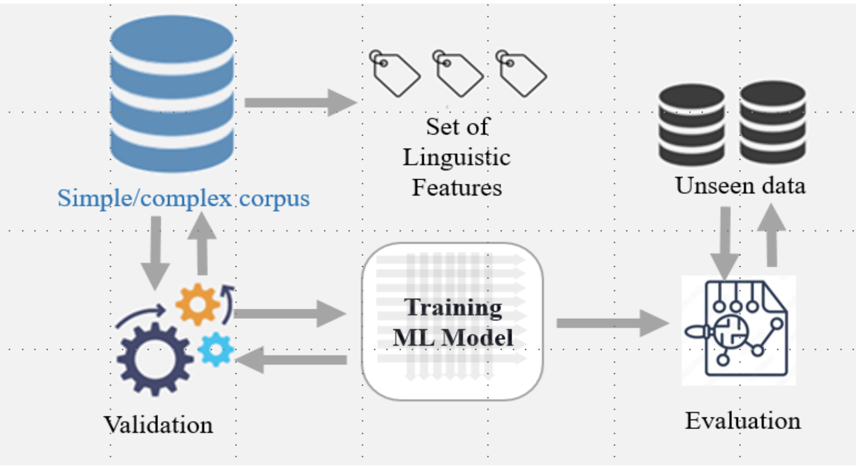


Figure 2.4 Automatic TR pipeline representation (Vajjala, 2021, p.3)

Either creating the corpus from scratch using available web content by crowdsourcing by machine-learning techniques or applying other resources already graded on text or sentence level. Vajjala (2021) classified them into two main categories: expert annotated and non-expert annotated.

The expert annotated corpus was mainly textbooks or graded texts used in education, such as school-graded textbooks for L1. This kind of corpus was well developed for several languages, such as English (Heilman et al., 2007), Japanese (Sato et al., 2008), German (Berendes et al., 2018), Swedish (Pilán et al., 2016), French(François and Fairon, 2012) and Bangla (Islam et al., 2012). Another method Xia et al. (2016), complying with a CEFR-level corpus, is extracting the reading comprehension passages from language exams conducted at different proficiency levels for L2 learners. The limitation of such a method is that most school textbooks are not available in a machine-readable format or are not accessible due to copyright permissions. To overcome these limitations, researchers build an Automatic TR corpus using publicly available news articles and encyclopaedia articles. They tend to modify and rewrite these articles to fit different graded readers and/or use various unrelated documents at each reading level. For example, in the English language, a widely used WeeBit (Vajjala and Meurers, 2012), a recent Newsela corpus (Xu et al., 2015a), and Onestopenglish (Vajjala and Lučić, 2018). Newsela was compiled as a parallel corpus that not only aligned on the document level but also aligned on the paragraph and sentence level. This aligning method allowed using this corpus in automatic text simplification (Štajner and Nisioi, 2018). A similar corpus is Complex/Simple Wikipedia [10] .Both researchers used automatic text readability/simplification to build easy versus complex systems. Other researchers follow this approach by using other websites to compile Complex/Simple corpus for English (Vajjala and Meurers, 2013), German (Hancke et al., 2012), Italian (Dell'Orletta et al., 2011), and Basque (Gonzalez-Dios et al., 2014). For example, Vajjala and Meurers (2014a) compiled a corpus from BBC channels' program subtitles grouped into three age groups. This method is commonly used to overcome the unavailability of a graded text corpus.

De Clercq et al.(2014) followed another approach using crowdsourcing from web texts in Dutch and asking readers to compare the difficulty of the presented texts' counterparts. In addition, ask an expert to annotate each text with difficulty level. After that, they compared both judgments to provide a final levelling of the texts.

A similar approach of using a non-expert to assign the document/sentence with a levelled difficulty grade for the German language is by Vor der Brück et al. (2008), using a 7-point Likert scale (Likert, 1932). In addition, Pitler and Nenkova (2008) asked college students to assign news articles on a scale. This approach is referred to as 'user studies', which rely on readers'/students' judgment and/or expert approval. Such as Kate et al. (2010), where both readers and experts classified the described dataset. In contrast, Shen et al. (2013) used a dataset collected and classified by experts in four languages - Arabic, Dari, English, and Pashto. In Nisioi et al. (2017) TS study, they collected user judgments of sentence-level text complexity for original, manually, and automatically simplified sentences. The limitations of the user studies approaches include time and effort-consuming and producing small data sets.

2.5.1. Readability wordlists

Thorndike (1921) provided the first frequency list of English words based on their use in general literature. He considered that the words that readers encountered frequently were more accessible to comprehend than the ones that

[10] https://www.english-corpora.org/wiki/

occurred infrequently. Naturally, familiarity breeds comprehension indicating that vocabulary is a strong predictor of text difficulty (Zamanian and Heydari, 2012). Hence, the research on the creation of wordlists annotated with some form of difficulty level (Gala et al., 2013; Francois et al., 2014; François et al., 2016), which are then used as features for Automatic TR (e.g., percentage of complex words in a text).

In Arabic, there are two established lists in ARA. These lists are Buckwalter and Parkinson's list and the KELLY Project list, which will be explained in detail in the following sections.

2.5.1.1. Buckwalter and Parkinson

Buckwalter and Parkinson's vocabulary list is a widely used Arabic frequency dictionary developed for language learners. It is part of Routledge Frequency Dictionary series, which includes dictionaries for 13 different languages such as Spanish, French, Russian, and Mandarin Chinese. The Arabic frequency dictionary contains the 5000 most frequent words in the Arabic language based on a 30-million-word corpus of academic/non-academic and written/spoken texts (Buckwalter and Parkinson, 2014). The corpus consists mostly of texts published in the 2006-2007 period, with some academic and well-known fiction resources from the 1990s and late 1950s.

The frequency dictionary is organized in a way that facilitates language learners to understand and use the words in context. Word entries are represented by their vowelized lemmas, which are base forms with several derivations based on unigrams appearing in descending frequency order. The main word list is arranged in alphabetical order based on the root system of the Arabic language. Entries in this list include the headword, its POS tagging, word derivational forms, English translation, and frequency in the last column.

In addition to the main word lists, the frequency dictionary also includes three grouped lists. The first list provides word frequency accompanied by a word lemma, POS tagging, and real context examples with linguistic information, including different word pronunciations based on 21 dialects, including Modern Standard Arabic (MSA). The second list consists of thematic lists or boxes in which words are grouped by their semantic classes into 30 semantic categories. These lists display the words with frequency and English translation. The final list is based on 12 classes of part of speech tagging. The statistical calculation of the frequency of those words is based on the MSA corpus and the most spoken dialect form, ensuring that the analysis is grounded in empirical data.

Overall, Buckwalter and Parkinson's Arabic frequency dictionary is a valuable resource for language learners and researchers alike. It provides a comprehensive list of the most frequent Arabic words in various contexts, and the organization of the dictionary makes it easy to use and understand.

2.5.1.2. The KELLY project’s Arabic

The KELLY project is a comprehensive linguistic endeavor that has produced aligned vocabulary lists across Arabic, Chinese, English, Russian, Italian, Swedish, Norwegian, Greek, and Polish. Its core objective, akin to the Buckwalter and Parkinson list, is to aid language learning. Each language list includes the 9,000 most frequent words and is freely available for download[11].

The Arabic list was developed based on an internet-based corpus of approximately 100 million words built by wide crawling following the same method as other Web corpora \citep{sharoff06ijcl}. It features lemmas associated with their CEFR levels and part of speech tagging. The CEFR levels in the KELLY project were established through both computational methods and human evaluation. Initially, a frequency analysis was conducted on the corpus to assign preliminary CEFR levels to the words based on their frequency of occurrence. Then, it was aligned with the frequency levels in corpora for other languages, such as Chinese, Greek, Italian or Russian (Kilgarriff et al., 2014a). Subsequently, linguistic experts reviewed and adjusted these levels, considering factors such as word difficulty, usefulness for learners, and relevance in various contexts.

[11] http://corpus.leeds.ac.uk/serge/kelly/

Thus, the KELLY project's Arabic list is not only a comprehensive tool for language learners and researchers, but it also represents a significant stride in aligning Arabic language learning with the CEFR standards. The balance between computational methods and expert input ensures the list's reliability and usability, making it a valuable resource for both learners and educators.

This list is produced from KELLY's project in Leeds, which includes nine languages bilingual vocabulary lists covering Arabic, Chinese, English, Russian, Italian, Swedish, Norwegian, Greek, and Polish. Those vocabulary lists are designed for the same purpose as the former list for language learning. Each list is composed of the 9,000 most frequent words in each language and is freely downloadable[12]. The Arabic list was obtained from an approximate 100-million-word internet-based corpus and contained only one language variety, which is MSA. It is a frequency word list represented in lemmas associated with their CEFR levels and part of speech tagging.

2.5.2. Feature extraction

"Identifying text properties that are strongly correlated with text complexity is itself complex."(Feng et al., 2010)

The research on building readability models, like many NLP tasks, was initially resolved based on traditional machine-learning approaches, which required extensive feature extraction. These features originally come from the easy-to-calculate features used previously in readability formulae, such as an average number of words per sentence. Then these features extended to more complex syntax. And semantics applying POS tagging and parsing for linguistic feature extraction, following the trend of the latest NLP approaches, applying deep learning approaches, word embeddings, and language transformers models instead of using a massive list of features. Feng et al.(2010) provided a detailed comparison between features extracted to be applied in ARA. These features range from shallow features such as word/sentence length to more complex features divided into five distinct groups' traditional shallow features, language modelling, part-of-speech-based grammatical features, parsed syntactic

[12] http://corpus.leeds.ac.uk/serge/kelly/ *accessed on 20/1/2019*

features, and discourse features. Their findings state that some shallow features (Average sentence length) were more effective than sophisticated syntactic features. These features are categorised as follows:

Shallow features (**Table 2.8** Shallow featuresTable 2.8), most of the researchers compiled a readability tool using features expressed by traditional readability metrics.

Table 2.8 Shallow features (Feng et al., 2010, p.280)

Average number of syllables per word
Percentage of poly-syll. words per doc.
Average number of poly-syll. words per sent.
Average number of characters per word
Chall-Dale difficult words rate per doc.
Average number of words per sentence
Flesch-Kincaid score
Total number of words per document

POS-based features (Table 2.9), adopting a morphological analyser to get informative linguistic calculations as presented in *Table 2.9.* POS features were proved to be effective in measuring text readability (Heilman et al., 2007; Leroy et al., 2008).

Table 2.9 POS-based features (Feng et al., 2010, p.280)

Percent of tokens per document
Percent of types per document
The ratio of Tokens/Types per total unique words in a document
The average number of adjectives/nouns/verbs/proper nouns per sentence
The average number of unique adjectives/nouns/verbs/proper nouns per sentence

Syntactic features (Table 2.10), using various syntactic analysis parse trees, dependency parsing (Schwarm and Ostendorf, 2005)

Table 2.10 Syntactic-based features (Feng et al., 2010, p.279)

Total number of phrases per document
Average number of phrases per sentence

Average phrase length measured by several words and characters, respectively
Average tree height
an average number of non-terminal nodes per parse tree
An average number of non-terminal nodes per word (terminal node).

Language modelling features *(LM)* training three language models (unigram, bigram, and trigram) on two paired complex/simplified corpora (Schwarm and Ostendorf, 2005)

Entity grid features refers to text/discourse coherence. This feature was intensively studied in research concerned with NLP tasks such as modelling text ordering and text generation (Lapata, 2005; Soricut and Marcu, 2006; Barzilay and Lapata, 2008) rather than readability. This feature was adopted by Barzilay and Lapata(2008), using a two-dimensional array grid model to represent the entities in each sentence in relation to other sentences' entities. One dimension corresponds to the text's most influential entities, while the other corresponds to each sentence. Each grid cell indicates whether the indicated entity is a subject (S), object (O), neither of the two (X) or absent from the phrase (-). Barzilay and Lapata (2008), reported that it helps to recognise the original text from the simplified version when compared.

Co-reference Inference (Table 2.11), implicit discourse relations, refers to the referential relations devices in the text. Research tends to focus on the automatic resolution of anaphoric devices in the text, e.g, pronominal references. Each entity and pronoun reference found in the text and related to the same person or object is extracted and linked to construct a semantic chain.

Table 2.11 Co-reference Chain Features (Feng et al., 2010, p.279)

Total number of co-reference chains per document
Avg. number of co-reference per chain
Avg. chain span
Num. of co-reference chains with span _ half doc. length
Avg. inference distance per chain
Num. of active co-reference chains per word
Num. of active co-reference chains per entity

Lexical Chain features, a more insightful text relation to represent text coherence. These features represent the semantic relations among words, e.g. synonym, hypernym, hyponym, etc. some researchers extracted these features and represented them as linked lexical-semantic relations chains (Galley and McKeown, 2003; Feng et al., 2009; Feng et al., 2010). For example, Feng et al. (2010) implemented six features based on linked entity chains, as shown in Table 2.12.

Table 2.12 Lexical Chain features (Feng et al., 2010, p.278)

Total number of lexical chains per document
Avg. lexical chain length
Avg. lexical chain span
Num. of lexical chains with span _ half doc. length
Num. of active chains per word
Num. of active chains per entity

Entity-Density features (Table 2.13), based on Feng et al.(2009) study of assessing the readability of a text for people with intellectual disabilities. They studied cognitive abilities with the assumption that the number of general nouns and named entities (proper nouns) and their relation affect the comprehension flow of the text. These basic entities are essential entities in text comprehension (Feng et al., 2009).

Table 2.13 Entity-Density features (Feng et al., 2010, p.278)

percentage of named entities per document
percentage of named entities per sentence
percentage of overlapping nouns removed
average number of remaining nouns per sentence
percentage of named entities in total entities
percentage of remaining nouns in total entities

Feng et al. (2010) performed comparisons between all these sets of linguistic/non-linguistic features. They concluded that discourse features have the least impact on text readability among all features. The entity density feature, primarily based on nouns and proper nouns, measuring noun phrases, stands in the second position in the prediction performance of classification algorithms. Furthermore, POS features stand at the top of all features providing a better prediction of text complexity level. Generally, POS features correlate more with text complexity than syntax and most discourse features. Emphasising measuring text readability requires linguistic analysis, yet a basic analysis rather than an intensive one. However, verbal phrases are highly correlated with text readability more than any other phrase type. They also reported that sentence length is dominant among all shallow features and has predictive power for text complexity. LM feature shows higher discriminating power only when trained on the testing corpus's relevant domain.

2.5.3. Readability models

Even though any text is composed of several sentences, which vary in their difficulty, research to date has tended to focus on assigning readability levels to the whole text rather than to individual sentences (Schumacher et al., 2016). Automatic TR research over the last 20 years is intricately linked to other areas of NLP. In short, traditional feature engineering-based methods dominate most of the early work, and recent work tends towards the deep learning model (Vajjala, 2021). Automatic TR is usually modelled as a supervised ML task, namely classification, and uncommonly modelled as regression (Vajjala and Meurers, 2014b) or ranking (Ma et al., 2012). However, Heilman et al. (2008) demonstrated that ordinal regression is better suited for Automatic TR tasks by comparing various methods.

In contrast, Xia et al. (2016) demonstrated that the ranking model may perform better compared to classification. In contrast to these approaches, Jiang et al. (2019) proposed a unique approach using graph propagation that can consider the inter-relationships between documents while modelling readability. Moreover, Martinc et al. (2021) compared different supervised and unsupervised approaches to neural text readability.

Applying neural and deep neural network-based approaches has recently dominated Automatic TR studies. For example, Mohammadi and Khasteh (2019) proposed a multilingual readability assessment model using deep reinforcement learning, and Meng et al.,(2020) proposed ReadNet, a hierarchical self-attention-based transformer model for ARA. Most recently, BERT (Devlin et al., 2019) dominated all NLP research and took over all ML architectures. Deutsch et al. (2020) demonstrated how BERT could resolve the Automatic TR task better than using linguistic features, which dominated Automatic TR research to date. Generally, most readability approaches have been resolved as a language-specific task. However, Azpiazu and Pera (2019) and (2020) study the development of multilingual and cross-lingual approaches to Automatic TR using deep learning architectures.

2.5.4. Evaluation methods

Evaluation is the last step in any NLP model, which aims to test the performance of the model architecture. Vajjala (2021) defined two methods for evaluation:

- *The intrinsic approach* refers to evaluating the Automatic TR model individually.
- *The extrinsic approach* refers to evaluating the Automatic TR model within a more extensive system.

Most Automatic TR models have been intrinsically evaluated on testing data regarding classification accuracy, Pearson/Spearman correlation for regression/ranking approaches, and root mean square error for regression (Vajjala, 2021). However, most commonly, the evaluating supervised machine learning approaches are held out on test data which is a part of the adopted corpus or as a cross-validation approach. At the same time, Pera and Ng(2012) and Kim et al. (2012) deployed a readability approach in a search engine and its plication to personalised search and reported an extrinsic evaluation of their experiments. In this case, they assess whether the easy-to-read (simple) predicted text leads to a better understanding for the target audience. Although this evaluation method appeared in TS research, it has yet to be performed on the Automatic TR model.

Validation is an optional step of assessing the performance of NLP architecture while compiling the method to tune the model accordingly. In this case, the adopted corpus is divided into three parts training, validation/tuning, and evaluation/testing. Yet, it is not common among Automatic TR systems to apply a validation process. The validation process aims to check if the features used in the model architecture can produce a reliable Automatic TR model that assigns readability levels to correlate with the reader's comprehension levels. Most studies focus on validating and assessing the Automatic TR results ignoring the reader comprehension factor. However, a few studies assessed the assigned readability level concerning the reader's comprehension perspective as such (Crossley et al., 2014; Vajjala et al., 2016; Vajjala and Lucic, 2019). Vajjala and Lucic (2019) found that the reading level annotations assigned to texts in a paired graded corpus did not have a measurable effect on readers' comprehension, indicating that factors other than these annotations may be more influential in determining the text's level of difficulty. In another study, François (2014) performed a qualitative and quantitative analysis of the French textbook corpus as a foreign language. He raised two issues: (i) there was no consistent correlation between expert annotations of the exact text, and (ii) no significant shared parameters among the texts assigned by the same level regarding linguistic features. Berendes et al. (2018) obtained similar results using a multidimensional corpus of graded German textbooks. Sheehan et al. (2015) and Sheehan (2017) provided a text evaluation tool named *TextEvaluator* for English teachers and test developers to select the appropriate text to the readers' levels.

2.6. Arabic (L2) automatic readability systems

Compared to English, Automatic TR focuses on assigning readability levels for L1 learners, and research on Arabic readability systems focuses on levelling the text targeting L2 learners. This assumption is initiated by the fact that reading Arabic poses difficulties and challenges for people born and growing up in Arabic-speaking countries as Arabic for them is a second language because their mother tongue is the colloquial variety of Arabic in that country (Habash, 2010). Other studies were conducted to measure text readability by modelling different ML algorithms targeting either L1 or L2 learners of the Arabic language. Most of these studies applied their methods to the GLOSS corpus because it is a rarely free Arabic L2 corpus. They aimed to construct a benchmark that future studies could modify or evaluate.

The early first study, the 'Arability' prototype system Al-Khalifa and Al-Ajlan (2010), used 150 texts from the Saudi Arabian school curriculum to classify them into three readability levels: easy, medium, and difficult. They used a bigram language model of their corpus, achieving an accuracy of 77.77%. This research is followed by Forsyth's (2014) study in his master's research on Automatic readability prediction/detection for MSA. He explored new readability factors and readability assessments. Applying a supervised machine learning technique on a selected 179 documents from the open-access curriculum corpus of the Defense Language Institute (DLI)[13] Foreign language levels are classified to five levels (1, 1+, 2, 2+, 3) of the Interagency Roundtable Levels (IRL) standard. Adopting the 5000-Arabic word frequency dictionary developed by Buckwalter and Parkinson[14]. This list is generated using a 30-million-word corpus of academic/non-academic and written/spoken texts. Words in the frequency dictionary are represented by their lemmas as a base form with several derivations of this form. Therefore, a morphological feature extraction has been done using MADA to annotate the corpus with lemma, clitics, and POS tags, to match the corpus with the dictionary entries. Developing a list of 165 features from which he used 162 features that affect the readability level, he isolated the nine main Features (See Table 2.14)

Table 2.14 Readability Feature set developed by Forsyth (2014, Table 4.2, p.30)

1.	POS-based Frequency Features
2.	Frequency-Based Discourse Connective Features
3.	Discourse Connective Features
4.	Token Count Features
5.	Type-To-Token Ratio Features
6.	Homographic Features
7.	Type-To-Token Features
8.	Token & Type Frequency Features
9.	Word Length Features

For the classifier's training, he used the TiMBL machine learning system using TiMBL's overlap metric to calculate K-nearest neighbours to rank the similarities and estimate the distance score between two feature vectors. Regarding the

[13] https://gloss.dliflc.edu/

[14] Available online: https://archive.org/details/AFrequencyDictionaryOfArabic/

evaluation, he conducted a range of experiments on an 80-20 train-test split of the corpus by training an instance-based classifier while varying 3-fold, 5-fold, or 10-fold cross-validation and on a 3- class or 5-class classifiers. The final test was on a 3-class dataset where adjacent levels were grouped with an F-score of 71.9% and 51.9% with a 5-class).

The following three studies represent continuous research, started by Cavalli-Sforza et al. (2014) using 71 texts from 'Al-Kitaab' and comparing them against the word lists introduced in the same book chapter labelling the words by (target, known, unknown), adding some averaging word/sentence features along with morphemes per word average.

They argued that the primary input in reading proficiency is vocabulary and the actual use of these words in context attached to different word senses. Accordingly, they classified the vocabulary list into three subcategories, ***Known***, which is already seen in previous lists; ***Target***, which is introduced in the target list; and ***Unknown***, which is unseen and untagged in this module. For tokenisation and corpus analysis, MADA (Morphological analysis and disambiguation for Arabic) (Habash et al., 2009) and SAMA 3.1, the LDC Standard Arabic Morphological Analyser, were used to analyse text and extract those factors. They experimented with previous readability factors along with newly introduced factors, as listed in Table 2.15.

Table 2.15 Newly introduced features by (Cavalli-Sforza et al., 2014, p.84)

Percentage of known, targeted, and unknown words in the text
Percentage of open-class words in the text
Percentage of closed-class words in the text
The Ratio of unique words over the total number of words in a text (lexical diversity)
Number of the unnecessary word token in Text (Text length) text length
Average sentence length in tokens
Average word length in syllables
The average number of attached clitics per word measures' word complexity.

They found that closed-class features, lexical diversity, and average word length do not affect the text readability level or the actual number of characters per word. However, the number of clitics/ morphemes attached to the word has a high impact on text readability. They continued their research by applying a machine learning classifier, the probabilistic decision tree (PDT), to the texts using the most relevant factors adopted from the previous experiment. They implement this module by using Python with a third-party Python library (scikit-learn 0.14)[15].

Regarding different word lists at the beginning of each module, they classify 25 different sets of known, unknown, and target words along with the other features of each text to attach each text to the best chapter. Like many other techniques, the ML model was exceptionally reliable in classifying the text in the slot of one of the first five chapters of Al-Kitaab's Part Two, second edition, and cannot predict if it is suitable for a specific stage in that span of chapters.

Their results were improved by grouping the levels into four classes using K-means clustering with an accuracy of nearly 87%, as reported. Then an attempt by Saddiki et al. (2015)*,* known as the Ibtikarat team. The purpose of their research was to analyze readability factors that could enhance the classification of L2 texts according to IRL levels, using a sample of 251 documents from the Gloss corpus. Their feature set consists of 35 set vectors performing the morphological analysis using MADAMIRA (Pasha et al., 2014). Those features are categorised into eight main factors, Sentence, Word, Morpheme, Character, Vocabulary load, Ambiguity, Word class, and Content word POS. Using WEKA[16] As a platform, Hall et al. (2009) studied various classification machine learning algorithms (e.g. Decision Tree, K-nearest- neighbour Support Vector Machine (SVM), and Random Forest). They adapted all features and training algorithms to a 3-class and a 5-class classification scheme. Their results indicated that features such as morpheme counts, type and token counts, measures of sentence length, and part-of-speech carried the most information gain and provided an economical and good baseline for building models. They were reaching a maximum accuracy of 73.31 on a 3-a class set. They were followed by a recent study by Saddiki et al. (2018), which highlights adding new syntactic features to

[15] https://scikit-learn.org/stable/

[16] Waikato Environment for Knowledge Analysis (WEKA) provide a workbench that allows researchers easy access to state-of-the-art techniques in machine learning. Available from: https://www.cs.waikato.ac.nz/ml/weka/

their features. Using two different datasets for both first and second Arabic language learning. This approach yielded an accuracy of 94.8% and 72.4% for L1 and L2, respectively.

The Oujda-NLP team (Nassiri et al., 2018b; Nassiri et al., 2018a) has also presented two linked types of research. The first was based on 170 features calculated and applied to 230 texts from the Gloss corpus as well as using the AraNLP library and MADAMIRA morphological analyser (Pasha et al., 2014). They reported the results with 3-class categories with an accuracy of 90.43%. The latter study used the same data set but analysed it with a different morphological analyser called AlKhalil and reduced the features to 133 features. They used the Buckwalter frequency list (Buckwalter and Parkinson, 2014) and reported an accuracy of 100% with 3-classes.

On the other hand, regarding the available tools for readability annotation for Arabic, Al-Twairesh et al. (2016) provided in their research a theoretical framework to build an interactive web-based Arabic readability annotation tool referred to as '*MADAD*'. This web-based framework for semi-automatic Arabic text annotation involves readability assessment. This framework supports a broad range of annotation tasks for various semantic phenomena by allowing users to create customised annotation schemes. The scale range for the text difficulty ranges from 0 easy to 100 difficult.

2.7. Arabic feature extraction tools

2.7.1. MADAMIRA Arabic morphological analyser

MADAMIRA is a toolkit used for Arabic morphological disambiguation and linguistic analysis (Pasha et al., 2014). The MADAMIRA[17] system architecture is depicted in Figure 2.5. The system consists of seven milestones of analysers and models. The input text enters first the Pre-processor and then travels through the system milestones, while each step adds analysis or information to be used in the following step. Thus, the system can provide various outputs based on the desired analysis output.

[17] https://github.com/owo/madamira_diac and demo is available: https://camel.abudhabi.nyu.edu/madamira/

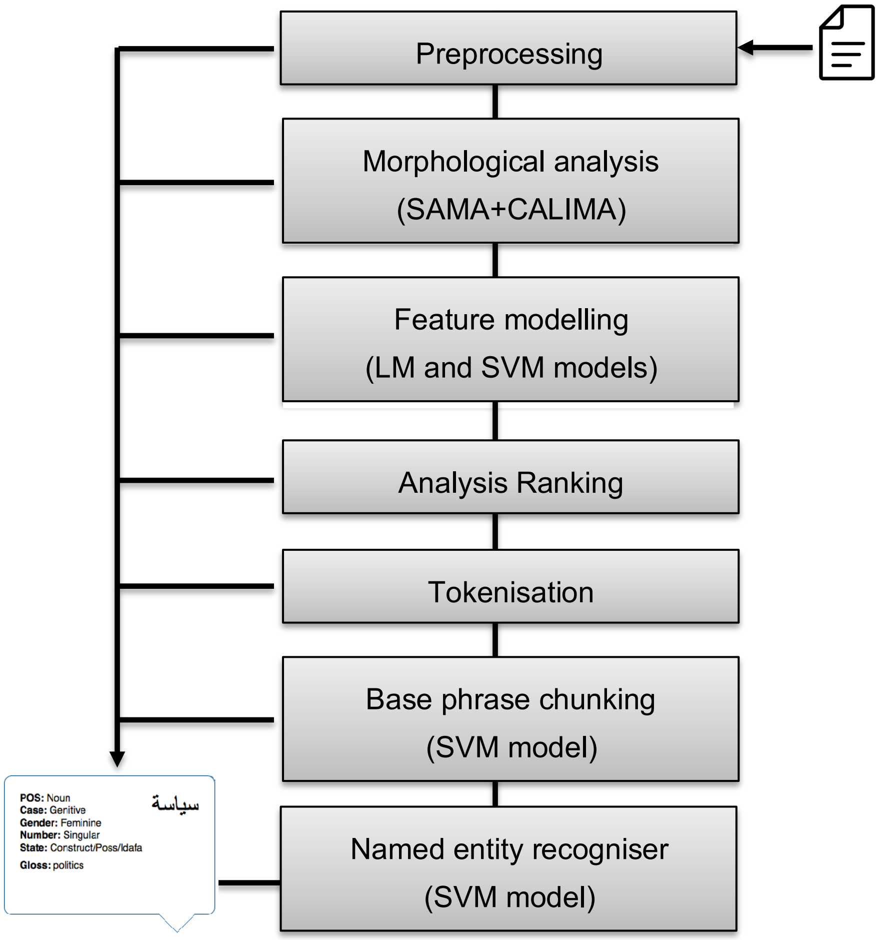

Preprocessing
Morphological analysis (SAMA+CALIMA)
Feature modelling (LM and SVM models)
Analysis Ranking
Tokenisation
Base phrase chunking (SVM model)
Named entity recogniser (SVM model)
POS: Noun
Case: Genitive
Gender: Feminine
Number: Singular
State: Construct/Poss/Idafa
Gloss: politics
سياسة

Figure 2.5: MADAMIRA architecture (Pasha et al., 2014, pp.1095, 1099)

MADAMIRA provides features for each word in a sentence based on various pr

- **Lemmatisation**: determining the lemma
- **Stemming**: provides the morphological stem
- **Diacritisation**: determining the fully diacritised form
- **Glossing**: determining the English translation
- **Part-of-speech** Tagging: determining the part-of-speech
- **Morphological Analysis**: identifying every possible morphological interpretation of input words.
- **Morphological disambiguation**: determining a complete or partial set of morphological features (either the most likely feature values for each word given its context or a ranked list of all possible analyses for each word).
- **Tokenisation**: segmentation of clitics with attendant spelling adjustments according to form.

MADAMIRA toolkit provides a POS tagset comprising 15 main tags such as gender, number, person, state, case, etc. This toolkit is considered state-of-the-art in Arabic automatic linguistic analysis tasks.

2.7.2. Farasa morphological analyser

Farasa[18] is an open-source project fast and accurate Arabic morphological analyzer and part-of-speech tagger. It is developed by the Qatar Computing Research Institute (QCRI). The tool is designed to provide several functionalities essential for Arabic language processing, including segmentation, part-of-speech tagging, and morphological analysis (Darwish and Mubarak, 2016).

Farasa primarily uses a machine-learning approach to morphological analysis, which allows it to handle the high degree of inflectional and derivational morphology found in Arabic. The system is trained on a large corpus of Arabic text, which helps it recognize and analyze a wide range of morphological patterns. It also segments words into their individual morphemes, which is particularly useful for processing Arabic, a language that often combines multiple morphemes into a single orthographic word. It includes a diacritization feature which is an important tool for various NLP tasks. The diacritization process in Farasa can add missing diacritics to the text, which helps in disambiguating words that have similar forms but different meanings depending on the diacritics.

Farasa stands out for its efficiency and accuracy in handling Arabic text. It has been evaluated on standard benchmarks and has achieved state-of-the-art performance, making it a valuable tool for researchers and developers working on Arabic language processing.

2.7.3. Arabic syntactic parsers

Green and Manning (2010) argue that the challenge in parsing Arabic sentences is the ambiguity at the discourse level. The Arabic sentence structure may compose of many subordinate words and phrases with variant word orders such as VSO, SVO, VOS, and VO (Green and Manning, 2010). Unlike English, the Arabic sentence may continue to appear in more than four lines in the text, and most Arabic text is written without punctuation. Therefore, counting the number of sentences for each text according to punctuation marks such as (' / . / , / ; / ?)

[18] QCRl-organization http://qatsdemo.cloudapp.net/farasa/

is not accurate for Arabic sentences. Hence, sentence chunking needs to be performed to measure the number of phrases inside each sentence.

In order to choose the most appropriate dependency parser for extracting features to perform the TS task. There are three main Arabic parsers, Stanford parser(Green and Manning, 2010)[19] and UDpipe parser (Straka and Straková, 2017)[20] and CamelParser (Shahrour et al., 2016) described in the following section.

1. *Stanford Arabic Parser* (Green and Manning, 2010)

This parser is a part of the Stanford Core-NLP system, one of the most popular toolkits used in NLP research (Green and Manning, 2010; Manning et al., 2014). The Stanford toolkit provides several NLP tasks, such as text preparation, normalisation, tokenisation, segmentation, part-of-speech tagging, sentence splitting, constituency parsing, and semantic annotation. The Stanford toolkits were developed initially for English NLP research, and later, the toolkit developers provided partial support for other languages, including Chinese, Germany, Arabic, Italian, Bulgarian, and Portuguese. The Stanford Arabic parser provides Universal Dependencies (v1) and Stanford Dependencies output as well as phrase structure trees (Nivre et al., 2016). The Arabic parser was based on the first three parts of the Penn Arabic Treebank (PATB) (Maamouri and Bies, 2004). These corpora contain newswire text.

2. *UDPipe Parser* (Straka and Straková, 2017)

UDPipe is a free software trainable pipeline for tokenisation, tagging, lemmatisation and dependency parsing of CoNLL-U files. UDPipe is language-agnostic and can be trained given annotated data in CoNLL-U format. Trained models are provided for nearly all UD treebanks. UDPipe is a fast transition-based neural dependency parser. The parser is based on a simple neural network with just one hidden layer that makes use of FORM, UPOS, FEATS and DEPREL embeddings. The form embeddings are precomputed with word2vec using the training data, the other embeddings are initialised randomly, and all embeddings

[19] http://nlp.stanford.edu:8080/parser/index.jsp
[20] https://lindat.mff.cuni.cz/services/udpipe/

are updated during training. It generates only one root node and only uses the root dependency relation for this node.

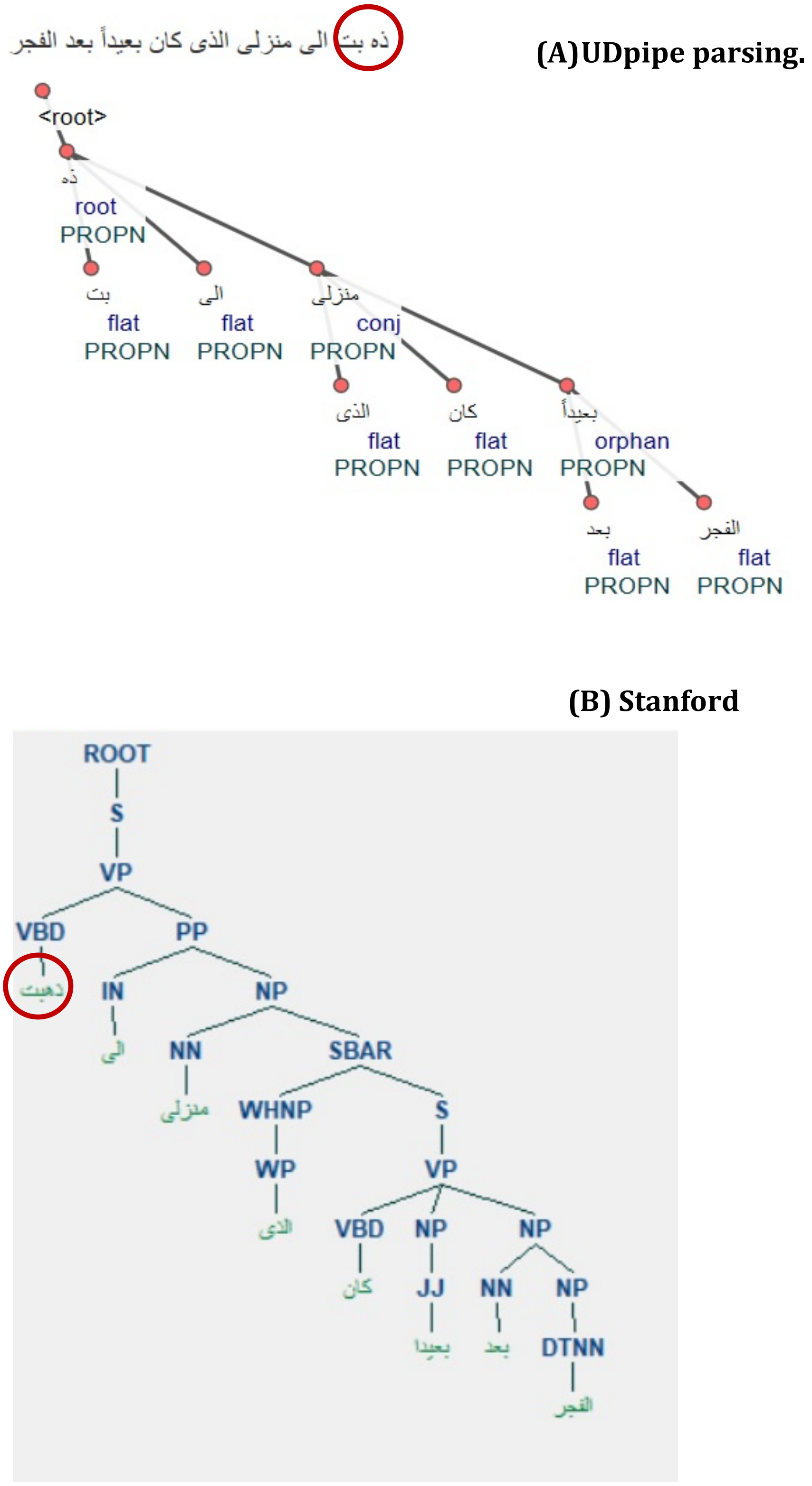
To demonstrate the performance of UDpipe and Stanford parser after parsing a simple Arabic sentence, consider the following example:

Arabic	"ذَهَبْتُ إِلَى مَنْزِلِي الَّذِي كَانَ بَعِيدًا بَعْدَ الْفَجْرِ"
Transliteration	ḏahabtu ʾilā manzilī allaḏī kāna baʿīdan baʿda alfajri
Translation	'I went to my home which was far, after fajjr'

In parsing this sentence, the UDpipe morphological analyser miss-analyse the first word 'ذهبت' 'ḏahabtu , I went' as 'ذه' and 'بت', which are both non-sense Arabic words and do not exist in the Arabic language which of course led to parsing mistakes (see Figure 2.6). On the other hand, the Arabic Stanford statistical parser performed well in the same sentence (Figure 2.6-B) by labelling the verb 'ذهبت' 'I went'.

3. *CamelParser* (Shahrour et al., 2016)

CamelParser is a state-of-the-art system for Arabic syntactic dependency, which is aligned with contextually disambiguated morphological features. It uses a MADAMIRA morphological disambiguator and improves its results using syntactically driven features. The parser trained an Arabic dependency parser using MaltParser(Nivre et al., 2005) on the Columbia Arabic Treebank (CATiB) version of the PATB (Habash and Roth, 2009). This parser provides several output formats, including basic dependency with morphological features, two-tree visualisation modes, and traditional Arabic grammatical analysis.



(A)UDpipe parsing.

(B) Stanford

Figure 2.6 Shows two different parse trees for the sentence (A) represents the UDpipe's parse tree, while (B)Shows Stanford's proper parse tree for the sentence.

2.8. Limitations and challenges

The limitations in Automatic TR studies are not limited to Arabic and extend to many other languages, as presented in (Vajjala, 2021). These limitations are:

1- *Availability of corpus resources:* few corpora assigned with text/sentence readability levels have yet to be published. Although there are accessible corpora, they may not be appropriate for the desired assessment task. However, there was much work done on Automatic TR in many languages by adopting and tuning available publicity corpora for other NLP tasks to perform ARA. This lack of available and diverse corpora can limit the development of Automatic TR models tailored to specific application scenarios. For example, the correlation between the corpus and the target users' comprehension may result in applying the same corpora for the Automatic TR model to levelling text for the L1 readers and dyslexic readers simultaneously. Because analysing problems and complexity via dyslexic readers are exclusive to first-language readers and vice versa.

2- *Availability of ready-to-use tools:* although there is extensive research in ARA, only some available implemented tools or access codes can be executed for the researcher tools (Vajjala, 2021). For example, some researchers shared code to reproduce their experiments (Ambati et al., 2016; Howcroft and Demberg, 2017).

3- *Reader and Task considerations:* From an educational and psychological point of view, there are many factors associated with the text that affect text comprehension, text properties, reader characteristics, and task complexity (Goldman and Lee, 2014; Valencia et al., 2014). The Automatic TR research was expected to consider all these parameters while assessing the text level. However, they were mainly focused on the text's linguistic features while ignoring all non-linguistic features and the reader/end-user characteristics. Kim et al. (2012) presented a research modelling reader perspective, whereas Kühberger et al. (2019) initiated modelling task complexity. Yet, up to date, no research combines all three factors in one model.

4- *Lack of validation and interpretation*: in connection to lack of reader characteristics representation in Automatic TR models. This is because more research needs to be conducted to check if the used corpus is suitable for the task and whether the model result's interpretation is connected to the target users' demands. Also, it is difficult to fully understand the attributes that the model learns exactly about the complexity of the text. These issues make it difficult for researchers in other disciplines to adopt the latest Automatic TR methods instead of turning to traditional formulas that are easy to compute and interpret (Vajjala, 2021).

5- *Lack of extrinsic evaluation:* as mentioned before, most Automatic TR systems were evaluated intrinsically and isolated from any applied scenario. This lack makes it difficult to understand how the model work to solve real-life situation.

6- *Lack of the same theoretical background* to compare different Automatic TR systems. In addition, most English Automatic TR models use other corpora, making it impossible to compare results across systems.

The main challenge appeared in building a multidimensional Automatic TR model that represents all factors affecting the complexity of the text beyond the textual features. Another level is embedding the Automatic TR model in a more extensive system to solve another NLP task.

2.9. Conclusion

The first section of this chapter reviews a 20-year study of Automatic TR in NLP and related research areas, identifying the limitation and challenges for each step in the Automatic TR pipeline. Despite extensive research, there is still no clear understanding of what works best with automatic text simplification, especially when it comes to applying it to real-world tasks. One of the primary challenges of automatic text simplification is ensuring that the simplified text remains readable and understandable. While advancements in deep neural NLP techniques have been made, there is still a lack of understanding of how to apply them effectively in the context of automatic text simplification.

Moreover, evaluating the effectiveness of automatic text simplification models remains a challenge. Intrinsic evaluation is the primary method of assessment, but there is a lack of validation work, which demands more attention. In addition, there are no available tools or resources for the diverse types of researchers and practitioners interested in ARA. One of the obstacles Arabic readability research faces is the insufficient training datasets for which annotators provide labels with sufficient readability assessments. This creates a significant problem, as different researchers and practitioners may have different perspectives on what constitutes a readable text.

In order to measure text complexity, there is a need to discover the linguistic phenomena that define the complexity of the text. Therefore, ***Chapter 4*** aims to answer these questions in a series of readability classification experiments. This involves the construction of an open and accessible corpus, which can serve as a gold standard to test new readability assessment models for different application scenarios. Furthermore, developing a new approach that targets various domains/target groups. Finally, providing an evaluation of the readability-prediction algorithm in a more significant NLP scenario for the task of Automatic Text Simplification (Chapter 5).

Chapter Three: Literature review (Text Simplification)

"Language is situational. Every utterance fits in a specific time, place, and scenario, conveys specific characteristics of the speaker, and typically has a well-defined intent." (Jin et al., 2021)

TS aims to control the readability attribute of the text and make it more accessible to different readers with various intellectual abilities. Therefore, TS is essential in natural language generation (NLG). TS is an active NLP research area, and as with much other ongoing research, its techniques show a drift from manually hand-crafted rules toward deep learning techniques (Sikka and Mago, 2020; Al-Thanyyan and Azmi, 2021). Most of these techniques were borrowed from closely related NLP tasks (Sikka and Mago, 2020). For example, considering TS as a translation task, in which the translation within the same language, the complex sentence as the source and the simple sentence as the target (Zhu et al., 2010). Rather than implementing new techniques, these similarities encourage researchers to use Machine Translation (MT) and monolingual text-to-text generation methods and techniques. Applying these techniques is not limited to system implementation but also system evaluation. Some studies utilise text summarisation and paraphrase generation framework while treating summarisation tasks as a type of TS (Sikka and Mago, 2020). However, there are different views on the correlation between TS and text summarisation; will discuss this later in this chapter.

This chapter is dedicated to presenting a systematic background review of the research on TS. Primarily discussing the research progress in European languages, especially the English language, reviewing over 100 research/studies since the initiative by Blum and Levenston (1978). Followed by presenting primitive studies on Arabic as the scope of this research.

Blum and Levenston (1980) completed one of the first studies introducing LS for Teaching English as a Second Language (TESOL). They are followed by the Easy-to-Read movement, which aims to produce simpler and more understandable documents by a wider group of people with different intellectual capabilities.

The movement uses the Easy-to-read criteria introduced by the european guidelines (Freyhoff et al., 1998). Freyhoff et al. demonstrated the importance of considering both text format and content in order to improve comprehension. Their findings indicated that the inclusion of visual aids such as images, charts, diagrams, and tables can enhance the structure and formatting of text, ultimately leading to better comprehension.Later work by Hervás et al.(2014) focused on text cohesion criteria, including the use of simple sentence structure, expression of a single idea per sentence, avoidance of technical terms, abbreviations, and initialise.

As with other NLP tasks, TS starts with hand-crafted rule-based and manual annotation, which is time and effort-consuming, and the need arises to automate this task. This generates new terminology, "Automatic Text Simplification" (ATS), with the extended aim to reduce, where possible, the time and human effort of development. One of the significant studies that set the milestones for an ATS system was Chandrasekar et al. (1996) which focused on text readability and understandability.

The majority of the literature in the TS is based on the premise of English language processing, reflecting the historical bias and the profound impact of this trend on the development and refinement of NLP tools and techniques. This skewness presents a significant challenge when addressing the requirements of other languages, particularly those with distinct linguistic characteristics like Arabic. However, Arabic, despite being one of the world's most widely spoken languages, has received far less attention in NLP research relative to English. To balance the narrative, the focus of this thesis will include discussing trials and progress made in Arabic language processing, which will be detailed at the end of each section. These discussions are meant to illuminate the strides made in Arabic TS, and identify the gaps that this thesis aims to address, thus underlining the significance and necessity of the present research in contributing to a more inclusive and language-diverse.

The rest of the chapter is organised as follows; section 3.1 provides an overview of TS approaches and how they correlate to similar NLP tasks. Section 3.2 gives an overview of the available data sets and parallel corpora applied in

TS research in different languages. Section 3.3 includes a discussion of the state of the art in LS, followed by a similar discussion of the text, sentence, and syntactic simplification approaches and techniques applied by other European languages in section 3.5. Section 3.6 discusses primitive Arabic TS studies. Finally, section 3.7 focuses on evaluation methods and metrics used in TS.

3.1. Text Simplification approaches

Sikka and Mago (2020) argued that there are two main TS approaches, i) The Extractive approach involves text summarisation; ii) The Abstractive approach, which includes lexical, syntactic, and semantic simplification (referred to as sentence compression). However, Shardlow, 2014, pointed out notable differences between text summarisation and text simplification. Text summarisation involves shortening the text while focusing on critical key ideas by deleting unimportant or redundant information. Whereas, in Text simplification, a deletion process could be performed on the text along with substitution and addition processes. In text simplification, the difficult words are replaced by more frequently understandable words, adding descriptive phrases for complex terms, adding connectors, and splitting sentences by adding explicit anaphora. All these operations would result in a longer simplified text than the original text. In this case, the target text will be short in the summarisation process; in contrast, the exact text will be longer after simplification. Therefore, while summarisation and simplification share the goal of maintaining the original information, they have different approaches to reaching their goals. Improving readability is a crucial factor in simplification; this gives a shred of convincing evidence to exclude text summarisation as an extractive approach from text simplification's sub-tasks.

The abstractive approach includes lexical, syntactic, and semantic simplification. Most studies approach simplification and focus on LS by replacing complex vocabularies or phrasal chunks by suitable substances (Paetzold and Specia, 2017). The first text simplification studies focused on lexical item modifications rather than other simplification tasks (Siddharthan, 2002; Shardlow, 2014).

Alva-Manchego et al., 2020, in their recent survey regarding data-driven sentence simplification, mentioned different sentence modification tasks concerning TS. **Abstractive sentence compression** is another related task that involves phrasal replacement, addition, and reordering (Cohn and Lapata, 2013). This task aims to reduce the text despite enhancing sentence readability. As per this definition, abstractive sentence compression is closely related to summarisation rather than simplification. On the other hand, the **sentence compression** task, also referred to as semantic simplification, comprises sentence length reduction while maintaining both primary information and grammatical structure. This task focuses on deleting unnecessary and redundant words. As a result, sentence compression might be classified as a type of TS. Narayan et al.(2017) introduced in their study not only deletion but also paraphrasing by splitting a sentence into simple ones, which involves syntactic and semantic simplification while preserving the meaning. Their **Split-and-rephrase** task focuses on deleting unnecessary words or phrases to make the text more understandable while conveying the main idea without any distractors. As such, Split-and-rephrase is considered in the context of simplification. Sentence simplification approaches are classified into monolingual MT and hybrid techniques(Al-Thanyyan and Azmi, 2021).

3.2. Text Simplification datasets and corpora

Lexical resources are essential in the development and evaluation of simplification systems. Only some datasets are available and reliable for LS. Most of these data sets are English language specified; however, the following list includes some attempts for other languages, e.g., Spanish.

3.2.1. Lexical Simplification datasets

The majority of LS datasets were manually annotated and identified the complex words based on human judgments. At the same time, presenting a ranked list of the possible substances of complex words. Table 3.1 summarises the efforts of building LS lexical resources in other languages and then presents the available Arabic lexical complexity classified lists.

Regarding Arabic LS resources, recently, Al Khalil et al. (2020) published the first Arabic readability list tailored for TS project(Al Khalil et al., 2017; Al Khalil et al., 2018). The list named SAMER[21] , this list consists of 26,000-lemmas five-level readability lexicon for MSA. It was manually annotated with three different language speakers of three Arabic dialects. A1 (Egypt), A2 (Syria/Levant), and A3 (Saudi Arabia/Gulf). and then they took the average of the labelling the Five-levels as follows: Level 1: Generally corresponding to Grade 1, Level 2: Generally corresponding to Grades 2-3, Level 3: Generally corresponding to Grades 4-5, Level 4: Generally corresponding to Grades 6-8, Level 5: This level reflects specialist language use beyond the eighth grade. However, this research also introduces a new Arabic CEFR-level vocabulary list, this list explained later in section (4.4).

3.2.2. TS parallel corpora

A complex/simple parallel corpus that consists of complex and simple aligned sentences is necessary for seq2seq modelling for text generation. This section presents the various efforts and methods in the automatic generating of such a corpus, as summarised in Table 3. 2. It lists the parallel corpora that have been used in the TS literature. In English, Newsela and Simple English Wikipedia were highlighted in English TS research. Some have one-to-one sentence alignments, and others include one-to-many sentence alignments to allow for sentence splitting.

It should be highlighted that most English ATS research applied on either one of the following available large TS datasets: (1) Simple English Wikipedia (SEW) parallelised to the original English Wikipedia (EW), which is available as a parallel sentence version or parallel documents version (William Coster and Kauchak, 2011). The complex-simple parallel sentences version contains 167,686 pairs of aligned sentences (Will Coster and Kauchak, 2011); (2)The second recent resource is the Newsela corpus contains 1,911 news articles which manually simplified up to five times (Xu et al., 2015a).

[21] https://camel.abudhabi.nyu.edu/samer-readability-lexicon/

Table 3.1 List of Benchmarks Lexical Resources publicly available [modified version of **(Al-Thanyyan and Azmi, 2021, p.4)**]

Lexical simplification resources	Size	Description
		English
SemEval-2012 **(Specia et al., 2012)**	2,010 contexts	The data set is based on SemEval 2007'sEnglish Lexical Substitution Task and covers 210 target words, including names, verbs, adverbs, and adjectives (McCarthy and Navigli, 2007). Each word appears in 10 different contexts. This is considered a primary LS data set, widely used in research and considered as the benchmark because it consistently captures the concept of simplicity recognised by non-English speakers. ***Target audience*** : None Native English speakers ***URL***:https://mailman.uib.no/public/corpora/2011-November/014319.html
LSeval **(De Belder and Moens, 2012)**	430 sentences	The sentences are classified according to difficulty by 46 Amazon Mechanical Turk (MTurk) and nine different PhD students. The data set is generated from one reference set (McCarthy and Navigli, 2007) and consists only of words not labelled as easy. Originally the list of easy words was a combination of simple English word lists from Simple English Wikipedia (SEW) and the Dale-Chall readability measure. The intensive annotation process used to create this data set has actually enabled the complexity of words to be simplified (Dale and Chall, 1948). LSeval uses the same base data as the SemEval 2012. ***Target audience*** : None Native English speakers
CW corpus **(Shardlow, 2013b)**	731 sentences	The sentences are extracted from SEW edit histories, each with one complex word. In order to maintain a balanced corpus, a negative example (i.e. only an example of a simple word) is provided by random selection of the word from the CW-occurrence sentence. ***Target audience*** : Evaluation of CWI systems

Lexical simplification resources	**Size**	**Description**
LexMTurk **(Horn et al., 2014)**	500 sentences	The sentences were randomly selected from the complex aligned corpus (EW) with SEW. 50 MTurk is used to provide simpler replacements for complex target words for each sentence in the dataset. ***URL***: cs.pomona.edu - /~dkauchak/simplification/lex.mturk.14/
SemEval2016 **(G. Paetzold and Specia, 2016b)**	9,200 Sentences	Manually annotated dataset for complex word identification shred task by 400 non-native English speakers annotated the shared-task dataset. These sentences were taken from three sources, CW Corpus, LexMTurk Corpus, and Simple Wikipedia. ***Target audience*** : None Native English speakers
BenchLS **(G. Paetzold and Specia, 2016a)**	929 instances	Using two datasets, LexMTurk and LSeval, that were automatically corrected spelling and inconsistencies. Each instance consists of a sentence with a complex word and seven ranked substations provided by English speakers. ***URL***: ghpaetzold.github.io/ data/BenchLS.zip
NNSeval **(G. Paetzold and Specia, 2016d)**	239 instances	This is created by filtering LexMTurk and LSeval datasets in two dimensions ; first, removing all substitutions synonyms considered complex by non-native speakers. Second, deleting the instances containing target words that were not considered complex by non-native speakers. This makes NNSevalis more accurate in capturing non-native English users than other data sets. ***Target audience***: None Native English speakers ***URL***: ghpaetzold.github.io/ data/NNSeval.zip
		Spanish
PPDB-S **(Štajner et al., 2019)**	5,709	Select a subset of sentence pairs from the paraphrases database (PPDB) (Ganitkevitch et al., 2013). PPDB-S dataset is a relatively small set of paraphrases that have the same meaning.

Lexical simplification resources	**Size**	**Description**
PPDB-M **(Štajner et al., 2019)**	15,524	Unlike PPDB-S, it is generated in the same way, but the coverage is high and the accuracy is lower than PPDB-S data set.
Synonyms from Spanish OT **(Štajner et al., 2019)**	21,635	Synonyms are extracted from Spanish Open Thesaurus (OT), filtering words from multiple senses and arranging them by their frequency or length in a corpus.
EuroWordNet synonyms **(Štajner et al., 2019)**	13,970	Synonyms were extracted from the Spanish EuroWordNet (Vossen, 1998) in the same way as OT.
CASSA **(Štajner et al., 2019)**	5,640,694	The output is produced by extracting all 5-gram pairs of the CASSA resource (Baeza-Yates et al., 2015), where the target word is not infinite.
		French
FLELex **(François et al., 2014)**	777,000 words	This is obtained from textbooks available and simplified readers for learners of French as a second language. It reports the frequency of the words in the form of lemmas standardised at each level of the Common European Languages Reference Framework (CEFR). ***URL***: https://cental.uclouvain.be/cefrlex/flelex/download/

Lexical simplification resources	**Size**	**Description**
ReSyf **(Billami et al., 2018)**	121,182 synonyms	The synonyms were extracted from the lexical network JeuxDeMots (Lafourcade, 2007) and then semantically disambiguated and ranked based on their reading difficulty for French learners. ***URL***: https://cental.uclouvain.be/resyf/
		Japanese
SNOW E4 **(Kajiwara and Yamamoto, 2015)**	2,500 instances	They were extracted from a newswire corpus. Moreover, manually provided the set of ranked substitutions by a set of annotators using a crowdsourcing service. ***Target audience***: Children & language learners ***URL***: www.jnlp.org/SNOW
BCCWJ **(Kodaira et al., 2016)**	2,100 instances	Ranked substitutions were provided and ranked using crowdsourcing services and by computer science students. As a result, the BCCWJ dataset overcomes the limitations of the SNOW E4 dataset, where sentences are extracted from a balanced corpus and constraint candidates are allowed in simple rankings. ***Target audience***: Children & language learners ***URL***: https://github.com/KodairaTomonori/EvaluationDataset
		Portuguese
LexSubNC **(Wilkens et al., 2017)**	1,500 substitutes	is a list of 180 Portuguese nominal compounds and their substitutions that had been manually checked. They are grouped into one of three types: synonym, near-synonym (such as hypernyms, hyponyms, and meronyms), and paraphrase or definition. ***URL***: https://pageperso.lis-lab.fr/~carlos.ramisch/?page=downloads

Lexical simplification resources	**Size**	**Description**
SIMPLEX-PB	1,719 instances	is a list of 757 complex words as a target of simplification with manually annotated replacements filtered and suggested by three linguists experts. ***Target audience:*** Children ***URL***: https://github.com/nathanshartmann/SIMPLEX-PB
SIMPLEX-PB-2.0 **(Hartmann et al., 2020)**	1,719 instances	Enhancement of SIMPLEX-PB on the number of synonyms for its target complex words (7,31 synonyms on average). With a manual ranking produced by the target audience itself – children between 10 and 14 years. ***Target audience***: Evaluation of LS for Children ***URL***: https://github.com/nathanshartmann/SIMPLEX-PB-2.0
SIMPLEX-PB-3.0	1,719 instances	Another new version of SIMPLEX-PB, enriched with linguistic features, added 38 new columns containing lexical features of word complexity. Currently, the corpus has 52 columns of information. **URL**: https://github.com/nathanshartmann/SIMPLEX-PB-3.0
		German
WaCKy **(Cholakov et al., 2014)**	2,040 words (includes 153 target words)	Frequency ranked German word list extracted from a large German corpus. Synonym substitutions were provided by German native speakers using a crowdsourcing service. (German) **URL**: https://tudatalib.ulb.tu-darmstadt.de/handle/tudatalib/2436

Table 3. 2 List of parallel corpora available for Text Simplification

Parallel corpora	Aligned pairs	Generation Method
		English
EW-SEW	137,000	It was generated by aligning "equivalent" articles and sentences from the regular English Wikipedia (EW) and the simplified versions of articles (SEW). This resource is publicly available and widely used, allowing seq2seq data-driven text simplification. The data covers the main simplification operations: reordering, inserting and deleting. **URL**: EW -- https://en.wikipedia.org/wiki/Main_Page SEW -- https://simple.wikipedia.org/wiki/Main_Page
Parallel Wikipedia Simplification Corpus (PWKP) (Zhu et al., 2010)	108,016	It was extracted from 65,133 articles in EW and SEW. It was using the dump files in Wikimedia to pair the articles and TF-IDF similarity measure for sentence aligning. **URL**: https://fileserver.ukp.informatik.tu-darmstadt.de
Coster-Kauchak Dataset (William Coster and Kauchak, 2011)	137,362	Extracted by pairing the Simple English Wikipedia with the English Wikipedia. The data has three different varieties of pairing, either one-to-one sentence alignment, or one to many by performing sentence splitting, or many-to-one representing summarisation. **URL**: https://cs.pomona.edu/~dkauchak/simplification/
Wikipedia-Simple Wikipedia WikiSmall (Kauchak, 2013)	167,689	This dataset consists of aligned sentences from 60,000 Wikipedia-aligned articles. It generated for implementing TS Language model. This updated dataset, WikiSmall, uses sentence alignment with updated Wikipedia data and improved text processing. A subset of SEW created by unified and standardised SWE

		sentence-level and paragraph-level aligned. It is widely used to evaluate the performance of simplification systems using BLEU metric. URL: https://cs.pomona.edu/~dkauchak/simplification/data.v2/sentence-aligned.v2.tar.gz
WikiLarg Dataset	3,856K	The dataset contains two files : normal.txt and simple.txt, and both files have the same number of lines and are aligned by lines. **URL**: https://github.com/XingxingZhang/dress
Newsela (Xu et al., 2015)	10,787	Contains news articles in English simplified to 4 or sometimes 5 different reading levels by human experts. Newsela contains parallel simple-complex news articles with 11 grade levels. Corpus-level simplification is available; however, it has to be processed for sentence-level simplification. This corpus is reviewed and corrected manually by human experts, which enhances the structure and reliability of the corpus. URL: https://newsela.com/data/
SS Corpus (Kajiwara and Komachi, 2016)	492,993	Extracted from 126,725 article pairs obtained by aligning articles from EW and SEW by exact matching of article's titles. **URL**: https://github.com/tmu-nlp/sscorpus
Turk Corpus (Xu et al., 2016)	2,350	This dataset was created using Amazon Mechanical Turk with their SARI evaluation metric. It is composed of aligning 8 simplified reference sentences for each complex sentence, which allows the SARI statistic to be calculated. This dataset is usually used for tuning and testing.
ASSET	2,350	Dataset for assessing sentence simplification in English aligned with Turk Corpus (ASSET). It contains the same set of original complex sentences found in Turk Corpus. ASSET has one-to-one and one-to-many alignments, with 10 simplification references per original complex sentence collected by Amazon Mechanical Turk. **URL**: https://github.com/facebookresearch/asset

OneStopEnglish (Vajjala and Lučić, 2018)	Up to 3,154	Consists of 189 English texts, each in three different reading levels: elementary (ELE), intermediate (INT), and advanced (ADV). It has 1,674, 2,166, and 3,154 sentence-aligned pairs for ELE-INT, ELE-ADV, and INT-ADV, respectively.
		Italian
SIMPITIKI (Tonelli et al., 2016)	1,166	Composed of two sets of simplified pairs: (a) those extracted in a semi-automatic way from the Italian Wikipedia revision history and (b) manually created sentence-by-sentence from documents belonging to an administrative domain. URL: https://github.com/dhfbk/simpitiki
PaCCSSIT (Brunato et al., 2016)	63,000	Automatically produced from a large raw corpus. (described in detail in the following section)
		Other Languages
Alector (Gala et al., 2020)	79 texts and their simplified equivalent	It is extracted from authentic literary and scientific texts that were commonly used for students in French primary schools. Experts manually simplified the texts at different linguistic levels: morpho-syntactic, lexical, and discourse levels. (French)
Simplext (Saggion et al., 2015)	200 news texts	The parallel corpus contains news from four domains covering national, international, cultural, and societal news. (Spanish)

3.2.3. TS parallel corpora extraction methods

Most of the complied parallel corpora were aligned either limited to document-level or paragraph-level simplification. However, considering TS as a text generation process requires a sentence-level aligned corpus. Thus, several researchers tackled this limitation by using various methods to extract parallel complex/simple sentences from an aligned corpus or from raw data. The following section presents some of these methods classified based on the extraction or generation of resources.

3.2.3.1.Extraction from an existing resource

Kajiwara and Komachi(2016) produced **SS Corpus**[22], they propose an automatic unsupervised method to extract 492,993 aligned sentence pairs from 126,725 article pairs obtained by aligning articles from EW and SEW by exact matching of article's titles. They use a many-to-one method to align each word in the complex sentence with the word that is most similar in the simple sentence, and then they compute sentence similarity by averaging these word similarities.

Scarton et al. (2018) proposed a combined method to extract pairs from the Newsela corpus in two steps. First, using traditional readability metrics to extract complex-simple sentence pairs from the corpus. Second, using this parallel corpus to train an ML model to classify sentences into binary classification complex versus simple and to predict complexity levels.

3.2.3.2.Extraction based on similarity measures

1) Semantic Similarity

The first method is measuring how similar two words, phrases, or expressions are based on how likely it is that they have the same meaning. There are two main approaches used for determining the similarity of phrases: (i) Corpus-based or Distributional Semantic models (DSMs), which identify similarity based on the presumption that similar words appear in similar articles, (ii) and Knowledge-based models methods evaluate the similarity between expressions using word senses, POS,

[22] https://github.com/tmu-nlp/sscorpus

and taxonomic information. The limitation of the corpus-based method appears in it does not consider different word senses based on the context. In contrast, the knowledge-based approach is limited by the availability of dictionaries composed by humans. The main drawback of relying on the semantic similarity measures is that words with opposite meanings also have a high similarity score based on word relatedness (Bollegala et al., 2007; Cilibrasi and Vitanyi, 2007).

Recent researchers employed semantic similarity measures by the use of deep language representation such as Word2Vec (Mikolov et al., 2013), Sent2Vec, Doc2Vec, Glove (Pennington et al., 2014), Gensim (Řehůřek and Sojka, 2010), and fastText (Grave et al., 2018) to convert words or sentences to vectors (word vectors/word embeddings) and the (cosine) similarity between these vectors are considered as the semantic similarity of the words.

2) Aligner models

Paetzold and Specia (2016), developed the MASSAligner, an open-source Python library that allows for the retrieval of aligned phrases or paragraphs from a document based on a specified similarity criterion. Aligning the sentences follows two main steps, first, measuring semantic similarity: which converts documents to a bag of words, forms word vectors, then uses TF-IDF and calculates the cosine similarity between word vectors. Second, align sentences based on similarity; the model follows a vicinity-driven approach to extract the sentences that are similar based on a threshold value provided as a hyperparameter. The aligner then makes its way through the similarity matrix to create a path for alignment that looks for the best pair of similar sentences.

The shared task of quality assessment for text simplification (QATS) aimed to establish a quantitative measure of successful extraction. Kajiwara and Fujita (2017) explore the usefulness of semantic functions based on word alignments to estimate the quality of text simplification. They introduced seven types of alignment-based functions that were calculated based on word embeddings and paraphrase lexicons and achieve state-of-the-art performance on the QATS[23] dataset. The training part of the QATS dataset training set consists of 505 sentence Complex/Simple pairs and four

[23] http://qats2016.github.io/shared.html

human scores: Grammaticality score (G), Simplicity score (S) assigned to the simplified sentence (Simplified), while Meaning preservation score (M), and Overall score (Overall) take into account both complex and simple pairs.

3.2.3.3.Generating synthetic corpus

This involves generating a complex synthetic sentence instead of an authentic simple source sentence (Aprosio et al., 2019). This was mainly inspired by work done in Generating Synthetic Data for keyword-to-question answering by Ding and Balog (2018). Aprosio et al. (2019) proposed a novel technique to overcome the limitation of data availability, which involves Simple-to-simple synthetic pair creation and Simple-to-complex synthetic pair creation. First, extracting the simplest sentences from a monolingual corpus using heuristical techniques. Then pair these sentences with their replications to be added to the actual complex-simple sentence pairs training set. This allowed for better word embeddings and created a bias in the system towards simpler sentences. Second, the Simple-to-complex synthetic pair creation, in which the extracted simple sentences passed through a "complexifier" to generate synthetic complex sentence pairs.

3.2.3.4.Parallel dataset mining

Brunato et al. (2016) managed to acquire PaCCSS–IT, a parallel corpus of complex–simple aligned sentences for Italian. It consists of 63,000 parallel pairs automatically produced from a large raw corpus. Their methodology concentrated on sentence extraction with structural transformations rather than lexical ones. They have proposed three steps to compile the corpus; first, develop a collection of a large number of sentences that share words but in a different structure. The sentences are subsequently ranked based on a similarity metric that is designed to evaluate the degree of similarity between the words contained within each sentence. Second, manually revised the top sentence pairs in order to build a sentence pairs classifier. Third, rank the sentences' lexical complexity using an automatic TR tool.

Another method by Martin et al. (2022) proposed a large-scale mining of sentence-level paraphrases from the web instead of a parallel simplification dataset. These sentences were semantically concatenated based on index embeddings for each sequence using *faiss*[24] (Johnson et al., 2017) for finding the nearest neighbour search. They applied their approach to three languages: English, French, and Spanish.

Furthermore, for the language in our concern here, Arabic (Al-Raisi et al., 2018) introduced the first automatically complied Arabic parallel sentences. They relied on Google Translate API for Europarl-v7 corpus, an English-French parallel legal corpus (Koehn, 2005), to build parallel sentences pair share the similar meaning with different grammatical forms. The corpus is provided in two different non-overlapped sizes, small and large, with 765 and 100,000 sentence pairs, respectively[25]. However, 200 sentence pairs of the corpus were manually verified by two native speakers of Arabic; the first analysis of the corpus shows numerous ungrammatical Arabic sentences in the corpus. These grammatical errors could be a result of their method of using Machine Translation (MT) in translating the English section to Arabic (considered complex) and the French section to Arabic (considered simple). This consideration is English/Complex – French/Simple because the average length of Arabic sentences translated from the English section tends to be longer than the ones translated from French.

3.3. Lexical Simplification pipeline (LS)

LS is the task of identifying and substituting complex, difficult words and expressions with simpler words equivalent in meaning without changing the sentence's grammatical structure (Shardlow, 2014; Paetzold and Specia, 2015; Saggion, 2017). However, some recent studies considered some simplification on the phrasal level besides the LS. Paetzold and Specia (2015) and Shardlow (2014), have identified four primary tasks that are involved in LS operation. : (i) Complex word identification [CWI] to extract the complex word from a text; (ii) Substitution Generation [SG], substitution and generation of alternatives ; (iii) Substitution Selection [SS], word-sense disambiguation according to the given context ; (vi) Substitution Ranking [SR], ranking of the alternatives in the order of simplicity. Figure

[24] https://github.com/facebookresearch/faiss

[25] The corpus is available at http://www.cs.cmu.edu/~fraisi/arabic/arparallel/

3.1 illustrates the LS pipeline, as mentioned in recent studies. Moreover, Paetzold and Specia (2017) added a fifth task called "Confidence Checker", which double-checks the selected simple word against the regular use of this word in the corpus to ensure the readability of the simplification. They had earlier observed Paetzold and Specia (2016e) that In "the Jitterbug Perfume novel written by Tom Robbins" (Robbins 2003, the author writes, "There are no such things as synonyms! He practically shouted. Deluge is not the same as a flood". Thus, the LS is an incredibly challenging task: how to identify a complex word and which is the best substitute that would preserve the meaning and sentence well-formedness (Paetzold and Specia, 2015; G. Paetzold and Specia, 2016e). The latter preservation requirement indicates that LS should be performed in more than an isolated word scale to preserve the collocations and multi-word expressions that affect the overall cohesion and coherence of the text (Saggion, 2017). Research on TS can be classified into two main approaches, the rule-based approach and the data-driven approach.

Carroll et al. (1998) introduced one of the first automatic TS systems, which targeted simplifying English newspapers targeting aphasic readers. They applied linguistic analysis, consisting of lexical tagging, morphological analysis of the words, and text parsing before the simplification process. Their simplifying stage consists of a lexical simplifier and a syntactic simplifier. They performed LS by following the "simplify everything" approach, which considers each word in a sentence as a target word that needs to be substituted by a simpler word. The lexical simplifier component skips the CWI step, generates a list of substances using WordNet, and ranks this list based on the Oxford Psycholinguistics Database frequencies. The best replacement is the word with the highest frequency. As stated in Sikka and Mago (2020), their continuum research by Devlin and Tait (1998) reported that in their system, 16.60% of all simplified content in their system had its grammatical structure altered, and over 44% had its meaning drastically modified.

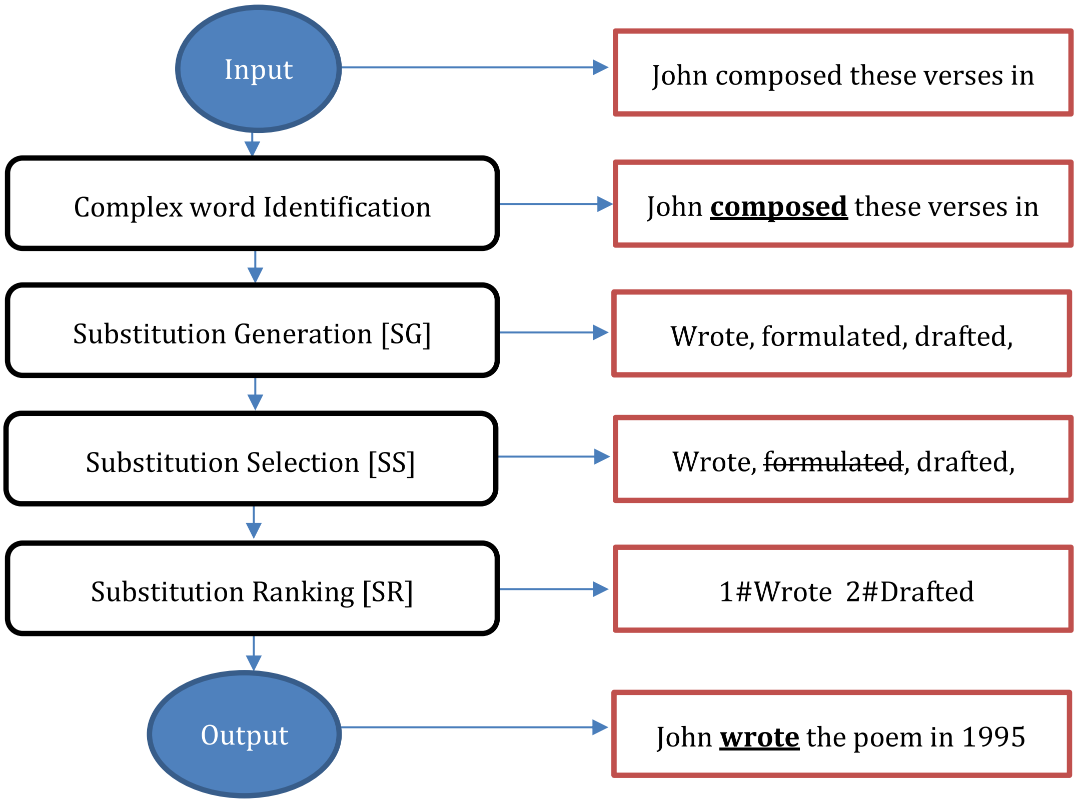

Input
John composed these verses in
Complex word Identification
John **composed** these verses in
Substitution Generation [SG]
Wrote, formulated, drafted,
Substitution Selection [SS]
Wrote, ~~formulated~~, drafted,
Substitution Ranking [SR]
1#Wrote 2#Drafted
Output
John **wrote** the poem in 1995

Figure 3.1 The LS pipeline as illustrated in (Shardlow, 2014, Paetzold and Specia, 2016a, Paetzold and Specia, 2016d, Paetzold, 2015, Paetzold and Specia, 2015, Paetzold and Specia, 2016f, Paetzold and Specia, 2016c, Paetzold and Specia, 2016e), along with an example from (Saggion, 2017)

3.3.1. Complex Word Identification (CWI)

The foremost step performed at the top of the pipeline for LS is Complex word identification (CWI) - to recognise the difficult, complex words to perform the simplification process. Unfortunately, early LS attempts did not consider CWI before performing the actual simplification process (Biran et al., 2011, Horn et al., 2014, Glavaš and Štajner, 2015).

A reliable CWI methodology identifies the complexity but prevents the LS system from miss-performing an unneeded and/or ungrammatical simplification (G. Paetzold and Specia, 2016e; G. Paetzold and Specia, 2016b; Yimam et al., 2018). However, G. Paetzold and Specia (2016e) argued that most searches, such as those (Biran et al., 2011, Horn et al., 2014, Glavaš and Štajner, 2015) do not provide a detailed methodology for CWI.

Recently, Paetzold and Specia (2016b), in the SemEval 2016 Task11 for Complex Word Identification, applied different methodologies to perform the best CWI. The results of their study suggest that the quality of the corpus is the primary factor that influences the success of the CWI task. This is because the complexity of a given word, as identified by the CWI system, is largely dependent on its frequency within the corpus being used.These methodologies ranged from Threshold-Based approaches and Lexicon-based approaches to Deep Recurrent Neural Networks and word embeddings. Recent studies proposed strategies for approaching CWI tasks are classified as follows (Paetzold and Specia, 2017):

a. **Threshold-based approaches**

Threshold-based techniques seek a threshold *t* for a word w over a given metric of simplicity *M*, such that if *M(w)< t*, the word *w* may be more reliably classified as complex or simple (Paetzold and Specia, 2017). Threshold-based metrics study the effect of the length of the words, word frequency, and/or level of synonyms on word readability. Keskisärkkä (2012) explored the affection of the previous three strategies. They reported that the more the replacement decision depends on the word length rather than word frequency, the more readable the produced sentence is. Nevertheless, word frequency has been applied frequently in metrics, identifying the less frequent words in a large corpus as the complex words giving a significant score in threshold-based metrics. Bott et al. (2012) described the first LS system for the Spanish language (**LexSiS**) based on a list of complex words that appeared in 1% of a large contemporary Spanish corpus (Corpus de Referencia del Español Actual, CREA)[26]. They reported results that outperform all other baseline systems, producing new simple sentences with simple and right synonym replacement achieving improvements in meaning preservation. Leroy, Endicott, et al.(2013) Leroy, Kauchak, et al.(2013), a similar study identifies the complex words in a list of the 5,000 least frequent words in the Google IT corpus (Michel et al., 2011). They aimed to simplify medical texts for patients with low literacy levels. They reported a human manual evaluation of the produced output with

[26] http://corpus.rae.es/creanet.html

significantly improved readability from the original text. Shardlow (2013a) introduced a threshold based on an evaluation corpus for CW dedicated to making a clear division between what is complex and what is simple(Shardlow, 2013b). Their corpus was compiled using word frequency based on the SUBTLEX corpus (Brysbaert and New, 2009). Wróbel (2016) represented another CWI model, which achieved the highest F-scores in SemEval 2016 task (G. Paetzold and Specia, 2016b) also and learned a word frequency threshold over Simple Wikipedia (Kauchak, 2013).

Although, Threshold-based approaches are easy to implement, using only word length or frequency metrics as a single feature to separate between complex and simple words. Bott et al.(2012) and Shardlow (2014), in their studies, provides evidence that using word length is not efficient when measuring word complexity. Although Bott et al. (2012) stated that a simple word does not mean it is a short word, they reported that less than 70% of manually simplified words are shorter than their original complex counterparts. Moreover, Shardlow (2014) evaluated their threshold method and found that 65% of complex words were wrongly identified, and 99 out of 119 mistakes resulted from being identified as complex. This leads to either dispensable replacement of mistaken identification of simple words or ignoring short but complex words.

b. Lexicon-based approaches

Using domain-specific, manually crafted lexicons to identify complex words through texts. Medical researchers applied this approach using different methods for lexicon extraction. Deléger and Zweigenbaum (2009) compiled a lexicon composed of medical technical terms. They identified pairs of aligned paraphrases in technical medical articles, and their selection was based on vector cosine similarity between pairs being higher than 0.33. Elhadad and Sutaria (2007) present another method for creating a medical terms lexicon based on the Unified Medical Language System (UMLS), a database of complex medical terms (Bodenreider, 2004). Another technique that Elhadad (2006) applies assumes that shared words across disciplines are simple. He considered a cross-reference technique between UMLS and Brown corpus.

Based on their results, all abbreviations should be complex to any reader and needs simplification.

PorSimples project Aluísio and Gasperin (2010) presented using a compiled lexicon of simple words rather than complex words for low literacy readers of the Portuguese language. This lexicon's simple entries were extracted from books for children. FACILITA tool is designed to simplify web pages by using this lexicon to identify complex words (Watanabe et al., 2009). The tool has been proven to be effective in assessing low literacy readers in understanding complex material, as in news articles. Kajiwara et al.(2013) presented another automatic method of compiling a lexicon of simple words for Japanese children. The lexicon consists of 5,404 simple Japanese words, a manually collected set of the Basic Vocabulary to Learn.

Despite the positive improvements in CWI using the lexicon-based approach, we must note some limitations. Generating a lexicon of complex or simple words is a highly complicated and arduous task. These lexicons are limited from two perspectives; first, these lexicons are domain-specific; second defining what is complex or difficult varies as the different target audiences would consider different words to simplify.

c. Implicit /Inherent CWI

Systems adopting this approach inherit CWI steps within other LS pipeline steps. It is similar to simplify everything approach as these systems target all words in a sentence; however, they do not perform substitution unless the substant is simpler than the original target word. For example, the word w_i is replaced by w_j ($w_i \rightarrow w_j$) only if w_i is more complex than w_j (Paetzold and Specia, 2017). Some researchers used word length or word frequency or both in their word complexity evaluation metrics to distinguish between complex and simple words (Biran et al., 2011; Bott et al., 2012). Bott et al. (2012), in their study to simplify words for people with Dyslexia, found that long and unfamiliar words tend to be challenging to comprehend. Glavaš and Štajner (2015) approach was that a target word is replaced by another alternative only when the new word is more frequent than the target. Another strategy was performed in the substitution selection step by Horn et al.(2014). Their strategy was to add the target word to the potential substitution list, and the system discarded the simplification if the target word was considered the simplest substance for itself.

Implicit CWI was more recognised when LS systems started to exploit MT methods. The earliest examples of this approach by (Specia, 2010; Zhu et al., 2010) applied phrase-based and tree-based MT models, which are trained over newly developed complex-simple parallel corpora. Following these initial steps, Wubben et al. (2012) employed the same MT models by adding a re-ranking step that uses the Levenshtein distance as a metric. Finally, in a continuum of applying new MT methods, Xu et al. (2016) complement a typical statistical MT model trained over a complex-simple parallel corpus for TS. Their results show that their approach outperforms a similar approach by Wubben et al.(2012); it better suits TS and the need for a large complex-to-simple parallel corpus. However, adopting implicit CWI allows focusing on simpler substitution selection instead of spending time and effort on complex word identification.

d. Machine learning-assisted

Expanding on MT and Machine learning developed methods, with providing the availability of parallel complex-to-simple data and a complexity degree labelled corpora. For example, if there is a sentence corpus with words labelled as complex 1 or simple 0, this allows the training of a binary classifier for CWI. Moreover, if the data is labelled with a degree of complexity on a scale (0 to 5), it allows employing a regression model to train and measure the degree of complexity. Despite using traditional methods, most submitted systems applied different machine learning methods, such as typical support vector machines (SVMs), decision trees, and neural networks (Bingel et al., 2016; Kuru, 2016; S.P et al., 2016). Moreover, others applied more complex ML techniques (Mukherjee et al., 2016; Nat, 2016; Choubey and Pateria, 2016).

Many studies obtained machine learning-assisted approaches that were initialised in the CWI task of SemEval 2016 (G. Paetzold and Specia, 2016b). The task involves developing a system to identify difficult words for non-native English speakers in a set of sentences manually annotated by 400 annotators.

It was divided into a small training set of 2,237 sentences containing 20 binary complexity labels and a testing set of 88,221 sentences containing a target word per sentence. Among the 42 systems developed by 21 teams, G. Paetzold and Specia (2016e), using the Performance-Oriented Soft Voting ensemble strategy, outperforms other systems in identifying complex words. Their approach combined different methods as merging lexicon-based, threshold-based, and machine learning approaches.

3.3.2. Substitution Generation [SG]

The task of Substitution Generation SG involves generating all possible substitutions without including ambiguous substances that would confuse the system in the Substitution Selection step. LS systems have tackled SG problems with many different approaches and methodologies with the challenge of producing all but reasonable alternatives to the complex target word. Existing SG approaches follow either of the following two categories,

a. Linguistic database querying

They were resolving SG task started by using manually crafted linguistic databases. Since most work for the SG task has been done in English, many of the LS systems rely on the WordNet or Oxford Psycholinguistic Database (Kučera Francis frequency list) to identify the list of synonyms, hypernyms, and phrases(Carroll et al., 1998; De Belder and Moens, 2010; Biran et al., 2011). However, such resources are expensive and time-consuming to establish and are limited to common languages. Additionally, Shardlow (2014) argued that thesauri such as WordNet do not contain either all English words or the semantic relation between these words, and relying only on WordNet limits the number of generated possible alternatives. Founding that 42% of the errors in the output resulted from the inability to predict a simple variant extracted from WordNet.

To overcome this limitation, researchers started combining resources altogether for better coverage of words in relation. For example, Leroy, Kauchak, et al.(2013), to provide a simplification system in the medical domain, combined three databases, WordNet, Wiktionary (Wikimedia, 2017), and UMLS.[27] (Unified Medical Language System). The latter database was also used by (Chen et al., 2012) in their LS module that used as a base to improve statistical MT systems. Elhadad (2006) also used the "define:" function of Google's search engine to retrieve medical terms' definitions from various dictionaries on the engine database.

b. Automatic generation

Later, researchers start using corpora, mainly parallel corpora, for word alignment. Simple English Wikipedia (PWKP) (William Coster and Kauchak, 2011) was the significant corpus used (Biran et al., 2011, Horn et al., 2014). In other languages, such as Portuguese and Spanish, researchers try to build corpora similar to the Simple English Wikipedia to apply the same algorithms as researchers investigating English. G. Paetzold and Specia (2016c) noted that all the parallel corpora provide only a minimal resource for SS and the final CWI results.

Each of the researchers performed the SG task with different perspectives. One such involved adding a part-of-speech tagger as a pre-processing stage to help in the SS task (Wubben et al., 2012, Kajiwara et al., 2013). Another approach involved word embedding by Glavaš and Štajner (2015) to extract a suitable substitute for complex words. An unsupervised learning approach was also tried (Paetzold and Specia, 2016e). The latter researchers claimed that their approach overcomes the limitation of the other approaches and resolves the problem of word ambiguity. They added two main constraints in selecting the word substitute: (i) the complex and substitute word have the same POS tag; (ii) the substitute word is an unfamiliar word with a novel word root obtained using their published tool LEXenstein[28].

3.3.3. Substitution Selection [SS]

In the process of Substitution Selection (SS), the system is tasked with choosing the most suitable substitute from a list generated by the SG component. This selection process is performed with the goal of maintaining the original meaning and grammatical structure of the sentence while also taking into consideration the

[27] It provides a large ontology containing semantic relations between pairs of medical terms.
[28] http://ghpaetzold.github.io/LEXenstein/

surrounding context. However, considering the fact that a word may have multiple meanings, and different meanings will have different relevant substitutions, the SS task may generate a miss-substitution, which may lead to meaning corruption. Thus, a Word sense disambiguation (WSD) algorithm is needed to best select the substitution list and prevent a loss of meaning and coherence. In that sense, LS cannot be isolated from the discourse and semantic level and adding a semantic module could solve many ambiguity problems(Collados, 2013). The major methodologies for WSD were done through a language model or a word vector or were based on the WordNet dataset.

3.3.4. Substitution Ranking [SR]

After selecting the substitution list, a ranking operation is performed to arrange the substitutes according to the possibility of their occurrence in the given context. Substitution Ranking (SR) is accomplished through many machine-learning approaches. For example, Wubben et al. (2012) used language modelling by the SRILM language model for this ranking task. In contrast, Paetzold and Specia (2017) present a whole LS approach that applies Neural Networks to learn substitutions from a parallel corpus accompanied by the former word embedding technique (Paetzold and Specia, 2016e) and then applying the "Confidence Checker" task as mentioned earlier.

3.4.Syntactic Simplification (SS)

Syntactic simplification, the second primary task in TS, aims to identify the complex grammatical structure in a sentence and regenerate a new simpler sentence that is easy to comprehend. Syntactic simplification is a paraphrasing operation that may involve sentence splitting, anaphora resolution, changing passive voice to active, and simplifying some complex structures such as coordinate clauses, relative clauses, adverbial clauses, dependent infinitives, and complex nominal(Shardlow, 2014). Collados (2013) defined a complex sentence as a sentence that has at least two conjugated verbs. Three techniques used in TS are rule-based, statistical MT, and the latest learning techniques. An example of a simplified English sentence is illustrated in Table 3.3.

Table 3.3 Syntactic Simplification Example(Siddharthan, 2002)

Original Sentence	Simplified Sentence
A) Also contributing to the firmness in copper, the analyst noted, was a report by Chicago purchasing agents, which **precedes the full purchasing agents** report that **is due out today and indicates what the full report might hold.**	B) Also contributing to the firmness in copper, the analyst noted, was a report by Chicago purchasing agents. **The Chicago report** precedes the full purchasing agents' **report. The Chicago report** gives an indication of what the full report might be. **The full report is due out today.**

The first attempt at syntactic simplification was proposed by (Chandrasekar and Srinivas, 1997), applying a rule-based approach to improve the parser performance. They presented the basic pipeline for syntactic simplification, providing the foundation for later rule-based simplification approaches. Their initial pipeline was composed of two main stages: (i) analysis to identify the complex structure; (ii) transformation and simplification. However, Siddharthan (2002) added a remarkable third stage to the pipeline: the generation/regeneration stage.

Some of the other TS systems followed the rule-based system by adding different modifications, such as (Petersen and Ostendorf, 2007; Evans, 2011). Some used a monolingual parallel-aligned corpus of original and simplified texts and applied a different machine-learning algorithm (Petersen and Ostendorf, 2007; Caseli et al., 2009; Aluisio et al., 2010; Specia, 2010; Camacho Collados, 2013). Nevertheless, others considered the TS problem as a monolingual translation problem, best solved by applying the Statistical Machine Translation (SMT) framework (Specia, 2010; Zhu et al., 2010; Woodsend and Lapata, 2011; Wubben et al., 2012). Siddharthan (2002) provides a pipeline for the alternative TS system architecture illustrated in Figure 3.2. Nisioi et al. (2017) expand the use of neural networks in LS (G. Paetzold and Specia, 2016b) to model an entire TS system processing both syntactic and lexical simplification. They applied both word embedding and Neural Machine Translation (NMT).

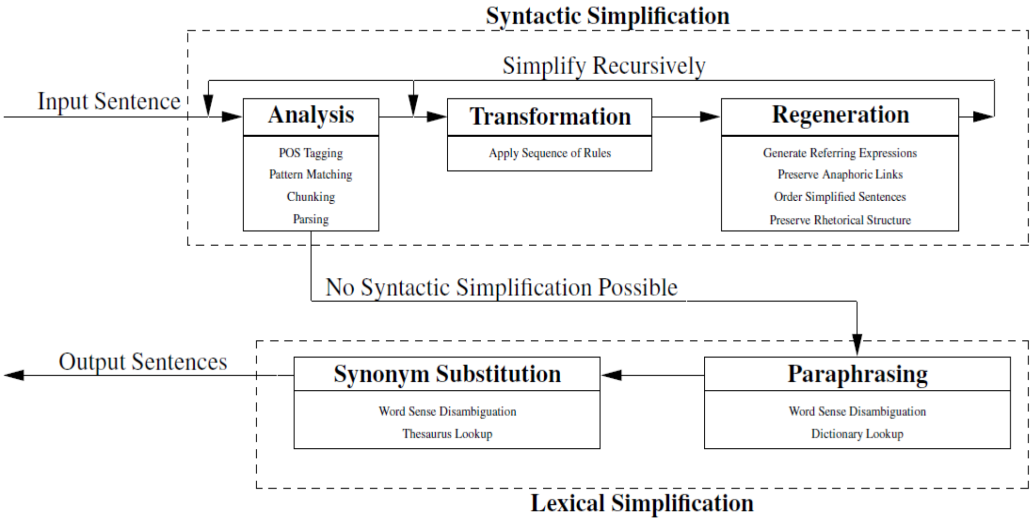


Figure 3.2 TS system architecture ((Siddharthan, 2002), Figure1)

3.4.1. Rule-based syntactic simplification

Rule-based simplification includes defining some complex sentence structures with simplification rules. This method adopts annotated corpus and automatically learns and extracts rewrite simplification rules (Chandrasekar and Srinivas, 1997). Evans and Orăsan (2019) introduced a rule-based method for sentence splitting while preserving the semantic structure. They are using a recursive top-down approach. Each rule specifies (1) how to break down complex statements into structurally simplified statements and reformulate them. (2) How to establish a context hierarchy between elements, and (3) how to identify the semantic relationship that holds exist these components. Syntactic simplification was performed following three main steps, as represented in Figure 3.2.

1- The *linguistic analysis* contains both POS tagging and parsing, and most researchers use a dependency parser in order to identify complex sentences. Therefore, performing the parse tree of each sentence is mostly phrase parsing or dependency parsing while keeping the chunks' relations. During this step, the complex sentences were identified, and the decision was made on which sentences could be simplified. This was done mainly using predefined automatic matching rules or a binary complex/simple machine learning classifier (Shardlow, 2014).

2- They used *transformational Rules*, or the rewritten rules to perform sentence simplification such as sentence splitting, phrase rearrangement (Siddharthan, 2004), phrase deleting (Specia, 2010), and adding new connectors according to rewrite predefined rules (Hervás et al., 2014). However, most of the transformational rules were handwritten rules; some systems used automatic rules generated from a parallel annotated corpus.

3- *Regeneration*, as the transformational stage, may result in a new, wrongly simplified structure that needs to be fixed in the regeneration stage. The fixing or Regeneration stage is the most challenging task, and it must consider relations within the sentences and the anaphoric references. Regeneration is done by applying rules of how to connect the new sentences with new word order and new agreement rules to ensure a simplified, well-formed structure (Siddharthan, 2011).

However, rule-based systems are known for their accuracy, and the creation and validation of these rules are time and effort-consuming. That convinces moving toward deep learning and automation of simplification rules identification and application (Niklaus et al., 2021).

3.4.2. Statistical Machine Translation (SMT)

As Zhu et al. (2010) stated: "*consider the sentence simplification as a special form of translation with the complex sentence as the source and the simple sentence as the target*". In other words, treat syntactic simplification as a monolingual text-to-text generation task. This involves a Sequence-to-Sequence transformation trained on a parallel corpus. TS has been done by applying SMT for English (Zhu et al., 2010; William Coster and Kauchak, 2011; Wubben et al., 2012), Brazilian and Portuguese (Specia, 2010), German (Klaper et al., 2013), Chinese (Chen et al., 2012) and Swedish (Stymne et al., 2013). SMT-based sentence simplification systems used the major tools for regular translation purposes, such as GIZA++ and Moses (Vaidya, 2014).

NMT techniques have dominated the field of TS in recent years, producing simpler sentences while preserving the meaning and grammatical well-formedness (G. Paetzold and Specia, 2016e).

Wang, Chen, Amaral, et al.(2016) build a sentence-level TS model applying the Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Encoder-Decoder model. Their proposed model trained on a parallel complex/simple sentence was able to learn rules such as sentence pattern changes, word replacements, and word deletions. However, the model performance was limited by the lack of aligned parallel complex/simple sentence pairs (Wang, Chen, Rochford et al., 2016).

Bingel and Søgaard (2016) approached TS using linear-chain conditional random fields trained over top-down dependency syntactic graphs. This method allowed both sentence compression and sentence paraphrasing by learning the entire syntactic trees and subtrees using the dependency graphs to reduce the production of ungrammatical output.

Zhang and Lapata (2017) provided a Sentence Simplification model based on reinforcement learning to overcome the seq2seq output issues. Their model learning to optimise a reward function provides LS and grammatical and semantic meaning preservation to the output. They reported good results based on BLEU (Papineni et al., 2002) and SARI (Xu et al., 2016) evaluation metrics on three different datasets. Furthermore, they found that reinforcement learning provides an excellent way to put prior knowledge into the task of simplification.

Another neural Seq2Seq model proposed by Nisioi et al. (2017) is the model trained on simple English Wikipedia(William Coster and Kauchak, 2011). Their model provides LS with content reduction. They reported through extensive human testing and, based on evaluative measures, they have shown that their Neural Text Simplification System (NTS) achieves near-perfect preservation of grammar and the meaning of the output sentences while creating a higher degree of simplification.

Sulem et al. (2018b) proposed both a simplification system targeting structural modification and a structural-aware evaluation metric [Called SAMSA], showing that it outperforms existing lexical and structural systems. Furthermore, they experimentally proved that robust measures of the quality of LS as SARI metric are not correlated with human judgments when structural simplification is performed. Vu et al. (2018) propose using a memory-augmented RNN architecture called Neural Semantic Encoders (NSE) rather than the traditional LSTM seq2seq model. The results obtained from both automatic and human evaluation of various datasets show that those models perform well in terms of grammatical and semantic retention while significantly decreasing the difficulty of reading the input.

Since the lack of training data is the main limitation affecting TS's accuracy, recent advances and research directions are aimed at solving this problem. Aprosio et al. (2019), utilising large amounts of heterogeneous data, automatically select simple sentences and uses them to create synthetic simplification pairs. These techniques provide better performance than the basic seq2seq setup.

Surya et al. (2019) propose an unsupervised NTS model for TS using unlabeled data from regular and simple Wikipedia. The framework they proposed is with a standard encoder with pair of a pair of attentional decoders supported by identification-based loss and denoising, which can perform TS at both lexical and syntactic levels.

Recently, Qiang and Wu (2021) proposed a new phrase-based unsupervised TS system based on phrase tables from regular Wikipedia and initialised two language models (Complex LM and Simple LM) without the need for parallel sentence pairs. Instead, they use Wikipedia as a vast source of information to enter data into the phrase table and get word embeddings that capture the frequency of words that reflect the semantic characteristics and difficulty of the words. They reported that the model is superior to some supervised models based on BLEU and SARI evaluations.

Shardlow and Alva-Manchego (2022) introduce the application of TS in MT scenario. They manually simplified the Spanish translation portions of around 6,000 sentences of English TICO-19 corpus. Then they experimented with the translation performance of the models when using Simplified sentences as input and as references. Their results proved that the prior simplifications of the original texts led to an overall increase in readability. (Felice et al., 2022)

Martin et al. (2022) proposed a sentence simplification approach using large-scale mining of sentence-level paraphrases from the web instead of a parallel simplification dataset. When evaluating the unsupervised TS system using SARI scores, they reported that the results either outperformed or matched the baseline performance of other studies. In addition, their system gives 95% confidence for English-produced simple sentences using native speakers' judgment evaluation on a 5-point Likert scale measuring adequacy, fluency, and simplicity. However, there still exist challenges with these approaches; the main limitation is that there are too many multi-word expressions and named entities present in the source text (G. Paetzold and Specia, 2016e; Martin et al., 2022).

Other TS systems applied the word embedding technique in reforming the new sentences. Word embedding is a technique used to represent words in a valued vector space in which words with similar meanings are located in the same space (Pennington et al., 2014). In this case, word embedding representation is used to identify the list of synonyms of complex words based on the semantic similarity and context similarity between the target and the original word.

Qiang and Wu (2021) and Sikka and Mago (2020) address challenges in generating simple sentences from complex ones. Qiang and Wu's approach focuses on using phrase-tables generated from word embeddings and word frequency to achieve this goal, while Sikka and Mago identify issues with seq2seq models that rely on word embeddings and parallel corpora.

It's interesting to note that both methods share a common limitation in that they heavily depend on the quality and quantity of the training corpus. This suggests that more research is needed to develop models that can generalize better across different types of complex sentences. Sikka and Mago's identification of issues with word embeddings and repetition in generated sentences also highlights the importance of designing models that can accurately capture contextual information and avoid over-reliance on frequent words. This can potentially be addressed through incorporating techniques like attention mechanisms and using different types of embeddings, such as contextualized embeddings like BERT.

3.5. Deep learning embeddings

One of the latest techniques in NLP is Word embeddings (WE) and Pre-training of Deep Bidirectional Transformers (BERT). WEs are technique of identifying and categorising semantic similarities between words based on their distributional features in a large corpus. Word embedding refers to a class of language modelling and feature learning approaches used in NLP in which words or phrases from the lexicon are mapped to a continuous d-dimensional vector (Lebret, 2016).

Recent Word Embeddings models are word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), Gensim (Řehůřek and Sojka, 2010), and fastText (Grave et al., 2018). There are two main learning algorithms in Word2Vec: continuous bag-of-words and continuous skip-gram. The word2vec algorithm includes a bag of words model that involves predicting the context words using a centre word. Both algorithms learn the representation of a word that is useful for the prediction of other words in the sentence. The main limitation of word2vec is the out-of-vocabulary (OOV), as the embedding is created for each word. As such, it cannot handle any words it has not encountered during its training. While the skip grams model, as in fastText, involves predicting the word using the context words. fastText was to use the internal structure of a word to improve vector representations obtained from the skip-gram method character embedding's list of character n-grams for a word. However, any word embedding technique cannot capture the meanings of multi-word expressions and phrases, which may have a different meaning from their constituent individually.

The ***fastText*** model is a python library used for learning text classification and word embedding. It comprises an unsupervised learning algorithm based on character n-grams to obtain vector representations for words. Facebook research centre developed a fastText tool which classifies text using a supervised as well as an unsupervised learning algorithm. This embedding was trained on Common Crawl and Wikipedia using the fastText[29] tool.

Followed by the Sent2Vec and Doc2Vec are extensions of Word2Vec, where the model calculates the average of the word vector representations of all the words in the sentence or documents. As such, ***the Universal sentence encoder*** (Yang et al., 2019), a multilingual model, requires modelling the meaning of word sequences (sentences) rather than just individual words. ***Multilingual BERT*** (Devlin et al., 2019) mBERT is a pre-trained transformers models that proved their ability to learn successful representations of language inspired by the transformer model presented by (Vaswani et al., 2017) — who introduced using attention instead to incorporate context information into sequence representation. ***XLM-R*** (Conneau et al., 2019) is another multilingual BERT-like model, which is different from mBERT by being

[29] https://fasttext.cc/docs/en/crawl-vectors.html

trained on Common Crawl (instead of Wikipedia's) with slightly different parameters. It is trained on multiple languages, including Arabic.

AraBERT (Antoun et al., 2020) and ***Arabic-BERT*** (Safaya et al., 2020) are two popular BERT-based pre-trained transformer models specifically designed for the Arabic language. Both models contain both Modern Standard Arabic (MSA) and Dialectal Arabic (DA) and have been trained on large corpora of Arabic text.

AraBERT was trained on 70 million sentences, while Arabic-BERT was trained on a combination of filtered Arabic Common Crawl and a recent dump of Arabic Wikipedia, containing approximately 8.2 billion words. Both models have demonstrated impressive performance on various NLP tasks, including sentiment analysis, named entity recognition, and text classification.

The ***Masked Language Modeling*** (MLM) task is one of the two pre-training tasks used in BERT. The other task is Next Sentence Prediction (NSP). In the MLM task, BERT randomly masks a certain percentage of the input tokens and then tries to predict the original value of those tokens based on the context provided by the surrounding tokens. The MLM task helps BERT to learn contextual representations of words and phrases, which are useful for various NLP tasks.

In the TS task, the Masked Language Modeling (MLM) task of BERT is used. In the MLM task, a certain percentage of the input tokens are randomly masked, and the model is trained to predict the masked tokens based on the context provided by the other tokens in the input sequence. The MLM task helps BERT to learn contextual representations of words and phrases, which are useful for various NLP tasks, including text summarization. When BERT is fine-tuned on the TS task, it is trained to generate a simple output of the input text based on the learned contextual representations. An extensive explanation of how MLM is used for TS task is presented in chapter 5 section 5.1.2.1.

Recently, Raffel et al., 2020 presented ***T5 "Text-to-Text Transfer Transformer***"[30]. T5 is a powerful text-to-text transfer transformer model that has been developed by Google's AI team, and it has shown to achieve state-of-the-art results on various

[30] https://simpletransformers.ai/docs/ t5-specifics/

natural language processing (NLP) tasks. The T5 model is pre-trained on a large corpus of text data using a sequence-to-sequence (seq2seq) approach.

T5 is a BERT-like seq2seq transformer that takes input a text and trains it on the model to generate target text. The pre-training process involves training the model to perform a range of tasks, such as language modeling, machine translation, summarization, and others, using a large amount of text data(Raffel et al., 2020).

3.6. Arabic Text Simplification

Automatic Arabic TS is a particularly challenging process because Arabic is a highly morphologically rich language with flexible word order; the Arabic nouns are multi-functional; most text lacks vocalisation diacritics and has pro-drop nature or 'hidden pronouns'. Habash (2010) explained the Arabic pro-drop nature in his book, expressing the dropping of the pronouns in Arabic verbal sentences (VB). These types of VBs with verbs with implicit or empty subjects are referred to as "*Empty pronoun*" or "*Hidden pronoun.*" [31] that the subject is pronominal and integrated into the verb itself. Much complexity arises from the fact that many syntactic structures are expressed by changing the morphological pattern of the word (Habash, 2010). This author states that "*Arabic rich morphology allows it to have some degree of freedom in word order since the morphology can express some syntactic relations*". Moreover, there is an absence of consensus on the reliability of NLP tools and corpora for the Arabic language.

Arabic syntactic characteristics lead to many challenges for Arabic lexical and syntactic simplification. Some of these characteristics are: (i) Arabic is a free word order language (it allows three possible structures verb-subject-object (VSO), subject-verb-object (SVO) and object-verb-subject (OVS); (ii) Arabic is a pro-drop language (which means that the property of dropping the subject pronoun and allowing subject-less or prepositional sentences); (iii) Arabic is an agglutinative language (a single Arabic word may contain up to four different morphemes, for example, a verb

[31] An example of the hidden pronoun extracted form (Habash, 2010)
(ضمير مستتَر) in كَتَبَ+هَاKatab + a + ha
Wrote+3rd person singular masculine + it
Verb + Subject + Object [He wrote it]

may embed within itself its subject and object as well as other clitics signifying tense, gender, person, number, and voice) (Farghaly and Shaalan, 2009; Fehri, 2013).

An example from Farghaly and Shaalan (2009) clarifies the syntactic ambiguity in a simple sentence with a prepositional in the sentence “قَابَلْتُ مُدِيرَ الْبَنْكِ الْجَدِيدِ” as illustrated in Table 3.4. That the sentence may have two different translations, first: if the adjective “الْجَدِيدِ” (aljadīdi, new) refers to either the “الْبَنْكِ” (albanki, bank) or “مُدِيرَ الْبَنْكِ” (mudīra albanki, bank manager).

These characterastics make it challenging to specify different phrase patterns that will be simplified for syntactic simplification tasks. This is because it is rich in syntax and semantics, and it is difficult to anticipate all the possible variations in language that may occur.

Table 3.4 Different analyses depend on the noun phrase's internal analysis.

Sentence	قَابَلْتُ مُدِيرَ الْبَنْكِ الْجَدِيدِ	
Transliteration	qābaltu mudīra albanki aljadīdi	
Possible Translation	I met with the ***new manager of the bank***	I met with the manager of the ***new bank***

The following Table 3.5 is an example of Arabic TS on both lexical and syntactic levels.

Table 3.5 Complex Arabic sentence with two possible simplifications

Original	**Arabic**	**دَخَلَ رَجُلٌ يَحْمِلُ عَدَداً كَبِيراً مِنْ الْكُتُبِ إِلَى الْغُرْفَةِ**
	Translitration	daḳala rajulun yaḥmilu ʿadadun kabīrun min alkutubi ʾilā alġurfati
	Translation	A man carrying a large number of books entered the room
Simplification1 Split into two sentences	Arabic	دَخَلَ رَجُلٌ إِلَى الْغُرْفَةِ وَهُوَ يَحْمِلُ كَثِيرٌ مِنْ كُتُبٍ
	Translitration	daḳala rajulun ʾilā alġurfati wahuwa yaḥmilu kaṯīrun min kutubin
	Translation	A man walked into the room and he was carrying a lot of books.
Simplification2 Simpler grammar stucture	Arabic	دَخَلَ رَجُلٌ إِلَى الْغُرْفَةِ، مَعَهُ كُتُبٌ كَثِيرَةٌ
	Translitration	daḳala rajulun ʾilā alġurfati maʿahu kutubun kaṯīratun
	Translation	A man walked into the room with many books.

As presented in Figure 3.3, the object is a whole verbal phrase (VP) in the original complex sentence combined two verbs "دَخَلَ"(daḳala, 'entered ') and "يَحْمِلُ" (yaḥmilu, 'carrying'). In simplification1, performing a syntactic simplification appeared in splitting the sentence (S) into two sub-phrases; each phrase is a VP starting with one of the original verbs in the complex sentence. In addition, a LS by replacing " عَدَداً كَبِيراً " ('adadan kabīran , a large number of) with "كَثِيرٌ مِنْ" (kaṯīrun min, a lot of).

In simplification2, the syntactic simplification appeared in the deletion of the verb "يَحْمِلُ" (yaḥmilu, 'carrying') and added adverb "+مَعَ" (maʿa+, with the) attached to the pronoun "+هُ"(hu, he) referring to the man. Whereas, the LS preseneted in replacing " عَدَداً كَبِيراً" ('adadan kabīran , a large number of) with one word "كَثِيرَةٌ" (kaṯīrun min, many).

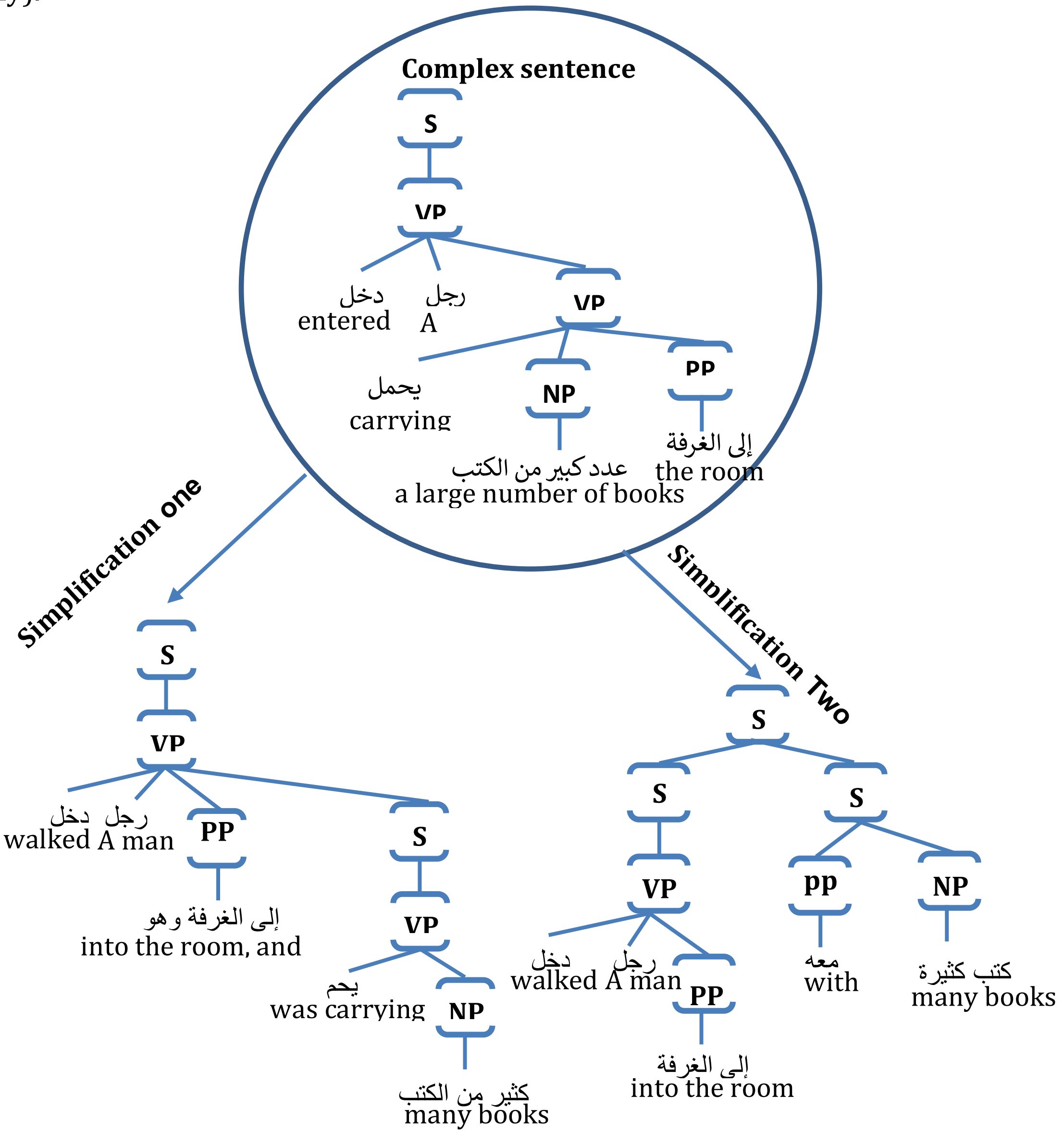


Figure 3.3 A representation of two possible simplified sentences for the complex sentence "دَخَلَ رَجُلٌ يَحْمِلُ عَدَداً كَبِيراً مِنْ الْكُتُبِ إِلَى الْغُرْفَةِ "A man carrying a large number of books entered the room."

Unlike English, only a few researchers have been tackling the problems of Arabic ATS. Founding only a prototype unreleased ATS system by (Al-Subaihin and Al-Khalifa, 2011) at King Saud University, which is inaccessible, and another starting project by (Al Khalil et al., 2017) at New York University in Abu-Dhabi provided a description of unreleased Arabic TS such *as "a levelled reading corpus of modern standard Arabic"* (Al Khalil et al., 2018) and "*a levelled readability lexicon for standard Arabic*"(Al Khalil et al., 2020).

Al-Subaihin and Al-Khalifa (2011), in their paper, highlighted the importance of implementing an Arabic ATS (AATS). The proposed system architecture for AATS in light of the state of the art of systems for other languages. They were targeting a different group of end-users, such as *SYSTAR*, a syntactic simplification system for the English aphasic or inarticulate population(Carroll et al., 1998). Another system, *SIMPLIFICA*, is a simplification tool for Brazilian Portuguese (BP) targeting those with low literacy levels (Scarton et al., 2010). Adopting methods and techniques used in these systems while stating the Arabic alternative available resources, they proposed a design for AATS named ***Al-Baseet***. Their design was constructed of four main stages: i) measuring complexity; ii) vocabulary (lexical) simplification; iii) syntactic simplification; iv) diacratisation. In the first stage, measuring complexity, they offer two different techniques to calculate text readability. They would adopt a statistical language model based on a machine learning technique called *ARABILITY* (Al-Khalifa and Al-Ajlan, 2010). Rather than using a traditional technique by applying Arabic readability formulae (AI-Dawood) and (AI-Heeti), as cited in Cavalli-Sforza et al. (2018). Their decision was based on two factors; first, most readability formula barely measures the syntactic complexity as they are based on words and sentence lengths. The second factor was their system targeting 3-way classification: simple, moderate, and complex, which could be reflected easily using *ARABILITY* classification: easy, medium, and difficult. The second stage, vocabulary simplification, referred to as LS, follows the state of the art of LS. This stage is composed of four steps:

i) Text tokenisation applying *MADA* (Habash et al., 2009), a stem-based approach that provides dicratisation ;
ii) Identify the complex word; they follow 'simplify everything' except words tagged as proper nouns, numbers, and prepositions.

iii) List of synonyms, they suggested either building a new dictionary combining available dictionaries and treasures (such as *Al-Baheth Al-Arabi* (Naeem, n.d.) and *Al-Radif* (Anon, n.d.)) or using *Arabic WordNet*[32](Elkateb et al., 2006), which stores the words in clusters of synonyms;

iv) Select the most common and possible synonym; using the Google API online search, they picked the best synonyms through the most frequent search keyword.

Their third stage, syntactic simplification, involves syntactic analysis of the sentences first. Then identifying the complex structures by applying a look-up approach to a predefined list of Arabic complex structures, indicating that using a set of manual simplification rules is the only way to perform syntactic simplification. Linguists could propose these rules when building Arabic text for Arabic learners. A language generation module follows this to ensure the word location and referring pronouns to resolve any sentence breakdown caused by partial sentence structure replacement. The last stage is diacratisation, using *MADA* diacritiser task. They suggested that this stage to add vowelisation would facilitate and improve the text's readability effortlessly. The main limitation of implementing this system at this point is the unavailability of essential Arabic resources and tools. Such as dictionaries, corpora, and parallel complex-simple structures are the main components of any ATS system.

The second attempt to build an AATS system was by (Al Khalil et al., 2017). They aimed to provide essential Arabic resources for building ATS and formulating manual simplification rules for Arabic fiction novels using TS stat-of-the-art. 1M tokens of the 12-grade curriculum, and 5.6M tokens of the adult novels (original and simplified counterparts)(Al Khalil et al., 2018). However, this resource is not available due to copyrights.

This corpus would be broadly graded using a readability scale driven from the graded part of the corpus, which is the 12-grade curriculum and applied to other corpus parts. Their readability scale uses a new classification based on the ACTFL language proficiency levels. Also, they provided a proposal to the SAMER

[32] http://globalwordnet.org/resources/arabic-wordnet/

(Simplification of Arabic Masterpieces for Extensive Reading) project based on the corpus analysis. Their guidelines invoke both the MADAMIRA (Pasha et al., 2014) for part of the speech tagger and Camel dependency parser (Shahrour et al., 2016) for data analysis and classification of their corpus. They aimed to build a readability measurement identifier to formulate a 4-levelled graded reader scale (GRS) by applying various machine-learning classifiers. Their simplification system was designed to be semi-automatic to simplify modern Arabic fiction; it involved a linguist using a web-based application to apply ACTFL guidelines for simplifying five Arabic novels.

After that, they produced a SAMER levelled list that consists of a 26,000-lemma five-levelled readability lexicon for MSA(Al Khalil et al., 2020). It was manually annotated with three different language speakers who speak different dialects A1 (Egypt), A2 (Syria/Levant), and A3 (Saudi Arabia/Gulf). Then the words were labelled by averaging the assigned label from the three experts. These levels were:

- Level 1: Generally corresponding to Grade 1,
- Level 2: Generally corresponding to Grades 2-3,
- Level 3: Generally corresponding to Grades 4-5
- Level 4: Generally corresponding to Grades 6-8
- Level 5: This level reflects specialist language use beyond the eighth grade.

It should be noted that, this project is still under creation with auspicious initial results.

3.7. Evaluation methods

Likewise, most TS evaluation approaches have been driven from other similar NLP research areas. Various evaluation methods have been applied across research to measure the three main aspects of the newly generated text. These aspects are: i) *fluency,* referring to the grammatically well-formedness and structure simplicity; ii) *adequacy*, meaning preservation; iii) *simplicity*, more readable. There are three major TS evaluation algorithms, which could be applied separately or in combination. Either automatically by applying borrowed MT evaluation techniques and using readability classifiers measurements or manually by direct human judgments. Many existing TS evaluation methods do not generalise across systems, as (Xu et al., 2016) stated, because they fail to represent the cumulative effects of the various simplification processes. Most often, TS researchers combine methods for robust evaluation. A more detailed account of TS various metrics is given in the following section.

3.7.1. Manual TS evaluation

Manual TS evaluation refers to assessing the system by human judgments of the output's three main aspects: fluency, adequacy, and Simplicity (Wubben et al., 2012; Saggion, 2017). Grammaticality refers to whether the simplified sentence is grammatically well-formed, whereas simplicity refers to how simple the new simplified sentence is compared to the complex sentence. Adequacy defines the degree to which the original meaning is preserved after simplification is applied. These three features are usually rated on a Likert scale of 1–5 or 1–3, with a higher score indicating better simplicity.

The manual TS evaluation approach adopted by many studies such as (Specia, 2010; Biran et al., 2011; Woodsend and Lapata, 2011; Wubben et al., 2012; Glavaš and Štajner, 2015; G. Paetzold and Specia, 2016e; Nisioi et al., 2017; Štajner and Glavaš, 2017), with the basic concept of presenting the complex sentences with their simplifications to the participants and asking for their evaluation. Nisioi et al. (2017) proposed two types of human evaluation according to the linguistic criteria under investigation. For both fluency and adequacy, the participants were three native English speakers giving a simple sentence on a Likert scale from 1-to 5. Whereas assessing the simplicity of sentences, the participants were three non-native fluent English speakers who were shown both original sentences and simplified counterparts as pairs.

A few issues limit the application of TS manual evaluation. First, it requires native speakers with linguistic background knowledge to assess the simplified sentence. Second, humans are inconsistent even if they have predefined measuring guidelines. In addition, human evaluation is typically performed with a small number of sentences (18 to 20) randomly chosen from the test set (Wubben et al., 2012; Narayan and Gardent, 2014; Narayan and Gardent, 2015). This makes it difficult to compare different TS systems, especially when different individuals are involved. Furthermore, the manual examination is costly and time-consuming. These limitations motivate the TS researchers to investigate automated ways of evaluating the output; several metrics have been developed to evaluate the quality of simplification discussed in the following section.

3.7.2. Automatic TS evaluation

There are two basic approaches currently being adopted in TS automatic evaluation. The first technique is simply by measuring readability using the standard readability formulae, which gives an estimated measure of text difficulty; for example, studies (Siddharthan, 2004; Zhu et al., 2010; Wubben et al., 2012; Zhang and Lapata, 2017; Štajner and Glavaš, 2017). However, most of these formulae are superficial as they derive the estimation based on; i) words, sentences, and syllables as in Flesch Reading Ease, Flesch Kincaid Grade; ii) characters per word used in Coleman Liau; iii) characters, words, and sentences in Automated Readability Index; iv) regular and complex words per sentence, Gunning FOG; v) "easy" and "hard" words per sentence, Linear Write Formula. One of these metrics' limitations is that it gives higher weight to short sentences based on the average length of sentence *(ASL), ASL (Average Sentence Length) an*d the average number of syllables in a word *(ASW)* [as explained earlier in the readability section]. Also, it is worth noting that neither the number of sentences nor the number of words would account for any of the three main aspects of measuring simplicity.

Secondly, leading studies promote using MT evaluation metrics in the same way; they adopted MT approaches for the simplification task; for example, studies (Specia, 2010; Wubben et al., 2012; Wubben et al., 2012; Narayan and Gardent, 2014). Different MT evaluation methods have been proposed, such as Bilingual Evaluation Understudy- ***BLEU*** (Papineni et al., 2002), National Institute of Standards and Technology ***NIST*** (Doddington, 2002), and Translation Edit Rate-***TER*** (Snover et al., 2006) and Translation Edit Rate-plus –***TERp*** (Snover et al., 2009). A more detailed account of these methods is given in the following section.

- ***BLEU*** (Papineni et al., 2002), an evaluation metric used originally to measure the accuracy of the MT output, is based on exact n-gram matching word reordering and sentence shortening. It measures the similarity between the system's simplification and the gold standard reference. It is widely used to evaluate MT, text summarisation systems, and text simplification. It is based on a "weighted average of similar length phrase matches" (n-grams), and it is sensitive to longer n-grams (the baseline is the use of up to 4-grams). BLEU score could be calculated according to the following formula:

$$Bleu(\mathrm{S},\mathrm{R}) = K(S,R) * e^{Bleu1(S,R)} \qquad (1)$$

$$Bleu(\mathrm{S},\mathrm{R}) = \sum_{i=1,2,\ldots n} wi * \lg\left(\frac{(|S_i \cap R_i|)}{S_i}\right) \qquad (2)$$

$$K(\mathrm{S},\mathrm{R}) = \begin{cases} 1 & if\ |S| > |R| \\ e^{\left(1\frac{|R|}{|S|}\right)} & otherwise \end{cases} \qquad (3)$$

$$w_i = \frac{i}{\sum_{j=1,2,..n} j} \qquad for\ i = 1,2,\ldots,n, \qquad (4)$$

- [S] the system set, [Si] is the bag of i-grams for the system

- [R] reference set, [Ri] is a bag of i-grams for reference

- n is the size of the n-gram

BLEU is the most widely applied MT evaluation method for TS evaluation by many studies, e.g. (Woodsend and Lapata, 2011; Xu et al., 2016; Nisioi et al., 2017; Ma and Sun, 2017). On the other hand, some studies reported a correlation with human evaluation, such as spearman correlations between the ranking of the automatic metrics and the human judgments. On the one hand, (Wubben et al.(2012) used 20 source sentences from the PWKP, and test corpus with five simplified sentences for each of them. They reported a positive correlation of BLEU with measuring simplicity, yet it fails to consider adequacy. On the other hand, an LS study by (Xu et al., 2016) claims that BLEU fails to capture simplicity even when using multiple gold-standard references; however, BLEU scores achieve a rational correlation between fluency and adequacy. Štajner et al.(2014) investigated the correlation of six automatic metrics with human judgment, cosine similarity with a bag-of-words representation, METEOR (Denkowski and Lavie, 2011), TERp (Snover et al., 2009), TINE (Rios et al., 2011), and two sub-components of TINE: T-BLEU (a variant of BLEU introduced by using lower n-grams with maximum 4-grams) and SRL (based on semantic role Labelling). Their experiment considered only sentences with structural changes in 280 pairs of a source sentence and their simplification. In this case, BLEU was found to provide a reasonable positive correlation for adequacy (meaning preservation) and a weak positive correlation for fluency (grammaticality). They did not report any correlation with simplicity.

- ***NIST*** (National Institute of Standards and Technology) (Doddington, 2002) is a metric based on BLEU. Also, it is a method for evaluating the quality of MT-generated text. NIST is based on n-gram matching like BLEU between gold standard reference and system simplifications, adding weights to different n-grams. The main advantage over BLEU is that the slight differences in the length of the system's output and the human reference do not impact the overall score. However, it is not widely used across studies. The following formula calculates NIST's score,

$$\boldsymbol{NIST\ Score} = \sum_{n=1}^{N} \left\{ \frac{\sum_{all\ w_1 \ldots w_n\ that\ co-occur} Info(w_1 \ldots w_n)}{\sum_{all\ w_1 \ldots w_n\ in\ sysoutput}(1)} \right\} . exp \left\{ \beta\ log^2 \left[min \left(\frac{L_{sys}}{\bar{L}_{ref}}, 1 \right) \right] \right\},$$

Where,

- $Info(w_1 \ldots w_n) = \log_2 \left(\frac{\#\ occurances\ of w_1 \ldots w_{n-1}}{\#\ occurances\ of w_1 \ldots w_n} \right)$
- N =5
- Β is chosen to make the brevity penalty factor = 0.5
- $\bar{L}_{ref}$ = the average number of words in a reference translation averaged over all reference translations.
- L_{sys} = the number of words in the translation being scored

- Translation Edit Rate-***TER*** (Snover et al., 2006) and the extension metric Translation Edit Rate–plus–***TERp*** (Snover et al., 2009). TERp, an MT evaluation method, adopts a bottom-up approach that measures the number of modifications needed to transform the simplified text back into the original complex text. TERp components measures consider major sentence editing types, such as using phrasal substitutions, paraphrasing, morphological changes, synonyms, and relaxed shifting constraints. The higher TERp value indicates less similarity between the simple output and the original text.

- ***iBLEU*** (Sun and Zhou, 2012), a revised BLEU score, is a log-linear model that combines translation and language models. Measures and ranks the quality of a set of candidates of paraphrased sentences compared to the reference and to the input in choosing the best candidate. iBLEU adds α as a parameter taking the balance between adequacy and dissimilarity, calculated according to the following,

$$iBLUE(s, r_s, c) = \alpha BLEU(c, r_s) - (1 - \alpha) BLUE(c, s)$$

- ***FKBLEU*** and ***SARI (System output Against References and the Input)*** (Xu et al., 2016) proposed the first two metrics designed specially to evaluate the TS system's production. **FKBLEU** is a geometric mean of the combination of iBLEU (Sun and Zhou, 2012), as a measure of the paraphrasing quality, and of the Flesch-Kincaid Index (FK) (Kincaid et al., 1975), as a readability measure. In contrast, **SARI** (System output Against References and against the Input sentence) metric compares system output against several human references and original input sentences. Separately measuring the quality of three LS operations involves word deletion, addition, and retention. These evaluation metrics have been used in studies by(Zhang et al., 2017; Nisioi et al., 2017; Sulem et al., 2018c). It has been proved that FKBLEU and SARI have a higher correlation with Simplicity than BLEU. However, these metrics require a set of human simplified references to compare with the system output, which is difficult to obtain. Moreover, providing BLEU with this set of simplified references outperforms other metrics for measuring output adequacy and fluency.

Besides these significant automatic evaluation techniques, several studies used their own combined methods to evaluate the structural modifications in the output in-depth. (Clarke and Lapata (2006), in their study of sentence compression (similar to syntactic simplification), found that a metric based on syntactic dependencies analysis highly correlates with human evaluation better than a metric based on surface sub-strings. (Clarke and Lapata, 2006; Toutanova et al., 2016), while using structure-aware evaluation metrics applying syntactic analysis on both reference and output. Toutanova et al. (2016) found that these measures highly correlate with human grammatical judgments better than bi-gram models. A recent study by (Sulem et al., 2018a) argued that BLEU is unsuitable for TS evaluation because it cannot capture sentence paraphrasing that involves sentence splitting and structural changes. In their experiment, they investigate the behaviour of BLEU while measuring the simplicity of splitting sentences by using a manually complied parallel corpus with four simplified references. Their finding proved that BLEU correlates negatively with simplicity, with a low correlation for grammaticality or meaning preservation.

Recently, a final suggestion has been to include semantic-based measurements. This started by using the structural simplicity and semantically evaluation metric SAMSA Simplification Automatic evaluation Measure through Semantic Annotation (Sulem et al., 2018b). SAMSA metric is based on semantic structural analysis instead of syntactic ones; this analysis is applied to both the input and output. Also, it requires only the input and the output without manually crafted references to perform the comparative evaluation. Sulem et al. (2018b) proved the effectiveness of using semantic-based measurements to evaluate TS, mainly when including sentence splitting.

This is followed by presenting **BERTScore** evaluation, a more extensive semantic-based evaluation following the broader use of BERT transformers. BERTScore is an automatic evaluation metric that computes cosine similarity scores using BERT embedding (Zhang et al., 2020). BERTScore leverages the pre-trained contextual embeddings from BERT and matches words in candidate and reference sentences by cosine similarity. It has been shown to correlate with human judgment on sentence-level and system-level evaluation. Moreover, BERTScore computes precision, recall, and F1 measure, which can be useful for evaluating different language generation tasks.

As BERT provides a better representation of the language's contextual structure and is less sensitive to natural variation. BERTscore evaluation correlates better with human judgments regarding the measurements of sentence similarity. For example, it accepts 'brilliant' as a word replacement when the source says 'excellent', while BLEU will count this as an error.

BERTSCORE evaluation metrics overcome the limitations of the previous MT evaluation metrics, such as BLEU and SARI, n-gram-based evaluation metrics. These methods were not able to capture two main simplification features: 1) changing word order as paraphrasing simplification method, 2) maintaining the deep structure meaning, despite changes in the surface form structure. Basically, n-gram models match the exact order of the words. So that an evaluation system may either give a high similarity score if the two sentences share the same sentence chunk despite the actual occurring context, or it provides a lower score; however, the two sentences share the same meaning expressed in different words and word order. Thus, any BERT model would likely capture a complete representation of deep sentence structure.

3.8. Conclusion

This chapter outlines various approaches and techniques toward TS. Most current TS approaches are abstract and include either LS or Novel Text Generation. On the one hand, LS studies followed the pipeline of identifying complex words, generating synonyms, ranking them, and selecting the best substitute by applying various NLP techniques. On the other hand, generating new text is approached by syntactic simplification, SMT, and seq2seq modelling, using deep neural learning techniques. The main challenges were the availability of a parallel corpus consisting of complex and simple sentence pairs and the development of evaluation metrics that can measure the subjective nature of the language and the readability levels of a simplified text.

The advent of more affordable computing resources and developments in language software support has boosted interest in TS research in recent years. However, as a field and part of NLP, TS is still in its infancy. This creates issues, including a lack of access to the diverse set of high-quality data sources necessary for automated TS, particularly when employing Artificial Intelligence (AI) and deep learning technology.

Moreover, the linguistic component of TS research presents additional obstacles, such as the inaccuracy of measuring simplifications.

Furthermore, simplifying complex sentences while preserving their meaning is a difficult problem, and there is no one-size-fits-all solution. Different types of complex sentences may require different types of simplification strategies. For example, some complex sentences may require splitting into multiple simpler sentences, while others may require the substitution of complex phrases with simpler ones, or the use of simpler sentence structures altogether.

To address these challenges, chapter 5 experiment a variety of techniques and algorithms that can automatically identify complex sentences and apply appropriate simplification strategies. These techniques involve machine learning models, such as sequence-to-sequence models, that are trained on large datasets of complex and simplified sentences to learn how to generate simplified versions of complex sentences.

Chapter Four: Arabic Sentence Readability

Section A: Datasets

At this stage, the research aims to provide a classification detection of any Arabic text based mainly on lexical complexity and the total measurement of subordinate phrases in the text.

The literature presented in ***Chapter Two*** leads us to the questions regarding Automatic TR models and suggested work to fill the Arabic Automatic TR research gap. These questions are:

1. What is the notion of text complexity?
2. Which grading levels are appropriate and linked to represent the text readability level?
3. Are the available corpora valid for the Automatic TR task? If not, how can they be improved?

Therefore, this chapter aims to answer these questions in a series of readability classification experiments. This involves the construction of an open and accessible corpus, which can serve as a gold standard to test new readability assessment models for different application scenarios. Furthermore, developing a new approach that targets various domains/target groups.

Before measuring the text readability, there is a need to standardise the classification measurement. For that matter, the research follows the classification of the Common European Framework of Reference for Languages, often referred to as CEFR or CEFRL[33] (see section *2.1.2. CEFR Levels*). The choice of relying on CEFR was initiated from KELLY's List classification, which was the main vocabulary list adopted for ATS task. These levels were treated as the measuring scale according to which a text can be classified. These levels will be used in the classification process, as each text will be graded with one of those levels as part of the proposed readability measurement. Indication of the specific language proficiency features attached to each of those levels guided is given informal schema of second language reading provided by Hudson (2007).

[33] https://www.fluentin3months.com/cefr-levels/

In order to measure text complexity, there is a need to discover the linguistic phenomena that define the complexity of the text. Here, it is necessary to answer three questions:

1. What is lexical complexity?
2. What is syntactic complexity?
3. What are the features maximising sentence readability assessment?

In order to answer all questions, a series of experiments were performed to select the appropriate to use and the best features to achieve a robust measurement of Arabic text readability. These experiments will be presented in this chapter as follows:

1. First, to test the most common readability formula in Arabic text.
2. Compare existing Arabic word frequency lists and compile a New Frequency List.
3. Finally, check the Arabic corpora's availability and rationale for building an Arabic sentence readability classifier.
4. Explore the basis for an Arabic sentence readability classification system.

4.1. Experiment one: Applying traditional readability formulae

While in Chapter 2: I have provided an overview of the previous works in measuring readability in Arabic and English. This experiment examined the Flesch–Kincaid readability, the most usable readability formula for the English language, along with the SMOG formula and the Dale-Chall formula. The formulae were applied as illustrated before in section *2.2 Measuring Text readability*.

4.1.1. Experiment procedure

The experiment procedure involved applying the Flesch Reading Ease test's parameters to Arabic text and assessing to what extent it would apply to the Arabic language. One of the main challenges in this process was the absence of diacritic marks in Arabic text, which indicate the vowels in the syllabification process and are necessary to measure the average number of syllables per word. To test the effectiveness of the Flesch Reading Ease test in Arabic, two corpora were used, representing both ends of the reading scale. The first corpus was the Arabic Learner

Corpus (ALC), which consists of Arabic written text produced by learners of Arabic in Saudi Arabia (Alfaifi and Atwell, 2013). The ALC corpus was considered to represent the easy-to-read end of the scale. The second corpus used was the Arabic Internet Corpus (***I-AR***), which represents college-level text and was considered to be at the difficult end of the reading scale. By applying the Flesch Reading Ease test to these two corpora, the experiment aimed to assess the test's applicability to Arabic text and its effectiveness in measuring the readability of Arabic text across different levels of complexity.

4.1.2. Experiment architecture

The system architecture for the experiment comprises four main steps, as shown in Figure 4.1: Arabic text diacritisation, text normalisation, text syllabification, and Flesch Reading Ease calculation. Since Arabic text lacks diacritic marks, the first step involves diacritising the text to indicate the vowels in the text. There are various Arabic Automatic Diacritisation systems available, including MADAMIRA (Pasha et al., 2014), Farasa, and Alserag[34].

However, for this experiment, the MADAMIRA diacritisation module was used for three reasons. Firstly, it is easy to use as it processes text in a .txt file format. Secondly, it is a statistics-based tool that produces diacritised text quickly. Finally, the primary requirement was to obtain diacritised text, regardless of the accuracy of the diacritisation, to use as a base for the syllabification analysis.

The next steps in the system architecture involve, syllabifying the text, then noramlise it again ,and finally calculating the Flesch Reading Ease score. By following these steps, the experiment aimed to assess the applicability of the Flesch Reading Ease test to Arabic text and evaluate its effectiveness in measuring the readability of Arabic text.

[34] User-friendly interface: https://alserag.bibalex.org/

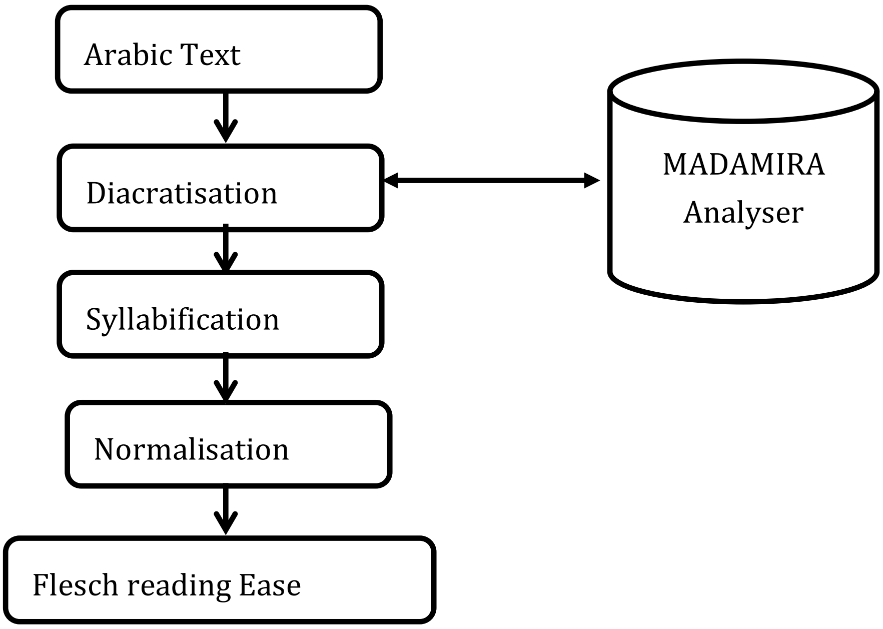


Figure 4.1 Experiment one, Classification system based on Flesch reading Ease formula

After diacritisation the text undergoes syllabification, which is based on the prominent syllables in Arabic, including CV, CVC, CVV, CVVC, and CVCC. The next step involves several pre-processing steps to make it machine-readable. These steps include normalising the spacing between words, removing punctuation marks, and transliterating the text using the Buckwalter system. Finally, the system performs the necessary calculations for the Flesch Reading Ease score. Following the sentence example presented in Table 4.1 "دَخَلَ رَجُلٌ يَحْمِلُ عَدَداً كَبِيراً مِنْ الْكُتُبِ إِلَى الْغُرْفَةِ" considering it as part of a larger Arabic text. This sentence consists of nine words and is broken down into eighteen syllables through syllabification. Following syllabification, the sentence undergoes normalization. This process strips away diacritics, case endings, and other non-essential elements, leaving only the core components of each word.

The final step involves calculating the sentence's readability score using the Flesch Reading Ease formula. Although this formula is primarily designed for English, it has been adapted here for use with Arabic. This calculation considers the total number of words and syllables in the sentence, giving us an estimated readability score. This score can provide insights into the sentence's complexity and readability level.

Table 4.1: Following an example Flesch calculation

Arabic	**دَخَلَ رَجُلٌ يَحْمِلُ عَدَداً كَبِيراً مِنْ الْكُتُبِ إِلَى الْغُرْفَةِ**
Translitration	daḳala rajulun yaḥmilu ʿadadun kabīrun min alkutubi ʾilā alġurfati
Translation	A man carrying a large number of books entered the room
Syllabification	دَخَ-لَ: CV-C رَجُ-لٌ: CV-CV يَحْ-مِ-لُ: CVC-CV-CV عَدَ-داً: CV-CV كَبِي-راً: CV-CV-CV مِنْ: CV الْكُ-تُبِ: CVC-CV-CV إِلَى: CV-CV الْغُرْ-فَةِ: CVC-CVC
Normalization	دخل رجل يحمل عددا كبيرا من الكتب الى الغرفة
Flesch Reading Ease calculation	Reading Ease score = 206.835 - (1.015 * (total words/total sentences)) - (84.6 * (total syllables/total words)) = 206.835 - (1.015 * (8/1)) - (84.6 * (18/8)) = 206.835 - 8.12 - 190.35 = 8.365

Following these procedures on each file , the output of the system is presented in an Excel sheet, with the text's name, Flesch score, reading level, and scores for three other reading tests (the SMOG formula, the Dale-Chall formula, and the Gunning fog) listed in Table 4.2. The experiment aimed to assess the applicability of the Flesch Reading Ease test to Arabic text and evaluate its effectiveness in measuring the readability of Arabic text by comparing the results with the scores of other readability tests.

Table 4.2 Part of the results file specifies the score for each Arabic text file

File	Grade	Dale_Chall	Smog Index	Gunning fog	Flesch
File_1	5th grade very easy to read	8.82	17.4	25.40	101.15
File_2	8th and 9th grade plain text	10.68	21.2	31.08	63.7
File_3	5th grade very easy to read	7.64	12.8	17.41	119.63
File_4	5th grade very easy to read	8.61	14.2	20.26	109.72
File_5	5th grade very easy to read	7	11.4	15.06	130.84

4.1.3. Results and conclusion

The assigned scores to the texts do not match the actual difficulty of the text, which means that this formula and methodology have been proven to fail in measuring the complexity of an Arabic text. Since Arabic relies on diacratisation and is a highly morphologically complex language, it is better to consider the number of word forms in the text rather than the token/type ratio.

Across previous studies have proven that using traditional methods such as statistical formulae (Flesch–Kincaid Grade, the SMOG formula, and the Dale-Chall formula) to measure the text complexity has been proven to be insufficient on its own in measuring the accurate readability level. This is because they ignore many factors affecting text readability beyond the frequency and average sentence length. For example, readability formulas cannot measure the text's context, prior knowledge, interest level, difficulty of concept, or coherence.

In summary, while traditional methods like the Flesch Reading Ease test may provide some insights into the complexity of a text, they are not sufficient on their own. It is important to consider additional factors such as word forms, context, target audience, and reader background when evaluating the readability of a text. Machine learning techniques and multiple evaluation metrics can would yield an adequate readability measure.

4.2. Experiment two: Arabic frequency lists

Vocabulary lists are essential resources in various language studies, ranging from language learning to all applied linguistics disciplines. Corpus-derived frequency lists are the typical way of producing vocabulary lists, in which "words are arranged according to the number of times they occur in particular samples of language" (Richards, 1974). Sharoff et al. (2014) have pointed out a significant challenge in using frequency lists for pedagogic purposes, which arises from the variation in corpus sources from which the lists are derived. If a corpus contains specialist texts, some technical words are listed at higher positions in the lists at the expense of everyday basic words. They argued that having a pedagogical reference for second language learning can isolate such unique words.

4.2.1 Experiment procedure

This Experiment involves selecting the best Arabic frequency list to represent the simplicity or the difficulty of the words in the Arabic text. It requires compiling a new Arabic vocabulary list from available Arabic word lists and classifying the newly developed list with the Common European Framework of Reference for Languages proficiency (CEFR).

Initially, I compiled a word list from ‘Al-Kitaab’(Brustad et al., 2013). Then I analysed the two available Arabic word lists derived from corpora: the Buckwalter and Parkinson Arabic frequency list (Buckwalter and Parkinson, 2014) and the Arabic Kelly's list (Kilgarriff et al., 2014). These lists are described in section ***2.3 (Wordlists***).

Originally ‘Al-Kitaab’ (Brustad et al., 2013), the second edition in three parts, is the most usable textbook for teaching Arabic as a second language that is considered a pedagogical reference. The Al-Kitaab vocabulary list is not a frequency list yet is a list presented in a teaching Arabic textbook based on the language proficiency levels of the students. It is compiled by manually extracting the word lists presented at the beginning of each chapter. It resulted in a list composed of 4024 words.

As well as adopting the KELLY project's Arabic vocabulary list to perform this experiment. Although KELLY’s entries are classified as lemmas, the list’s analysis has proven that it consists of words or even multi-word expression words in some entries. The entry representation does not consider the linguistic analysis for the Arabic lemma. Some of these entries are listed in Table 4.3.

Table 4.3 Some entries in Kelly’s list

The Entries	**Arabic**	**Transliteration**	**English**
Many entries start with a regular prefix as the definite article in Arabic	‘ال’	[al]	‘the’
Entries appear with two different entries; however, they belong to the same lemma.	اخْتَارَ	[akṭār]	‘choose’
	اخْتَارَهُ	[akṭārahu]	‘he chooses’
	رَأْسِمَالِيّ	[raʾsimāliyy]	‘capitalist’
	رأسمالية	[raʾsimāliyya]	‘capitalism’
There are the singular and plural forms of the same lemma	انْتِخَابُ	[antikābu]	‘election’
	انْتِخَابَاتُ	[antikābātu]	‘elections’
Some multi-word expressions appear as entries	مَرَضُ الْإِيدْزْ	[maradu al’iydz]	‘AIDS’
	يَوْمُ الْأَحَدِ	[yawmu al’aḥadi]	‘Sunday’
	رَحْمَةُ اللَّهِ عَلَيْهِ	[raḥmatu Allahi ‘alayhi]	‘God bless him’

Some entries of non-MSA words appear in the list; for example, some belonged to the colloquial Egyptian dialect	عَاوْزْ	[ʿāwz]	‘I want’
	عَايْزْ	[ʿāyz]	‘I want’
	عَاوْزِين	[ʿāwzīn]	‘We want’

Knowles and Don (2004), emphasized the significance of Arabic lemmatisation in the analysis of Arabic text, which is unlike English and can be used as a methodology for constructing dictionaries. So, there was a need to analyse the words in the frequency list to lemmas before matching the analysed text and adopting MADAMIRA. After lemmatising the KELLY’s 9000 entries, it is reduced to 7765 unique entries.

4.2.2 Experiment architecture

The experiment architecture for comparing the two existing vocabulary lists involved several steps. First, a thorough analysis of both lists was conducted to identify their strengths and weaknesses. Second, the "Al-Kitaab" list was included in the analysis to ensure that the selected list would complement the language taught in the textbook. Third, a comparison was made between the lists' entries, taking into account their frequency, relevance, and accuracy. Fourth, the lists were modified and cleaned according to the lemma definition used in the study. Finally, the selected list was integrated into the main system for further analysis and evaluation.

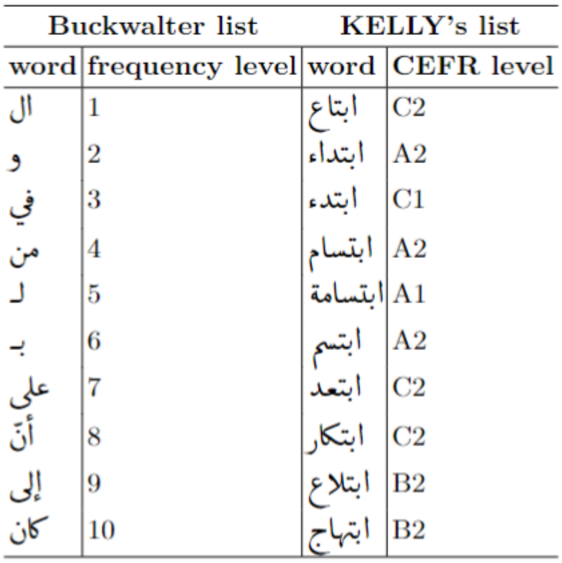
4.2.2.1 Data normalisation stage

In the Arabic script, diacritical marks play a significant role in the pronunciation and meaning of words. Among these, Shaddah and nunation marks are noteworthy.

Shaddah (also called the gemination mark) is a diacritical mark shaped like a small written " ـّ". It is placed above a letter to indicate that the consonant is doubled or geminated. The doubling implies that the consonant is pronounced for a longer duration. For example, the word "مدرسة" (madrasa, meaning "school") can have a shaddah added to the "d", becoming "مدّرسة" (maddirasa), which changes the meaning to "female teacher."

Nunation, on the other hand, is the addition of one of three vowel diacritics (tanwin) that represent a short-vowel sound (a, i, u) followed by an "n" sound. For example, the word "كتاب" (kitaab, meaning "a book") can take nunation to become “كتابٌ" (kitaabun), indicating an indefinite noun in the nominative case.

This noralisation stage is needed to remove any noise affecting the comparison results. The main difference between the two lists was that KELLY’s list (Kilgarriff et al., 2014) was not as fully vowelised as Buckwalter’s. As a result, it was removing all diacritisation marks from both lists except the Shaddah "ـّ” the gemination mark and also the nunation marks “ـًـٍـٌ" as these three appeared in both lists and are considered as a part of the word. The use of POS arranged Buckwalter’s list and minimised it to word and its correspondence frequency (see Figure 4.3). At the same time, it was selecting only words and attaching CEFR columns from the KELLY's frequency list (see Figure 4.3).



Buckwalter list		KELLY’s list	
word	frequency level	word	CEFR level
ال	1	ابتاع	C2
و	2	ابتداء	A2
في	3	ابتدء	C1
من	4	ابتسام	A2
لـ	5	ابتسامة	A1
بـ	6	ابتسم	A2
على	7	ابتعد	C2
أنّ	8	ابتكار	C2
إلى	9	ابتلاع	B2
كان	10	ابتهاج	B2

Figure 4.2 The first 10 words of Buckwalter and KELLY’s word lists after data normalisation

4.2.2.2 Data exploration

In this section, I will make comparisons between the two discussed Arabic frequency lists to find the common words and see how they could be matched and combined. In addition, exploring how to classify non-annotated words with the right CEFR level.

The KELLY’s list (Kilgarriff et al., 2014) classified words into six CERF levels (A1, A2, B1, B2, C1, C2) in KELLY's list. Figure 4.3 illustrates how words are distributed among the six levels. The figure reveals that there has been an inequivalent distribution of the words. The height of the column shows the number of words that appears on each level. For example, the C2 level represents the highest peak of the graph containing approximately 2000 words, while the number of words in A1 reached a low point of the graph to have only 750 words. Between levels A2– C1, the words are almost equally classified, ranging from approximately 1200 to 1500.

The association of the first 1000 words in the Buckwalter list with the CEFR levels is shown in Figure 4.4.The first 1000 of Buckwalter’s entries do not all appear in the A1 and A2 levels. Moreover, there are 285 words that appear in the C level. Table 4.4 shows that while most of these words cannot be classified as C levels according to the Arabic teaching syllabus, they are still essential in any speech or writing performance, particularly pronouns and prepositions. On the other hand, some other words which appear in the top 1000 in Buckwalter's list are classified at the superior level in the learning scale, such as 'صَهْيُونيّ’ ‘Zionist’.

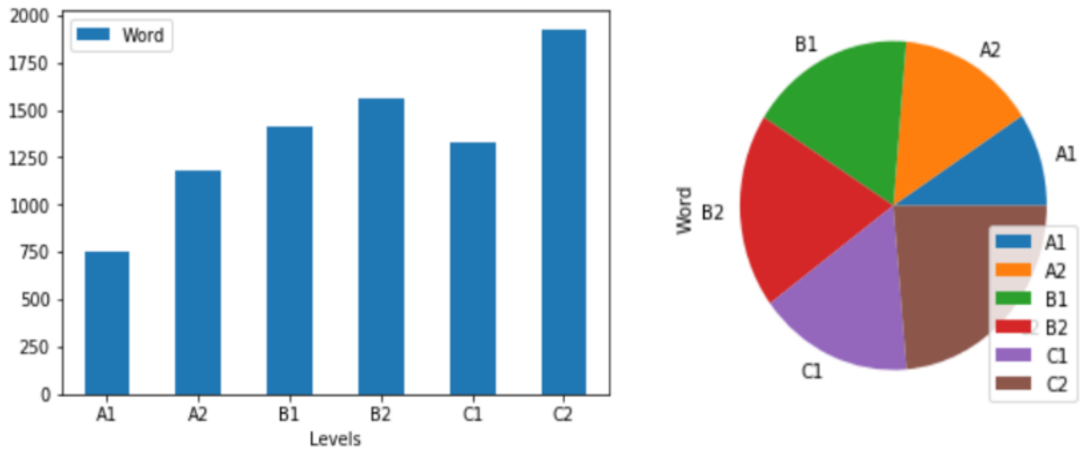
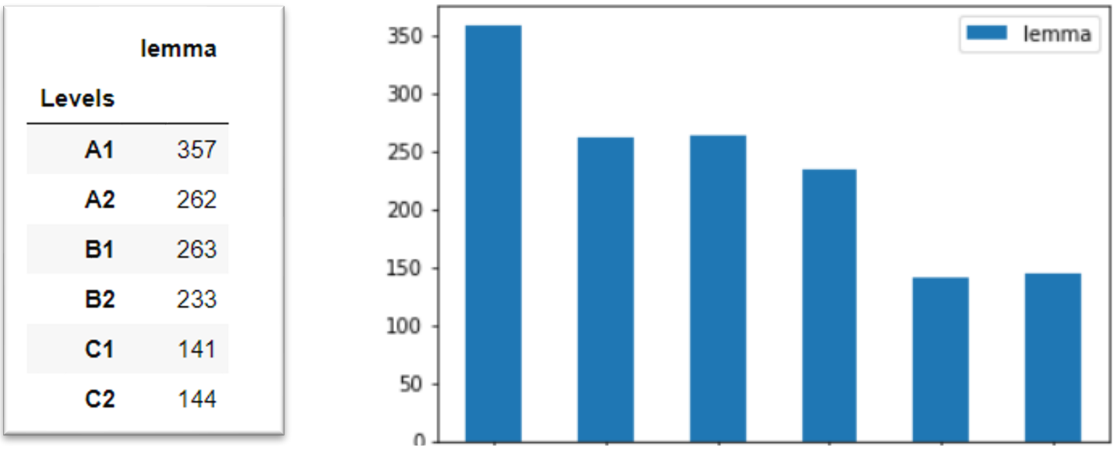


Figure 4.3 KELLY's word list distribution



	lemma
Levels	
A1	357
A2	262
B1	263
B2	233
C1	141
C2	144

Figure 4.4 The association of the first 1000 words in Buckwalter’s list with the CEFR

Table 4.4 Part of the top 1000 words classified in C-level proficiency

word	Transliteration	Translation	CEFR_level
مِن	min	from	C1
عَلَى	ʿalā	above	C1
كان-ُ	kān-u	He was	C2
الله	allāh	God	C2
قال-ُ	qāl-u	He said	C1
هٰذا	hāḏā	this	C1
مع	maʿ	with	C1

4.2.3 Newly vocabulary list[35]

Manually creating a list of Arabic lemmas categorized by CEFR is a time-consuming and labor-intensive task. Therefore, compiling automatic readability ranked list was reasonable at this stage. This involves creating a dictionary of common words across the available lists to develop a common dataset with better word coverage. This is done by merging the three lists to compile an Arabic list containing only MSA variety. Hence, only MSA words from Buckwalter and Kelly's list were considered. A comparison between these two lists shows inconsistency between the lemmas entries, making it difficult to align them. Though, to validate the results, the words presented in all lists were analysed by MADAMIRA. Merging the lists and aligning them with the MADAMIRA lemmatiser led to the new wide-coverage Arabic frequency list, which can be used to predict difficulty as the entropy of the probability distribution of each label in a sentence.

The result of previous analysis of each list urges us to rely on Kelly's list classification for the words that do not exist in the 'Al-Kitaab' list. For the compilation of the final list, the following steps were taken:

1- Removed all dialectical words (70, 95 words from Buckwalter's and KELLY's lists, respectively)
2- Removed duplicated entries to get unique values (777, 1708 from Buckwalter's list and KELLY's list, respectively)

[35] The full list is available at https://github.com/Nouran-Khallaf/Arabic_CEFR_Classified-List

3- Started compiling the list according to the Arabic linguistics classification ‘Al-Kitaab’ book chapters by adding 4024 words that added 1947 new lemmas to the final list.
4- Then averaged 3669 intersected lemmas in both KELLY’s and Buckwalter’s lists.
5- After that, manually classify 525 and 616 lemmas from KELLY’s and Buckwalter’s lists, respectively.

This resulted in a new classified Arabic Vocabulary list consisting of 8834 distinct lemmas, as illustrated in Figure 4.5. The list of Arabic lemmas was carefully compiled and classified according to the six CEFR levels, ensuring that each *lemma* was assigned the appropriate level from A1 to C2 (the lemma is the best representation for Arabic word forms (Knowles and Don, 2004)). This list is the first step in building such an Arabic language profile based on CEFR Level classification. It is an essential resource for sentence readability measurement, and it proved to be effective as one of the sentence features that can rely on measuring sentence readability. This enables selecting sentences from large corpora to represent each language proficiency level. The current list shows some consistency with the English profile list[36] regarding the percentage of words allocated to each CEFR level, as shown in

Figure 4.6.

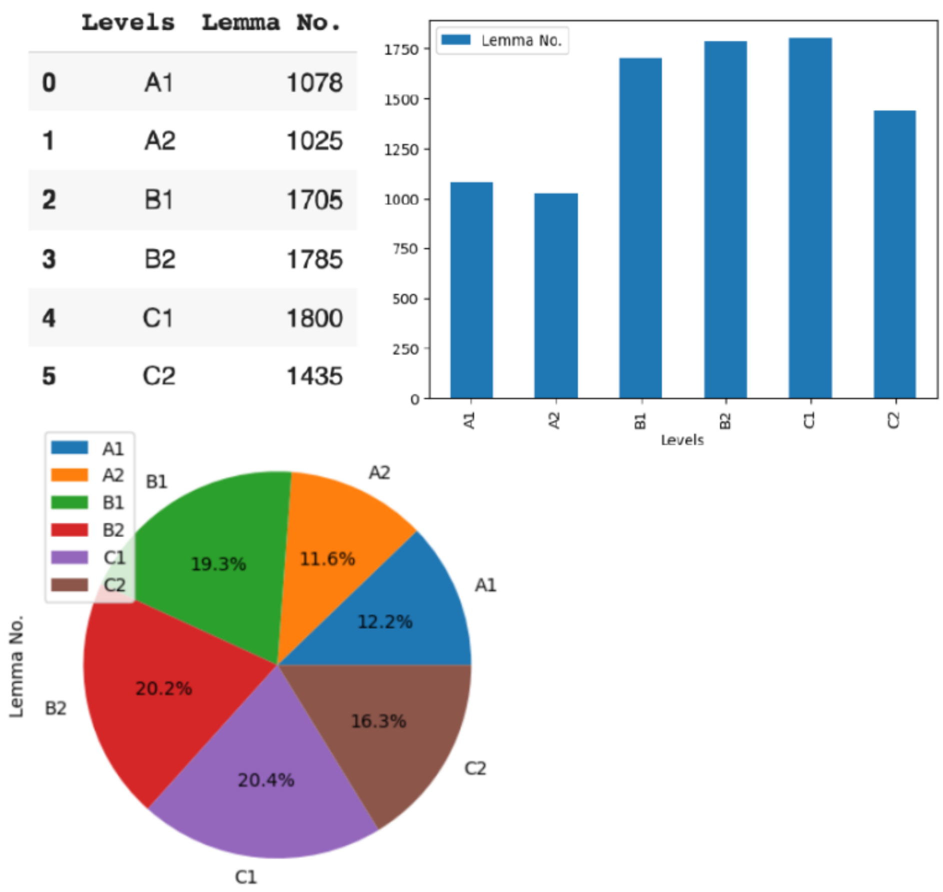
Some insights from the list:

- The word مُجَرِّد meaning 'abstract' is a C2-level word according to CEFR. It is listed at position 3073 in the CEFR-classified list with a frequency count of 76. Although it may not be frequently used in spoken Arabic, it is still a crucial word for advanced learners who need to comprehend and express abstract concepts through written communication.
- The word مُتَعَدِّد which translates to 'multifaceted' is a B2-level word according to CEFR. It appears at position 3537 with a frequency count of 55 in the CEFR-classified list. Although it may not be commonly used in everyday Arabic

[36] https://languageresearch.cambridge.org/wordlists

communication, it is still an essential word for learners at the B2 level, particularly those who need to describe intricate ideas.

- The word مُشْتَقّ meaning 'derived' is classified as a C1-level word in CEFR. It is listed at position 3207 in the CEFR-classified list with a frequency count of 68. While it may not be commonly used in everyday conversation, it is an important term for learners who need to read and understand technical or scientific papers.
- The word تَصْنِيف which means 'classification' is a C1-level word in CEFR. This indicates that learners at this level should be able to comprehend and employ this term in context. It is listed at position 5228 with a frequency count of 60 in the CEFR-classified list. Although it may not be one of the most frequently used terms in Arabic, it is still essential for learners at the C1 level to comprehend its meaning and usage.
- The word سَامِع meaning 'listener' is classified as a B2-level word in CEFR. It appears at position 347 with a frequency count of 1340 in the CEFR-classified list. This suggests that the word is commonly used in Arabic communication and is therefore a crucial word for learners at the B2 level to be familiar with.



	Levels	Lemma No.
0	A1	1078
1	A2	1025
2	B1	1705
3	B2	1785
4	C1	1800
5	C2	1435

Figure 4.5 The word frequency distribution across the CEFR levels for the New modified List

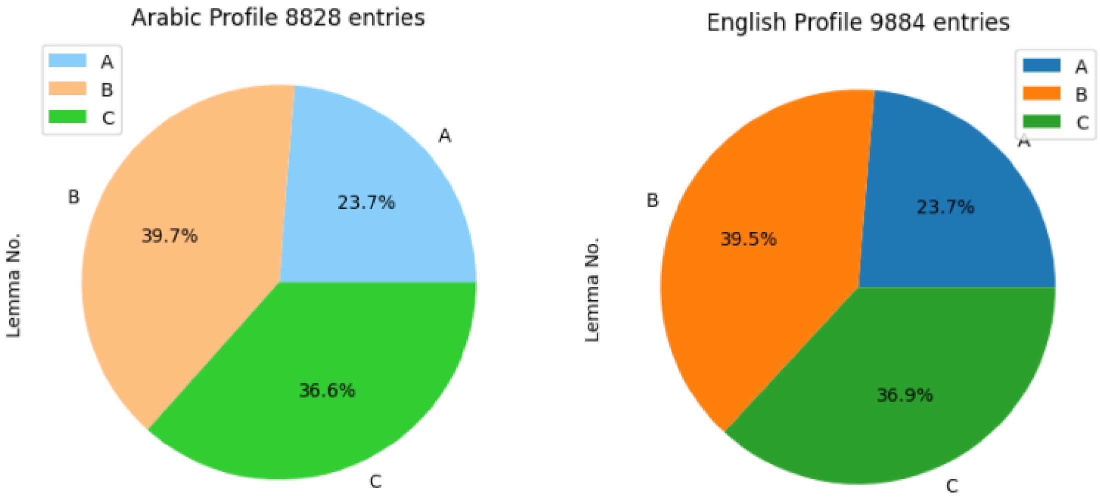


Figure 4.6 The distribution of the words across the CEFR levels

4.3. Corpora

This section details the creation and compilation of two corpora I developed for this thesis: the Arabic Sentence Complexity Level Annotated Corpus and the Parallel Complex-Simple Arabic Sentence Corpus.

I curated the Arabic Sentence Complexity Level Annotated Corpus, consisting of 16,045 Arabic sentences, each individually annotated for complexity level. This corpus, vital for training and evaluating various models, enables the prediction of Arabic sentence complexity. For each sentence, I've assigned a complexity score ranging from 1 (least complex) to 5 (most complex)

The Parallel Complex-Simple Arabic Sentence Corpus, also created by me, comprises 2,980 parallel Arabic sentences. Each complex sentence in this corpus is paired with a simpler version of the same sentence. This corpus serves to train and evaluate different models for sentence simplification, with its parallel structure facilitating direct comparison among various simplification models.

The two corpora I developed play a pivotal role in the progression and assessment of natural language processing models. These models aim to enhance the accessibility and comprehension of Arabic text, particularly for non-native speakers and individuals with reading difficulties.

4.3.1 Dataset One: Sentence-level annotation[37]

Arabic Sentence Complexity Level Annotated Corpus was used for Arabic sentence difficulty classification. The aim was to build a new dataset by compiling a corpus from three available sources classified for readability on the document level and a sizeable Arabic corpus obtained by Web crawling.

The first corpus source is the reading section of the ***Gloss*** Corpus developed by the Defence Language Institute (DLI). It has been treated as a gold standard and used in the most recent studies on document-level predictions (Forsyth, 2014; Saddiki et al., 2015; Nassiri et al., 2018b; Nassiri et al., 2018a). Texts in Gloss have been annotated on a six-level scale of the Inter-Agency Language Roundtable (IL). The CEFR levels

[37] This corpus is publicity available at https://github.com/Nouran-Khallaf/Arabic-Readability-Corpus/tree/main

have been matched to the Gloss list based on the schema proposed by Tschirner et al. (2015). Gloss corpus is organized into ten different genres (culture, economy, politics, environment, geography, military, politics, science, security, society, and technology) and four competence areas (lexical, structural, socio-cultural, and discursive). The second corpus source is the ***ALC***, which consists of Arabic written text produced by learners of Arabic in Saudi Arabia (Alfaifi and Atwell, 2013). Each text file is annotated with the proficiency level of the student. They were mapping these student proficiency levels to CEFR levels. The third corpus source comes from the textbook Al-Kitaab (Brustad et al., 2015), which was compiled from texts and sentences from parts one and two of the third edition but only texts from the third part second edition. This book is widely used to teach Arabic as a second language. These texts were initially classified based on ACTFL guidelines, which mapped to CEFR levels. As these corpora have been annotated on the document level and not on the sentence level, the rule was assigning each sentence to the document via re-annotation through machine learning, see the dataset cleaning procedure below. A counterpart corpus of texts not produced for language learners in mind is provided by I-AR, 75,630 Arabic web pages collected by wide crawling (Sharoff, 2006). A ***random snapshot*** of 8627 sentences longer than 15 words was used to extend the limitations of C-level sentences coming from corpora for language learners. Which added to the tortal number of sentences to be 24,672 sentenecs. Table 4.5 shows the distribution of the number of used sentences and tokens per each Common European Framework of language proficiency Reference [CEFR] Level.

Table 4.5 The distribution of the number of used (S) sentences and (T) tokens per each CEFR Level.

CEFR Level		**Number of S**			**Total S**	**Total T**
		Gloss	ALC	Al-Kitaab		
A1	A1.1	874	1877	161	4479	142682
	A1.2	460	963	144		
A2		2231	1829	106	4166	111340
B1	B1.1	2210	1690	317	5672	91649
	B1.2	1310	145	0		
B2		747	98	381	1226	55031
C		0	145	357	502	26156
Total		7832	6747	1466	16045	426858

4.3.2 Dataset Two: complex- simple parallel corpus

Dataset Two, which I meticulously compiled, is a collection of parallel simple/complex sentences derived from the internationally renowned Arabic novel "Saqq Al-Bambuu" (Al-Sanousi, 2013). This novel has an official simplified version intended for Arabic language learners, produced by (Familiar and Assaf, 2016). The dataset's primary objective is to test a classifier's capability to identify sentences in the original text that necessitate simplification.

In this dataset, which I have put together, there are 2,980 parallel sentences; each intricate sentence is paired with a simpler rendition of the same sentence. The sentences are divided into two groups: Simple A+B and Complex C, following the Common European Framework of Reference (CEFR) for language proficiency. Simple A+B pertains to levels A1 and A2, whereas Complex C corresponds to levels B1, B2, and C1.

Table 4.6 gives a concise summary of the number of sentences and tokens available for each CEFR level in Dataset Two, which I constructed. A snapshot of the Dataset is provided in Appendix A, which showcases a portion of Dataset Two: the complex-simple parallel corpus snapshot.

Table 4.6 Number of Sentences and Tokens available per each CEFR Level in Dataset two

Levels	Sentence	Token
Simple A+B	2980	34447
Complex C	2980	46521
Total	5690	80968

The corpus used in this study was compiled through a multi-step process. First, both the original complex novel and the simplified version were manually scanned to identify parallel sentences. Second, an online Arabic Optical Character Recognition (OCR) tool[38] was used to digitize the text. Third, only the sentences that occurred in the simple version were manually aligned with their corresponding sentences in the original complex novel. This is because not all sentences in the original novel have an

[38] https://www.i2ocr.com/

equivalent simple version, as one of the simplification processes is deletion. Finally, the words in the parallel "Saqq Al-Bambuu" corpus were aligned using the Eflomal[39] word aligning tool for each sentence pair.

and English	**Arabic**	**languages**	**both**		**in addition to**		**Filipino**	**He is fluent in**	Translation
والإنكليزية	العربية	اللغتين	من	كلا	إلى	بالإضافة	الفلبينية	يُجيد	Complex
والإنكليزية	العربية	اللغتين	-	-	-	جانب	الفلبينية	يجيد	Simple
K	K	K	D	D	D	R	K	R	Label

To label these alignments, four different operations were used, inspired by the labeling algorithm described in (Alva-Manchego et al., 2017). The NLTK alignment tool was used to identify simplification types at both the word-level and sentence-level. Table 4.7 represents an example of a parallel complex/simplified sentence pair while aligning and labelling the word changes across the sentences as following:

1. Deletions, DELETE (D) in the complex sentence. [word-level]
2. Additions, ADD (A) in the simplified sentence. [sentence-level]
3. Substitutions, REPLACE (R), a word in the complex sentence, is replaced by a new word in the simplified sentence. [word-level]
4. Keep, and KEEP (K) words shared in both complex and simple sentence pairs. [sentence-level]

Table 4.7 Represents the simplification operations labelling using Eflomal alignment along with NLTK alignment.

Manual verification of a random sample of 50 parallel sentences suggested performing corpus cleaning includes removing vowelisation and punctuation to eliminate miss-classification for the operations such in Table 4.7 example, the first word ‘يجيد ’/’يُجيد’[He is fluent in] was recognised as a substitution; however, the only difference was in removing the diacritic mark unless we consider the removing of vowelisation or adding it is a simplification process. Also, in some other examples, the Eflomal aligner itself missed the correct word alignments. However, despite these

[39] https://github.com/robertostling/eflomal

alignment errors, most of the revised sentences were correctly aligned. Also, in some other examples, the Eflomal aligner itself missed the right word alignments. For example, in those 100 sentences, there were 402 replacement and addition processes in which only 83-word pairs were wrongly aligned.

The overall simplification processes in the "Saqq Al-Bambuu" corpus are shown in Figure 4.7. The most frequent operation is "Keep," where 21,899 words were copied in the simplified version. This is followed by "Deletion," with 12,561 words deleted to simplify the sentence. The third most common operation is "Replacement," where 9,082 words were substituted with their simple counterparts. Only 362 words were added to simple sentences, recognized as an "Addition" process. Overall, the simplification process in the "Saqq Al-Bambuu" corpus involved a combination of these four operations to make the language more accessible to non-native speakers.

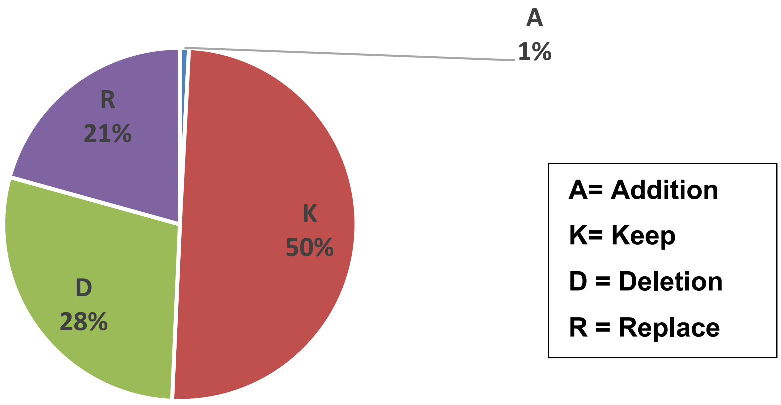


Figure 4.7 Percentage of each simplification process on Saqq al bambuu corpus

Section B: Arabic Sentence Difficulty Classifier

This section focusses on experiments aimed at measuring to what extent a sentence is understandable by a reader, such as a learner of Arabic as a foreign language, and at exploring different methods for readability assessment. Research to date has tended to focus on assigning readability levels to whole text rather than to individual sentences, even though any text is composed of several sentences, which vary in difficulty (Schumacher et al., 2016). Assigning readability levels for a text is a challenging task, and it is even more challenging on the sentence level as much less information is available. Also, the sentence difficulty is influenced by many parameters, such as genre or topics and grammatical structures, which need to be combined in a single classifier. Therefore, difficulty assessment at the sentence level is a more challenging task in comparison to the better-researched text-level task. However, the availability of a readability sentence classifier for Arabic is vital since this is a prerequisite for research on ATS (Saggion, 2017).

The main aim of this section lies in developing and testing different sentence representation methodologies, which range from using linguistic knowledge via feature-based machine learning to modern neural methods. In summary, the contributions of this section are:

1. Using the compiled dataset for training on the sentence level presented in (Chapter4- Section A).
2. Developing a range of linguistic features, including POS, syntax, and frequency information.
3. Evaluating a range of different sentence embedding approaches, such as fastText, BERT, and XLM-R, and comparing them to the linguistic features.
4. Casting the readability assessment as a regression problem as well as a classification problem.
5. This model is the first sentence difficulty system available for Arabic.

4.4. Features and extraction methods

Assigning the following groups of features in Table 4.8: Part of speech tagging features (POS features); Syntactic structure features (Syntactic features); CEFR-level lexical features; Sentence embeddings.

Table 4.8 The Feature set. (All measures are for the rate of tokens on the sentence levels)

	POS Features		
1	TTR of word forms	**12**	Numeric Adj Tokens
2	Morphemes word Tokens	**13**	Comparative Adj
3	TTR of Lemma	**14**	Conjunction Tokens
4	Nouns Tokens	**15**	Conjunction Subordination Tokens
5	Verbs Tokens	**16**	Proper noun Tokens
6	Adj Tokens	**17**	Pronoun Tokens
7	Verb pseudo-Tokens	**18**	Punc Tokens
8	Passive verbs Tokens Tokens	**19**	Simple Connector
9	Perfective verbs Tokens	**20**	Complex Connector Tokens
10	Imperfective verbs Tokens	**21**	All Sent Connector Tokens
11	3rdperson verb Verbs		
	Syntactic Features		
22	Incidence of subjects	**25**	Incidence of coordination
23	Incidence of objects	**26**	Average phrases/sentence
24	Incidence of modifier/root	**27**	Average phrases depth
	CEFR Word Features		
28	Incidence of Level A1	**32**	Incidence of Level C1
29	Incidence of Level A2	**33**	Incidence of Level C2
30	Incidence of Level B1	**34**	Word entropy concerning CEFR
31	Incidence of Level B2		
35	**Sentence Embeddings Features**		

4.4.1 Linguistic features

While the sentence-level classification task is novel, borrowing some features from previous studies of text-level readability (Forsyth, 2014; Saddiki et al., 2015; Nassiri et al., 2018c; Nassiri et al., 2018a). Deciding to exclude the sentence length from the feature set creates an artificial skew in understanding what is difficult: more difficult writing styles are often associated with longer sentences, but it is not the sentence length that makes them difficult. Specifically, many long Arabic sentences contain shorter ones, which are connected by conjunctions such as '/wa /'= 'and'. In their book "Arabic Grammar in Context", Mohammad

T. Alhawary and Kristen Brustad note that such compound sentences are a common feature of Arabic and are generally not difficult for learners to understand(Alhawary and Brustad, 2016). Similarly, Mohammad Abu-Rabia argues that the use of conjunctions in Arabic sentences helps to make the language more cohesive and easier to comprehend (Abu-Rabia, 2008). Overall, it seems that learners of Arabic need not be overly concerned about longer sentences with ‘and’ conjunction as they are a natural part of the language's structure.

4.4.2 The POS-features

[Table 4.8 features (1-21)], these features represent the distribution of different word categories in the sentence and the morpho-syntactic features of these words. Knowles and Don (2004), argue that Arabic lemmatisation is crucial for analysing Arabic text and constructing dictionaries, unlike English lemmatisation. Therefore, using the Lemma/Type ratio instead of Word/Type ratio. Adding features represents the different verb types (Verb pseudo, Passive verbs, Perfective verbs, Imperfective verbs, and 3rdperson). As conjunction is one of the key features in representing sentence complexity in Arabic (Forsyth, 2014), adopting the annotated discourse connectors introduced by Alsaif (2012) by splitting this list into 23 simple connectors and 56 complex connectors referring to non-discourse connectors and discourse connectors, respectively. POS features are extracted by using MADAMIRA (Pasha et al., 2014).

4.4.3 Syntactic features

Features (22-27) from Table 4.8 provide some information about the sentence structures and the number of phrases as well as phrase types. These features are derived from a dependency grammar analysis. Because dependency grammar is based on word-word relations, it assumes that the structure of a sentence consists of lexical items that are attached by binary asymmetrical relations, which are known as dependency relations. These relations will be more representative of this task. For this purpose Camel Parser (Shahrour et al., 2016), explained in chapter 2 section 2.7. Arabic Feature extraction tools.

4.4.4 CEFR-level lexical features

Features (28-34) from Table 4.8 are used to assign each word in the sentence with an appropriate CEFR level. For this, create a new Arabic word list consisting of 8834 unique lemmas labelled with CEFR levels. This list was a combination of three frequency lists, 1) Buckwalter and Parkinson's 5000 frequency word list based on a 30-million-word corpus of academic/non-academic and written/spoken texts (Buckwalter and Parkinson, 2014) KELLY's list, which is produced from the KELLY project (Kilgarriff et al., 2014), which directly mapped a frequency word list to the CEFR levels using numerous corpora and languages, 3) lists presented at the beginning of each chapter in 'Al-Kitaab' (Brustad et al., 2011; Brustad et al., 2015). Merging the lists and aligning them with the MADAMira lemmatiser led to the development of this new wide-coverage Arabic frequency list, which can be used to predict difficulty as the Entropy of the probability distribution of each label in a sentence.

4.4.5 Sentence embeddings

In addition to the 34 traditional features, the sentence could be represented as embedding vectors using different neural models as follows:

fastText is a straightforward way to create sentence representations is to take a weighted average of word embeddings of each word, for example, using fastText vectors. Using the Arabic ar.300.bin file, each word in word embeddings is represented by the 1D vector mapped of 300 attributes (Grave et al., 2018). The sentence vectors were normalised to have the same length concerning dimensions. The idea behind vector normalization is to adjust the magnitude of each sentence vector so that all vectors have the same length.

For this, tf-idf weights were calculated for each word in the corpus to use them as weights:

$$s = w_1 w_2 \dots . w_n$$

$$\text{Embedding[s]} = \frac{1}{n} \sum_i tfidf[w_i] \times Embedding[w_i]$$

Universal sentence encoder (Yang et al., 2019) was generated to be used on the sentence level, which after sentence tokenisation, it encodes the sentence to a 512-dimensional vector. Considering here the large version[40].

Multilingual BERT(mBERT) (Devlin et al., 2019)***, AraBERT*** (Antoun et al., 2020) and ***Arabic-BERT*** (Safaya et al., 2020), here using the last layer produced by BERT transformers while padding the sentences to the maximum length of 128 tokens.

XLM-R (Conneau et al., 2019) At the same time, using the same setup for classification as in the case of mBERT while also testing a different setup of combining its output with linguistic features and using it as a joined vector of features for traditional ML classification.

4.5. Experiments

CEFR language proficiency levels can be presented as labels or as a continuous scale. The former is solved as a classification task with macro-averaged F-1 as the primary measure for accuracy. The latter is solved as a regression task (Vajjala and Lõo, 2014). The experiments presented in the following section are divided into two main phases. ***Phase one*** was testing CEFR classification with 7-way, 5-way, and 3-way datasets. The experimental results, along with the error analysis, urged to clean the corpus, redo the experiments with new data, and apply new machine learning approaches as presented in ***Phase two***.

4.5.1 Phase one

All ML experiments were done using Python 3.6 toolkits Natural Language Processing Toolkit (NLTK[41]), and Scikit-learn[42]. Using Dataset one , this sentence corpus was split: 80% for training and 20% for testing. Using the data presented in Table 4.3[section 4.4.1 Dataset One: Sentence-level annotation], perform a series of experiments on 3, 5 and 7 data sets. Using all presented features in the previous section but using only fastText to represent a sentence-embedding feature.

[40] https://tfhub.dev/google/universal-sentence-encodermultilingual/1

[41] http://www.nltk.org/

[42] https://scikit-learn.org/stable/index.html

4.5.1.1 Readability as a classification problem

After scaling the data and applying the random splitting for training/testing splitting, the experiments were performed on a 5-data set. Training seven different ML classifiers, as reported in Table 4.9, shows that the Xgboost classifier provides the best results through all models, with Precision=0.44, Recall=0.46, and F-measure=0.43.

Table 4.9 Evaluation results for Classification ML models applied on 5-data set categories.

Classification ML Model	Acc.	Per.	Rec.	F-1
Naive_bayes	0.20	0.41	0.21	0.25
SVM, rbf kernel	0.45	0.42	0.45	0.42
Linear SVM	0.45	0.41	0.46	0.41
Random Forest	0.40	0.42	0.40	0.39
Decision Tree	0. 41	0.33	0.41	0.32
KNeighbors	0.43	0.42	0.43	0.42
XgBoost	**0.45**	**0.44**	**0.46**	**0.43**

The 5-data set results lead to performing two other experiments using the Xgboost classifier on both 3-dataset and 7-dataset classes. According to the results presented in Table 4.10, the XgBoost classifier was applied on both 3-dataset and 7-dataset classes. The evaluation metrics used to assess the performance of the classifier were accuracy (Acc.), precision (Per.), recall (Rec.), and F-1 score (F-1). For the 3-dataset classes, the XgBoost classifier achieved an accuracy of 0.59, precision of 0.60, recall of 0.60, and F-1 score of 0.59. These results suggest that the classifier's overall performance is moderate, with slightly better performance in terms of precision and recall.

On the other hand, for the 7-dataset classes, the XgBoost classifier achieved an accuracy of 0.32, precision of 0.31, recall of 0.33, and F-1 score of 0.31. These results indicate that the classifier's performance is poor, with lower accuracy, precision, recall, and F-1 score values.

Overall, the results suggest that the XgBoost classifier performs better on the 3-dataset classes compared to the 7-dataset classes. However, the performance of the classifier is still relatively low, indicating that further improvements are necessary for accurate and reliable predictions.

Table 4.10 Evaluation results for XgBoost classifier applied on 3-dataset and 7-dataset categories.

XgBoost classifier	Acc.	Per.	Rec.	F-1
3-dataset classes	0.59	0.60	0.60	0.59
7-dataset classes	0.32	0.31	0.33	0.31

4.5.1.2 Readability as a regression problem

Regression allows us to make ranked predictions along with the discrete CEFR levels, thus assessing which text is more complex than the other. Table 4.11 presents the evaluation results for different regression machine learning models applied to 5-data set classes. The performance of each model was assessed using mean absolute error (MAE) and Pearson correlation coefficient.

Derived from the results, the SVM regression model achieved the best performance among all the models with a Pearson correlation coefficient of 0.137 and an MAE of 0.72. This indicates that the model's predicted levels were moderately correlated with the actual level, with an average difference of 0.72 units from the actual levels.

Among the other models, the random forest regression model performed the second-best with an MAE of 0.742 and a Pearson correlation coefficient of 0.090. The other models, including linear regression, decision tree regressor (2T and 5T), and Xgboost regression, performed worse than the SVM regression and random forest regression models in terms of both MAE and Pearson correlation coefficient.

Overall, these results suggest that SVM regression is the most effective method for predicting the complexity level of texts in the 5-data set classes. However, there is still room for improvement in the accuracy of the predictions, as indicated by the MAE values. Further optimization and refinement of the model may be necessary to achieve more accurate predictions.

Table 4.11 Evaluation results for Regression ML models applied to 5-data set categories.

Regression ML model	**MAE**	**Pearson Corr.**
Linear Regression	0.77	0.092
DecisionTreeRegressor,[2T]	0.807	0.06
DecisionTreeRegressor, [5T]	0.765	0.10
Random Forest Regression	0.742	0.090
Xgboost regression	0.78	0.113
SVM	**0.726**	**0.137**

Again, applying the best-performed ML regression technique SVM (support vector machines) on 3 and 7 dataset classes. The results as reported in Table 4.12. For the 3-dataset classes, the SVM regression model achieved a MAE of 0.41 and a Pearson correlation coefficient of 0.20. This means that the model's predictions were off by an average of 0.41 units from the actual values. The Pearson correlation coefficient of 0.20 suggests a weak positive correlation between the predicted values and the actual values. For the 7-dataset classes, the SVM regression model achieved a MAE of 1 and a Pearson correlation coefficient of 0.18. The higher MAE value indicates that the model's predictions were off by a larger amount compared to the 3-dataset classes. The Pearson correlation coefficient of 0.18 suggests a weak positive correlation between the predicted levels and the actual levels.

Table 4.12 List of the most effective features through different feature selection algorithms

SVM Regression model	**MAE**	**Pearson Corr.**
3-dataset classes	0.41	0.20
7-dataset classes	1	0.18

In this first experiment, although applying different ML techniques, the results obtained from all previous experiments show a low f-measure. This indicates inconsistent annotations and wrong instances in the dataset. As a result, to test the reliability of the training corpus and classification, an Error Analysis needs to be undertaken to identify the dominant/significant errors in the corpus. Hence provide guidance to what needs to be improved or modified on the training corpus.

4.5.1.3 Error analysis

Manual error analysis is an expensive task in terms of time and effort. So, the choice was to run a Semi-Automated Error Analysis experiment. The section is inspired by the linguistic annotation error analysis strategy introduced by Di Bari et al. (2014); their sentiment analysis schema for English is based on detecting agreement between classifiers belonging to different Machine Learning paradigms. The cases when most of the classifiers agreed on predicting a label while the gold standard was different were inspected manually by a specialist in teaching Arabic. Using the following classifiers: SVM (with the rbf kernel), Random Forest, KNeighbors, Softmax and XgBoost using linguistic features discussed in *Section 3.4.1*, I trained them via cross-validation and compared their majority vote to the gold standard.

Table 4.13 presents the performance of different classification machine learning models trained on the 5-class data set. The performance of each model was evaluated using precision (Per.), recall (Rec.), and F-1 score (F-1).

In accordance with the results, the SVM classifier with the rbf kernel achieved the best performance among all the models, with a precision of 0.47, recall of 0.46, and F-1 score of 0.45. This indicates that the model was able to identify the correct CEFR level for a sentence with a moderate level of accuracy.

The KNeighbors classifier with N=7 achieved the second-best performance, with a precision of 0.45, recall of 0.46, and F-1 score of 0.45. The XgBoost classifier performed slightly worse than the KNeighbors classifier, with a precision of 0.43, recall of 0.45, and F-1 score of 0.43.

Finally, the random forest classifier achieved the lowest performance among all the models, with a precision of 0.41, recall of 0.41, and F-1 score of 0.40. Overall, the results suggest that the SVM classifier with the rbf kernel is the most effective method for classifying sentence readability levels in the 5-class data set.

These models rely on different strategies, which makes the analysis more reliable. This done by comparing them against the gold standard correct level to investigate to what extent the agreement between the different ML models in sentence readability labelling.

Table 4.13 Performance of the classifiers trained on the 5-class data set.

Classification ML Model	Per.	Rec.	F-1
SVM, rbf kernel	0.47	0.46	0.45
Random Forest	0.41	0.41	0.40
KNeighbors(N=7)	0.45	0.46	0.45
XgBoost classifier	0.43	0.45	0.43

Adopting the error classification tags introduced by Di Bari et al. (2014) with some modifications as follows:

A) ***Wrong [W]***: if the classifiers have wrongly labelled the data and the gold standard is correct.

B) ***Modify [M]***: if the classifiers are correct and the gold standard needs to be modified.

C) ***Ambiguous [A]***: considering both labels are possible based on different perspectives.

D) ***False [F]***: False is an added label that represents the disagreement between the gold standard and the classifiers when neither is correct. This label was added because the sentence classifications themselves were automatically classified based on the hypothesis that each sentence in a text would be considered to have the same overall text level in which it appears.

For each sentence, five different predictions are assigned. Compared to the gold standard CEFR label, the classifiers agreed to predict 10204 instances. Then what needs to be considered is when all classifiers agree on the predicted label, and it contradicts the gold standards. In this case, the classifiers agreed on the 1943 sentence classification. After conducting a thorough investigation of randomly selected sentences, I manually assigned error classification tags. The results indicated that the main classification confusion was in Level B instances, suggesting a need for further analysis and potentially revised guidelines for identifying and categorizing errors.

Out of the 1605 random testing sentences, there was a classification agreement on 302 sentences, which gives us 1303 sentences to analyse. First, considering the agreement between the Four ML models, in about 198 sentences, all ML agreed on their classification against the gold standard. The results as shown in Figure 4.8 label A was assigned 35 times, label F was assigned 14 times, label M was assigned 110 times, and W was assigned 45 times. After the previous analysis, the wrong predicated sentences are now 1105 instances. To get more instances that the ML concur so that drop the KNN approach and get other instances besides 198 in the above experiment. Figure 4.8 also represents the distribution of each error type occurring across the three different corpora. In the F label, only the ALC instances show the false agreement to the manual annotation. This is explained as a reason that the ALC corpus is an Arabic second language student's writing, in addition to it was missing most of the punctuation marks that result in wrongly sentence splitting. Gloss corpus represents most of the M label; in fact, most of these sentences were in the A2 category in the gold standard and correctly classified by ML models to the B1 category. In this case, raising this issue as a wrongly classified Level2 in Gloss corpus to be A2 while it should have added to the B1 category instead. Away from this case, the ALC corpus appeared to be the most problematic instance through the four Error types.

In contrast, the Al-Kitaab corpus seemed to be well categorised, despite the increased number of wrongly classified level in Al-Kitaab corpus as those 20 incidences were only in level C classified as B2. This could be a result of a small representation of level C in the corpus. The analysis results show the distribution of categories where each error type occurred. In the end, 380 instances had to be assigned to the lower level (usually from B to A).

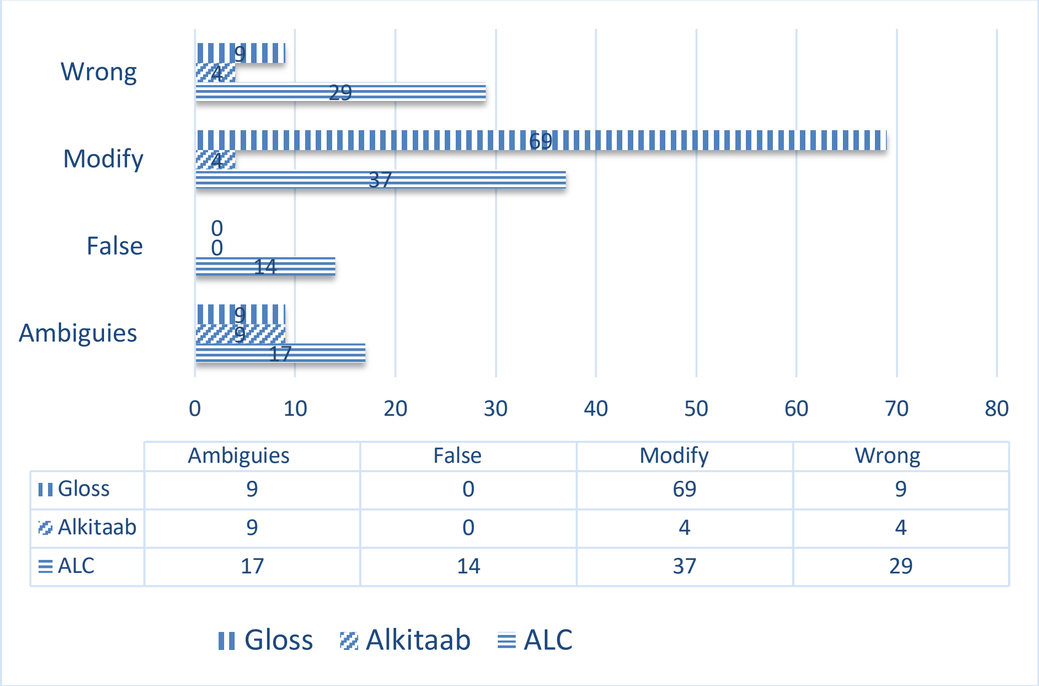


Figure 4.8 Error types of distribution across sentences from different corpora

4.5.1.4 Dataset one: Data-adjustments

Therefore, the decision was made to improve the quality of the training corpus considering the previous results. Table 4.14 shows the new distribution of the number of used sentences and tokens compared to dataset one version one per each [CEFR] Level.

Table 4.14 (S) sentences and (T) tokens available per each CEFR Level in the two versions of the corpus

CEFR	**Old**		**New**	
	S	**T**	**S**	**T**
A	8661	187225	9030	195343
B	5532	126805	5083	117825
C	8627	287275	8627	287275
Total	22820	601305	22740	600443

In the first experiments, the confusion matrix of the classifier trained on dataset one in Table 4.15 represents the confusion matrix of the Xgboost classifier. This also indicates that the classification guidelines need to be more explicit, especially between the A1, A2 and B1 labels.

Table 4.15 Confusion Matrix of XgBoost Classification prediction.

	A1	A2	B1	B2	C
A1	691	128	295	2	5
A2	272	315	354	0	4
B1	337	163	664	12	1
B2	70	13	39	8	5
C	16	11	24	4	6

After reanalysing the Gloss level2, most of the presented text was classified under the categories of politics, technology, and science. These categories are present in B1 and B2 in the actual interpretation of the representation of the CEFR levels in a real-life context. The error analysis results suggest new adjustments to the corpus. These adjustments involve reallocating the Gloss level2 from A2 to B1 category and dropping all ALC corpus instances from the data. In Table 4.16, an example from Gloss Level 2 which represents a political discourse. First, it contains a grammatical structure 'لا يزال'- [still] which is appeared in Chapter 3 in 'Al-Kitaab part two'. Second, it has the word 'مـحْتجزين'- [detained] which is the passive participle 'اسم مفعول' that appeared in Chapter 8 in 'Al-Kitaab', indicating a classification to B1 level.

Table 4.16 An Example from Gloss Level 2

Sentence	**الرهائن الخمسة عشر الذين *لايزالون مـحْتجزين* في الصحراء الجزائرية**
Translation	The 15 hostages who are still detained in the Algerian desert
	arrahāʾinu alḳamsata ʿašara allaḏīna lā yazālūna muḥtajazīna fī aṣṣaḥrāʾi aljazāʾiriyyati

Running an Xgboost classifier trained on the dataset version two with 5-classes and 3-classes, reaching F-measure 0.72, 0.80 respectfully as represented in Table 4.17. The results indicate an improvement of 27% from the initial data and reach 0.72 F-1. These results imply the problem was not in the feature representation but noise in the dataset arising from the ALC corpus instances.

Table 4.17 Results obtained using the XgBoost Classifier on the new dataset (without ALC corpus)

XgBoost Classifier	Acc.	Per.	Rec.	F-1
5- dataset classes	0.76	0.72	0.76	0.72
3- dataset classes	0.82	0.80	0.82	0.80

On the other hand, running the SVM regression on dataset-VII with 3 and 5 datasets. The result in Table 4.18 shows an improvement as in previous classification models. In which the 5-dataset performance is better than applying 3-classes on saqq albambuu dataset without ALC corpus on both 3 and 5 classes.

Table 4.18 Results of SVM Regression for both 3 and 5 classes based on Dataset Two without ALC corpus

SVM Regression model	MAE	Pearson Corr.
5- dataset classes	0.44	0.33
3- dataset classes	0.39	0.32

Upon removing the ALC sentences, I observed a deficit in the presentation of Level A1. Consequently, I decided to merge Levels A1 and A2, thereby streamlining our approach to just three classes: A, B, and C. The confusion matrix in Table 4.19 shows the results after applying the Error analysis modification on 3-dataset classes. The results indicate a considerable improvement in the performance; however, it highlights an ambiguity between levels A and B. This issue will be considered in the following corpus annotation modification.

Table 4.19 Confusion Matrix of XgBoost Classification prediction after Error Analysis modification.

	A	B	C
A	995	948	0
B	611	1945	0
C	0	0	8616

Despite other TR research focusing on measuring the overall readability of text, this study concentrated solely on sentences trained using the Gloss corpus. The study resulted in the creation of 5-way and 3-way classifiers based only on Gloss sentences. The system achieved an accuracy of 0.83, precision of 0.80, recall of 0.83, and F-measure of 0.81 on the 3-classes dataset. However, the F-measure of the system dropped to 0.51 on the 5-classes dataset. This suggests that adding 'Alkitaab' instances improved the 5-classes classification with an F-measure of 0.72 which had not been attained previously.

To conclude, the results from ***Phase one*** experiments lead to enhancing the training dataset for better classification and improving annotations in the Gloss, and Al-Kitaab corpora have led to better interpretation of the links between language proficiency and linguistic features. At this stage, to improve the classifier, new features, or different representations of the sentence for a better understanding of the training dataset. This led to running experiments presented in the following section, ***Phase two***.

4.5.2 Phase two

Here, the decision was to work with the three main CEFR levels (A, B, and C) because it was quite challenging to determine the boundary between the inner sub-levels as in the boundary between B1 and B2. Yet, the other binary classification is either Simple (A+B) or Complex (C). Here there is a problem for evaluation since the gold standard labels are represented as integers 1, 2, and 3 (for the A, B, and C levels, respectively), which leads to a large number of ties. Out of the standard correlation measures, Kendall's tau-b is designed to handle ties, so in addition to Pearson's ρ, this is the adopted measure for regression (Maurice and Dickinson, 1990). In principle, the dataset is classified for 5-way (A1, A2, B1, etc.), 3-way (A, B, or C), and binary (A+B vs C) classification tasks, but here in phase two, experiments focus on the 3-way and binary (simple vs complex) classification tasks.

4.5.2.1 Readability as a classification problem

Table 4.19 presents the results of classification using an updated version of dataset one after the application of error analysis. The classification was performed using different machine learning (ML) approaches with 10-fold cross-validation on a three-way multi-class classification problem.

The classification results in Table 4.20 are divided into two categories:

1. Linguistics: This category represents the results obtained by adding XLM-R vectors to the original set of linguistic features and training the model with 1058 features (1024 XLM-R dimensions + 34 linguistic features).
2. Neural: This category represents the results obtained by representing the sentence only by sentence embeddings with neural models.

The results show that using linguistic features along with sentence embedding vectors, an SVM with rbf kernel classifier provides the best F-1 score of 0.75 on the updated corpus version. The SVM classifier is slightly better than both Xgboost and Softmax in precision, and they have roughly the same recall value.

The comparison between different types of sentence embeddings for Arabic text, including XLM-R, mBERT, fasText, and UCS, as well as two BERT-based models, AraBERT and Arabic-BERT. The results show that Arabic-BERT outperformed the other models, achieving an F-1 score of 0.80. It is suggested that the better performance of Arabic-BERT is due to the use of a more diverse corpus for training, including Common Crawl and Wikipedia for Arabic-BERT, compared to Common Crawl XML-R and Wikipedia for BERT, AraBert, and UCS.

The performance of the models is further analyzed using a confusion matrix in

Table 4.21, which shows a clear separation between the lower and higher levels of proficiency in the dataset. The majority of errors occur between neighboring levels, and the number of errors decreases as the predicted class moves further away from the true class. The most problematic level is B, which tends to be misclassified as CEFR Level A.

Table 4.20 3-way classification using weighted macro averaged precision, recall, and F-1, Dataset One VII Using all features versus neural models.

Classification model	P	R	F-1
Features			
KNeighbors	0.51	0.55	0.52
Naïve bayes	0.68	0.65	0.65

Classification model	P	R	F-1
Decision Tree	0.75	0.77	0.74
Random Forest	0.59	0.75	0.66
XgBoost	0.74	0.77	0.74
Softmax	0.74	0.77	0.74
SVM, Linear	0.75	0.77	0.74
SVM, rbf kernel	**0.75**	**0.77**	**0.75**
Neural			
FastText	0.57	0.59	0.58
UCS	0.52	0.53	0.52
mBERT	0.53	0.54	0.53
ArabicBERT	**0.78**	**0.80**	**0.80**
AraBERT	0.73	0.73	0.73
XLM-R	0.56	0.70	0.61

Table 4.21 Confusion Matrix of SVM (rbf) on 3-way classification with XLM-R.

Predicted	A	B	C
A	7485	1021	156
B	4506	1112	0
C	0	0	8627

4.5.2.2 Readability as a regression problem

The training, just as in the previous experiment, use 10-fold cross-validation to train various machine learning models for predicting sentence readability as a regression problem, similar to the previous experiment. The performance of these models is evaluated using mean absolute error (MAE) from the gold standard and correlation coefficients, including Pearson, Spearman, and Kendall's tau. Table 4.22 lists the results of the different models, which demonstrate improved performance across all methods through error analysis. The best-performing model achieves an MAE rate of 0.34, indicating that the predicted sentence difficulty is quite close to the gold labels. The best MAE rate of 0.34 shows that sentence difficulty prediction is quite close to the gold labels. As mentioned before, this model has an exceptional number of ties for the gold labels (which can only take three values), so the preferred evaluation measure for regression is Kendall’s tau-b.

In accordance with the neural model for predicting sentence readability as a regression problem, I used only Arabic-BERT as it performed the best among Arabic BERT transformers in the previous classification experiment. The results of the neural model using Arabic-BERT show an RMSE of 0.31, an R^2[43] value of 0.82, and an accuracy of 0.87, indicating that this model achieved high accuracy in predicting sentence difficulty.

Table 4.22 Regression using all features and XLM-R for sentences.

Model	Pearson	Spearman	Kendall
Decision Tree 2T	0.82	0.62	0.44
Decision Tree 5T	0.83	0.64	0.47
Random Forest	0.82	0.70	0.54
Xgboost	0.78	0.56	0.37
Linear	0.74	0.67	0.49
MLP	0.81	0.68	0.49
SVR,rbf kernel	0.78	0.69	0.52
SVR, Linear	0.8	0.71	0.54
	Neural		
	RMSE	**R^2**	**Acc.**
ArabicBERT	0.31	0.82	0.87

In conclusion, the first experiment treated readability as a classification problem, where the goal was to predict the CEFR level of a given sentence. The findings indicated that the highest performance was achieved when utilizing a combination of linguistic features and sentence embeddings in conjunction with an SVM classifier with an rbf kernel, resulting in an F-1 score of 0.75. Additionally, the study found that Arabic-BERT outperformed other sentence embedding models in the task of sentence classification, achieving an F-1 score of 0.80.

The second experiment treated readability as a regression problem, where the goal was to predict the difficulty level of a given sentence using machine learning models. The results indicated that various machine learning models performed well, with the best-performing model achieving an MAE rate of 0.34.

[43] r-squared is how well the regression model explains observed data

Additionally, using Arabic-BERT as the neural model achieved high accuracy in predicting sentence difficulty, with an accuracy of 0.87.

Overall, these experiments demonstrate the feasibility of using machine learning models to predict the readability of Arabic sentences. The results suggest that combining linguistic features with sentence embeddings can improve classification performance, and Arabic-BERT is a particularly effective sentence embedding model for Automatic TR task. Additionally, machine learning models can also perform well when predicting sentence difficulty as a regression problem, which has important implications for language learning and text simplification applications.

4.5.2.3 Feature selection

Interpreting feature importance is indeed a valuable method for gaining insights into how a machine learning model is making its predictions. This process provides a ranking of the features by assigning a score for each feature that represents its contribution to the target label prediction. These scores provide insights into data representation and model performance.

In this case, the Recursive Feature Elimination (RFE) approach using an SVM classifier has been applied to identify the most important features for sentence difficulty scoring. RFE works by recursively removing some features and testing the remaining features to select the best feature set affecting the classifier decisions. The results in Table 4.23 show that sentence embedding using XLM-R is the most useful feature, indicating that it has the greatest impact on the model's predictions. This suggests that the way sentences are represented in the embedding space is a crucial factor in determining their difficulty.

The CEFR word frequency features also appear to be important, with four features (Label A1, Label B2, Label C2, and Entropy) ranking high in the feature importance list. These features are related to the frequency of words in the sentence and their CEFR level of difficulty.

Finally, the syntactic set of features is also considered important, suggesting that the model places significant weight on the sentence's syntactic structure in determining its difficulty. Overall, these insights into the most important features can be used to optimize the model's efficiency and effectiveness by focusing on the most relevant variables and removing the noise or irrelevant features.

Table 4.23 Ten most effective features using the REF approach based on the SVM classifier

35 Sentence embedding
26 Average phrases/sentence
31 Incidence of Level B2
27 Average phrases depth
28 Incidence of Level A1
24 Incidence of modifier/root
23 Incidence of objects
22 Incidence of subjects
32 Incidence of Level C1
34 Words CEFR levels entropy

4.5.2.4 Ablation

Going further, performing feature ablation experiments by excluding certain sets of features using a SVM rbf classifier on a dataset with different groups of features. The results of the experiment indicate that the sentence embeddings (specifically, those produced by the XLM-R model) play a significant role in the classification results, even when compared to the other hand-crafted features (such as POS, syntactic features, and CEFR-level lexical features). This suggests that the transformer models used for generating sentence embeddings provide a rich representation of the sentences that can contribute significantly to classification accuracy as presented in Table 4.24.

At the same time, the experiment also suggests that the linguistic features (such as POS, syntactic features, and lexical features) can still be useful in interpreting the results of neural classification, even if their impact on classification accuracy is relatively smaller compared to the sentence embeddings.

Overall, these results suggest that combining both hand-crafted linguistic features and neural representations (such as sentence embeddings) can lead to more accurate and interpretable sentence classification models. These results encourage to continue experimentation by applying only the sentence embedding feature to reduce the number of features which consequently decreases the data analysis and training time. This approach can have several benefits. By relying solely on the sentence embeddings, could simplify the feature set and reduce the computational overhead associated with processing and analyzing multiple types of linguistic features. This can also allow focusing more on optimizing the neural model architecture and hyperparameters.

Table 4.24 SVM Classification ablation experiment on 3-way classification

Feature set	P	R	F-1
Exclude XLM-R	0.49	0.63	0.55
Exclude POS	0.55	0.71	0.62
Exclude Syntactic	0.57	0.69	0.59
Exclude CEFR	0.55	0.71	0.62
Only XLM-R	0.75	0.77	**0.75**

4.5.2.5 Classification testing on Saqq al bambuu

For the binary classification, the classifier reached F-1 of 0.94 and 0.98 for Arabic-BERT and SVM XML-R, respectively. However, when testing the binary classifiers trained from Dataset one on Saqq al bambuu, the accuracy drops considerably, see Table 4.25. As the confusion matrix in Table 4.26 shows, both classifiers performed better in identifying the complex instances rather than simple ones, so the F1 measure drops. However, the initial results on dataset two show that the XLM-R classifier performed better than Arabic-BERT, still considering Arabic-BERT classifiers [both 3-way and binary] as the best classifier so far. The primary interpretation for these confusions is because of the fictional nature of Dataset Two. First, fiction is well represented in the training data for the A+B levels in Dataset One. In contrast, the C level (Snapshot corpus) contains texts of many diverse types from the internet, so the classifiers could not handle the mismatch in genres. The other probable reason is that the developers of simplified part in Dataset Two recommended simplifying what they considered complex sentences, which may not necessarily be considered complex by all readers. Therefore, such sentences may not be suitable only for C-level students. Further research is needed to identify the difference between the two datasets.

Table 4.25 Fine-tuned Arabic-BERT versus SVM XLMR Classifier's performance on Dataset two

	Arabic-BERT	XLM-R
P	0.60	0.56
R	0.50	0.53
F-1	0.53	0.54

Table 4.26 Confusion Matrix with binary classifier Arabic-BERT versus XLM-R on Dataset Two

	Arabic-BERT		XLM-R	
Predicted	A	C	A	C
A	19	2961	138	2842
C	46	2934	223	2757

4.5.2.6 Sentence similarity testing on Dataset Two

The experiment aimed to test the sentence similarity of simplified and non-simplified sentences in Dataset Two. To achieve this, I duplicated the 2980 complex sentences without any simplification and aligned them with the exact sentence without modification, labelling them with 0 to indicate that they were not paraphrased or simplified. This resulted in a dataset of 5960 sentences, with 2980 correctly simplified sentences labelled as 1 and 2980 non-simplified sentences labelled as 0. Two models, AraBert and Arabic-Bert, were trained on this similarity task, and they both achieved an F-1 measure of 0.98, indicating their ability to detect sentences that meet the simplification standards set by Dataset Two.

4.6. Conclusions

This chapter presents the first attempt to build a methodology for Arabic difficulty classification on the sentence level. It has been found that while linguistic features, such as POS tags, syntax, or frequency lists, are valid for prediction, Deep Learning is the most significant contribution to performance. However, the traditional features can help in interpreting the black box of Deep Learning alone. For this specific task and the Arabic language, fine-tuned Arabic-BERT offers better performance than other sentence embedding methods. Also, the application of the classifiers trained on one dataset to a vastly different evaluation corpus shows that the classifiers learn some essential properties of what is difficult in Arabic. However, the classifier is more successful for the feature-based models than for the BERT-based ones. The best results have been achieved using fined-tuned Arabic-BERT. The accuracy of our 3-way CEFR classification is F-1 of 0.80 and 0.75 for Arabic-Bert and XLM-R classification respectively and 0.71 Spearman correlation for regression. While the binary difficulty classifier reaches F-1 0.94 and F-1 0.98 for sentence-pair semantic similarity classifier.

In the end, the best classifier is reasonably reliable in detecting complex sentences; however, it is less successful in separating between the lower learner levels. Still, the binary classifier provides the functionality for filtering out complex sentences not suitable for learners.

Chapter Five: Using neural models to detect and simplify difficult sentences

The literature presented in chapter 3 leads us to the questions regarding TS models and suggested work to fill the gap in Arabic TS research. These questions are:

1. **What is the reliable Arabic corpus for TS task?**

   As per the literature, in contrast to English, Arabic and many other languages lack extensive monolingual resources that are similar and conducted manually. Considering the current approaches, the most feasible solution for such languages with limited resources is to automatically acquire a corpus of paired sentences, which are complex and simple, from a large pool of web texts.

2. **What are the principles for Arabic text simplification?**

   The principles for Arabic text simplification involve identifying and modifying linguistic features that affect the readability of the text. Some of these features include the use of complex syntactic structures, multi-functional nouns, attached pronouns, and the lack of vocalisation diacritics.

   One of the main challenges in automatic Arabic text simplification is the fact that many syntactic structures are expressed through changes in the morphological pattern of words, making it difficult to identify and modify these structures automatically (Habash, 2010).

   Another challenge is the lack of consensus on the reliability of NLP tools and corpora for the Arabic language. This makes it difficult to develop accurate and effective automatic text simplification algorithms for Arabic text.

   To address these challenges, the study may need to investigate the availability of reliable Arabic corpora and develop new NLP tools specifically tailored to the unique characteristics of the Arabic language. Additionally, it needs to consider the different types of reading texts and the specific needs of the readers when simplifying Arabic text in a very precise manner.

3. **How is Arabic LS performed?**

   The Arabic LS component will follow a pipeline that includes four steps: (i) identifying the complex word; (ii) substitution generation [SG]; (iii) substitution selection [SS]; and (vi) substitution ranking [SR]. These steps will be performed as the pipeline on a suitably selected corpus with some final steps added. Selecting the synonyms of the complex word could be performed by using the Arabic WordNet Corpus for word sense disambiguation to discover the most appropriate senses or Word Embeddings or BERT transformers.

4. **Why are some texts difficult to simplify?**

   Like MT, some texts are more difficult to simplify than others. Therefore, we need to explore those texts to answer this question after performing the simplification process, identifying complex simplification texts by performing an error analysis of the simplification sentences and re-measuring the readability level. The aim is to decide if errors arise from the numerical size of the complex structure of the lexical complexity or the text genre.

Therefore, this chapter presents an attempt to present an Arabic sentence-level simplification method that aims to 1) explore text components that lead to lexical and syntactic complexity; 2) find the principles for Arabic sentence simplification 3) answer why some texts are difficult to simplify. This chapter describes the experimentation with SS using two approaches: (i) a classification approach leading to LS pipelines which use Arabic-BERT, a pre-trained contextualised model, as well as a model of fastText; and (ii) a generative approach, considering SS as an MT task, a Seq2Seq technique by applying both mT5 and OpenNMT. Also, this section provides description of an attempt for automatic building monolingual parallel corpus of complex/simple sentences. However, as this attempt did not result in an accurate reliable parallel corpus to conduct this research, I have added a method to enhance it. It should be stressed that the LS state-of-the-art and some common features of the previous literature of research on Sentence Simplification are adopted in this chapter. However, the novelty of the presented methodology lies in the fact that it combines different techniques while applying the latest NLP techniques using a new Arabic dataset.

The general framework of both approaches is based on the hypothesis that not all sentences in the text need simplification; neither all words in a sentence are required to be simplified.

This chapter will describe the details of these two approaches, including an outline of the methods applied to evaluate the results on both automatic and manual metrics. Along with the findings in the resulting datasets, tools, and their discussion. The current chapter is divided into four main sections: Section 5.1 describes the classification approach procedures. It outlines the steps applied to perform the LS pipeline. Section 5.2, firstly it is dedicated to proposing the methodology for mining the monolingual parallel corpus with a discussion of the resulted corpus. Then it presents the generation approach methods. Section 5.3 presents the analysis and evaluation of the results via manual and automatic methods. It discusses the approaches to evaluating produced simplified sentence versions to decide on the most congruent paraphrases of the original Arabic complex and the target simplified version. Finally, Section 5.5 gives a conclusion to this chapter.

5.1. Method One - Classification approach

The reference for this approach is the pipeline of the LS task as it is composed of 1) Complex word identification [CWI], (2) Substitution Generation [SG], (3) Substitution Selection [SS], and (4) Substitution Ranking [SR]. The aim of this approach focuses on LS by replacing complex vocabularies or phrasal-chunks with suitable substances (Paetzold and Specia, 2017).

Classification Approach SS is considered a classification task that requires a decision on which word to replace or syntactic structure to regenerate in each complex sentence. This approach allows the application of the LS task pipeline, i.e., aims to control the readability attribute of the text and make it more accessible to different readers with various intellectual abilities. LS particularly involves word change, thus experimenting with the effect of different embedding representations on word classification decisions. This approach highlights the impact of how the text is simplified either by applying word embedding or contextualised embedding such as ***BERT*** (Devlin et al., 2019).

Considering the definition of the four main steps applied in the pipeline for LS is as follows: Complex word identification [CWI] is the main first step performed at the top of the pipeline that is employed to distinguish complex words from simple words in the sentence. Substitution Generation [SG] involves generating all possible substitutions but without including ambiguous substances that would confuse the system in the Substitution Selection step. Substitution Ranking [SR] is to order the newly generated substitution list to ease the selection step by giving a high probability of the most appropriate highly ranked word. Finally, Substitution Selection [SS] is responsible for selecting the most suitable substitute from the generated ordered list of SG, taking into account the context, while preserving the same meaning and grammatical structure. Considering the fact that a word may have multiple meanings, and different meanings will have different relevant substitutions, then the SS task may generate a miss-substitution, which may lead to meaning corruption.

Sentence pre-processing was the first step in the classification approach. In this step, each complex sentence was analysed by using MADAMIRA, which allows generating word features such as 1) the word's ***lemma***; 2) the ***gloss*** representing the English translation of the word; 3) morphological analysis splitting the affixes from the stem (prc3, prc2, prc1, prc0, enc0). The latter feature attempts to count the number of syllables per word, representing the affixes attached to the word. Figure 5.1 represents a summary of the classification simplifier approach to the framework.

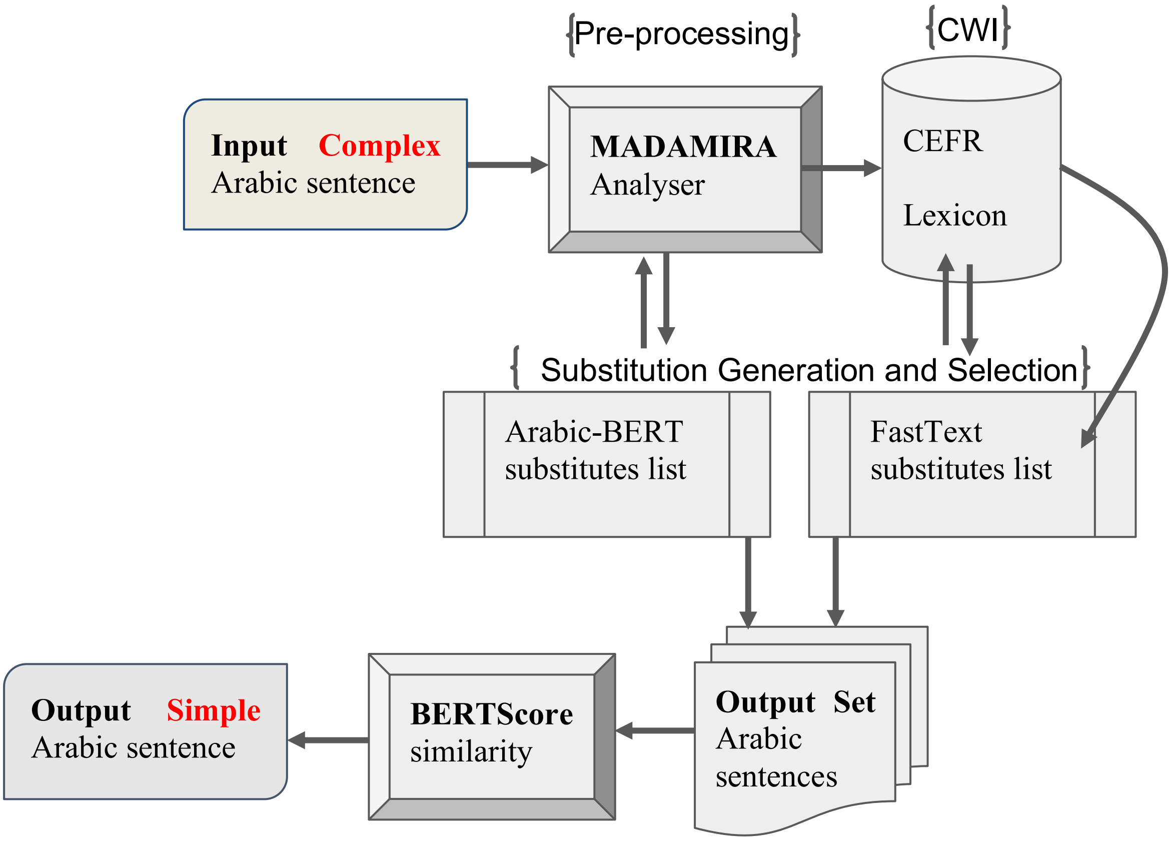

{Pre-processing}
{CWI}
Input Complex Arabic sentence
MADAMIRA Analyser
CEFR Lexicon
{ Substitution Generation and Selection}
Arabic-BERT substitutes list
FastText substitutes list
Output Set Arabic sentences
BERTScore similarity
Output Simple Arabic sentence

Figure 5.1 Lexical simplification system- Classification approach

5.1.1. Complex word identification (CWI)

To identify the complex word in the sentence, various analysis approaches were utilized at different stages. As per the literature, the complexity of a word is commonly evaluated based on four factors: length, familiarity, ambiguity, and context. It is believed that the number of syllables per word is a good indicator of complexity. Hence, the more syllables there are, the more complex the word will be. The word familiarity also refers to word frequency measures using a large corpus. Ambiguity can be measured empirically by counting the number of synonyms for the target word. The context indicates that a word's complexity is not a static notion but is influenced by the surrounding words. For each word in a complex sentence, the set of feature categories presented in Table 5.1 as extracted to indicate the word frequency and familiarity, whereas ambiguity is resolved in later steps.

Table 5.1 Feature categories for each complex word

Word length	○ Syllable Count (morphological structure MADAMIRA)
Frequency	○ TF-IDF
Familiarity	○ Using CEFR List as an indicator of complex words, words ranked B2, C1, and C2 are selected as a target to be simplified.

CWI step could be viewed as a layered analysis to opt for a better understanding of word complexity. Hence, applying a lexicon-based approach. Considering one sentence per time, the first level relates to identifying the number of syllables per word in the target sentence keeping a record of its POS-tag along with other features produced by MADAMIRA to be used in further steps. The second layer of analysis moved to assign each word a CEFR complexity level, adopting a Lexical-based approach using CEFR vocabulary List as a reference to allocate each word in the target sentence to a readability level. At CWI, with identifying the complex words, these words become the targets to simplify. First, by ordering words according to their CEFR level and considering each of these words as the target per time to deploy the simplification process. For example, if a sentence has three complex words assigned with B2, C2, and C1, firstly order them to be C2, C1, and B2 and then start the simplification process by targeting C2 tagged words, followed by C1 and so on. In this example, this process results in generating three sentences, each with a different masked word slot, while keeping the original sentence. One of the major word categories that needs to be treated carefully in simplification is the Named entities. These words were identified by MADAMIRA, considering words tagged as 'proper noun'.

5.1.2. Substitution generation and substitution ranking

After the complex words are identified, the next step is to generate a list of substitutions for the target word and rank them based on the original context. Substitution generation and ranking steps were considered in one process using different methodologies to generate the substitution list and rank them considering semantic similarity measures. For this purpose, obtaining different sentence embedding to the top-ranked substitution list of the complex token. This involves providing an arranged synonyms list of the target word.

To reach this goal, the decision was to implement three classification models:

1. The classification model, which is based on word embedding, thus applies ***fastText*** word embedding tool that represents words as vectors embedding. Those vectors embedding was trained on Common Crawl and Wikipedia. Adopting the Arabic ar.300.bin file in which each word in word embeddings is represented by the 1D vector mapped of 300 attributes (Grave et al., 2018);
2. The classification model is based on transformers using ***Arabic-BERT*** (Safaya et al., 2020);
3. Classification model combining both ***fastText*** and ***Arabic-BERT*** results with post-editing rules.

The following subsections are dedicated to describing those models and how they work and interpreting the resulting simplified sentences. The aim here was to compare Arabic-BERT and fastText produced lists to obtain better word substitutes and to find which model is dealing better in resolving lexical ambiguity.

It should be noted that Arabic WordNet does not have enough synsets for many complex words, which makes it unreliable for generating substitutions. For example the word 'التجاعيد' wrinkles, in Arabic WordNet does not have any synsets,however, in English WordNet version has 7 different sysnset as wn.synsets('Wrinkles'),[Synset('wrinkle.n.01'),#Synset('wrinkle.n.02'), #Synset('wrinkle.n.03'),#Synset('purse.v.02'),#Synset('wrinkle.v.02'), #Synset('furrow.v.02'), #Synset('rumple.v.03')].

Therefore, it makes sense to exclude it from the system and use other embeddings methods for generating substitutions.

5.1.2.1. Arabic-BERT embedding

Arabic-BERT model has different tasks to use in various NLP tasks. For each complex word, apply BERT's task *MaskedLanguageModeling (MLM)*. This task predicts a substitution list of a masked [not shown, complex] token in a sequence given its left and right context. In this process, the MLM requires a concatenation between the original sequence and the same sentence sequence where the target word is replaced by [MASK] token as a sentence pair and feed the sentence pair into the BERT to obtain the probability distribution of the possible replacements corresponding to the MASK word. To use any pre-trained BERT model, there is a need to convert the input data (sentence's tokens) into an appropriate format so that each sentence can be sent to the pre-trained model to obtain the corresponding embedding using modules and functions available in Hugging Face's transformers package.

For this task, in the ne xt sentence prediction, the beginning and end of each sentence need to be marked before feeding them to the BERT model. For this purpose, a general token [CLS] was added as a first token to represent the hidden state of the whole sentence, along with adding another generated token [SEP] identifying the end of a sentence. For example, any input could be represented by: [CLS] original sentence [SEP] sentence with a masked token [SEP], in which [CLS] is the beginning of the sentence, the first [SEP] a mark for the end of the first sentence and the beginning of the following one and, a last [SEP] identifying the end of the whole input.

By using this approach, consider not only the complex word but also the surrounding context of the complex word.

For instance, given this sentence from Arabic Wikipedia:

تَتَطْلَّبُ مِنْ هَيْئَةِ الْمَحْكَمَةِ **وُجُوبَ** تَحْدِيدِ الْحُقُوقِ
tataṭllabu min hayʾati almaḥkamati **wujūba** taḥdīdi alḥuqūqi
[It is **obligatory** from the court to declare the rights]

The sentence pair construction before feeding into BERT (shortening the original sentence for clarification) is as follows:

[CLS]تَتَطْلَّبُ مِنْ هَيْئَةِ الْمَحْكَمَةِ وُجُوبَ تَحْدِيدِ الْحُقُوقِ [SEP] تَتَطْلَّبُ مِنْ هَيْئَةِ الْمَحْكَمَةِ[MASK] تَحْدِيدِ الْحُقُوقِ [SEP]

The complex word in this sentence is "وُجُوبَ" (wujūba , 'obligatory'), to get the simplest replace candidates, the sentence will be fed into Arabic-BERT, replacing the complex with [MASK] as represented in Figure 5.2 showing the first five BERT-prediction candidates. Also, this figure illustrates part of the prediction list of the [MASK] word “وُجُوبَ”) wujūba , ‘obligatory’) applying MLM task [BertForMaskedLM] from the hugging face library.

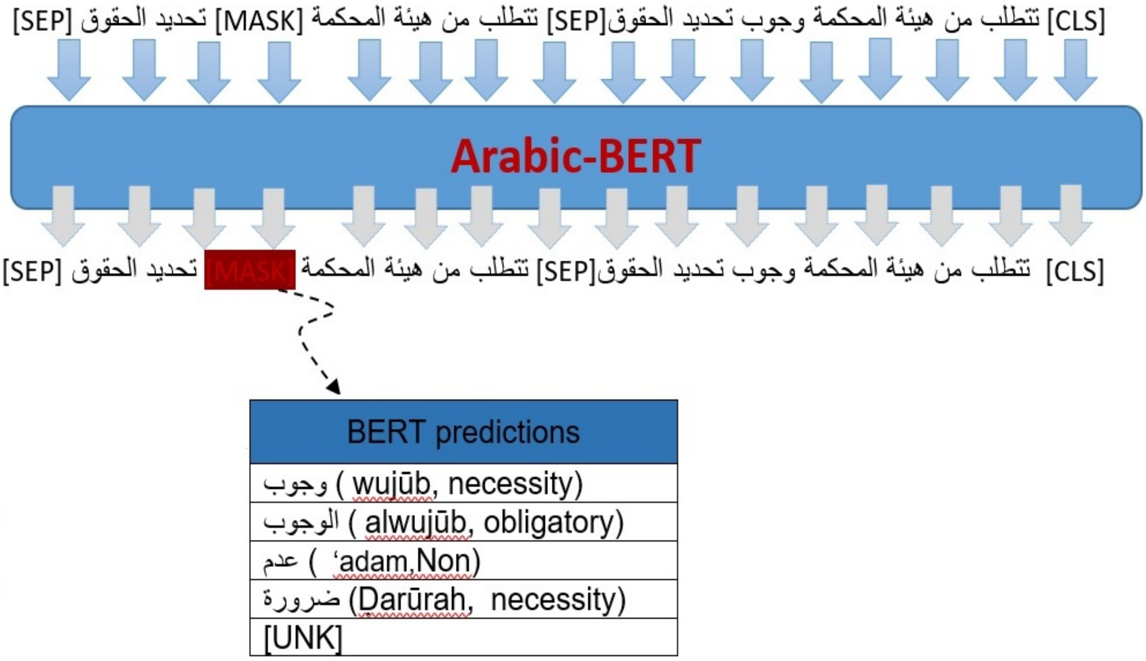

[CLS] تتطلب من هيئة المحكمة وجوب تحديد الحقوق[SEP] تتطلب من هيئة المحكمة [MASK] تحديد الحقوق [SEP]
Arabic-BERT
[CLS] تتطلب من هيئة المحكمة وجوب تحديد الحقوق[SEP] تتطلب من هيئة المحكمة MASK تحديد الحقوق [SEP]
BERT predictions
وجوب (wujūb, necessity)
الوجوب (alwujūb, obligatory)
عدم (‘adam,Non)
ضرورة (Darūrah, necessity)
[UNK]

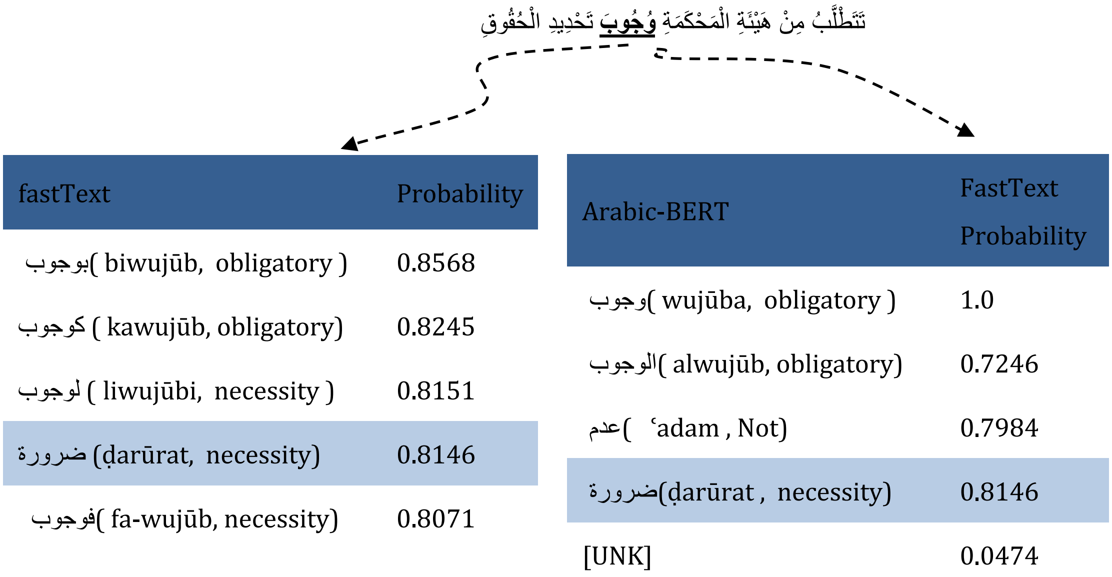
Figure 5.2 Sentence fed structure to Arabic-BERT

One of the most noticeable aspects of BERT is sentence tokenisation which is an initial step before converting tokens into their corresponding unique IDs [embedding vector]. A key point to highlight about the BERT-tokenizer algorithm is the common out-of-vocabulary (OOV) problem. Since the model is pre-trained on a specific corpus, the words are limited to ones that appeared in this training corpus. As a solution, in testing and prediction processes, BERT models are designed to replace the unseen tokens with a unique token [UNK], which stands for unknown token. However, converting all unseen tokens into [UNK] will take away much information from the input data. Hence, the BERT tokeniser adopts the *WordPiece* algorithm that splits the sentences into words and breaks out words into several subwords. This splitting technique is represented by the model by adding ‘##’ as a start for each consecutive word part. In other words, a token starting with ‘##’ could be appended to the previous token to reform the original word. For example, the word Tatṭlb (“تتطلب – ‘require’) in the previous example does not appear in the training corpus. Without tokenising the sentence, it is directly replaced with the token [UNK] with the ID 100. Nevertheless, when applying the BERT tokeniser would tokenise this word as [',[’طلب##‘ ,’تتـ which matches the word that appears in the training corpus.

Where the first token is a more commonly seen word (prefix) in a corpus, and the second token is prefixed by two hashes ## to indicate that it is a suffix following some other subpart. If there is no way to split the token into subwords, the whole word becomes [UNK]. After this tokenisation step, all tokens can be converted into corresponding IDs. Before calling the function named **[BertForMaskedLM]**, considering the previous concatenation as input and feeding it to Arabic-BERT-model using the MLM task, it will be able to predict the [MASK] word. This will produce an output of the ranked set of most probably occurring words given the target context, as presented in Figure 5.3 as the BERT-***prediction*** list.

5.1.2.2. fastText- embedding

Using this word embedding model in two folded processes, first ranking the previously produced substitutions obtained by MLM Arabic-BERT by getting the semantic cosine similarity between each word in the produced list to the target complex word. The second is using ***fastText*** word embedding to generate a list of replacements [***Substitution Generation***] and then rank [***Substitution Ranking***] by the nearest neighbour. For instance, applying ***fastText*** to generate the list of synonyms given the target word in the previous example “وجوب”(wujūba, ‘obligatory’), the predictions were shown on the left side of Figure 5.3. In contrast, the ranking probability using ***fastText*** for ***Arabic-BERT*** list prediction was shown on the right side of the figure. The use of these probabilities, represented in Figure 5.3, will be explained later in the following section ***5.1.3 substitution selection***.



تَتَطْلَّبُ مِنْ هَيْئَةِ الْمَحْكَمَةِ **<u>وُجُوبَ</u>** تَحْدِيدِ الْحُقُوقِ

fastText	Probability
بوجوب(biwujūb, obligatory)	0.8568
كوجوب (kawujūb, obligatory)	0.8245
لوجوب (liwujūbi, necessity)	0.8151
ضرورة (ḍarūrat, necessity)	0.8146
فوجوب(fa-wujūb, necessity)	0.8071

Arabic-BERT	FastText Probability
وجوب(wujūba, obligatory)	1.0
الوجوب(alwujūb, obligatory)	0.7246
عدم(ʻadam , Not)	0.7984
ضرورة(ḍarūrat , necessity)	0.8146
[UNK]	0.0474

Figure 5.3 BERT and fastText prediction lists along with the probability obtained from *FastText* for the word "وُجُوبَ" (wujūba, 'obligatory')

5.1.3. Substitution selection

At this stage, each complex word in the sentence has differently ordered substituted lists based on ***Arabic-BERT*** and ***fastText.*** It is a very crucial stage, and the system needs to be careful when selecting the best substitute based on different measures. Each prediction list was considered individually to analyse and select the logical substitute based on the semantic similarity measures. This allowed the system to generate a set of simplified versions of the target sentence. In addition, it kept a record of the semantic similarity and the readability level of the newly produced sentences. The system produces three simple sentences based on ***Arabic-BERT*** substitute selection, ***fastText***, and combined decisions from both generated lists.

Given that the lists for the word "وجوب" (wujūba, 'obligatory') are presented in Figure 5.3. Starting with the ***Arabic-BERT*** list, the greater the value the most common or familiar the word for a person referring to simple words. If the decision were to replace the first word with the highest probability, the replaced word would be بوجوب (bi+wujūb, by necessity) or الوجوب (al-Wujūb, obligatory) as predicted by Arabic-BERT and fastText respectively. In this case, the word would remain the same either by adding the prefix " بـ bi, by" or the definite article "الـ al, the", which gives the idea of ranking Arabic-BERT's list using ***fastText*** semantic similarity measures. As presented in Table 5.2, this first word appeared in both the original ***Arabic-BERT*** list and the re-ranked ***Arabic-BERT*** is وجوب (wujūb, obligatory) which is the same as the original complex word. In such case, it is easy to ignore the first instance in a list if it is the same as the original masked word. Following the second choice in the original ***Arabic-BERT*** list, the replacement word would be الوجوب(alwujūb, obligatory of as explained before with the probability 0.7246 as presented in Table 5.3 (BERT 1st choice). In theory, this instance could be easily rejected because it is a different morphological form of the original complex word, and it is not logical to change the complex word with a word with the same lemma. This would direct to the third instance عدم (ʿadam, lack of) as shown in Table 5.3 (BERT 3rd Choice); this choice will give the complete opposite meaning of the sentence. Whereas, using the second instance ضرورة**(ḍarūrat, necessity)** from the re-ranked list with the probability of 0.8146 **would** give an easy way to find the simple word replacement. While, using the list from ***fastText*** and applying a simple rule of rejecting any instance that shares the same lemma with the complex word would direct to the fourth choice of the word ضرورة **(ḍarūrat, necessity)** as in Table 5.3 (fastText Choice). The last sentence in (fastText Choice) is the target simple synonym.

Table 5.2 The ranked substitution list for the word “وُجُوبَ” (wujūba, ‘obligatory’)

Original Arabic-BERT	**Re-ranking by fastText**	**Original fastText**
“وُجُوبَ” (wujūba , ‘obligatory’)	“وُجُوبَ” (wujūba , ‘obligatory’)	"بوجوب"(biwujūb, necessity)
"الوجوب"(al-wujūb, obligatory)	“ضَرُورَة” (ḍarūrat, necessity)	"كوجوب"(kawujūb, obligatory)
"عدم"(ʿadam ,lack of)	" عدم"(ʿadam, lack of)	"لوجوب"(liwujūb, necessity)
"ضرورة"(ḍarūrat, necessity)	"الوجوب"(alwujūb, obligatory)	"ضرورة"(ḍarūrat, necessity)
[unk]	[unk]	"فوجوب"(fawujūb, necessity)

Table 5.3 The set of newly generated sentences using *Arabic-Bert* and *fastText* list

Original	تَتَطْلَّبُ مِنْ هَيْئَةِ الْمَحْكَمَةِ وُجُوبَ تَحْدِيدِ الْحُقُوقِ	the court is obligatorily required to declare the rights	
BERT 1st choice	تَتَطْلَّبُ مِنْ هَيْئَةِ الْمَحْكَمَةِ الْوُجُوبَ تَحْدِيدَ الْحُقُوقِ	the court requires the obligation to the declaration of rights	Ill- formed
BERT 3rd Choice	تَتَطَلَّبُ مِنْ هَيْئَةِ الْمَحْكَمَةِ عَدَمَ تَحْدِيدِ الْحُقُوقِ	The court requires not to declare the rights	Opposite meaning
fastText Choice	تَتَطَلَّبُ مِنْ هَيْئَةِ الْمَحْكَمَةِ ضَرُورَةَ تَحْدِيدِ الْحُقُوقِ	The court requires the necessity to declare rights	Right simplification

As shown in the previous example, relying only on Arabic-BERT will not give an appropriate paraphrase of the original sentence. However, using either the ***re-ranked Arabic-BERT*** list or the original ***fastText*** will better represent the possible meaningful replacements; this will also be proved in further examples.

Therefore, the selection of the best substitute could be controlled by combining both ***Arabic-BERT*** and ***fastText*** results along with a set of selection rules to limit incorrect selection as follows:

1. ***Rule1***: *"if [UNK] is a top-ranked Arabic-BERT substitute representing unrecognised word by BERT, then go to fastText results*."

Check if the first substitute is [UNK]; in this case, the system completely ignores BERT results and keeps the original, then relies on *fastText* results immediately.

2. ***Rule2***: *"if any word's lemma in the generated list equals the lemma of the original word, exclude these words from the list*."

Check if the lemmas in the predicted list match the same lemma of the target word. In this case, exclude these words from the potential replacement for the target word and keep only the words with a different lemma. These replacements should also share the same POS and Number with the target word.

3. ***Rule3:*** “*CEFR list placement for difficulty*.”

Check the word CEFR level of the new substitute word. The new word's CEFR level should be equal to or less than the CEFR level of the target word. Because sometimes, the generated list may have a more frequent substitute which is more difficult than the original word but more frequent.

4. ***Rule4: “check if the new substitute shares the meaning.”***

The system uses this rule as it gives a level of confidence to the system selection. After the system makes the final decision, either keep the target word or select the suggested substitute based on previous rules. At this stage, the target translation is compared to the substitute’s translationappeared in ***Gloss*** feature]. If both words share part or all possible translations, the system is confident of replacing the target with the substance.

The following examples illustrate how the system follows the rules for optimum selection of the new simple substitute while preserving the original meaning.

Example 1 is in Table 5.4. shown original complex word “أُحْدِّقُ” (ʾuḥaddiqu , 'staring’) and generated sentence with the simple substitute”أَتَأَمَّلُ” (‘ata’ammalu ,‘muse’)

Table 5.4 Example 1, Rule1 application

كُنْتُ **أ◌ُحْدِّقُ** فِي الطَّبَقِ وَالصَّمْتِ يَكَادُ يَبْتَلِعُ الْمَكَانَ.	**Original sentence**
kuntu ʾuḥaddiqu fī aṭṭabaqi waṣṣamti yakādu yabtaliʿu almakāna	
[I was ***staring*** at the plate, and the silence almost swallowed up the place.]	
كُنْتُ **أَتَأَمَّلُ** فِي الطَّبَقِ وَالصَّمْتِ يَكَادُ يَبْتَلِعُ الْمَكَانَ.	**Generated sentence**
kuntu ʾataʾammalu fī aṭṭabaqi waṣṣamti yakādu yabtaliʿu almakāna	
[I was ***staring*** at the plate, and the silence almost swallowed up the place.]	

In this context, for the word “أُحْدِّقُ ” (ʾuḥaddiqu , 'staring’), ***Arabic-BERT*** predicted the first substitute for the target word is [UNK], which indicates that this word does not exist in the training corpus. Whereas the ***fastText*** list of the same word represents rational and considerable substitutes as shown in Table 5.5 represented by fastText probability measures. This gives a strong reason to the system to apply ***Rule1*** and divert its selection process to *fastText* prediction’s list, and then applying ***Rule2*** the system will reject all.

Table 5.5 Arabic-BERT and fastText substitution list for the word “أُحْدِّقُ ” (ʾuḥaddiqu , 'staring’)

Arabic-BERT	fastText Probability	fastText	fastText Probability
[unk]	-0.0645	"أُحْدِّقُ" (ʾuḥaddiqu , ‘staring’)	0.7795
"أِنَا "(ʾanā ,'i')	0.2430	"وَأَحْدَقُ "(waʾaḥdaqu, ‘staring’)	0.7683
"الْآنَ "(alʾāna,'now')	0.1851	" أَتَأَمَّلُ "(ʾataʾammalu, ‘muse’)	0.7468
"هُنَاكَ"(hunāka,'right now')	0.2061	"أُحمِلْقُ" (ʾuḥamliqu ,’gaze’)	0.7381
"الْيَوْمَ"(alyawma ,'today')	0.0338	”أَتَفِرُسُ ” (ʾtafarrasu , ‘gaze’)	0.7335

In this case, the word “أُحْدِّقُ ” (ʾuḥaddiqu , 'staring’) was replaced with “أَتَأَمَّلُ” (‘ataʾammalu ,‘muse’), which is more frequent and simpler, which generates the sentence presented in Table 5.4.

Example 2 is the second example for applying the first and second selection rules, as shown in Table 5.6.

Table 5.6 Example 2 shows the application of *Rule1* and *Rule2*

تُوجَدُ بَعْضُ الْحَالَاتِ الَّتِي **يُوصَى** فِيهَا بِعَدَمِ الضَّحِكِ وَتَجَنُّبِهِ	**Original sentence**
tūjadu baʿḍu alḥālāti allatī **yūṣā** fīhā biʿadami aḍḍaḥiki watajannubihi	
[There are some situations in which it is ***recommended*** not to laugh and avoid it]	
تُوجَدُ بَعْضُ الْحَالَاتِ الَّتِي **يُنْصَحُ** فِيهَا بِعَدَمِ الضَّحِكِ وَتَجَنُّبِهِ	**Generated**

tūjadu baʿḍu alḥālāti allatī yunṣaḥu fīhā biʿadami aḍḍaḥiki watajannubihi	**sentence**
[There are some situations in which it is ***advised*** not to laugh and avoid it]	

The word “يُوصَى” (yūṣā , 'recommended/advised'), is assigned with the C1 level in *Example2*. Arabic-BRET produces a list of substitutes starting with an unknown replacement [UNK], as shown in Table 5.7. Whereas the fastText suggests a different list that starts with a word and shares the lemma with the complex word. In this case, applying ***Rule1*** will divert the system to rely on the *fastText* list, while ***Rule2*** will reject any substitutes that are the same as the original word. This results in selecting the second choice from the *fastText* list '“يُنْصَحُ” (yunṣaḥu , ‘advise’).

Table 5.7 Arabic-BERT and FastText substitution lists for the word ‘yunṣaḥ’ ('ينصح’, 'advise') ranked by FastText probability measures.

Arabic-BERT	FastText Probability of BERT	FastText	FastText Probability
[unk]	0.0877	“ويُوصَى ” (wayūṣā , 'recommended/advised')	0.8782
“يُنْصَحُ” (yunṣaḥu , ‘advise’)	0.8341	“يُنْصَحُ” (yunṣaḥu , ‘advise’)	0.8341
“اشْعُرُ” (ašuʿuru ,'feel')	0.8341	“ويَنْصَحُ” (wayanṣaḥu ,'advise')	0.7894
“ننْصَحُ ” (nanṣaḥu , 'advise')	0.3820	“يُوصَى” (yūṣā , 'recommended/advised')	0.7870
"يَتَمَيَّز"(yatamayyaz, ’characterized’)	0.4746	“فيُنْصَحُ ” (fa yunṣaḥ,'advise')	0.7800

Example 3 is the third example for applying the first and second rules in Table 5.8.

Table 5.8 Example 3 shows the application of Rule1 and Rule2

بَيْنَمَا سَيَقُومُ الْمُسْتَوْرِدُونَ بِدَفْعِ قِيمَةِ وَارِدَاتِهِمْ **لِلْمَصْرِفِ** الْمَالِيزِيِّ بِالرِّنْغِيتْ	**Original sentence**
baynamā sayaqūmu almustawridūna bidafʿi qīmati wāridātihim lilmaṣrafi almālīziyyi birrinġīt	
[While importers will pay the value of their imports to the Malaysian ***bank*** in Ringgit.]	
بَيْنَمَا سَيَقُومُ الْمُسْتَوْرِدُونَ بِدَفْعِ قِيمَةِ وَارِدَاتِهِمْ لِلْبَنْكِ الْمَالِيزِيِّ بِالرِّنْغِيتْ	**Generated sentence**
baynamā sayaqūmu almustawridūna bidafʿi qīmati wāridātihim lilbanki almālīziyyi birrinġīt	
[While importers will pay the value of their imports to the Malaysian ***bank*** in Ringgit.]	

The word “لِلْمَصْرِفِ” (lilmaṣrifi, ‘bank’), assigned with C1 CEFR level in Example 3, presents an unrelated prediction list produced by Arabic-BERT, which starts with [UNK] and is followed by either prepositions or punctuation marks as shown in Table 5.9. Conversely, fastText generates more accurate substituents for this word. This list is initialised with the simplest replacement “لِلْبَنْكِ” (lilbanki, ‘bank’) with A1 CEFR level with the highest probability of generating the new sentence in Table 5.8. This is another example when the lexical item can be directly replaced with another without impacting the structure.

Table 5.9 Arabic-BERT and fastText substitution list for the word “لِلْمَصْرِفِ” (lilmaṣrifi, ‘bank’) represented by fastText probability measures.

Arabic-BERT	**FastText Probability of BERT**	**FastText**	**FastText Probability**
[UNK]	-0.1130	“لِلْبَنْكِ” (lilbanki , ‘bank’)	0.8713
“في”(fī,'in')	0.1181	“لِلْمَصْرِفِ” (lilmaṣrifi, ‘bank’)	0.8238
, (punctuation mark)	0.0340	“لِلْمُسْتَثْمِرِ ” (lilmustaṯmiri ,’investor’)	0.7835
،(punctuation mark)	0.1920	“لِمَصْرِفِ” (limaṣrifi, ‘bank’)	0.7582
“و” (wa, ‘and’)	0.0081	“لِ◌صُّنْدُوق ” (liṣṣundūqi,’box’)	0.7429

Example 4 represents an example for applying the second and third rules, as shown in Table 5.10.

Table 5.10 Example 4 shows the application of Rule2 and Rule3

تَطْوِيرُ وَاسْتِخْدَامُ اللَّقَاحَاتِ فِي سَبِيلِ تَأْسِيسِ مَنَاعَةٍ لِلْمَرَضِ	**Original sentence**
taṭwīru wastiḳdāmu allaqāḥāti fī sabīli taʾsīsi manāʿatin lilmaraḍi	
[***Development*** and use of vaccines to establish immunity to disease]	
بِنَاءُ وَاسْتِخْدَامُ اللَّقَاحَاتِ فِي سَبِيلِ تَأْسِيسِ مَنَاعَةٍ لِلْمَرَضِ	**Generated sentence**
binā'u wastiḳdāmu allaqāḥāti fī sabīli taʾsīsi manāʿatin lilmaraḍi	
[***Build*** and use vaccines to establish immunity to disease]	

The target lemma “تَطْوِيرُ” (taṭwīru, development) was assigned to the B2 level. In this context, Arabic-BERT generates a substitution list starting with five words that share the same lemma with the target word “ تَطْوِير_1 ” (taṭwīr, development) as illustrated in Table 5.11. In this case, applying ***Rule2***, the system excludes all these words from the possible subsistence. This leaves the system with fastText predictions and applies ***Rule2*** on the new list. The system removes the first three words from the list because they have the same lemma as the target word. As a result, the substitution list is limited to only two words ”دَعْم ” (daʿm ,support) and “بِنَاءٍ” (bināʾ ,build). However, the word Da‘m (دَعْم,support) with the highest probability, the system selects the word “بِنَاءٍ” (bināʾ ,build) after applying ***Rule3*** by checking the CEFR level of the new substitute word and ensuring its level is equal to or less than the level of the original target word. In this example, the word “دَعْم” (daʿm ,support) was assigned with C1, whereas the word “بِنَاءٍ” (bināʾ , build) was assigned with A1, as illustrated in Table 5.12. When using this information, the system will favour generating a new sentence with the word “بِنَاءٍ” (bināʾ ,build), as shown in the generated sentence in Table 5.10. The new substitute was assigned with an A1 difficulty level which is two levels lower than the target word. However, the newly generated sentence is simpler than the target complex one; the newly added word is not used in such a context which affects the sentence's meaning. Yet, this newly generated sentence could be accepted as a useful simplified version.

Table 5.11 Arabic-BERT and fastText substitution list for the word “تَطْوِيرُ” (taṭwīru, development) aligned with MADAMIRA Lemma.

Arabic-BERT	Lemma	fastText	Lemma
“تَطْوِيرُ” (taṭwīru, development)	1_تَطْوِير	“لِتَطْوِيرُ” (litaṭwīru, to develop)	1_تَطْوِير
“التَطْوِيرُ” (altaṭwīru, development)	1_تَطْوِير	“وتَطْوِيرُ” (wataṭwīru, and development)	1_تَطْوِير
“وتَطْوِيرُ” (wataṭwīru, and development)	1_تَطْوِير	“بِتَطْوِيرُ” (bitaṭwīru, by development)	1_تَطْوِير
“لِتَطْوِيرُ” (litaṭwīru, to develop)	1_تَطْوِير	”دَعْم ” (daʿm ,support)	1_دَعْم
“بِتَطْوِيرُ” (bitaṭwīru, by development)	1_تَطْوِير	“بِنَاءٍ” (bināʾ ,build)	1_بِناء

Table 5.12 List of substitutions lemma, gloss, and the CEFR level

	Word	lemma	Gloss	CEFR
Target word	“تَطْوِيرُ” (taṭawwur, ’development’)	1_تَطَوُّر	progress; development	B2
Substitute1	”دَعْم ” (daʿm ,support)	1_دَعْم	support	C1
Substitute2	“بِنَاءٍ” (bināʾ ,build)	1_بِناء	Build; development	A1

Example 5 This is another example providing evidence about how it is essential to apply ***Rule 2*** on both predicated lists. As well as applying ***Rule 3*** for the selection assurance, the system decides not to simplify and keeps the original sentence as in Table 5.13.

Table 5.13 The word “يَتَعَلَّقُ” (yataʿallaqu, regard) in a sentence where the system chooses to keep the sentence without simplification.

امَا فِيمَا **يَتَعَلَّقُ** بِالْمَهْرَجَانَاتِ الْكُبْرَى لِمُوسِيقَى	**Original sentence**
amā fīmā *yataʿallaqu* bilmahrajānāti alkubrā limūsīqā	
[while ***regarding*** the major music festivals]	
امَا فِيمَا **يَتَعَلَّقُ** بِالْمَهْرَجَانَاتِ الْكُبْرَى لِمُوسِيقَى	**Generated sentence**
amā fīmā yataʿallaqu bilmahrajānāti alkubrā limūsīqā	
[while ***regarding*** the major music festivals]	

In Table 5.14, the word “يَتَعَلَّقُ” (yataʿallaqu, regard) with the target lemma " تَعَلَّق_1" (*taʿallaqu, regard)* is assigned with C1. After applying **rule 1** and **rule 2**, the target word was limited to be replaced with either ***“ يَخُصُّ” (yaḳṣṣu , regards)*** or “يَخْتَصُّ”(yaḳtaṣṣu , specializes) suggested by Arabic-Bert and fastText respectively as shown in Table 5.13. In this case, applying ***rule 3*** will give the system a better selection vision as both words were assigned with the same CEFR level as the target word; thus, the system chooses to keep the target word without modification.

Table 5.14 BERT and fastText substitution list for the word yataʿallaqu (“يَتَعَلَّقُ” , Regard).

BERT predictions	**Lemma**	**fastText predictions**	**Lemma**
“يَتَعَلَّقُ” (yataʿallaqu, regard)	1_تَعَلَّق	“يَتَعَلَّقُ” (yataʿallaqu, regard)	1_تَعَلَّق
”يَخْصُّ ” (yaḳṣṣu , regards)	1_خَصّ-ُ	“مُتَعَلِّق ” (mutaʿalliq , related)	1_متَعَلَّق
“# # عَلَّق”(##ʿallaq , ## comment)	1_عَلَّق	"تَتَعَلَّقُ"(tataʿallaqu, related)	1_تَعَلَّق
"تَتَعَلَّقُ"(tataʿallaqu, related)	1_تَعَلَّق	“يَخْتَصُّ ” (yaḳtaṣṣu , specializes)	1_اِخْتَصّ
“يَخْتَصُّ ”(yaḳtaṣṣu , specializes)	1_اِخْتَصّ	"لا يَتَعَلَّقُ "(lā yataʿallaqu, not related to)	1_تَعَلَّق

Example 6 in Table 5.15 is another example explaining the effectiveness of Rule3 in limiting the selection of more complex words than the target word. As well as applying Rule 4 in giving a confidence percentage of the system selection.

Table 5.15 The word *Taṭawwur (‘تطور' development/ evolution')* in a sentence where the system chooses to keep the sentence without simplification.

وَفْقًا لِهُولْوَايْ يَتَمَثَّلُ مِفْتَاحُ السِرّ فِي فَهْمِ تَطَوُّرِ◌ شَبِيهِ الاِنْسَانِ	**Original sentence**
wafqan lihūlwāy yatamaṯṯalu miftāḥu assirri fī fahmi taṭawwuri šabīhi aliānsāni	
[According to Holloway, a secret key is understanding the ***evolution*** of hominids]	
وَفْقًا لِهُولْوَايْ يَتَمَثَّلُ مِفْتَاحُ السِرّ فِي فَهْمِ نُمُوِّ شَبِيهِ الاِنْسَانِ	**Generated sentence**
wafqan lihūlwāy yatamaṯṯalu miftāḥu assirri fī fahmi numuwwi šabīhi aliānsāni	
[According to Holloway, a secret key is understanding the ***growth*** of hominids]	

The word “تَطَوُّرِ◌” (taṭawwuri, 'development/ ***evolution'***) in Table 5.15, after applying ***rule1*** and ***rule2,*** the system selected the word ***"نُمُو" (numuwwi, ‘growth/development’)*** as the correct substitute for the target word. Then apply Rule 3 as a final checkpoint to prevent replacing the target word with a more difficult word. Table 5.16 shows that the CEFR level of the substitute assigned with A2 is lower than B2 of the target word. This example is also considered a useful simplification case as the new word carries some meaning from the original target word. Yet, it slightly changes the meaning as shown in the generated sentence in Table 5.15. After that, applying **rule 4**, checking the translation of both original and chosen words, gives the confidence percentage of the system selection. This allows the final selection for the substitute based on the translation. At this stage, Table 5.16 represents the MADAMIRA analysis for the words displaying lemma and gloss [the possible English translations], which shows that both words, the target and the substitute, share one of the translations, which is the underlined word [development]. This emphasises how the system adds more confidence in using the word ***"نُمُو" (numuwwi ,growth/development')*** in this context.

Table 5.16 Represents the target "تَطَوُّرِ" (taṭawwuri, 'development/ ***evolution')*** and the accepted substitute with underlining the shared translation.

	Word	lemma	Gloss	CEFR
Target word	تَطَوُّر (taṭawwuri,' development')	تَطَوُّر_1	progress, development	B2
Substitute1	نُمُوِ(***numuwwi*** , 'growth')	نُمُوّ_1	development, growth	A2

Example 7 is another instance showing how applying ***Rule 3*** limits the system from providing more complex sentences than the original one. A sentence with the complex word Mut'ārf ('متعارف, 'recognized') and the newly generated sentence with the word Shā'i' (''شائع' , 'common')

Table 5.17 Example 7 with the complex word Mut'ārf (' ,'متعارف'recognized')

ثَلَاثَةُ انْوَاعٍ مِنْ الْغَضَبِ مُتَعَارَفٌ عَلَيْهَا مِنْ قِبَلِ عُلَمَاءِ النَّفْسِ	**Original sentence**
ṯalāṯatu anwāʿin min alġaḍabi mutaʿārafun ʿalayhā min qibali ʿulamāʾi annafsi	
[three types of anger ***recognised*** by psychologists]	
ثَلَاثَةُ انْوَاعٍ مِنْ الْغَضَبِ شَائِعٌ عَلَيْهَا مِنْ قِبَلِ عُلَمَاءِ النَّفْسِ	**Generated sentence**
ṯalāṯatu anwāʿin min alġaḍabi šāʾiʿun ʿalayhā min qibali ʿulamāʾi annafsi	
[three types of anger ***common*** by psychologists]	

The word "مُتَعَارَفٌ "***(mutaʿārafun, 'recognised')*** in Table 5.17, in this example applying ***Rule1*** and ***Rule2*** limit the list to three possible substitutes, as shown in Table 5.18. The substitute1 "مُتَّبَع " (muttabaʿ ,'followed') is irrelevant in this context. Moreover, the substitutes here give different meanings and are also used differently, as some require different prepositions. Here, the system would suggest the substitute3 ***"شَائِعٌ" (šāʾiʿun, 'common')*** as a suitable replacement over "مَعْمُول " (maʿmūl, ' wrought') and "مُتَّبَع " (muttabaʿ ,'followed'), because substitute3 is the only one with lower CEFR level B1 than the target word CEFR level C2, as presented in Table 5.18. This results in the newly generated sentence in Table 5.17. In this example, there are two drawbacks appeared. First, the newly added word ***"شَائِعٌ" (šāʾiʿun, 'common')*** slightly changes the meaning of the sentence. Secondly, it does not match with the preposition in the original sentence "عَلَيْهَا" (ʿalayhā, 'as'). In this case, the sentence needs further syntactic modification to fit the new word by removing the unmatched preposition.

Table 5.18 Represents the target "مُتَعَارَفٌ"(mutaʿārafun ,'recognised') and the accepted substitute

	Word	lemma	Gloss	CEFR
Target word	"مُتَعَارَفٌ "(mutaʿārafun ,'recognised')	1_مُتَعارَف	conventional, recognized	C2
Substitute1	"مَعْمُول " (maʿmūl, ' wrought')	1_مَعْمُول	wrought	C2
Substitute2	"مُتَّبَع " (muttabaʿ ,'followed')	1_مُتَّبَع	followed	C2
Substitute3	"شَائِع" (šāʾiʿ, 'common')	1_شائِع	common	B1

Example 8, as presented in Table 5.19, the word "سَبِيلِ" (sabīli , 'way') is replaced by *"طَرِيقِ" (*ṭarīqi ***, 'way')***, based on the translation selection ***Rule 4***.

Table 5.19 Example 8 expresses the application of Rule 4 in replacing the word "سَبِيلِ" (sabīli , 'way') with *"طَرِيقِ" (*ṭarīqi *, 'way')*.

تَطْوِيرُ وَاسْتِخْدَامُ اللَّقَاحَاتِ فِي سَبِيلِ تَأْسِيسِ مَنَاعَة لِلْمَرَضِ	**Original sentence**
taṭwīru wastiḳdāmu allaqāḥāti fī sabīli taʾsīsi manāʿa lilmaraḍi	
[development and use of vaccines ***to*** establish immunity to disease]	
تَطْوِيرُ وَاسْتِخْدَامُ اللَّقَاحَاتِ فِي طَرِيقِ تَأْسِيسِ مَنَاعَة لِلْمَرَضِ	**Generated sentence**
taṭwīru wastiḳdāmu allaqāḥāti fī ṭarīqi taʾsīsi manāʿa lilmaraḍi	
[development and use of vaccines ***to*** establish immunity to disease]	

The word "سَبِيلِ" (sabīli, 'way') in **Example 8** shows a simplification based on the translation. In this situation, the system applies ***Rule 4*** by matching the English translation of the substituted list against the original target word, as illustrated in Table 5.20. As the target word's translation matches the substitute2 and the system would choose *"طَرِيقِ" (*ṭarīqi ***, 'way')*** based on translation. However, the word ***“أَجْل” (’ajl , ‘sake’)*** is a better substitute for the word “سَبِيلِ” (sabīli , ‘way’) in this context, proceeded by the preposition “فِي” (fī , ‘in’). Yet, the word ***“أَجْل” (’ajl , ‘sake’)*** would require a change of the preposition in the sentence form “فِي” (fī , ‘in’) to “مِنْ” (min , ‘of’) to form the right structure for the word***“أَجْل” (’ajl , ‘sake’)***. This could be resolved later in a post-generative model that could check the grammatical structure of the newly simplified sentences.

Table 5.20 Representing the target “سَبِيلِ” (sabīli , ‘way’) and the accepted substitute

	Word	lemma	Gloss
Target word	“سَبِيلِ” (sabīli , ‘way’)	سَبِيل_1	way; road
Substitute1	“أَجْل” (’ajl , ‘sake’)	أَجْل_1	for_sake_of; because_of
Substitute2	“طَرِيقِ” (ṭarīqi , 'way')	طَرِيق_1	road; way

Example 9, as presented in Table 5.21, the word “وُجُوبَ” (wujūba, ‘obligatory’) is replaced by “ضَرُورَةَ” (ḍarūrat, ‘necessity’) based on Rule 3 CEFR level difficulty limitation.

Table 5.21 Example 9 applying *Rule 3* to select the simplest word match from the generated list

لَكِنَّ كَثِيرًا مِنْ الِاوّضَاعِ تَتَطَلَّبُ مِنْ الْقَاضِي اوْ هَيْئَةِ الْمَحْكَمَةِ وُجُوبَ تَحْدِيدِ الْحُقُوقِ	**Original sentence**
lakinna kaṯīran min aliāwwiḍāʿi tataṭallabu min alqāḍī aw hayʾati almaḥkamati wujūba taḥdīdi alḥuqūqi	
[but many situations require the judge or the court to determine the rights necessarily]	
لَكِنَّ كَثِيرًا مِنْ الِاوّضَاعِ تَتَطَلَّبُ مِنْ الْقَاضِي اوْ هَيْئَةِ الْمَحْكَمَةِ ضَرُورَةَ تَحْدِيدِ الْحُقُوقِ	**Simplified sentence**
lakinna kaṯīran min aliāwwiḍāʿi tataṭallabu min alqāḍī aw hayʾati almaḥkamati ḍarūrata taḥdīdi alḥuqūqi	
[but many situations require the judge or the court to determine the rights necessarily]	

The target word “وُجُوبَ” (wujūba , ‘obligatory’) in Example 9 is another context where the new substitutes share the same English possible transitions, yet the

CEFR level determines the selection. This situation gives more confidence in the order of the rules, as applying Rule 3 by checking the word CEFR level saves the system another step and finalises the selection process. As shown in Table 5.22, the word “ضَرُورَة” (ḍarūrat, ‘necessity’) is assigned with A2; however, the word ***“أشترَاطُ” (Ĩšatarāṭ, necessity)***, which has the same gloss assigned with C2 CEFR level. In this case, the system selects *substitute1* over any word in the prediction list.

Table 5.22 Represent the target “وُجُوبَ” (wujūba , ‘obligatory’) and the accepted substitute

	Word	lemma	Gloss	CEFR
Target word	“وُجُوبَ” (wujūba , ‘obligatory’)	1_وُجُوب	necessity, need, imperative	B2
Substitute1	“ضَرُورَة” (ḍarūrat, necessity)	1_ضَرُورَة	duty, necessity, obligation	A2
Substitute2	“أشتَرَاطُ ” (Ĩšatarāṭ, necessity)	1_اِشْتِراط	duty,necessity,obligation	C2

The framework represented in Figure 5.4 provides pseudocode for the combined classification approach using *fastText* and *Arabic-BERT* while applying the selection rules.

```
Algorithm1 Simplify (sentence $S$, Complex word $w$)
1: for each Complex word w do
2:      p (.|$S$ \{$w$}← fastText($w$\ $S$ )
3:      subs_ft← 5-top-probability (p (.|$S$ \{w}))
4:      Replace $w$ of $S$ into [MASK] as $\hat{S}$
5:      Concatenate $S$ and $\hat{S}$ using [CLS] and [SEP]
6:      p (.|$S$, $\hat{S}$ \{$w$}← ArabicBERT($S$, $\hat{S}$)
7:      subs ←5-top-probability (p (.|$S$, $\hat{S}$ \{$w$}))
8: end for
9: ŵs ←Ø
10: for each substitute sub ϵ subs do
11:     If subs [0] == [UNK]→ Goto 17
12:     elseIf lemma (sub ) == lemma($w$) ←ŵs add $w$
13:     elseIf lemma (sub ) ≠ lemma($w$)
14:         If CEFR(sub)< CEFR ($w$) ←ŵs add sub
15:         elseIf trans(sub)== trans($w$) ←ŵs add sub
```

```
16: end for
17: for each substitute sub-ft ϵ subs-ft repeat 12:15
18: end for
19: all_ranks ← Ø
20: for each f ϵ subs ∩ subs_ft do
21:     ranks ← rank_numbers ( f )
22:     all_ranks ← all_ranks ∪ ranks
23: end for
24: avg_rank ← average (all_ranks)
25: best ←argmax (avg_rank)
26: $\hat{w}$s add best
27: Return $\hat{w}$s
```

Figure 5.4 LS classification approach simplify (sentence *S*, Complex word *w*)

5.2. Method Two - Generative approach

In the generative approach, the SS is considered a translation task, in which the translation is done within the same language from a complex sentence as the source to a simplified sentence as the target (Zhu et al., 2010). This perspective suggests that the SS generative model can be implemented using machine translation (MT) and monolingual text-to-text generation techniques. As such, it combines all LS steps into a single process, which learns how to generate the simple version from the complex sentence. For this purpose, use of the recent advances in neural machine translation (NMT) and BERT-like pre-trained transformer to perform a sequence-to-sequence (Seq2Seq) algorithm. This section will first introduce an attempt to compile a parallel complex/simple corpus to be undertaken in this approach. Secondly, it will present the generative approach steps and primary results.

5.2.1. Monolingual parallel corpus

At this stage, mining a monolingual parallel corpus of complex/simple sentences for Arabic was essential as a prerequisite for Arabic ATS. It is a long-running arduous task to manually build such a resource and time-consuming as well. Then the direction was to adopt automatically extracting sentences-to-sentence parallel pairs containing the same linguistic information and differ in their complexity level. Hence, the proposed approach relies on considering different methodologies used in the extraction of bilingual parallel sentences from corpora dedicated to the MT task. This extraction has been done using parallel or comparable corpora through different approaches such as word-embedding-based, machine translation based and deep-learning-based approaches (Maskara and Bhattacharyya, 2019), following the methodology introduced by Brunato et al. (2016). They managed to acquire PaCCSS–IT, a parallel corpus of complex–simple aligned sentences for Italian. Their methodology as mentioned earlier in chapter 3 section (3.2.3.4 parallel dataset mining) was concentrating on sentence extraction with structural transformations rather than lexical ones, compiled from a very large web corpus.

5.2.1.1. Corpus compilation methodology

The main technique here in mining a monolingual parallel sentence corpus relies on measuring the similarity between sentences' clusters. To find the best sentence pair that shares the same meaning yet is not identical. Extracting the sentence pairs from a large dataset. This involves four processes: 1) *Sentence clustering* for each dataset group sentences that share similar meaning using different methods; 2) *Sentence ranking* in each cluster by applying different similarity metrics to arrange the set according to their semantic similarity; 3) *Readability classification* is a standalone process that involves classifying each sentence using the 3-way Arabic-Bert readability classifier (introduced in Chapter4); 4) *Final similarity score* is the process in which avoiding sentence pairs having similar words but different meaning and identical sentences pairs.

5.2.1.2. Monolingual corpus dataset

The dataset selected here was a combination of three corpora; the Arabic Wikipedia corpus, the Arabic web Snapshot and, the ALTIC[44] corpus; in order to include as many genres as possible.

[44] The corpus consists of 10 genre-classified Arabic articles from the Arabic Wikipedia diacratised and annotated with a Named Entity schema.

To pre-process the dataset, I applied a 3-way Arabic-Bert Classifier. This classifier uses a pre-trained BERT model to classify the language proficiency level of each sentence in the dataset according to the CEFR scale. According to Table 5.23 and Figure 5.5, the distribution of CEFR levels across the dataset is skewed towards the B and C levels, with very few sentences classified as A-level. This suggests that the majority of the sentences in the dataset are of average readability, which is consistent with the normal distribution of sentences in written language. Later steps utilized this sentence corpus to extract the optimal sentence pairs based on various similarity metrics.

Table 5.23 Dataset used to compile the parallel Arabic corpus as classified according to the 3-way Arabic-Bert Classifier

	Level A	**Level B**	**Level C**	**Sentences**	**Words**
ALTIC	16724	136000	13163	165615	3562318
Wiki	319814	1003836	1362488	2686138	702651740
Snap Shot	470573	1606456	728592	2805376	71623097
Total	807,111	2,746,292	2,104,243	5,657,129	777,837,155

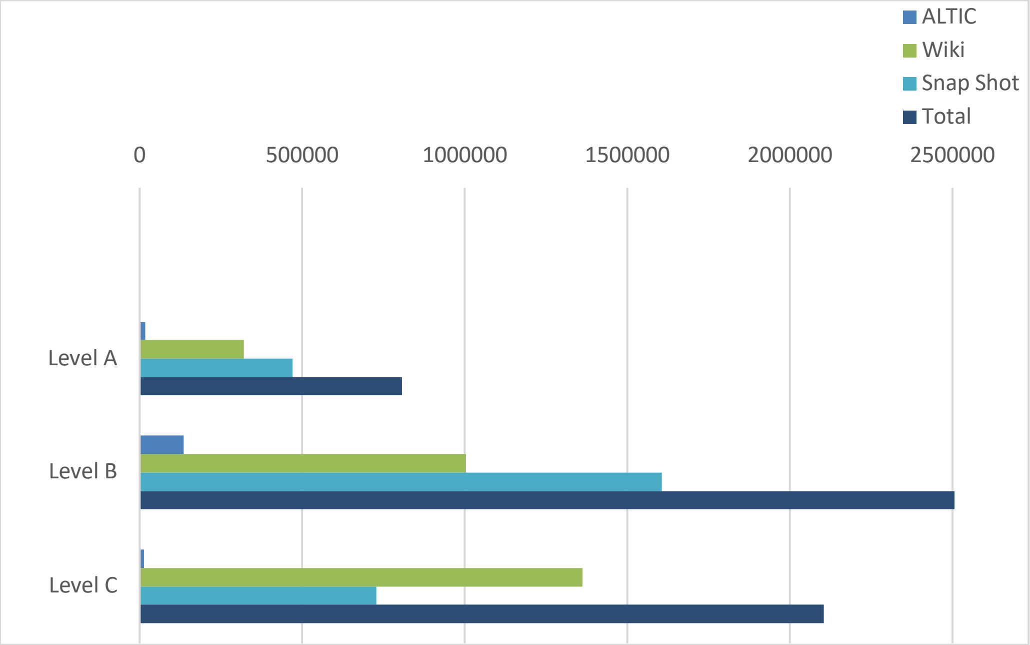


Figure 5.5 The CEFR 3-levels represented in each corpus

5.2.1.3. Sentence clustering

Sentence clustering is the grouping of semantically similar sentences together and this is obtained by calculating the distance between the sentence embedding vectors. This process is composed of six sub-processes: 1) Named Entity (NE) masking; 2) Sentence embedding generation; 3) Sentence clustering; 4) Readability classification; 5) Rank cluster candidate; and 6) Final similarity score. These steps are explained in the following sections.

1) NE-masking

Named Entity masking is a preprocessing task that involves masking or removing Named Entities (NEs) from a sentence while retaining their position in the sentence. NE masking is an essential sub-process to limit confusion in sentence clustering and classification. This is done to reduce the impact of NEs on sentence clustering and classification. Both clustering and classification can be mistaken affected by the vector representation of the NE in the sentence. Therefore, padding NEs reinforces the clustering process to group sentences that share similar meanings regardless of NEs occurred. The task here was to mask the NE in each sentence before performing any sentence clustering. Masking here means removing the word yet keeping the word slot in the sentence. Hence, to perform this task any NE in each sentence were masked using an available Arabic Named Entity recognition tool[45]. Here, I employed specialized Arabic NE tool over MADAMIRA tool because it is faster and easier to implement can be beneficial for NE-masking as it reduces the processing time and complexity of the task. The process was to pad any recognised NE with the label [Masked]. For example, as shown in Table 5.24, a sentence extracted from the "Saqq Al-Bambuu” corpus applying the NE masking task using the Arabic NE recogniser.

[45] https://github.com/EmnamoR/Arabic-named-entity-recognition

Table 5.24 The process of masking NE in a sentence.

Sentence	أَحْبَطَ الْجَمِيعُ فِي الدَّقِيقَةِ الْ 61 عِنْدَمَا سَجَّلَ يُوسُفُ نَاصِرٍ هَدَفًا لِصَالِحِ مُنْتَخَبِ الْكُوَيْتِ
Masked NEs	أَحْبَطَ الْجَمِيعُ فِي الدَّقِيقَةِ الْ 61 عِنْدَمَا سَجَّلَ [Masked] [Masked] هَدَفًا لِصَالِحِ مُنْتَخَبِ [Masked]
Transliteration	ʾaḥbaṭa aljamīʿu fī addaqīqati al 61 ʿindamā sajjala yūsufu nāṣirin hadafan liṣāliḥi muntaḳabi alkuwayti
Translation	Everyone was disappointed in the 61 minutes when Yusef Nasser scored a goal for Kuwait.

1) Sentence embedding generation

This task involves a representation of each sentence in the corpus with an embedding vector to enable the ML approaches searching and group sentences. For this purpose, adopting different sentence embedding as Word2Vec, XLM-R (1024 vector and 2 vectors) and Arabic-BERT.

2) Sentence Clustering

In this stage, semantic similarity was used to cluster sentences based on a single vector of fixed dimension (in this case, 768 for Arabic-BERT). The goal was to cluster sentences into groups and find sentence pairs with different readability levels. The clustering process required that sentences in each cluster share some lemmas in any order but cannot share all lemmas, as this would result in identical sentences.

Several techniques from the scikit-learn python library were experimented with to identify the best sentence clustering method. K-nearest and Agglomerative Clustering modules were not suitable for the large dataset with BERT vector representation as they require significant computational resources. To address this, mini_batch_k_means module was applied, which splits the data into batches before clustering. Mean_shift module, which uses an algorithm to shift each point in the data set until it reaches the top of its nearest surface peak, was also experimented with. Additionally, vector dimension reduction into 2D representation was attempted using PCA (principle component analysis). According to the findings, it was observed that the 'mini_batch_k_means' algorithm resulted in the most optimal clustering results.

3) Readability classification (Linguistic Complexity measurement)

Assigning a readability level for each sentence in each cluster. This process involves using an Arabic-BERT readability 3-way classifier. Applying this classifier allows measuring the complexity of each sentence in the cluster and splitting the cluster into two sub-clusters each one containing either the complex sentence labelled as C (complex) or the simple ones labelled as A/B (easy or intermediate) organized in ranked order according to complexity measures.

4) Rank the Candidates

In this stage, a ranking measure was applied to identify the most difficult sentences classified as C in the previous step. This was accomplished by using similarity metrics to determine the semantic similarity between all sentences in a cluster.

To accomplish this, all sentences within the matched clusters were paired and ranked for similarity by calculating the cosine distance between the sentence vectors. Cosine similarity was calculated within each sentence cluster as a group using Arabic-BERT sentence vectorization. This process resulted in the identification of the top 10 most similar simple sentences for each complex sentence. This approach was based on work by (Bouamor and Sajjad, 2018).

5) Final similarity score

At this step, after applying the previous steps, there is a defined parallel sentence pair. However, some of these sentence pairs may have similar words and located in the same vector space but with a different meaning, as such sentences have an antonym word. To eliminate such pairs among the dataset, apply an approach to capture if a candidate pair has highly similar words but has unparalleled parts. his approach was based on work by (Hangya and Fraser, 2019).

This involves applying a word alignment algorithm (in this case, the Eflomal[46] Bayesian HMM model) to align the words in the paired sentences in the same

[46] https://github.com/robertostling/eflomal

vector space. To score the similarity between the sentence pairs, you are using a method that considers both the presence of parallel segments and the alignment scores of the full sentence. If there are no parallel segments aligned between the sentences, the score is set to 0. Otherwise, the average word alignment scores of the full sentence are calculated and weighted by the ratio between the length of the longest complex parallel segment and that of the full sentence (Östling and Tiedemann, 2016).

Overall, this approach can be useful for measuring the similarity between parallel Arabic sentences However, it's important to note that the quality and accuracy of the word alignment algorithm can have a significant impact on the results.

Reaching this stage, to process a corpus of parallel complex-simple sentence pairs in Arabic following these steps:

1. Annotate each sentence pair with three features: cosine similarity, sentence readability, and word alignment similarity score.
2. Remove sentence pairs that are either identical or indicate a difference in meaning.
3. Rank all remaining candidate pairs in the corpus based on the three annotated features.
4. Select the top list of sentence pairs that have a semantic similarity of at least 85%.
5. Consider this set of sentence pairs as the parallel complex-simple sentence pairs Arabic corpus.

Overall, these steps are meant to create a high-quality corpus of parallel sentence pairs that can be used for various NLP tasks, such as machine translation, text simplification, and language learning. By filtering out identical or meaning-differentiating sentence pairs and selecting those with high semantic similarity, the resulting corpus is expected to have a high level of alignment between the complex and simple sentences, which can facilitate the development of effective TS models.

5.2.1.4. Experiment one: Proof of concept

A pilot sentence clustering and semantic similarity experiments have been performed to select the best sentence embedding representation and clustering method for compiling the Arabic parallel corpus (Al-Raisi et al., 2018). This pilot study was carried out on "*Saqq Al-Bambuu*" corpus on a set of 4,594 parallel sentences using two methods for clustering: 1) K-means cluster approach that uses the sum of distances of samples to their closest cluster centre to cluster the sentence; 2) faiss[47] semantic similarity search, which indexes the sentences' vectors according to their similarity (Johnson et al., 2017).

As illustrated in Table 5.25, the results on sentence clustering using faiss similarity search applied both cosine similarity and Euclidian distance with precision (true positive) on the 4,594 parallel sentences. In principle the ideal number of clustering is 2297, which indicates two sentences per cluster. However, in clustering related sentences could be assigned to one cluster. Therefore, a large number of clusters indicate a better classification of the semantically similar sentences. According to the information presented in Table 5.25, it can be inferred that the utilization of word2vec in sentences resulted in a superior clustering performance, as evidenced by the significantly greater number of produced clusters. But this assumption, by relying only on the number of clusters, is superficial and this required a manual error analysis to confirm which vectorisation method is reliable for the task. Manually revising the first 100 sentences clustering indicated that faiss was the fastest method yet the accuracy drops as the sentence pairs do not appear in the 10 nearest sentences. However, word2vec initially performed better, the rest of the sentence pairs do not exist in the 10 k-nearest. As for ArabicBERT, the majority of sentence pairs could be found in the 5 k-nearest neighbours. These results suggested using ArabicBERT to identify the similarity between sentences and then use the K-means clustering approach to cluster the sentences with the same context together.

[47] https://github.com/facebookresearch/faiss

Table 5.25 Number of sentence clusters using various sentence embedding representation

Sentence representation	Cosine Cluster	Euclidean Cluster
Word2vec	1058	1119
ArabicBert	477	423
XLM-R1024	66	21
XLM-R2	43	15

For example, as illustrated in Table 5.26 presenting the sentence clustering performance for 10 sentences from *Saqq Al-Bambuu* applying word2vec and ArabicBERT. The overall results indicated that ArabicBERT's clustering compromises the semantic meaning of the sentences rather than word2vec clustering, which reflects the matching of word occurrences in sentences in a cluster represented by numbers as clusters.

Table 5.26 Clustering performance using both word2vec and ArabicBERT

ArabicBERT	Word2Vec	Sentence	N
1	0	يجب عليكم دفع مبلغ من المال إلى الوكيل.	1
0	1	المترجم إبراهيم سلام، يعمل في حقل الترجمة.	2
3	2	همس باسم حفيدته عند أذن آيدا أذن آيدا.	3
0	1	المترجم ابراهيم سلام، يعمل في مجال الترجمة.	4
3	2	ميرلا همس باسم حفيدته عند أذن آيدا.	5
2	0	يستوجب عليكم دفع مبلغ من المال إلى الوكيل.	6
4	3	والدتي، لتضمن لهم حياة ليس بالضرورة أن تكون كريمة، بل حياة وحسب، بعد أن ضاقت بهم السبل.	7
4	3	والدي، لتكون لهم حياة ليس بالضرورة أن تكون كريمة، بل حياة فقط.	8
2	4	صُعق الجميع حين سمعوا الرقم من الجار، فلم يكن بمقدور العائلة توفير مثل هذا المبلغ.	9
2	4	لم يكن باستطاعة العائلة دفع مثل هذا المبل.	10

On the one hand, AraBERT clustering for these ten sentences as illustrated in Figure 5.6 using k-means clustering, the Kmeans score= -6173742289.75 and the silhouette score= 0.5367579893384077 (the mean Silhouette Coefficient of all samples). According to the figure, it can be seen that the 10 sentences were divided into 5 clusters, and among these clusters, three of them correctly identified two parallel sentences that were represented by the numbers (0, 3, 4). In contrast, one cluster consisted of one sentence (cluster number 1) and the last cluster consists of three sentences (cluster number 2). Although in cluster 2, these three sentences indicate similar semantic meanings in some way as the three sentences share the meaning of not being able to pay (’دَفْعَ ’, dafʿa) or provide (‘تَوْفِيرُ’, tawfīru) money (’مَبْلَغٍ مِنْ الْمَالِ ’, mablaġin min almāli).

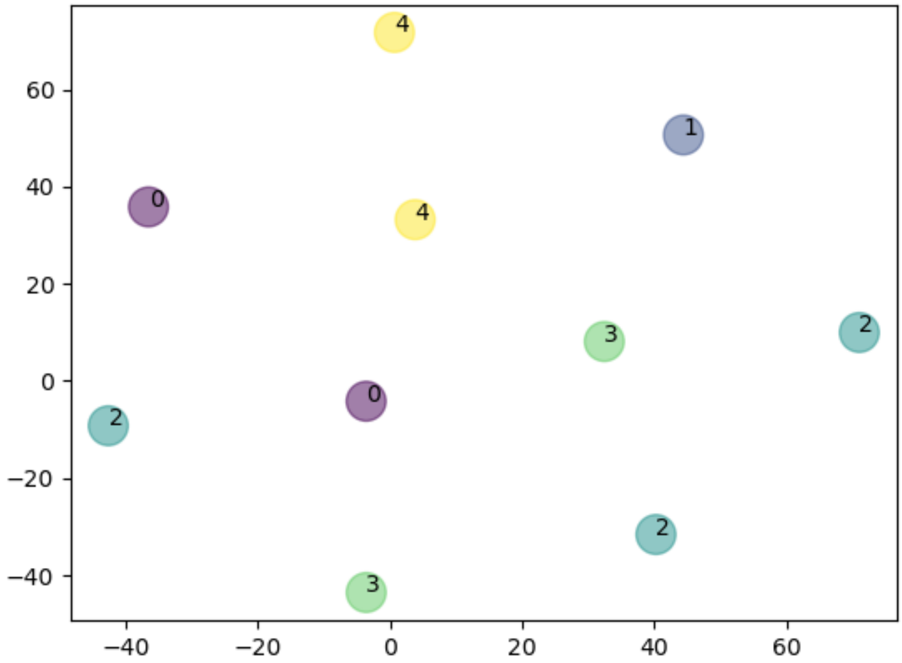


Figure 5.6 Arabic-BERT sentence clustering

On the other hand, for word2vec clustering as illustrated in Figure 5.7 the Kmeans score was 1.1868E-05 while the silhouette score was 0.43699676. In this case, I classified the ten sentences into five clusters of which each has two parallel sentences. Figure 5.8 presenting the hierarchical clustering using Euclidean distance shows the right five clusters of the parallel sentences.

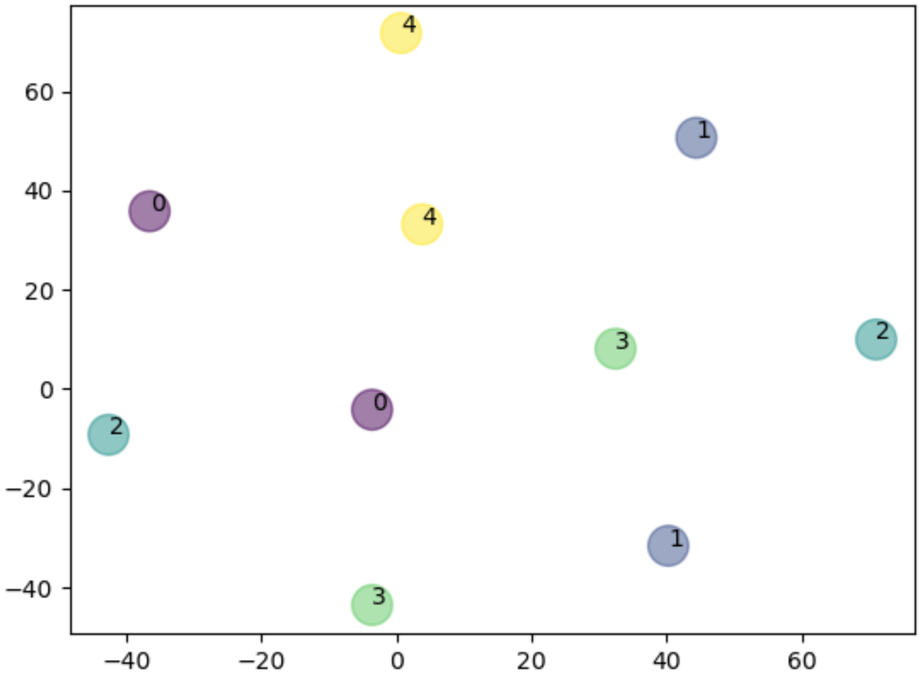


Figure 5.7 Word2vec sentence clustering

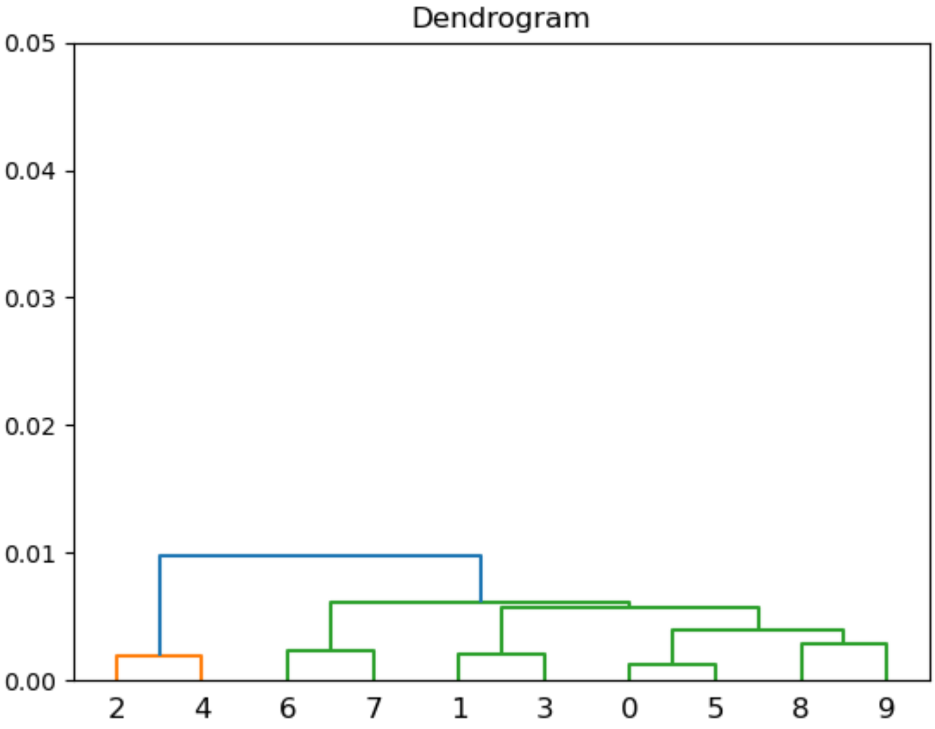


Figure 5.8 Hierarchical sentence clustering by Euclidean distance

5.2.1.5. Experiment two: Corpus clustering

Hence, it is proved prior in the pilot experiment the feasibility of the adopted approach. As illustrated previously in Table 5.1, all previous steps on the dataset compiled were applied (see section 5.1.1.1 Monolingual corpus dataset). First, the NE masking is applied, followed by generating sentence embeddings and performing the clustering.

Unfortunately, the clustering results measuring the cosine similarity in each cluster was disappointing. The manual analysis of random 50 cluster showed that none of the sentences in these clusters were meaning related. Table 5.27 presents an example of cluster number 5 which consists of 7 unrelated sentences which was manually identified as unrelated semantic cluster.

Table 5.27 A sentence cluster extracted from the dataset corpus measuring the Cosine Similarity

Cluster	Sentence	Cosine Similarity
5	بَلَغَتْ الْجُورْجِيَّةُ [Masked]ذُرْوَتُهَا فِي الْقَرْنِ الثَّانِيَ عَشَرَ الَى بِدَايَةِ الْقَرْنِ الثَّالِثَ عَشَرَ . balaġat aljūrjiyyatu [Masked] ḏurwatuhā fī alqarni aṯṯāniya ʿašara alā bidāyati alqarni aṯṯāliṯa ʿašara Georgian [Masked] reached its peak in the twelfth century to the beginning of the thirteenth century.	1.0
5	قَانُونِ [Masked] الثَّالِثُ qānūni [Masked] aṯṯāliṯu 3rd [Masked] Law	0.80
5	وَمِن امْثَلْتِهَا wamina amṯaltihā As an example	0.65
5	طِبْقًا لِهَذِهِ النَّظَرِيَّةِ فَالنِّظَامُ الثَّانِي هُوَ الْهَدَفُ الصَّحِيحُ مِنْ دِرَاسَةِ عُلَمَاءِ النَّفْسِ وَالنِّظَامِ الثَّالِثِ [Masked] وَالنِّظَامُ الرَّابِعُ [Masked] الِانْثُرُوبُولُوجْيَا الثَّقَافِيَّةُ ṭibqan lihaḏihi annaḓariyyati fanniḓāmu aṯṯānī huwa alhadafu aṣṣaḥīḥu min dirāsati ʿulamāʾi annafsi wanniḓāmi aṯṯāliṯi [Masked] wanniḓāmu arrābiʿu [Masked] aliānṯurūbūlūjyā aṯṯaqāfiyyatu According to this theory, the second system is the right goal of the study of psychologists, the third system [masked] sociology, and the fourth system [masked] cultural anthropology.	0.43
5	تَنْدَرِجُ مُعْظَمُ الِانْظِمَةِ الْحَاكِمَةِ وَمُفَكِّرِيهَا ضِمْنَ هَذَا التَّيَّارِ مَعَ بَعْضِ الِاسْتِثْنَاءَاتِ مِثْلَ [Masked] . tandariju muʿḓamu aliānḓimati alḥākimati wamufakkirīhā ḍimna haḏā attayyāri maʿa baʿḍi aliāstiṯnāʾāti miṯla [Masked] . Most of the ruling regimes and their thinkers fall within this trend, with some exceptions, such as [Masked].	0.39
5	نَجِدُهُنَّ اكْثُرَ حَسَاسِيَةً لِبَعْضِ الرَّوَايِحِ رَغْمَ انْ حَاسَّةً [Masked] لَدَيْهِنَّ تَقِلُّ اثِّنَاءَ هَذِهِ الْفَتْرَةِ . najiduhunna akṯura ḥasāsiyatan libaʿḍi arrawāyiḥi raġma an ḥāssatan [Masked] ladayhinna taqillu aṯṯināʾa haḏihi alfatrati. We find that they are more sensitive to some smells, although their sense of [Masked] decreases during this period.	0.33
	حُقْنٌ بِنَسْلَيْنِ [Masked] هِيَ الْعِلَاجُ الْوَحِيدُ ذُو التَّاثِيرِ الْمُوَثَّقِ اثِّنَاءَ فَتْرَةِ الْحَمْلِ .	0.25

Cluster	Sentence	Cosine Similarity
5	ḥuqnun binaslayni [Masked] hiya alʿilāju alwaḥīdu ḏū attāṯīri almuwaṯṯaqi aṯṯināʾa fatrati alḥamli	
	Penicillin injections [Masked] are the only treatment with documented effect during pregnancy.	

In this case, the resulted clustered dataset could not be used to perform the further steps as readability classification, sentence similarity ranking in order to select the subset of parallel semantic similarity complexity annotated sentences. These findings suggested using the Arabic-parallel corpora that was readily accessible corpus (Al-Raisi et al., 2018). However, since this corpus was originally a translation of English and French texts, a pre-verification procedure is necessary before using it for the intended tasks. The next section will discuss this pre-verification procedure in detail.

5.2.1.6. Arabic-Parallel corpus verification

However, 200 sentences pairs of this corpus were manually verified by two native speakers of Arabic, and the first analysis of the corpus shows numerous, ungrammatical, Arabic sentences in the corpus. These grammatical errors could be as a result of their method in using MT in translating the English section to Arabic (considered as complex) and the French section to Arabic (considered as simple). This consideration as English/Complex – French/Simple is because the average length of Arabic sentences translated from the English section tend to be longer than the ones translated from French. Moreover, in the corpus some of the parallel sentences cannot be considered as parallel simplified pairs as shown in Table 5.28 a parallel sentence pair selected from (Al-Raisi et al., 2018) corpus shows different words in the sentences leads to different meaning. This decision was made because the dataset was specifically designed for the TS task, and it had undergone rigorous annotation and verification processes to ensure its accuracy and reliability.

Table 5.28 A parallel sentence pair selected from (Al-Raisi et al., 2018) corpus shows different meaning

Complex	الْبَرْلَمَانُ ارْتَفَعَ، وَلَاحَظَ دَقِيقَةَ صَمْتٍ
	albarlamānu artafaʿa, walāḥaḏa daqīqata ṣamtin
	Parliament rose, and he noticed a minute of silence
Simple	ارْتَفَعَ الْبَيْتُ وَلَاحَظَ صَمْتَ دَقِيقَةٍ وَاحِدَةٍ
	artafaʿa albaytu walāḥaḏa ṣamta daqīqatin wāḥidatin
	The house rose and noticed a one-minute silence

The first verification stage was applying an automatic verification of sentence pair similarity using the sentence similarity classifier based on "Saqq Al-Bambuu" as a gold standard corpus. This similarity classifier was trained on Arabic-BERT achieving 0.98 F-1 measure. Out of 100,000 sentence pairs, 53,235 sentences pairs were classified as being semantically similar.

The second verification stage was a manual verification for 80 random selections of approved semantically similar sentences. The results shown that all sentence pairs were highly similar, however, thirty of these pairs were reversed in position regarding the complexity versus simple data side classification.

The third verification stage involved using the results from the previous manual stage and applying the 3-Way Classifier and Binary Classifier on 30 examples to measure complexity and determine which version is simpler. However, the classifiers were unable to correctly classify the sentences as only two out of 30 pairs were classified as simple, while the rest were classified as complex. Despite this outcome, the classifiers were applied to the full dataset to verify the data division. The results indicated that 2010 sentence pairs needed to be reversed in position, meaning that complex instances should become simple and vice versa. This finding suggests that the initial observation of the data division was incorrect and that the classifiers could be used to improve the accuracy of the dataset.

Based on the results of the corpus verification, it was concluded that building a whole seq2seq model based on MT sentences and readability classification was not convincing. This was due to the fact that readability was wrongly classified, indicating that the MT sentences were not reliable for this task. As a result, I decided to rely on the dataset used in this chapter, which is the parallel developed corpus of the Arabic novel "Saqq Al-Bambuu", as described in chapter 4.

5.2.2. Methodology: Generative approach

Here, I employed a Seq2Seq approach. First, use the recent NMT techniques, the OpenNMT framework (Klein et al., 2017) to allow a comparison with previous models. Second, adopting T5 and deploy a "multilingual Text-to-Text Transfer Transformer", Multilingual T5, mT5 (Xue et al., 2021). Trained on "Saqq Al-Bambuu" and parallel sentence pairs selected from Al-Raisi et al. (2018) [as explained in section 5.2.1.6]. Considering the multilingual capabilities of mT5 and the suitability of the Seq2Seq format for language generation. This gives it the flexibility to perform any NLP task without having to modify the model architecture. This experiment employs the 'MT5-For-Conditional-Generation' class that is used for language generation as in Figure 5.9. Training a TS model makes use of the "Saqq Al-Bambuu" parallel sentences corpus over the *mT5- base model*[48]. This approach was tested in a *Python3.8* environment using other toolkits such as *NLTK* and *Scikit–learn*. Our sentence corpus was randomly split into 80% for training and 20% for testing

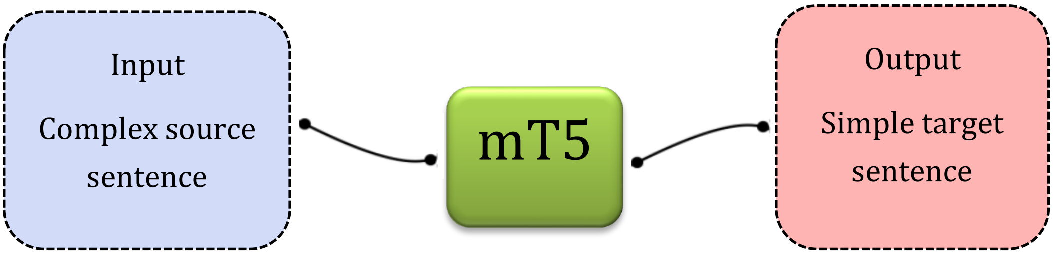


Figure 5.9 Using an mT5 model trained over Saqq Al-Bambu corpus as a translation model from the source complex sentence to the simple target counterpart

[48] google/mt5-base, is available through the Huggingface repository, https://huggingface.co/google/mt5-base

5.3. Evaluation

Likewise, most TS evaluation approaches have been driven from other similar NLP research areas. Various evaluation methods have been applied across research to measure the three main aspects of the newly generated text as presented in the literature review. These aspects are: i) fluency, referring to the grammatically well-formedness and structure simplicity; ii) adequacy, meaning preservation; iii) simplicity, meaning the text is more readable.

Both classification and generative methods were evaluated on the same test dataset containing 299 randomly chosen sentences excluded from training. Both automatic and manual evaluations were employed to compare both approaches. The rest of the chapter is focused on explaining the evaluation and error analysis of the classification and generative approaches.

5.3.1. Automatic evaluation

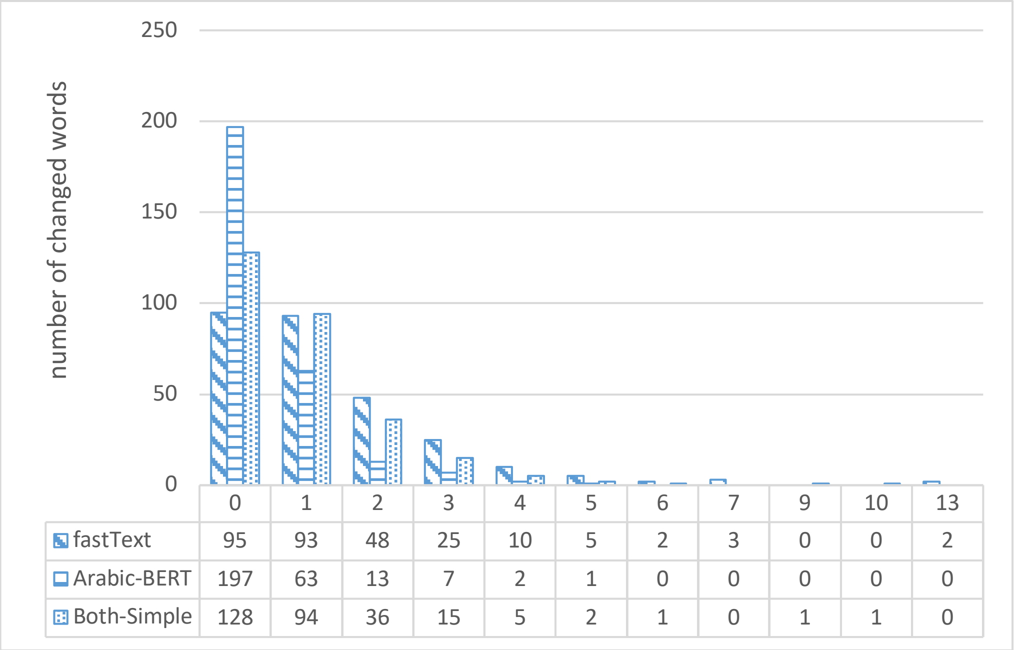
Automatic evaluation of TS models I applied BERTScore evaluation method and BLEU matrics as well. BERTScore allows the use of different pre-trained transformer models by applying baseline rescaling to adjust the output scores. This allowed determining the performance of different Arabic-language trained BERT models: (i) the default in multilingual BERT (mBERT) (Devlin et al., 2018) that is based on the selected language, which is Arabic in this case; (ii) ARBERT[49] (Abdul-Mageed et al., 2021); (iii) AraBERTv0.2-base model[50] (Antoun et al., 2020). However, AraBERT has been trained on a larger corpus than ARBERT; the latter uses WordPiece tokeniser, as illustrated before. Whereas AraBERT relies on SentencePiece tokeniser that uses spaces as word boundaries. These two parameters reflected in BERTScore metrics are carefully measured. Whereas, using BLEU allowed to compare the performance with other TS models. Furthermore, using multiple evaluation metrics can provide a more comprehensive understanding of the performance of the TS models.

Classification approach - Automatic Evaluation The classification system produced three simple versions of the target sentence using BERT-alone, fastText-alone, and combined versions. This automatic evaluation was applied to

[49] https://github.com/UBC-NLP/arbert
[50] https://huggingface.co/aubmindlab/bert-base-arabert

compare different BERT model resolutions of these sentences, as represented in Table 5.29. Figure 5.10 represents the number of changed words performed by each classification model. These primary results suggest that using fastText-alone performs unneeded simplification resulting in lower F-1. In contrast, a higher F-1 measure in Arabic-BERT-alone generated sentences suggests that using BERT eliminates necessary changes. At the same time, the combination of both tools' suggestions enhances the substitution ranking and choice process, which eliminates unnecessary changes and enhances performance. In this case, combined produced sentences achieved P 0.97, R 0.97, and F-1 0.97 using ARBERT.

These primary results showed that using fastText-alone or Arabic-Bert-alone either eliminates necessary changes or performs unneeded simplification using fastText. While the combination of both simplification suggestions enhances the substitution ranking and choice process.



	0	1	2	3	4	5	6	7	9	10	13
▧ fastText	95	93	48	25	10	5	2	3	0	0	2
□ Arabic-BERT	197	63	13	7	2	1	0	0	0	0	0
▦ Both-Simple	128	94	36	15	5	2	1	0	1	1	0

Figure 5.10 Number of changed words using fastText-alone, Arabic-Bert-alone and combining both in Both-simple

Table 5.29 Precision, recall, and F1 measures using BERTScore with different transformer models

Classification	**P**	**R**	**F1**
Default based on the language			
Target/fastText	0.962	0.966	0.964
Target /BERT	0.991	0.990	**0.990**
Target / Simple	0.974	0.975	0.975
UBC-NLP/ARBERT , num_layers= 9			
Target/fastText	0.958	0.960	0.959
Target /BERT	0.990	0.991	**0.990**
Target / Simple	0.976	0.976	0.978
bert-base-arabert , num_layers= 9			
Target/fastText	0.962	0.963	0.963
Target /BERT	0.989	0.989	0.989
Target / Simple	0.975	0.976	0.976

Generative –T5	**P**	**R**	**F1**
Default based on the language			
Original/Target	0.889	0.838	**0.862**
Generated/Original	0.806	0.725	0.762
Generated/ Target	0.754	0.723	0.736
UBC-NLP/ARBERT , num_layers= 9			
Original/Target	0.840	0.754	0.790
Generated/Original	0.647	0.529	0.573
Generated/ Target	0.570	0.524	0.538
bert-base-arabert, num_layers= 9			
Original/Target	0.879	0.823	0.848
Generated/Original	0.787	0.693	0.734
Generated/ Target	0.723	0.686	0.701

Generative T5 Approach - Automatic Evaluation testing the 299 sentences for evaluating the generated simplified sequences compared to the original and target simple sentences. Using three measures as presented in Table 5.29 shows the higher F1 score achieved is 0.862. Moreover, T5 generative approach achieved the highest BLEU score across the models with 20.372 score.

Original/Target, considering it as a reference to the mT5 system.

1. Generated/Original, comparing the newly generated sentence with the original complex sentence.
2. Generated/Target, comparing the newly generated sentence with the target simple sentence.

To further illustrate these three models' performance, Figure 5.11 represents the distribution of F-1 across the testing data instances using different Arabic BERT models. The default model F-1 plots skewed towards the right, reflecting strong similarity across the three parallel sentences (Original/Target/Generated). Whereas AraBERT plots Original/Target and Generated/Original skewed to the left, indicating less similarity across the data. While ARBERT's plots represent a normal distribution representing a more accurate similarity measure in the data.

These findings suggest ARBERT that applying a WordPiece sentence tokeniser BERT model performed better in sentence representation.

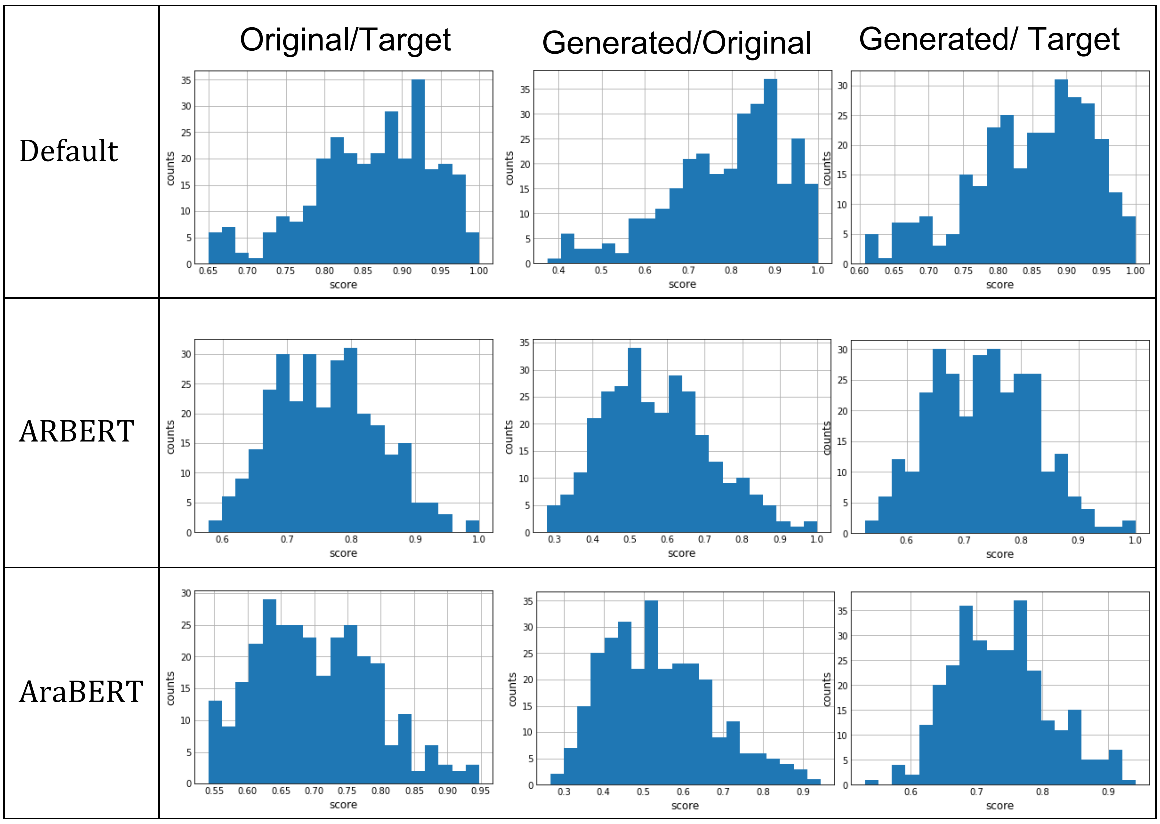


Figure 5.11 The F1 scores for each sentence pair are more spread out, making it easy to compare different methods.

Generative OpenNMT Approach-Automatic Evaluation

Following the same procedures as in evaluation of the generative-T5 approach. Table 5.30 illustrates the results comparing models trained on "Saqq al-Bambuu" alone , parallel sentence pairs selected from Al-Raisi et al. (2018) alone and a combined dataset of both. The combination of the data was a way to make bigger dataset as NMT requires a lot of data to train a good model.

The initial results shows that the use of bigger dataset improve the system accuracy. The Saqq al-Bambuu only model resulted in BERT F1 score of 0.690 and BLEU score of 0.65. Whereas Al-Raisi et al. (2018) parallel sentences reached F1 score of 0.790 and BLEU score of 8.84. However, these results are lower than the F1 score of 0.86 achieved by the generative BERT approach.

Table 5.30 OpenNMT results using BERTScore with different dataset models

BERT model	Generative -OpenNMT	P	R	F1	BLEU
Saqq al-Bambuu					
Default	Generated/ Target	0.696	0.703	0.699	0.651
NLP/ARBERT		0.494	0.513	0.501	
AraBERT		0.668	0.678	0.672	
Parallel dataset					
Default	Generated/ Target	**0.840**	**0.754**	**0.79**	7.623
NLP/ARBERT		0.622	0.617	0.619	
AraBERT		0.744	0.737	o.740	
Combined dataset					
Default	Generated/ Target	0.778	0.769	0.774	8.848
NLP/ARBERT		0.637	0.630	0.633	
AraBERT		0.752	0.746	**0.749**	

5.3.2. Manual evaluation

Classification Approach - Manual Evaluation, a manual analysis of the produced sentences of the combined system, has been performed by the researcher. The results are displayed in Figure 5.12 and Figure 5.13, on a scale of good, useful, a bit useful, and useless simplification. 55% of the new simplified sentences were either good, useful, or a bit useful, as majority. While 45% of the sentences were classified as useless simplification, the complex word was replaced by a more complex word or its antonym. For example, a useful simplification of the combined system as in the following sentence from "Saqq al-Bambuu" as explained before in Example 1:

كُنْتُ **أُحْدِّقُ** فِي الطَّبَقِ وَالصَّمْتِ يَكَادُ يَبْتَلِعُ الْمَكَانَ.

kuntu ʾuḥaddiqu fī aṭṭabaqi waṣṣamti yakādu yabtaliʿu almakāna

[I was ***staring*** at the plate, and the silence almost swallowed up the place.]

In this sentence, the word "أُحْدِّقُ " (ʾuḥaddiqu , 'staring') was replaced by "أَتَأَمَّلُ" ('ata'ammalu ,'muse') which is more frequent and simpler and generates

كُنْتُ **أَتَأَمَّلُ** فِي الطَّبَقِ وَالصَّمْتِ يَكَادُ يَبْتَلِعُ الْمَكَانَ.

kuntu ʾataʾammalu fī aṭṭabaqi waṣṣamti yakādu yabtaliʿu almakāna

[I was ***staring*** at the plate, and the silence almost swallowed up the place.]

Although the new word is simpler, it doesn't reach the exact target word ***"'anḏuru" (أَنْظُرُ, 'look').***

BERTScore also provides a function plot example to support sentence-level visualisation by plotting the pairwise cosine similarity in Figure 5.13.

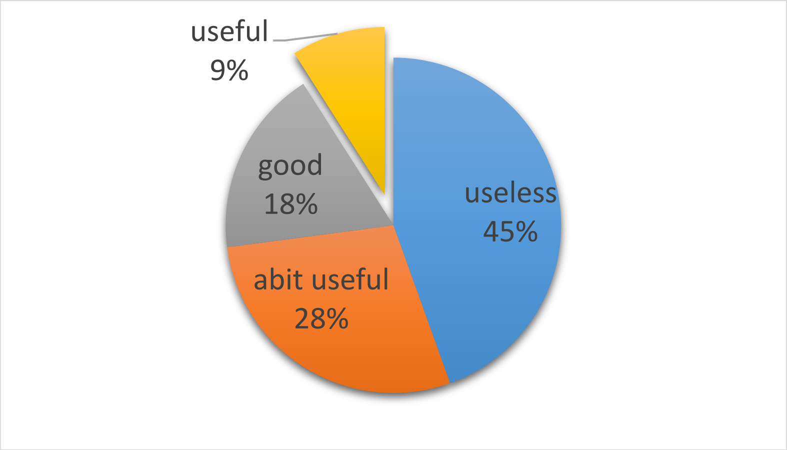
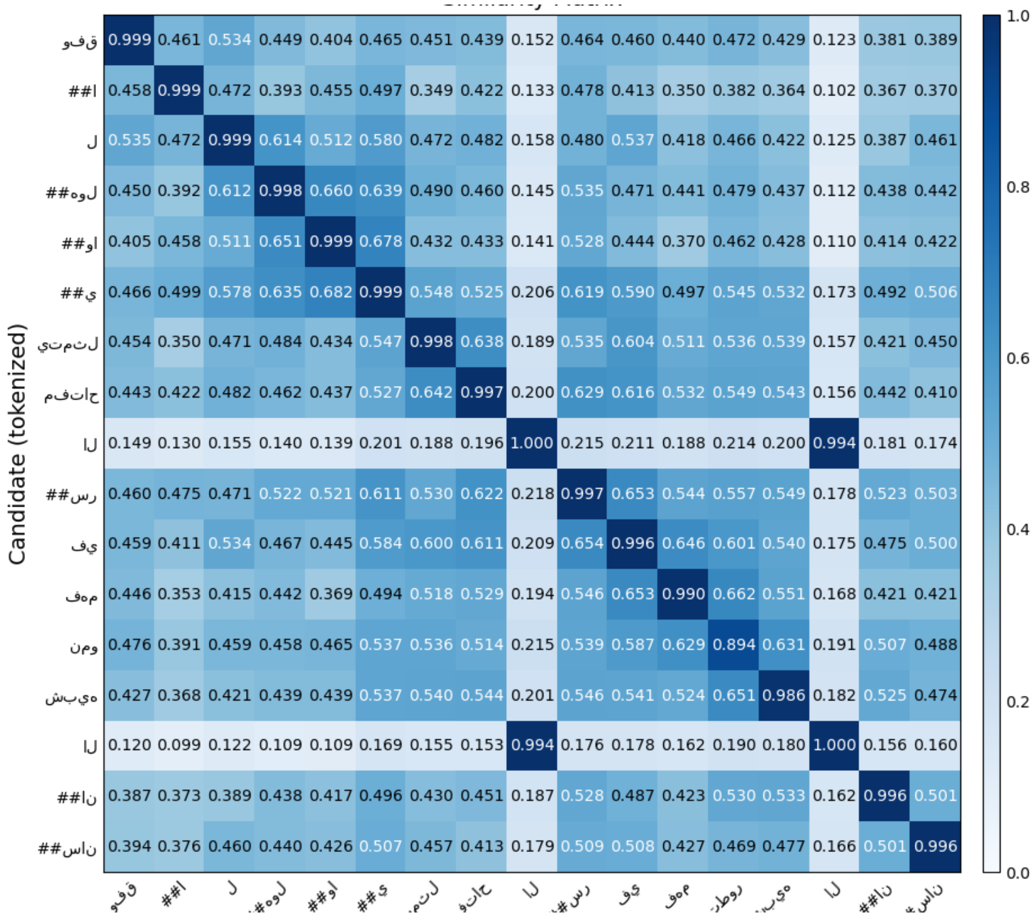


Figure 5.13 Simplified sentences analysis based on the usefulness of the lexical substitution processes.



	قفو	##ا	ل	لوه##	او##	ي##	لتم	حاتف	ال	رس##	يف	مهف	روطت	هيب	ال	ان##	ناس
قفو	0.999	0.461	0.534	0.449	0.404	0.465	0.451	0.439	0.152	0.464	0.460	0.440	0.472	0.429	0.123	0.381	0.389
##ا	0.458	0.999	0.472	0.393	0.455	0.497	0.349	0.422	0.133	0.478	0.413	0.350	0.382	0.364	0.102	0.367	0.370
ل	0.535	0.472	0.999	0.614	0.512	0.580	0.472	0.482	0.158	0.480	0.537	0.418	0.466	0.422	0.125	0.387	0.461
##لوه	0.450	0.392	0.612	0.998	0.660	0.639	0.490	0.460	0.145	0.535	0.471	0.441	0.479	0.437	0.112	0.438	0.442
##او	0.405	0.458	0.511	0.651	0.999	0.678	0.432	0.433	0.141	0.528	0.444	0.370	0.462	0.428	0.110	0.414	0.422
##ي	0.466	0.499	0.578	0.635	0.682	0.999	0.548	0.525	0.206	0.619	0.590	0.497	0.545	0.532	0.173	0.492	0.506
لثمتي	0.454	0.350	0.471	0.484	0.434	0.547	0.998	0.638	0.189	0.535	0.604	0.511	0.536	0.539	0.157	0.421	0.450
حاتفم	0.443	0.422	0.482	0.462	0.437	0.527	0.642	0.997	0.200	0.629	0.616	0.532	0.549	0.543	0.156	0.442	0.410
ال	0.149	0.130	0.155	0.140	0.139	0.201	0.188	0.196	1.000	0.215	0.211	0.188	0.214	0.200	0.994	0.181	0.174
##رس	0.460	0.475	0.471	0.522	0.521	0.611	0.530	0.622	0.218	0.997	0.653	0.544	0.557	0.549	0.178	0.523	0.503
يف	0.459	0.411	0.534	0.467	0.445	0.584	0.600	0.611	0.209	0.654	0.996	0.646	0.601	0.540	0.175	0.475	0.500
مهف	0.446	0.353	0.415	0.442	0.369	0.494	0.518	0.529	0.194	0.546	0.653	0.990	0.662	0.551	0.168	0.421	0.421
ومن	0.476	0.391	0.459	0.458	0.465	0.537	0.536	0.514	0.215	0.539	0.587	0.629	0.894	0.631	0.191	0.507	0.488
هيبش	0.427	0.368	0.421	0.439	0.439	0.537	0.540	0.544	0.201	0.546	0.541	0.524	0.651	0.986	0.182	0.525	0.474
ال	0.120	0.099	0.122	0.109	0.109	0.169	0.155	0.153	0.994	0.176	0.178	0.162	0.190	0.180	1.000	0.156	0.160
##ان	0.387	0.373	0.389	0.438	0.417	0.496	0.430	0.451	0.187	0.528	0.487	0.423	0.530	0.533	0.162	0.996	0.501
##ناس	0.394	0.376	0.460	0.440	0.426	0.507	0.457	0.413	0.179	0.509	0.508	0.427	0.469	0.477	0.166	0.501	0.996

Figure 5.12 BERTScore cosine similarity using AraBert tokenise

Generative Approach-Manual Evaluation, despite the initial automatic evaluation providing promising results, the manual evaluation of the generated text provides deeper insight into mT5's output for the Arabic simplification task. As indicated by the manual error analysis, as shown in Figure 5.14, only 31 sentences were correctly simplified from 299 testing instances. In addition, about 120 generated sentences were incomplete, and the system produced 64 meaningless or ill-formed sentences. A significant shortcoming is that the produced sentences tend to have the same repeated phrase. Moreover, one of the generated sentences was more complex than the original. Also, the unexpected errors produced simple sentences with different or opposite meanings.

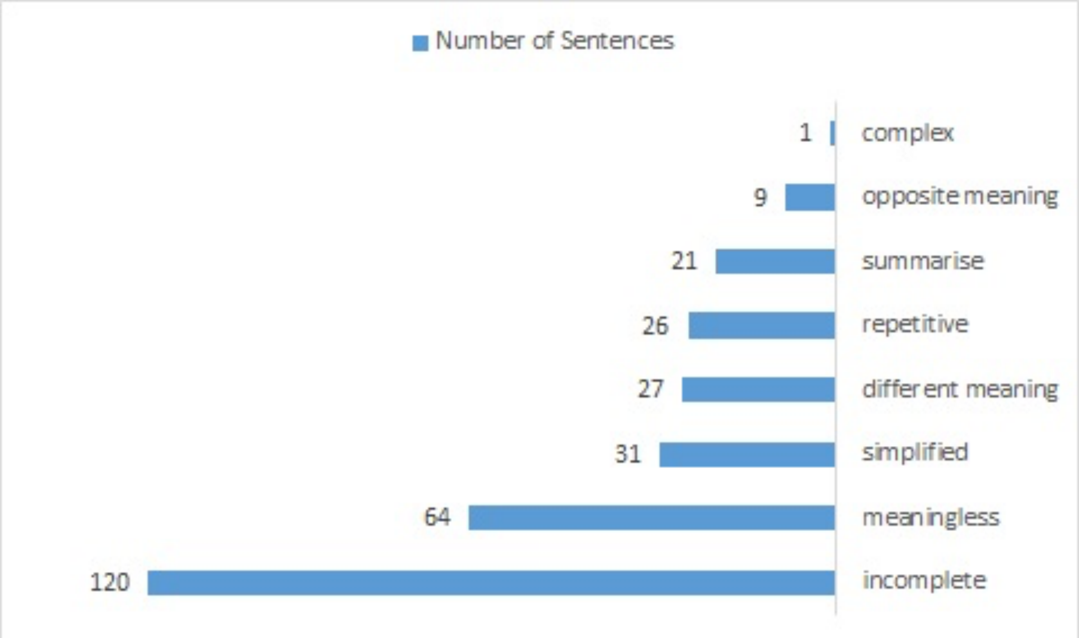


Figure 5.14 Manual error analysis distribution across testing

The following section presents an error analysis of the generative approach generated sentences.

Case 1: There was a tag inserted in many results ***<extra_id_0>***. This generated token appeared in many incomplete instances, which affected the meaning as follows:

Generated sentence	**Target sentence**	**Original sentence**
<extra_id_0> الذُّعْرُ تَمَّ	تَمَلَّكَنِي الرُّعْبُ.	تَمَلَّكُنِي الذُّعْرُ.
aḏḏuʿru tamma	tamallakanī arruʿbu	tamallakunī aḏḏuʿru
Panic is done.	Fear possesses me.	I panicked.

Case 2: The sentence is generated with a different form of the complex target lemma resulting in changing the whole sentence’s meaning. In the following example the word “نَظْرَةٌ” (naḏ̭ratun, look) was replaced by “نَظَرِيَّةٌ” (naḏ̭ariyyatun, theory):

Generated sentence	**Target sentence**	**Original sentence**
عَيْنِي تَنْتَقِلَانِ بَيْنَهُمَا **نَظَرِيَّةَ** اسْتِهْزَاءٍ.	عَيْنَايَ تَنْتَقِلَانِ بَيْنَهُمَا	لَا تَعْبِيرَ وَلَا حَرَكَةَ سِوَى عَيْنَيٍّ تَنْتَقِلَانِ بَيْنَهُمَا **نَظْرَةَ** اسْتِهْزَاءٍ
ʿaynī tantaqilāni baynahumā naḏ̭ariyyatu astihzāʾin	ʿaynāya tantaqilāni baynahumā	lā taʿbīra walā ḥarakata siwā ʿaynayyin tantaqilāni baynahumā naḏ̭rata astihzāʾin
My eyes shift between them, mockery theory.	My eyes shift.	There was no expression or movement except my eyes shifted with a look of mockery.

Another example was changing the word “قِرَاءَةَ” (qirāʾata , reading) to the word “قَر◌ّ◌َاءَ” (qarrāʾa , readers), resulting in changing the meaning completely.

Generated sentence	**Target sentence**	**Original sentence**
مِثْلُ حُبِّهَا **لِلْقُرَّاءِ**	هِيَ مِثْلُ أَبِي بِسَبَبِ حُبِّهَا **لِلْقِرَاءَةِ**	تَكَادُ تَكُونُ نُسْخَةً عَنْ أَبَيٍّ بِسَبَبِ الِانْكِبَابِ عَلَى **قِرَاءَةِ** كُتُبِهِ فِي غُرْفَةِ مَكْتَبِهِ.
miṯlu ḥubbihā lilqurrāʾi	hiya miṯlu ʾubī bisababi ḥubbihā lilqirāʾati	takādu takūnu nusḳatan ʿan ʾabayyin bisababi aliānkibābi ʿalā qirāʾati kutubihi fī ġurfati maktabihi
Like her love for readers.	She is like my dad because she loves to read	Almost a copy of my father's because she was reading his books in his office.

Case 3: Some of the generated text is considered a summarisation of the original sentence by dropping some phrases or information and repeating part of the original sentence. In the following sentence dropping the first part of the sentence, which contains the main verb, affects the full meaning.

Generated sentence	Target sentence	Original sentence
قَدَمَيْهَا فِي الْمَاءِ.	وَضَعَتْ **قَدَمَيْهَا فِي الْمَاءِ** الَّذِي أَخَذَ يَرْتَفِعُ حَتَّى مُنْتَصَفِ جَسَدِهَا.	خَاضَتْ **قَدَمَيْهَا فِي الْمَاءِ** الَّذِي أَخَذَ يَرْتَفِعُ كُلَّمَا خَطَّتْ إِلَى الْأَمَامِ، جَاوَزَ الْمَاءُ مُنْتَصَفَ جَسَدِهَا.
qadamayhā fī almāʾi	waḍaʿat qadamayhā fī almāʾi allaḏī ʾaḳaḏa yartafiʿu ḥattā muntaṣafi jasadihā	ḳāḍat qadamayhā fī almāʾi allaḏī ʾaḳaḏa yartafiʿu kullamā ḳaṭṭat ʾilā alʾamāmi, jāwaza almāʾu muntaṣafa jasadihā
Her feet are in the water.	She put her feet in the water, which began to rise to the middle of her body.	She waded her feet in the water that rose as she stepped forward, the water past the middle of her body.

Case 4: Adding repetitive words as in the following example by repeating the word "وَالِدِي" (wālidī, my father) twice at the end of the generated sentence.

Generated sentence	Target sentence	Original sentence
كَانَتْ لِقَاؤُهُمَا وَالِدِي وَالِدِي	فِي الْيَوْمِ نَفْسِهِ كَانَ لِقَاؤُهُمَا بِمَوْعِدٍ حَدَّدَهُ وَالِدِي.	فِي الْيَوْمِ ذَاتِهِ كَانَ لِقَاؤُهُمَا بِمَوْعِدٍ حَدَّدَهُ وَالِدِي.
kānat liqāʾuhumā wālidī wālidī	fī alyawmi nafsihi kāna liqāʾuhumā bimawʿidin ḥaddadahu wālidī	fī alyawmi ḏātihi kāna liqāʾuhumā bimawʿidin ḥaddadahu wālidī
They met my father my father.	On the same day, they met at an appointment set by my father.	On the same day, they met at an appointment set by my father.

Case 5: This produced simple sentences yet with different or opposite meanings. Such an example gives an opposite meaning to the generated sentence represented in the following example. A newly generated sentence expresses the opposite meaning of the original. For example, instead of mentioning that his relationship with the church was not strong, the system generated the opposite meaning by expressing a very strong relationship with the church.

Generated sentence	Target sentence	Original sentence
عَلَاقَتِي بِالْكَنِيسَةِ فِي بِلَادِ أُمِّي قَوِيَّةٌ جِدًّا.	لَمْ تَكُنْ عَلَاقَتِي بِالْكَنِيسَةِ فِي بِلَادِ أُمِّي قَوِيَّةً، فَزِيَارَاتِي لَهَا قَلِيلَةٌ جِدًّا.	لَيْسَ هُنَاكَ مَا يُمَيِّزُ عَلَاقَتِي بِالْكَنِيسَةِ فِي بِلَادِ أُمِّي، فَزِيَارَاتِي لَهَا قَلِيلَةٌ جِدًّا.
ʿalāqatī bilkanīsati fī bilādi ʾummī qawiyyatun jiddan	lam takun ʿalāqatī bilkanīsati fī bilādi ʾummī qawiyyatan, faziyārātī lahā qalīlatun jiddan	laysa hunāka mā yumayyizu ʿalāqatī bilkanīsati fī bilādi ʾummī, faziyārātī lahā qalīlatun jiddan
My relationship with the church in my mother's country is very strong.	My relationship with the church in my mother's country was not strong, as I visited it very few.	There is nothing that distinguishes my relationship with the church in my mother's country. My visits to it were very few.

Case 6: A generated sentence with the opposite meaning. However, the system catches the main idea of the original sentence better than the target sentence; it produces a well-structured simplified sentence with an opposite adjective. In the following example, the system replaces the adjective “صَغِيرَةٍ” (ṣaġīratin ,small) with its antonym “كَبِيرَةٍ” (kabīratin, large).

Generated sentence	Target sentence	Original sentence
يَظْهَرُ بِلِحْيَةٍ كَبِيرَةٍ تُشْبِهُ لِحْيَةَ أَمِيرِ الْكُوَيْتِ.	أَحَدُ الرَّجُلَيْنِ بِلِحْيَةٍ صَغِيرَةٍ.	يَظْهِرُ أَحَدُ الرَّجُلَيْنِ بِلِحْيَةٍ صَغِيرَةٍ تُشْبِهُ لِحْيَةَ أَمِيرِ الْكُوَيْتِ الَّذِي تُوُفِّيَ يَوْمَ وُصُولِي، إِلَّا انَّهُ لَا يَمْلِكُ ابْتِسَامَتَهُ.
yaḏharu biliḥyatin kabīratin tušbihu liḥyata ʾamīri alkuwayti	ʾaḥadu arrajulayni biliḥyatin ṣaġīratin	yaḏhiru ʾaḥadu arrajulayni biliḥyatin ṣaġīratin tušbihu liḥyata ʾamīri alkuwayti allaḏī tuwuffiya yawma wuṣūlī, ʾillā annahu lā yamliku abtisāmatahu
He appears with a large beard similar to the beard of the Emir of Kuwait.	One of the men with a small beard.	One of the two men appears with a small beard similar to that of the Emir of Kuwait, who died on the day I arrived ,but did not have his smile.

Case 7: One of the generated sentences was a more complex sentence. The system generated this sentence by keeping the complex verb “رَغِبَتْ” (raġibat, ‘beg’) rather than طَلَبَتَ (ṭalabata,‘asked’ or ‘wanted’).

Generated sentence	Target sentence	Original sentence
رَغِبَتْ أُمِّي رَاجِيَةَ بَقَائِي فِي الْكُوَيْتِ وَقْتٌ أَطْوَلُ.	رَفَضَتْ أُمِّي الْفِكْرَةَ رَغْمَ اشْتِيَاقِهَا لِي طَلَبَتَ مِنّي أَنْ أَبْقَى فِي الْكُوَيْتِ وَقْتًا أَطْوَلَ.	رَفَضَتْ أُمِّي الْفِكْرَةَ رَغْمَ اشْتِيَاقِهَا لِي، طَلَبَتِ مِنّي رَاجِيَةً بَقَائِي فِي الْكُوَيْتِ وَقْتًا أَطْوَلَ.
raġibat ʾummī rājiyata baqāʾī fī alkuwayti waqtun ʾaṭwalu	rafaḍat ʾummī alfikrata raġma aštiyāqihā lī, ṭalabata minnī rājiyatan baqāʾī fī alkuwayti waqtan ʾaṭwala	rafaḍat ummī alfikrata raghma ashtiyaāqihā lī, ṭalabat minnī rājiyatan baqāʾī fī alkuwayti waqtan aṭwala.
My mother begged me to stay in Kuwait for a longer time.	My mother rejected the idea, although she missed me. Instead, she asked me to stay in Kuwait for a longer time.	My mother rejected the idea, despite her longing for me. She asked me, begging me to stay in Kuwait for a longer time.

Case 8: An accurate simplified generated version which matches the target sentences in some cases. As in the following example, the generated sentence is replacing the verb “أَوَدُّ” (ʾawaddu, I want) with a simpler verb “أُرِيدُ” (ʾurīdu, I want)

Generated sentence	Target sentence	Original sentence
أَرْسَلَ رِسَالَةٌ تَحْمِلُ كُلَّ مَا أُرِيدُ قَوْلَهُ لِابْنَةِ خَالَتِي الْحَبِيبَةِ.	أَرْسَلَ رِسَالَةً تَحْمِلُ كُلَّ مَا أُرِيدُ قَوْلَهُ لِابْنَةِ خَالَتِي الْحَبِيبَةِ.	أَرْسَلَ رِسَالَةً تَحْمِلُ كُلَّ مَا أَوَدُّ قَوْلَهُ لِابْنَةِ خَالَتِي الْحَبِيبَةِ.
ʾarsala risālatun taḥmilu kulla mā ʾurīdu qawluhu libnati ķālatī alḥabībati	ʾarsala risālatan taḥmilu kulla mā ʾurīdu qawluhu libnati ķālatī alḥabībati	ʾarsala risālatan taḥmilu kulla mā ʾawaddu qawlahu libnati ķālatī alḥabībati
I sent a letter with everything I want to say to my beloved cousin	I sent a letter with everything I want to say to my beloved cousin.	I sent a letter with everything I want to say to my beloved cousin.

Case 9: mT5 in some cases can produce a perfectly valid paraphrase, which is better than the target simple sentence. In the following example, the generated sentence was syntactically simpler than the target while focusing on the main information.

Generated sentence	Target sentence	Original sentence
طَلَبَ مِنَّا الْجُلُوسَ فِي صَالُونِهِ الْمَلِيءِ بِالْكُتُبِ.	فِي صَالُونِهِ الصَّغِيرِ الْمَلِيءِ بِالْكُتُبِ، طَلَبَ مِنَّا الْجُلُوسَ أَمَامَ مَكْتَبٍ صَغِيرٍ.	فِي صَالُونِهِ الصَّغِيرِ الْمَلِيءِ بِالْكُتُبِ، طَلَبَ مِنَّا الْجُلُوسَ أَمَامَ مَكْتَبٍ صَغِيرٍ مَلِيءٍ بِالْأَوْرَاقِ وَأَقْلَامِ الرَّصَاصِ الْمَبْرِيَّةِ حَتَّى آخِرِهَا.

Generated sentence	Target sentence	Original sentence
ṭaliba minnā aljulūsa fī ṣālūnihi almalīʾi bilkutubi	fī ṣālūnihi aṣṣaġīri almalīʾi bilkutubi, ṭalaba minnā aljulūsa ʾamāma maktabin ṣaġīrin	ya ṣālūnihi aṣṣaġīri almalīʾi bilkutubi, ṭalaba minnā aljulūsa ʾamāma maktabin ṣaġīrin malīʾin bilʾawrāqi waʾaqlāmi arraṣāṣi almabriyyati ḥattā ʾāḳirihā
He asked us to sit in his salon ,which was full of books.	In his small salon full of books, he asked us to sit in front of a small desk.	In his little salon full of books, he asked us to sit in front of a small desk full of papers and sharpened pencils.

Another example involved word movement to simplify the question to form a direct question easier to comprehend.

Generated sentence	Target sentence	Original sentence
سَأَلْتُهُ إِلَى أَيْنَ؟	إِلَى أَيْنَ؟ سَأَلْتُهُ.	إِلَى أَيْنَ؟ سَأَلْتُهُ.
saʾaltuhu ʾilā ʾayna?	ʾilā ʾayna? saʾaltuhu.	ʾilā ʾayna? saʾaltuhu
I asked him, To where?	To where? I asked him.	To where? I asked him.

Generative OpenNMT Approach-Manual Evaluation, the initial automatic evaluation providing promising results, the manual evaluation of 50 generated (translated) sentences from the three models showed the weakness of the approach.

Model one – based on Saqq Al-Bambuu, only one sentence out of the 50 examples was correctly simplified even well than the target sentence (see Table 5.31). Whereas the rest 49 sentences were wrongly generated that disturb the general information conveyed.

Table 5.31 Simplified generated sentence from Model One

Generated sentence	**Target sentence**	**Original sentence**
Correct simplification		
مَاذَا تَقُولُ؟ بِغَضَبٍ سَأَلْتُ خَوْلَةً	مَاذَا تَقُولُ؟، بِغَضَبٍ سَأَلْتُ خَوْلَةَ	مَاذَا تَقُولُ؟، مَاذَا تَقُولُ؟، سَأَلْتُ خَوْلَةَ وَالْغَضَبُ يَتَمَلَّكُنِي.
māḏā taqūlu? biġaḍabin saʾaltu ḳawlatan	māḏā taqūlu?, biġaḍabin saʾaltu ḳawlata	māḏā taqūlu?, māḏā taqūlu?, saʾaltu ḳawlata walġaḍabu yatamallakunī.
What do you say? What do you say? Angry I asked Khawla	What do you say? I asked Khawla. While the anger possesses me.	What do you say? What do you say? I asked Khawla. While the anger possesses me.
Misinformation simplification		
بحثت في حقيبتي بين الأوراق	بحثت في جيب بنطلوني ولكنني لم أجد محفظتي	بحثت في جيب بنطلوني ولكنني لم أعثر على محفظتي
baḥaṯtu fī ḥaqībatī bayna alʾawrāqi	baḥaṯtu fī jaybi banṭalūnī walakinnī lam ʾajid maḥfaḓatī	baḥaṯtu fī jaybin banṭalūnī walakinnī lam ʾuʿṯur ʿalā maḥfaḓatī
I searched in my briefcase among the papers	I searched in my trouser pocket but could not find my wallet	I searched in my trouser pocket but could not find my wallet

Model Two –combined dataset, despite providing the model with bigger dataset, in the 50 sentences there were 6 sentences considered as good simplified, 3 accepted simplifications and the 41 rest was not accepted as simplified versions. Table 5.32 shows a generated sentences which is different from the original idea in the original sentence.

Table 5.32 Example of generated sentences using Model Two.

Generated sentence	Target sentence	Original sentence
يَجِبُ أَنْ نُعْطِيَ وَزْنًا لِلتَّخَلُّصِ مِنْ أَفْرَادِ شِبْهِ الْعَسْكَرِيَّةِ فِي سِلْسِلَةٍ مِنْ الضَّمَانَاتِ.	عَلَيْكَ أَنْ تَسْأَلَ بَاكِسْتَانَ لِوَقْفِ تَسَلُّلِ الْقُوَّاتِ شِبْهِ الْعَسْكَرِيَّةِ عُبُورَ خَطِّ السَّيْطَرَةِ مِنْ أَرَاضِيهَا.	وَيَنْبَغِي أَنْ تُطَالِبَ بَاكِسْتَانُ بِوَضْعِ حَدٍّ لِتَسَلُّلِ أَفْرَادٍ مِنْ الْقُوَّاتِ شِبْهِ الْعَسْكَرِيَّةِ مِنْ جَانِبِهِمْ مِنْ خَطِّ السَّيْطَرَةِ .
yajibu ʾan nuʿṭiya waznan lilttaḳalluṣi min ʾafrādi šibhi alʿaskariyyati fī silsilatin min aḍḍamānāti.	ʿalayka ʾan tasʾala bākistāna liwaqfi tasalluli alquwwāti šibhi alʿaskariyyati ʿubūra ḳaṭṭi assayṭarati min ʾarāḍīhā.	wayanbaġī ʾan tuṭāliba bākistānu biwaḍʿi ḥaddin litasalluli ʾafrādin min alquwwāti šibhi alʿaskariyyati min jānibihim min ḳaṭṭi assayṭarati .
We must give weight to get rid of paramilitary personnel in a series of safeguards.	You have to ask Pakistan to stop the infiltration of paramilitary forces crossing the control line from its territory	Pakistan should demand an end to the infiltration of its paramilitary forces to control line.

Out of the two generative methods, the generative T5 seq2seq method generated better results. Although text simplification can be framed as a machine translation task, it is not always the best approach. The main reason for this is that the syntax and structure of complex sentences can be very different from that of simpler sentences, even if they convey the same meaning (Zhang et al., 2017).

In machine translation, the aim is to preserve the meaning of the source text while producing a translation that is grammatically correct and idiomatic in the target language. However, in text simplification, the goal is not just to preserve the meaning of the complex sentence but also to produce a simplified version that is easier to read and understand for the intended audience.

Therefore, a simplification model that relies solely on a machine translation approach may not always produce simplified sentences that are easy to read and understand. Instead, text simplification models often require additional techniques and strategies to ensure that the output is both simpler and more readable.

5.4. Conclusion

This chapter presented an Arabic sentence simplification system by applying both classification and generative approaches. On the one hand, the classification approach focuses on LS. Looking at the different classification methods showed that a combined method generates well-formed simple sentences. In addition, using word embeddings and transformers prove to produce a reasonable set of substitutions for complex word more accurately than traditional methods such as WordNet. The interpretation of the limitation in the classification system arises from the fact that some of the generated sentence structures are not well-formed and that the system can misidentify what makes some complex words in the CWI step. Despite this limitation reveals the limitations of the Arabic CEFR vocabulary list in identifying complex words, the list is proven to be more useful in the substitution replacement step.

On the other hand, while the generative Seq2Seq approach provides a less accurate simplified version in most cases, in some cases, it outperforms the classification approaches by generating a simplified sentence, which can be even better than the target human simple sentence. Nevertheless, one of the generative approach’s limitations is the repetition of a part of the same phrase patterns. Future research is needed to address this issue. Overall, showing the advantages and limitations of the two approaches, both of which could benefit from building a larger parallel simple/complex Arabic corpus. Moreover, adding a post-handler language generation module could resolve some of the limitations even if only acting as a less accurate alternative fast solution, for example, by avoiding and removing repeated phrase patterns produced from the generative system. Another example is a post-syntactic checker to remove or change the preposition to match the new verb.

Chapter Six: Summary and Conclusion

In conclusion, this research investigated text readability (TR) and Text simplification (TS) approaches with a focus on their application to the Arabic language. The analysis presented in this thesis is expected to contribute to the existing knowledge in measuring text readability and sentence simplification.

Throughout the discussion, several key points and findings were highlighted. Firstly, the importance of measuring text readability in various contexts was discussed, and several existing TR measures were reviewed. Secondly, different TS techniques and their applications were explored, with a focus on the challenges of TS in Arabic. Thirdly, Arabic TS classification and generative models were developed and evaluated using various metrics, including BERTScore and BLEU.

To conclude the thesis, a summary of the key points and findings highlighted throughout the course of the discussion is presented in this chapter. The chapter will also discuss reflections from the experience of conducting this research project and the encountered challenges. Lastly, the chapter presents suggestions for future studies.

6.1. Summary

This thesis tackled text readability and text simplification which are NLP-related tasks aimed at building an Automatic Arabic SS system using robust NLP techniques targeting a wide range of users. Simplifying everything in a text may result in inconsistencies and incoherent simplified sentences. Hence, measuring the text readability was necessary to decide what to simplify.

This led to identifying the Automatic TR gap in Arabic that compromises the lack of resources and language-specific features that affect sentence readability. In the TR literature, most systems focused on measuring the readability of a whole text rather than each sentence alone. Hence, it directed the study to either enhance available resources or produce new Arabic resources specified for the Arabic Automatic TR task on sentence level.

As explained in section (A) in Chapter 4, this study firstly provides an Arabic CEFR-level classified word list based on enhancing two available Arabic frequency lists with a compiled learner's textbook vocabulary list as a pedagogical reference. This Arabic CEFR level classification is not limited to being used in ARA. However, it could be used as it is as a reference to select the appropriate text for L1 and L2 syllabus construction. Secondly, a CEFR sentence-level corpus is compiled from the available text-level readability classified corpora to be used as a gold standard in building a sentence readability classifier.

The second section in Chapter 4 presents a detailed and novel methodology for developing a sentence readability assessment. Which needs to use much less information than text-level approaches. This approach represents one of the main contributions of this thesis, includes a comparison of various ML approaches, error analysis and feature ablation to select the classifier with the best performance. This ended by developing a binary sentence complexity classifier that predicts if the sentence is easy or complex. Additionally, it produced a 3-way sentence readability classifier that predicts the sentence level based on the primary CEFR levels (A, B or C).

For Arabic Automatic TR tasks, fine-tuned Arabic-BERT offers better performance than other sentence embedding methods or linguistic features. If one thinks of Arabic learners, especially in higher education, one expects learners to graduate with a BA degree in the case of Arabic as a complex language with confidence in reading B2 texts, which implies that the tool for separating A+B vs C-level texts is helpful for undergraduate teaching. This tool provides a computational assessment of difficulty and will enable lecturers: i) to select the appropriate texts for students; ii) to access ever-larger volumes of information to find educational material of the right difficulty online; iii) to explore curriculum-based assessment to identify areas where students need support and improvement and to develop effective strategies to address these gaps.

Building on such resources allowed the study to apply the readability-prediction algorithm in a more significant NLP scenario for the task of Automatic Text Simplification. In the literature, TS is referred to as text simplification, sentence simplification, lexical simplification, sentence compression, text summarisation, paraphrasing and text style transfer. All methods tend to be used to refer to improving text readability using different procedures.

Despite having a clear state-of-the-art for TS and a defined pipeline for Arabic LS, TS is still in its infancy. The main challenges in TS were: i) the availability of a simultaneously simple/complex Arabic corpus and ii) measuring the subjective nature of the readability levels of a simplified text. Therefore, the study provides a manually compiled complex/simple parallel sentence pair considered a gold standard.

The goal of this corpus was twofold. First, it was used to train a TS algorithm and second, to test the TR binary classifier on newly unseen data. It was found that some texts are more difficult to simplify than others while exploring and experimenting with Arabic TS. This difficulty arises from the complex nature of the Arabic Language, specifically, in using long concatenated sentences using addition and reference connectors. At this stage, while having reliable resources, the research is directed towards exploring the Arabic TS application. For this purpose, the study adopts a hybrid method combining machine learning and rule-based techniques to provide a new approach to the Arabic sentence simplification methodology. The primary contribution of this thesis is to examine different approaches for Arabic sentence simplification tasks using automatic and manual evaluation. To our knowledge, this is the first available Arabic sentence-level simplification system.

6.2. Results

The studies presented in this thesis were performed as a series of experiments resulting in a series of answers, outcomes, and contributions. These experiments provide evidence for the effectiveness of applying various combined approaches. This section will discuss the final results of each experiment in order as presented in Chapters 4 and 5 reflected the aims and obejectives of this thesis.

Objective 1: To investigate how text complexity/readability can be measured

- The first experiment demonstrated that traditional readability measurement statistical formulae like Flesch–Kincaid Grade, the SMOG formula, and the Dale-Chall formula do not effectively measure the readability of an Arabic sentence. Instead, the 3-way ArabicBERT classifier outperforms these formulae, as it takes into consideration not just shallow linguistic features, but other non-linguistic features that affect readability. This leads to a shift in the study towards traditional and deep ML methods to approach Arabic Text Readability (TR).
- The fifth experiment went beyond the first objective by using the novel dataset CEFR classified sentence-level as a training set. This led to the development of an MSA sentence difficulty classifier, predicting the difficulty of sentences for language learners using either the CEFR proficiency levels or the binary classification as simple or complex. This experiment tested different sentence representation methodologies, from linguistic knowledge via feature-based machine learning to modern neural methods, suggesting that the Automatic TR task could be treated as a classification or regression task.

Objective 2: To explore possible approaches to simplify the Arabic text on lexical and syntactic levels

- The second experiment is directly related to this objective, where a new MSA CEFR classified list was created. This list involved combining three available Arabic vocabulary lists to develop a common dataset. This was instrumental in creating a resource to help simplify the Arabic text, ensuring that only an MSA variety is listed by removing dialectical words.
- The fourth experiment involved manually compiling a set of 2980 simple/complex parallel sentences, creating an Arabic parallel complex/simple sentence corpus. This corpus was used in both the evaluation of the developed binary TR classifier and as a training corpus to develop a seq2seq generative Arabic TS approach.

- The sixth experiment extended beyond the objective by presenting a reliable method for Arabic sentence simplification. This involved the application of both classification and generative approaches. The generative Seq2Seq approach tackled full sentence simplification, considering both lexical and syntactic simplification.

Objective 3: To investigate why some texts are more challenging to simplify than others

- The third experiment addressed this objective by exploring the available TR-oriented Arabic corpora and compiling a sentence-level CEFR corpus. This corpus became a fundamental resource of this research, and it helped understand why some texts were more challenging to simplify than others.

Beyond the objectives:

- The developed classifiers' performance on different datasets shows that the classifiers learn some essential properties of what is difficult in Arabic, providing valuable insights for Arabic language learning and pedagogy.
- The fine-tuned Arabic-BERT, which provides the best performance among the deep learning approaches, is a substantial contribution to the field of Arabic NLP.
- The Arabic TS system, which generates easy-to-read Arabic text, could be a significant resource for Arabic language learners and educators.

Overall, this thesis didn't just focus on achieving the objectives but extended its implications beyond, offering a more nuanced and detailed understanding of the complexities of Arabic TR and TS.

6.3. Impact

Reading is one of the essential life tasks we encounter every day. However, text can often be complex and difficult to read for certain groups of people who face several difficulties in comprehension. Approximately 10% of the world’s population has an intellectual disability. They face significant challenges in literacy and reading comprehension. In addition, each person's language literacy level is different. Measuring text readability/complexity and enhancing text accessibility to the vast majority of readers have attracted researchers from various fields.

This study's main impact yields different views of text readability and accessibility across fields such as education, psychology, and linguistics, highlighting the implications and connections between those fields. In addition, this study has pulled together established theoretical frameworks for measuring TR in different languages and applied the confirmed pipeline with the latest techniques in the Arabic Language. Moreover, the study proposed a new framework for Arabic Automatic TR and its application in ATS as a real-life NLP scenario. The finding of this study provides a promising application of Automatic TR and ATS in the Arabic Language.

The experiments conducted in this research provide evidence for the efficiency of different approaches built on either quantitative or qualitative analyses or a combination of both in the representation of text readability features. The implications of this study can be extended beyond the field of Arabic NLP, which has the potential to grow radically, to the field of teaching Arabic as a first and foreign language, as the results of this research are strongly related to selecting the best curriculum and simplifying the syllabus for various language disabilities for the sake of inclusion using data science. For example, it could be used to simplify the primary educational schools’ syllabus that works towards the inclusion of autistic children in mainstream schools to learn and reach advanced stages without being an economic and social burden on their families. Besides, teachers in inclusion schools may benefit from a simplified syllabus aiding them in teaching normal and disabled children without the need for trained teachers with supporting speech therapists to deal with autistic children. This leads to the creation of a new generation of language-disabled people considered productive and influential persons in society and not a burden on it.

To further highlight the implications that each of the thesis findings contributes to corpus linguistics, it should be added that the resources produced will help learners and teachers of Arabic as a foreign language. These tools (the CEFR frequency list, the sentence-level corpus, the readability classifier and the sentence simplifier) will assist them in understanding complex Arabic texts, leading them to master the Arabic Language. Moreover, corpus linguistic methods are used to learn the actual use of various Arabic words in an authentic situation.

Besides, the TS tool would also act as a sub-assisted application in many NLP tasks. For example, TS is believed to be a very effective pre-processing stage in machine translation. Moreover, most recent sentence simplification systems use basic machine translation models to learn lexical and syntactic paraphrases from a manually simplified parallel corpus.

6.4. Challenges

The challenges of this research were related to the nature of the study, as well as the application of the methodology. This research went through several challenges summarised in the following points, some of which are not limited to Arabic and extend to many other languages.

- This research, being interdisciplinary in nature, posed a challenge in maintaining a balance between two fields of study: Arabic NLP and Cognitive processes in reading to comprehend text.
- Assigning readability levels on sentence level is challenging, as much less linguistic information is available.
- The availability of corpora, resources and tools for both Automatic TR and ATS.
- The limited studies of reader's comprehension and needs from educational and psychological points of view.
- The main challenge is embedding the Automatic TR model in the ATS model.
- The availability of evaluation methods that are tailored especially for the ATS system.

6.5. Limitations

The research recognizes other potential limitations:

- The comprehension ability of Arabic second language learners was not extensively tested in the study. The difficulties they encounter while reading a new Arabic text are likely to be multifaceted, encompassing not just syntactic, but also semantic, cultural, and linguistic aspects, which were not fully covered in the research. It is necessary to have a more comprehensive understanding of these issues to facilitate the development of an effective Arabic text simplification system.
- While the study was limited to Modern Standard Arabic (MSA), it's important to note that there are various Arabic dialects, each with its unique complexities. The methodology adopted might not be directly applicable or as effective when dealing with these dialects. Further research is required to adjust the methods for these dialects.
- In addition to the current limitations concerning the Arabic complex/simple corpus and the syntactic simplification module, the study also did not account for the learner's individual background (such as their native language), which could significantly impact their reading skills and comprehension of Arabic text.
- There's a noted absence of human evaluative feedback in the study. The user experience, including the difficulties faced by Arabic second language learners and the ease of understanding the simplified text, is crucial in evaluating the system's effectiveness. Future work should incorporate human-centered evaluations to better understand how users perceive and interact with the simplified text.

6.6. Ongoing experiments and future work

The research journey in this area is far from complete; much more work is needed in this area to address the many open research problems. Nevertheless, the aforementioned linguistic resources and the insights produced by this research will be valuxable for future studies. Moreover, the journey of searching for a methodology that best suits this research's aims was fruitful, for it opened the researcher's mind to developments in the field of NLP and possible further avenues of research in this domain.

The future work involves building a much bigger parallel simple/ complex Arabic corpus for sentence simplification. The corpus will be classified based on how complex the sentences are in a Common Crawl snapshot of Arabic web pages. Using the text difficulty classifier, the corpus can be split into two groups for complex and simple sentences. The semantic similarity detection on "Saqq al-Bambuu" can also be considered as a benchmark, which could be used in the corpus compilation. In the readability study, only some ablation analysis was performed. However, because BERT-like models are more valuable as classifiers, but they operate as black-boxes, their performance via probing for linguistic features should be investigated following the BERTology framework (Rogers et al., 2020; Sharoff, 2021). In addition, the link between the difficulty assessments on the document vs sentence levels ought to be explored (Dell'Orletta et al., 2014).

One possible area for future work involves using methods for interpretability, such as Integrated Gradients, to better understand the decision-making process of the text difficulty classifier. This method can help to identify which input features are most important for the classifier's predictions, and can provide insights into how the model is making its decisions. Another area is to evaluate the readability ratio by measuring the difference between the original and simplified readability measure to measure the simplification ratio. Additionally, providing a human evaluation of the produced sentences can give a more complete picture of the effectiveness of the simplification technique.

Other likely postdoctoral work involves explaining how methods in TR and TS can be applied to teaching MSA. This may help learners of Arabic as a foreign language since the Leeds tools will assist them in mastering and translating complex Arabic texts.

Bibliography

Abu-Rabia, M. 2008. The Effect of Conjunctions on the Reading Comprehension of Arabic-speaking Students Learning Hebrew as a Second Language. *Language Learning*. **58**(2), pp.325–364.

Al Khalil, M., Habash, N. and Jiang, Z. 2020. A Large-Scale Leveled Readability Lexicon for Standard Arabic *In*: *Proceedings of the 12th Language Resources and Evaluation Conference* [Online]. Marseille, France: European Language Resources Association, pp.3053–3062. [Accessed 31 December 2020]. Available from: https://www.aclweb.org/anthology/2020.lrec-1.373.

Al Khalil, M., Habash, N. and Saddiki, H. 2017. Simplification of Arabic Masterpieces for Extensive Reading: A Project Overview. *Procedia Computer Science*. **117**, pp.192–198.

Al Khalil, M., Saddiki, H., Habash, N. and Alfalasi, L. 2018. A Leveled Reading Corpus of Modern Standard Arabic *In*: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)* [Online]. Miyazaki, Japan: European Language Resources Association (ELRA). [Accessed 24 June 2020]. Available from: https://www.aclweb.org/anthology/L18-1366.

Al-Ajlan, A.A., Al-Khalifa, H.S. and Al-Salman, A.S. 2008. Towards the development of an automatic readability measurements for arabic language *In*: *2008 Third International Conference on Digital Information Management*., pp.506–511.

Al-Badrashiny, M., Hawwari, A., Ghoneim, M. and Diab, M. 2016. SAMER: A Semi-Automatically Created Lexical Resource for Arabic Verbal Multiword Expressions Tokens Paradigm and their Morphosyntactic Features *In*: *Proceedings of the 12th Workshop on Asian Language Resources (ALR12)* [Online]. Osaka, Japan: The COLING 2016 Organizing Committee, pp.113–122. [Accessed 6 June 2023]. Available from: https://aclanthology.org/W16-5414.

Alfaifi, A. and Atwell, E. 2013. Arabic Learner Corpus v1: A New Resource for Arabic Language Research *In*: University of Leeds: Leeds.

Alhawary, M.T. and Brustad, K. 2016. *Arabic Grammar in Context*. Routledge.

Al-Khalifa, H. and Al-Ajlan, A. 2010. Automatic readability measurements of the arabic text: An exploratory study. *Arabian Journal for Science and Engineering*. **35**, pp.103–124.

Al-Raisi, F., Lin, W. and Bourai, A. 2018. A Monolingual Parallel Corpus of Arabic. *Procedia Computer Science*. **142**, pp.334–338.

Al-Sanousi, S. 2013. *Saqq Al-Bambuu*. Arab Scientific Publishers Inc., Lebanon.

Al-Subaihin, A.A. and Al-Khalifa, H.S. 2011. Al-Baseet: A proposed simplification authoring tool for the Arabic language *In*: *2011 International Conference on Communications and Information Technology (ICCIT)*., pp.121–125.

Al-Thanyyan, S.S. and Azmi, A.M. 2021. Automated Text Simplification: A Survey. *ACM Computing Surveys*. **54**(2), pp.1–36.

Al-Thubaity, A.O. 2015. A 700M+ Arabic corpus: KACST Arabic corpus design - ProQuest. *Language Resources and Evaluation*. **49**(3), pp.721-751.

Al-Twairesh, N., Al-Dayel, A., Al-Khalifa, H., Al-Yahya, M., Alageel, S., Abanmy, N. and Al-Shenaifi, N. 2016. MADAD: A Readability Annotation Tool for Arabic Text *In*: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)* [Online]. Portorož, Slovenia: European Language Resources Association (ELRA), pp.4093–4097. [Accessed 17 May 2021]. Available from: https://www.aclweb.org/anthology/L16-1646.

Aluísio, S. and Gasperin, C. 2010. Fostering Digital Inclusion and Accessibility: The PorSimples project for Simplification of Portuguese Texts *In*: *Proceedings of the NAACL HLT 2010 Young Investigators Workshop on Computational Approaches to Languages of the Americas* [Online]. Los Angeles, California: Association for Computational Linguistics, pp.46–53. [Accessed 13 November 2021]. Available from: https://aclanthology.org/W10-1607.

Aluisio, S., Specia, L., Gasperin, C. and Scarton, C. 2010. Readability Assessment for Text Simplification *In*: *Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications* [Online]. Los Angeles, California: Association for Computational Linguistics, pp.1–9. [Accessed 26 February 2022]. Available from: https://aclanthology.org/W10-1001.

Alva-Manchego, F., Bingel, J., Paetzold, G., Scarton, C. and Specia, L. 2017. Learning How to Simplify From Explicit Labeling of Complex-Simplified Text Pairs *In*: *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* [Online]. Taipei, Taiwan: Asian Federation of Natural Language Processing, pp.295–305. [Accessed 28 April 2020]. Available from: https://www.aclweb.org/anthology/I17-1030.

Alva-Manchego, F., Scarton, C. and Specia, L. 2020. Data-Driven Sentence Simplification: Survey and Benchmark. *Computational Linguistics*. **46**(1), pp.135–187.

Ambati, B.R., Reddy, S. and Steedman, M. 2016. Assessing Relative Sentence Complexity using an Incremental CCG Parser *In*: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* [Online]. San Diego, California: Association for Computational Linguistics, pp.1051–1057. [Accessed 21 February 2022]. Available from: http://aclweb.org/anthology/N16-1120.

Anon n.d. Al-radif: Arabic thesaurus. *SourceForge*. [Online]. [Accessed 26 October 2021]. Available from: https://sourceforge.net/projects/radif/.

Anon 2019. Assigning CEFR Ratings to ACTFL Assessments | ACTFL. *American Council on the Teaching of Foreign Languages*. [Online]. [Accessed 22 November 2019]. Available from: https://www.actfl.org/publications/additional-resources/assigning-cefr-ratings-actfl-assessments.

Antoun, W., Baly, F. and Hajj, H. 2020. AraBERT: Transformer-based Model for Arabic Language Understanding. *arXiv:2003.00104 [cs]*.

Aprosio, A.P., Tonelli, S., Turchi, M., Negri, M. and Di Gangi, M.A. 2019. Neural Text Simplification in Low-Resource Conditions Using Weak Supervision *In*: *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation* [Online]. Minneapolis, Minnesota: Association for Computational Linguistics, pp.37–44. [Accessed 13 April 2022]. Available from: http://aclweb.org/anthology/W19-2305.

Azpiazu, I.M. and Pera, M.S. 2020. Is cross-lingual readability assessment possible? *Journal of the Association for Information Science and Technology*. **71**(6), pp.644–656.

Azpiazu, I.M. and Pera, M.S. 2019. Multiattentive Recurrent Neural Network Architecture for Multilingual Readability Assessment. *Transactions of the Association for Computational Linguistics*. **7**, pp.421–436.

Baeza-Yates, R., Rello, L. and Dembowski, J. 2015. CASSA: A Context-Aware Synonym Simplification Algorithm *In*: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* [Online]. Denver, Colorado: Association for Computational Linguistics, pp.1380–1385. [Accessed 13 November 2022]. Available from: https://aclanthology.org/N15-1156.

Bailin, A. and Grafstein, A. 2001. The linguistic assumptions underlying readability formulae: a critique. *Language & Communication*., pp.285–301.

Barzilay, R. and Lapata, M. 2008. Modeling Local Coherence: An Entity-Based Approach. *Computational Linguistics*. **34**(1), p.34.

Beigman Klebanov, B., Knight, K. and Marcu, D. 2004. Text Simplification for Information-Seeking Applications *In*: R. Meersman and Z. Tari, eds. *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE*. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp.735–747.

Berendes, K., Vajjala, S., Meurers, D., Bryant, D., Wagner, W., Chinkina, M. and Trautwein, U. 2018. Reading demands in secondary school: Does the linguistic complexity of textbooks increase with grade level and the academic orientation of the school track? *Journal of Educational Psychology*. **110**(4), pp.518–543.

Billami, M.B., François, T. and Gala, N. 2018. ReSyf: a French lexicon with ranked synonyms *In*: *Proceedings of the 27th International Conference on Computational Linguistics* [Online]. Santa Fe, New Mexico, USA: Association for Computational Linguistics, pp.2570–2581. [Accessed 13 November 2022]. Available from: https://aclanthology.org/C18-1218.

Bingel, J., Schluter, N. and Martínez Alonso, H. 2016. CoastalCPH at SemEval-2016 Task 11: The importance of designing your Neural Networks right *In*: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)* [Online]. San Diego, California: Association for Computational Linguistics, pp.1028–1033. [Accessed 20 November 2021]. Available from: https://aclanthology.org/S16-1160.

Bingel, J. and Søgaard, A. 2016. Text Simplification as Tree Labeling *In*: *The 54 Annual Meeting of the Association for Computational Linguistics proceedings of the conference, vol. 1 (long papers): ACL 2016 : August 7-12, 2016, Berlin, Germany.* [Online]. Stroudsburg (PA), USA: Association for Computational Linguistics, pp.337–343. [Accessed 12 April 2022]. Available from: https://www.aclweb.org/anthology/P16-1.pdf.

Biran, O., Brody, S. and Elhadad, N. 2011. Putting it Simply: a Context-Aware Approach to Lexical Simplification *In*: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* [Online]. Portland, Oregon, USA: Association for Computational Linguistics, pp.496–501. [Accessed 26 April 2020]. Available from: https://www.aclweb.org/anthology/P11-2087.

Blum, S. and Levenston, E. 1980. Lexical Simplification in Second-Language Acquisition. *Studies in Second Language Acquisition*. **2**(2), pp.43–63.

Bodenreider, O. 2004. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Research*. **32**(Database issue), pp.D267–D270.

Bollegala, D., Matsuo, Y. and Ishizuka, M. 2007. *Measuring semantic similarity between words using Web search engines*.

Bott, S., Rello, L., Drndarevic, B. and Saggion, H. 2012. Can Spanish Be Simpler? LexSiS: Lexical Simplification for Spanish *In*: *Proceedings of COLING 2012* [Online]. Mumbai, India: The COLING 2012 Organizing Committee, pp.357–374. [Accessed 12 November 2021]. Available from: https://aclanthology.org/C12-1023.

Bruce, B., Rubin, A. and Starr, K. 1981. Why readability formulas fail. *IEEE Transactions on Professional Communication*. **PC-24**(1), pp.50–52.

vor der Brück, T., Hartrumpf, S. and Helbig, H. 2008. A Readability Checker with Supervised Learning Using Deep Indicators. *Intelligent Systems Guest Editors: Costin Badica*. **32**, pp.429–435.

Brunato, D., Cimino, A., Dell'Orletta, F. and Venturi, G. 2016. PaCCSS-IT: A Parallel Corpus of Complex-Simple Sentences for Automatic Text Simplification *In*: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp.351–361.

Brustad, K., Al-Baṭal, M. and Al-Tonsi, A. 2013. *Al-Kitaab fii Tacallum al-cArabiyya: A Textbook for Arabic.* Third Edition. USA: Georgetown University Press.

Brustad, K., Al-Baṭal, M. and Al-Tonsi, A. 2015. *Al-Kitaab fii Tacallum al-cArabiyya: A Textbook for Arabic* Third Edition. George-town University Press.

Brustad, K., Al-Baṭal, M. and Al-Tonsi, A. 2011. *Al-Kitaab fii Tacallum al-cArabiyya: A Textbook for Arabic.* second Edition. USA: Georgetown University Press.

Brysbaert, M. and New, B. 2009. Moving beyond Kucera and Francis: A Critical Evaluation of Current Word Frequency Norms and the Introduction of a New and Improved Word Frequency Measure for American English. *Behavior research methods*. **41**, pp.977–90.

Buckwalter, T. and Parkinson, D. 2014. *A Frequency Dictionary of Arabic: Core Vocabulary for Learners* 1 edition. Routledge.

Camacho Collados, J. 2013. *Syntactic simplification for machine translation*.[Online] Wolverhampton, United Kingdom. [Accessed 2 February 2022]. Available from: https://orca.cardiff.ac.uk/113068/.

Carrell, P.L. 1987. Readability in ESL *In*: *Reading in a Foreign Language*. University of Hawaii National Foreign Language Resource Center, pp.21–40.

Carroll, J., Minnen, G., Canning, Y., Devlin, S. and Tait, J. 1998. Practical Simplification of English Newspaper Text to Assist Aphasic Readers. *Proc. of AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*.

Caseli, H., Pereira, T., Specia, L., Pardo, T., Gasperin, C. and Aluisio, S. 2009. Building a Brazilian Portuguese parallel corpus of original and simplified texts. , pp.59–70.

Cavalli-Sforza, V., El Mezouar, M. and Hend, S. 2014. Matching an Arabic text to a learners' curriculum *In*: 5th Int. Conf. on Arabic Language Processing (CITALA): Oujda, Morocco, pp.79–88.

Cavalli-Sforza, V., Saddiki, H. and Nassiri, N. 2018. Arabic Readability Research: Current State and Future Directions. *Procedia Computer Science*. **142**, pp.38–49.

Chandrasekar, R. and Srinivas, B. 1997. Automatic Induction of Rules for Text Simplification.

Chen, H.-B., Huang, H.-H., Chen, H.-H. and Tan, C.-T. 2012. A Simplification-Translation-Restoration Framework for Cross-Domain SMT Applications *In*: *Proceedings of COLING 2012* [Online]. Mumbai, India: The COLING 2012 Organizing Committee, pp.545–560. [Accessed 21 November 2021]. Available from: https://aclanthology.org/C12-1034.

Cholakov, K., Biemann, C., Eckle-Kohler, J. and Gurevych, I. 2014. Lexical Substitution Dataset for German *In*: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)* [Online]. Reykjavik, Iceland: European Language Resources Association (ELRA), pp.1406–1411. [Accessed 13 November 2022]. Available from: http://www.lrec-conf.org/proceedings/lrec2014/pdf/545_Paper.pdf.

Choubey, P. and Pateria, S. 2016. Garuda & Bhasha at SemEval-2016 Task 11: Complex Word Identification Using Aggregated Learning Models

Cilibrasi, R.L. and Vitanyi, P.M.B. 2007. The Google Similarity Distance. *IEEE Transactions on Knowledge and Data Engineering*. **19**(3), pp.370–383.

Clarke, J. and Lapata, M. 2006. Models for Sentence Compression: A Comparison across Domains, Training Requirements and Evaluation Measures. *ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17-21 July 2006.*, pp.377–384.

Cohn, T. and Lapata, M. 2013. An abstractive approach to sentence compression. *ACM Transactions on Intelligent Systems and Technology*. **4**(3), p.41.

Collins-Thompson, K. 2014. Computational Assessment of Text Readability: A Survey of Current and Future Research. *ITL-International Journal of Applied Linguistics*. **165(2)**, pp.97–135.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L. and Stoyanov, V. 2019. Unsupervised Cross-lingual Representation Learning at Scale. *arXiv:1911.02116 [cs]*.

Coster, Will and Kauchak, D. 2011. Learning to Simplify Sentences Using Wikipedia *In*: *Proceedings of the Workshop on Monolingual Text-To-Text Generation* [Online]. Portland, Oregon: Association for Computational Linguistics, pp.1–9. [Accessed 30 April 2021]. Available from: https://www.aclweb.org/anthology/W11-1601.

Coster, William and Kauchak, D. 2011a. Simple English Wikipedia: A New Text Simplification Task *In*: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* [Online]. Portland, Oregon, USA: Association for Computational Linguistics, pp.665–669. [Accessed 30 April 2021]. Available from: https://www.aclweb.org/anthology/P11-2117.

Coster, William and Kauchak, D. 2011b. Simple English Wikipedia: A New Text Simplification Task *In*: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* [Online]. Portland, Oregon, USA: Association for Computational Linguistics, pp.665–669. [Accessed 30 April 2021]. Available from: https://www.aclweb.org/anthology/P11-2117.

Crossley, S.A., Skalicky, S., Dascalu, M., McNamara, D.S. and Kyle, K. 2017. Predicting Text Comprehension, Processing, and Familiarity in Adult Readers: New Approaches to Readability Formulas. *Discourse Processes*. **54**(5–6), pp.340–359.

Crossley, S.A., Yang, H.S. and McNamara, D.S. 2014. What's so simple about simplified texts? A computational and psycholinguistic investigation of text comprehension and text processing. *Reading in a Foreign Language*. **26(1)**, pp.92–113.

Dale, E. and Chall, J.S. 1948. A Formula for Predicting Readability: Instructions. *Educational Research Bulletin*. **27**(2), pp.37–54.

Darwish, K. and Mubarak, H. 2016. Farasa: A New Fast and Accurate Arabic Word Segmenter *In*: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)* [Online]. Portorož, Slovenia: European Language Resources Association (ELRA), pp.1070–1074. [Accessed 3 January 2021]. Available from: https://www.aclweb.org/anthology/L16-1170.

Daud, N.M., Hassan, H. and Abdul Aziz, N. 2013. A Corpus-Based Readability Formula for Estimate of Arabic Texts Reading Difficulty. *World Applied Sciences Journal*. **21**, pp.168–173.

De Belder, J. and Moens, M.-F. 2012. A Dataset for the Evaluation of Lexical Simplification *In*: A. Gelbukh, ed. *Computational Linguistics and Intelligent Text Processing*. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp.426–437.

De Belder, J. and Moens, M.-F. 2010. Text simplification for children.

De Clercq, O., Hoste, V., Desmet, B., Van Oosten, P., De Cock, M. and Macken, L. 2014. Using the crowd for readability prediction. *Natural Language Engineering*. **20**(3), pp.293–325.

Deléger, L. and Zweigenbaum, P. 2009. Extracting Lay Paraphrases of Specialized Expressions from Monolingual Comparable Medical Corpora *In*: *Proceedings of the 2nd Workshop on Building and Using Comparable Corpora: from Parallel to Non-parallel Corpora (BUCC)* [Online]. Singapore: Association for Computational Linguistics, pp.2–10. [Accessed 13 November 2021]. Available from: https://aclanthology.org/W09-3102.

Dell'Orletta, F., Montemagni, S. and Venturi, G. 2011. READ–IT: Assessing Readability of Italian Texts with a View to Text Simplification *In*: *Proceedings of the Second Workshop on Speech and Language Processing for Assistive Technologies* [Online]. Edinburgh, Scotland, UK: Association for Computational Linguistics, pp.73–83. [Accessed 19 February 2022]. Available from: https://aclanthology.org/W11-2308.

Dell'Orletta, F., Wieling, M., Venturi, G., Cimino, A. and Montemagni, S. 2014. Assessing the Readability of Sentences: Which Corpora and Features? *In*: *Proceedings of the Ninth Workshop on Innovative Use of NLP for Building Educational Applications* [Online]. Baltimore, Maryland: Association for Computational Linguistics, pp.163–173. [Accessed 23 November 2019]. Available from: https://www.aclweb.org/anthology/W14-1820.

Denkowski, M. and Lavie, A. 2011. Meteor 1.3: Automatic Metric for Reliable Optimization and Evaluation of Machine Translation Systems *In*: *Proceedings of the Sixth Workshop on Statistical Machine Translation* [Online]. Edinburgh, Scotland: Association for Computational Linguistics, pp.85–91. [Accessed 6 November 2021]. Available from: https://aclanthology.org/W11-2107.

Deutsch, T., Jasbi, M. and Shieber, S. 2020. Linguistic Features for Readability Assessment. *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications.*, pp.1–17.

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*.

Devlin, S. and Unthank, G. 2006. Helping aphasic people process online information *In*: *Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility - Assets '06* [Online]. Portland, Oregon, USA: ACM Press, p.225. [Accessed 5 February 2022]. Available from: http://portal.acm.org/citation.cfm?doid=1168987.1169027.

Di Bari, M., Sharoff, S. and Thomas, M. 2014. Multiple views as aid to linguistic annotation error analysis *In*: *Proceedings of LAW VIII - The 8th Linguistic Annotation Workshop* [Online]. Dublin, Ireland: Association for

Computational Linguistics and Dublin City University, pp.82–86. [Accessed 15 November 2019]. Available from: https://www.aclweb.org/anthology/W14-4912.

Ding, H. and Balog, K. 2018. Generating Synthetic Data for Neural Keyword-to-Question Models *In*: *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval* [Online]., pp.51–58. [Accessed 15 November 2022]. Available from: http://arxiv.org/abs/1807.05324.

Doddington, G. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics *In*: *Proceedings of the second international conference on Human Language Technology Research* - [Online]. San Diego, California: Association for Computational Linguistics, p.138. [Accessed 6 November 2021]. Available from: http://portal.acm.org/citation.cfm?doid=1289189.1289273.

Dreyer, L.G. 1984. Readability and Responsibility. *Journal of Reading*. **27**(4), pp.334–338.

DuBay, W.H. 2004. *The Principles of Readability* [Online]. [Accessed 22 November 2019]. Available from: https://eric.ed.gov/?id=ED490073.

Elhadad, N. 2006. Comprehending Technical Texts: Predicting and Defining Unfamiliar Terms. *AMIA Annual Symposium Proceedings*. **2006**, pp.239–243.

Elhadad, N. and Sutaria, K. 2007. Mining a Lexicon of Technical Terms and Lay Equivalents *In*: *Biological, translational, and clinical language processing* [Online]. Prague, Czech Republic: Association for Computational Linguistics, pp.49–56. [Accessed 13 November 2021]. Available from: https://aclanthology.org/W07-1007.

Elkateb, S., Black, W., Vossen, P., Farwell, D., Rodríguez, H., Pease, A., Alkhalifa, M. and Fellbaum, C. 2006. Arabic WordNet and the Challenges of Arabic *In*: *Proceedings of the International Conference on the Challenge of Arabic for NLP/MT* [Online]. London, UK, pp.15–24. [Accessed 15 October 2022]. Available from: https://aclanthology.org/2006.bcs-1.2.

Evans, R. and Orăsan, C. 2019. Identifying signs of syntactic complexity for rule-based sentence simplification. *Natural Language Engineering*. **25**(1), pp.69–119.

Evans, R.J. 2011. Comparing methods for the syntactic simplification of sentences in information extraction. *Literary and Linguistic Computing*. **26**(4), pp.371–388.

Familiar, L. and Assaf, T. 2016. *Saud al-Sanousi's Saaq al-Bambuu: The Authorized Abridged Edition for Students of Arabic.* Georgetown University Press.

Farghaly, A. and Shaalan, K. 2009. Arabic Natural Language Processing: Challenges and Solutions. *ACM Transactions on Asian Language Information Processing*. **8**(4), 14:1-14:22.

Fehri, A.F. 2013. *Issues in the Structure of Arabic Clauses and Words*. Springer Science & Business Media.

Felice, M., Taslimipoor, S., Andersen, Ø.E. and Buttery, P. 2022. CEPOC: The Cambridge Exams Publishing Open Cloze dataset. , p.6.

Feng, L., Elhadad, N. and Huenerfauth, M. 2009. Cognitively Motivated Features for Readability Assessment *In*: *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)* [Online]. Athens, Greece: Association for Computational Linguistics, pp.229–237. [Accessed 5 March 2022]. Available from: https://aclanthology.org/E09-1027.

Feng, L., Jansche, M., Huenerfauth, M. and Elhadad, N. 2010. A Comparison of Features for Automatic Readability Assessment *In*: *Coling 2010: Poster Volume*., pages 276-284.

Flesch, R. 1948. A new readability yardstick. *Journal of Applied Psychology*. **32**(3), pp.221–233.

Flesch, R. 1979. How to Write Plain English. *University of Canterbury*. **Available at http://www. mang. canterbury. ac. nz/writing_guide/writing/flesch. shtml.[Retrieved 5 February 2016]**.

Forsyth, J. 2014. *Automatic Readability Prediction for Modern Standard Arabic*. Brigham Young University. Department of Linguistics and English Language.

Fouad, M. and Atyah, M. 2016. MLAR: Machine Learning based System for Measuring the Readability of Online Arabic News. *International Journal of Computer Applications*. **154**, pp.29–33.

François, T. 2014. An analysis of a French as a Foreign Language Corpus for Readability Assessment *In*: *Proceedings of the third workshop on NLP for computer-assisted language learning* [Online]. Uppsala, Sweden: LiU Electronic Press, pp.13–32. [Accessed 21 February 2022]. Available from: https://aclanthology.org/W14-3502.

François, T. and Fairon, C. 2012. An "AI readability" Formula for French as a Foreign Language *In*: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* [Online]. Jeju Island, Korea: Association for Computational Linguistics, pp.466–477. [Accessed 19 February 2022]. Available from: https://aclanthology.org/D12-1043.

Francois, T., Gala, N., Watrin, P. and Fairon, C. 2014. FLELex: a graded lexical resource for French foreign learners. , p.8.

François, T., Gala, N., Watrin, P. and Fairon, C. 2014. FLELex: a graded lexical resource for French foreign learners *In*: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)* [Online]. Reykjavik, Iceland: European Language Resources Association (ELRA), pp.3766–3773. [Accessed 13 November 2022]. Available from: http://www.lrec-conf.org/proceedings/lrec2014/pdf/1108_Paper.pdf.

François, T., Volodina, E., Pilán, I. and Tack, A. 2016. SVALex: a CEFR-graded Lexical Resource for Swedish Foreign and Second Language Learners *In*: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)* [Online]. Portorož, Slovenia: European Language Resources Association (ELRA), pp.213–219. [Accessed 20 February 2022]. Available from: https://aclanthology.org/L16-1032.

Freyhoff, G., Hess, G., Kerr, L., Tronbacke, B. and Van Der Veken, K. 1998. Make it Simple.

Fry, E. 1968. A Readability Formula That Saves Time. *Journal of Reading*. **11**(7), pp.513–578.

Gala, N., François, T. and Fairon, C. 2013. Towards a French lexicon with difficulty measures: NLP helping to bridge the gap between traditional dictionaries and specialized lexicons *In*: *eLex - Electronic Lexicography* [Online]. Tallin, Estonia. [Accessed 20 February 2022]. Available from: https://hal.archives-ouvertes.fr/hal-03194427.

Gala, N., Tack, A., Javourey-Drevet, L., François, T. and Ziegler, J.C. 2020. Alector: A Parallel Corpus of Simplified French Texts with Alignments of Misreadings by Poor and Dyslexic Readers *In*: *Language Resources and Evaluation for Language Technologies (LREC)* [Online]. Marseille, France. [Accessed 11 June 2021]. Available from: https://hal.archives-ouvertes.fr/hal-02503986.

Galley, M. and McKeown, K. 2003. Improving Word Sense Disambiguation in Lexical Chaining. *In Proceedings of the 18th International Joint Conference on Artificial Intelligence.*, p.3.

Ganitkevitch, J., Van Durme, B. and Callison-Burch, C. 2013. PPDB: The Paraphrase Database *In*: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* [Online]. Atlanta, Georgia: Association for Computational Linguistics, pp.758–764. [Accessed 13 November 2022]. Available from: https://aclanthology.org/N13-1092.

Gasperin, C., Maziero, E., Specia, L., Pardo, T. and Aluisio, R.M. 2009. Natural language processing for social inclusion: a text simplification architecture for different literacy levels. , pp.387–401.

Glavaš, G. and Štajner, S. 2015. Simplifying Lexical Simplification: Do We Need Simplified Corpora? *In*: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)* [Online]. Beijing, China: Association for Computational Linguistics, pp.63–68. [Accessed 26 April 2020]. Available from: https://www.aclweb.org/anthology/P15-2011.

Goldman, S.R. and Lee, C.D. 2014. Text complexity: State of the art and the conundrums it raises. *The Elementary School Journal*. **115(2)**, pp.290–300.

Gonzalez-Dios, I., Aranzabe, M.J., Díaz de Ilarraza, A. and Salaberri, H. 2014. Simple or Complex? Assessing the readability of Basque Texts *In*: *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers* [Online]. Dublin, Ireland: Dublin City University and Association for Computational Linguistics, pp.334–344. [Accessed 19 February 2022]. Available from: https://aclanthology.org/C14-1033.

Graesser, A.C., McNamara, D.S., Louwerse, M.M. and Cai, Z. 2004. Coh-Metrix: Analysis of text on cohesion and language. *Behavior Research Methods, Instruments, & Computers*. **36**(2), pp.193–202.

Grave, E., Bojanowski, P., Gupta, P., Joulin, A. and Mikolov, T. 2018. Learning Word Vectors for 157 Languages *In*: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)* [Online]. Miyazaki, Japan: European Language Resources Association (ELRA). [Accessed 22 November 2019]. Available from: https://www.aclweb.org/anthology/L18-1550.

Green, S. and Manning, C.D. 2010. Better Arabic Parsing: Baselines, Evaluations, and Analysis *In*: *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)* [Online]. Beijing, China: Coling 2010 Organizing Committee, pp.394–402. [Accessed 23 November 2022]. Available from: https://aclanthology.org/C10-1045.

Gunning, R. 1968. *The technique of clear writing.* Rev. ed. New York: McGraw-Hill.

Habash, N., Rambow, O. and Roth, R. 2009. MADA+TOKAN: A toolkit for Arabic tokenization, diacritization, morphological disambiguation, POS tagging, stemming and lemmatization. *Proceedings of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR).*

Habash, N. and Roth, R. 2009. CATiB: The Columbia Arabic Treebank *In*: *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers* [Online]. Suntec, Singapore: Association for Computational Linguistics, pp.221–224. [Accessed 23 November 2022]. Available from: https://aclanthology.org/P09-2056.

Habash, N.Y. 2010. *Introduction to Arabic Natural Language Processing*. Morgan & Claypool.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I.H. 2009. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*. **11**(1), pp.10–18.

Hancke, J., Vajjala, S. and Meurers, D. 2012. Readability Classification for German using Lexical, Syntactic, and Morphological Features *In*: *Proceedings of COLING 2012*. Mumbai, India: The COLING 2012 Organizing Committee, pp.1063–1080.

Hangya, V. and Fraser, A. 2019. Unsupervised Parallel Sentence Extraction with Parallel Segment Detection Helps Machine Translation *In*: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* [Online]. Florence, Italy: Association for Computational Linguistics, pp.1224–1234. [Accessed 7 May 2020]. Available from: https://www.aclweb.org/anthology/P19-1118.

Hartmann, N., Paetzold, G. and Aluísio, S. 2020. SIMPLEX-PB 2.0: A Reliable Dataset for Lexical Simplification in Brazilian Portuguese *In*:, pp.18–22. [Accessed 11 October 2021]. Available from: https://aclanthology.org/2020.winlp-1.6.

Hayes, J., Flower, L., Schriver, K., Stratman, J. and Carey, L. 1989. Cognitive Processes in Revision *In*: *Advances in Applied Psycholinguistics* [Online]. S. Rosenberg (ed.), Cambridge, England: Cambridge University Press, pp.176–240. [Accessed 11 March 2022]. Available from: https://www.cambridge.org/core/product/identifier/S0142716400008584/type/journal_article.

Heilman, M., Collins-Thompson, K., Callan, J. and Eskenazi, M. 2007. Combining Lexical and Grammatical Features to Improve Readability Measures for First and Second Language Texts *In*: *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference* [Online]. Rochester, New York: Association for Computational Linguistics, pp.460–467. [Accessed 19 February 2022]. Available from: https://aclanthology.org/N07-1058.

Heilman, M., Collins-Thompson, K. and Eskenazi, M. 2008. An Analysis of Statistical Models and Features for Reading Difficulty Prediction *In*: *Proceedings of the Third Workshop on Innovative Use of NLP for Building Educational Applications* [Online]. Columbus, Ohio: Association for Computational Linguistics, pp.71–79. [Accessed 21 February 2022]. Available from: https://aclanthology.org/W08-0909.

Hervás, R., Bautista, S., Rodríguez, M., de Salas, T., Vargas, A. and Gervás, P. 2014. Integration of lexical and syntactic simplification capabilities in a text editor. *Procedia Computer Science*. **27**, pp.94–103.

Horn, C., Manduca, C. and Kauchak, D. 2014. Learning a Lexical Simplifier Using Wikipedia *In*: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* [Online]. Baltimore, Maryland: Association for Computational Linguistics, pp.458–463. [Accessed 26 April 2020]. Available from: https://www.aclweb.org/anthology/P14-2075.

Howcroft, D.M. and Demberg, V. 2017. Psycholinguistic Models of Sentence Processing Improve Sentence Readability Ranking *In*: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers* [Online]. Valencia, Spain: Association for Computational Linguistics, pp.958–968. [Accessed 21 February 2022]. Available from: https://aclanthology.org/E17-1090.

Hudson, T. 2007. *Teaching second language reading*. Oxford: Oxford University Press.

Intellaren n.d. Intellibe: An Arabic to Latin text transcriber. [Accessed 15 October 2022]. Available from: http://www.intellaren.com/intellibe/doc.

Islam, Z., Mehler, A. and Rahman, R. 2012. Text Readability Classification of Textbooks of a Low-Resource Language *In*: *Proceedings of the 26th Pacific Asia Conference on Language, Information, and Computation* [Online]. Bali, Indonesia: Faculty of Computer Science, Universitas Indonesia, pp.545–553. [Accessed 19 February 2022]. Available from: https://aclanthology.org/Y12-1059.

Jiang, Z., Gu, Q., Yin, Y., Wang, J. and Chen, D. 2019. GRAW+: A two-view graph propagation method with word coupling for readability assessment. *Journal of the Association for Information Science and Technology*. **70**(5), pp.433–447.

Jin, D., Jin, Z., Hu, Z., Vechtomova, O. and Mihalcea, R. 2021. Deep Learning for Text Style Transfer: A Survey. *arXiv:2011.00416 [cs]*.

Johnson, J., Douze, M. and Jégou, H. 2017. Billion-scale similarity search with GPUs.

Kajiwara, T. and Fujita, A. 2017. Semantic Features Based on Word Alignments for Estimating Quality of Text Simplification *In*: *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)* [Online]. Taipei, Taiwan: Asian Federation of Natural Language Processing, pp.109–115. [Accessed 15 November 2022]. Available from: https://aclanthology.org/I17-2019.

Kajiwara, T. and Komachi, M. 2016. Building a Monolingual Parallel Corpus for Text Simplification Using Sentence Similarity Based on Alignment between Word Embeddings *In*: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers* [Online]. Osaka, Japan: The COLING 2016 Organizing Committee, pp.1147–1158. [Accessed 7 May 2021]. Available from: https://www.aclweb.org/anthology/C16-1109.

Kajiwara, T., Matsumoto, H. and Yamamoto, K. 2013. Selecting Proper Lexical Paraphrase for Children *In*: *Proceedings of the 25th Conference on Computational Linguistics and Speech Processing (ROCLING 2013)* [Online]. Kaohsiung, Taiwan: The Association for Computational Linguistics and Chinese Language Processing (ACLCLP), pp.59–73. [Accessed 13 November 2021]. Available from: https://aclanthology.org/O13-1007.

Kajiwara, T. and Yamamoto, K. 2015. Evaluation Dataset and System for Japanese Lexical Simplification *In*: *Proceedings of the ACL-IJCNLP 2015 Student Research Workshop* [Online]. Beijing, China: Association for Computational Linguistics, pp.35–40. [Accessed 13 November 2022]. Available from: https://aclanthology.org/P15-3006.

Kate, R., Luo, X., Patwardhan, S., Franz, M., Florian, R., Mooney, R., Roukos, S. and Welty, C. 2010. Learning to Predict Readability using Diverse Linguistic Features *In*: *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)* [Online]. Beijing, China: Coling 2010 Organizing Committee, pp.546–554. [Accessed 20 February 2022]. Available from: https://aclanthology.org/C10-1062.

Kauchak, D. 2013. Improving Text Simplification Language Modeling Using Unsimplified Text Data *In*: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* [Online]. Sofia, Bulgaria: Association for Computational Linguistics, pp.1537–1546. [Accessed 13 November 2021]. Available from: https://aclanthology.org/P13-1151.

Keskisärkkä, R. 2012. *Automatic Text Simplification via Synonym Replacement.*[Online] Linkoping University. [Accessed 12 November 2021]. Available from: https://www.semanticscholar.org/paper/Automatic-Text-Simplification-via-Synonym-Keskis%C3%A4rkk%C3%A4/17b2c96c192020e44ec9550e03c7abcd162415b5.

Kilgarriff, A., Charalabopoulou, F., Gavrilidou, M., Johannessen, J.B., Khalil, S., Johansson Kokkinakis, S., Lew, R., Sharoff, S., Vadlapudi, R. and Volodina, E. 2014. Corpus-based vocabulary lists for language learners for nine languages. *Language Resources and Evaluation*. **48**(1), pp.121–163.

Kim, J.Y., Collins-Thompson, K., Bennett, P.N. and Dumais, S.T. 2012. Characterizing web content, user interests, and search behavior by reading level and topic *In*: *Proceedings of the fifth ACM international conference on Web search and data mining - WSDM '12* [Online]. Seattle, Washington, USA: ACM Press, p.213. [Accessed 21 February 2022]. Available from: http://dl.acm.org/citation.cfm?doid=2124295.2124323.

Kincaid, J.P., Fishburne, J., Rogers, R.L. and Chissom, B.S. 1975. *Derivation of New Readability Formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel* [Online]. NAVAL TECHNICAL TRAINING COMMAND MILLINGTON TN RESEARCH BRANCH. [Accessed 8 November 2021]. Available from: https://apps.dtic.mil/sti/citations/ADA006655.

Kirkwood, K.J. and Wolfe, R.G. 1980. *Matching Students and Reading Materials: A Cloze-Procedure Method for Assessing the Reading Ability of Students and the Readability of Textual Materials*. Ontario Government Bookstore, 800 Bay St.

Kitson, H.D. 1921. *The mind of the buyer: A psychology of selling*. Macmillan.

Klaper, D., Ebling, S. and Volk, M. 2013. Building a German/Simple German Parallel Corpus for Automatic Text Simplification *In*: *Proceedings of the Second Workshop on Predicting and Improving Text Readability for Target Reader Populations* [Online]. Sofia, Bulgaria: Association for Computational Linguistics, pp.11–19. [Accessed 12 April 2022]. Available from: https://aclanthology.org/W13-2902.

Klare, G.R. 2000. The measurement of readability: useful information for communicators. *ACM Journal of Computer Documentation*. **24**(3), pp.107–121.

Klein, G., Kim, Y., Deng, Y., Senellart, J. and Rush, A. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation *In*: *Proceedings of ACL 2017, System Demonstrations* [Online]. Vancouver, Canada: Association for Computational Linguistics, pp.67–72. [Accessed 8 February 2023]. Available from: https://aclanthology.org/P17-4012.

Knowles, G. and Don, Z.M. 2004. The notion of a "lemma": Headwords, roots and lexical sets. *International Journal of Corpus Linguistics*. **9**(1), pp.69–81.

Kodaira, T., Kajiwara, T. and Komachi, M. 2016. Controlled and Balanced Dataset for Japanese Lexical Simplification *In*: *Proceedings of the ACL 2016 Student Research Workshop* [Online]. Berlin, Germany: Association for Computational Linguistics, pp.1–7. [Accessed 13 November 2022]. Available from: https://aclanthology.org/P16-3001.

Koehn, P. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. , p.8.

Kondru, J. 2006. *Using Part of Speech Structure of Text in the prediction of Its Readability*. The University of Texas at Arlington, U.S.

Kühberger, C., Bramann, C., Weiß, Z. and Meurers, D. 2019. Task complexity in history textbooks: A multidisciplinary case study on triangulation in history education research. *History Education Research Journal*. **16**(1).

Kuru, O. 2016. AI-KU at SemEval-2016 Task 11: Word Embeddings and Substring Features for Complex Word Identification *In*: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)* [Online]. San Diego, California: Association for Computational Linguistics, pp.1042–1046. [Accessed 20 November 2021]. Available from: https://aclanthology.org/S16-1163.

Lafourcade, M. 2007. Making people play for Lexical Acquisition with the JeuxDeMots prototype. *The 7th International Symposium on Natural Language Processing (SNLP'07).*, p.8.

Lapata, M. 2005. Automatic evaluation of text coherence: models and representations *In*: *In the Intl. Joint Conferences on Artificial Intelligence.*, pp.1085–1090.

Lebret, R.P. 2016. *Word Embeddings for Natural Language Processing*. Ecole Polytechnique Fédérale de Lausanne.

Leroy, G., Endicott, J.E., Kauchak, D., Mouradi, O. and Just, M. 2013. User Evaluation of the Effects of a Text Simplification Algorithm Using Term Familiarity on Perception, Understanding, Learning, and Information Retention. *Journal of Medical Internet Research*. **15**(7), p.e144.

Leroy, G., Kauchak, D. and Mouradi, O. 2013. A user-study measuring the effects of lexical simplification and coherence enhancement on perceived and actual text difficulty. *International journal of medical informatics*. **82**(8), pp.717–730.

Leroy, G., Miller, T., Rosemblat, G. and Browne, A. 2008. A balanced approach to health information evaluation: A vocabulary-based naïve Bayes classifier and readability formulas. *Journal of the American Society for Information Science and Technology*. **59**(9), pp.1409–1419.

Likert, R. 1932. A technique for the measurement of attitudes. *Archives of Psychology*. **22 140**, pp.55–55.

Ma, S. and Sun, X. 2017. A Semantic Relevance Based Neural Network for Text Summarization and Text Simplification. *arXiv:1710.02318 [cs]*.

Ma, Y., Fosler-Lussier, E. and Lofthus, R. 2012. Ranking-based readability assessment for early primary children's literature *In*: *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* [Online]. Montréal, Canada: Association for Computational Linguistics, pp.548–552. [Accessed 21 February 2022]. Available from: https://aclanthology.org/N12-1063.

Maamouri, M. and Bies, A. 2004. Developing an Arabic Treebank: Methods, Guidelines, Procedures, and Tools *In*: *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages* [Online].

Geneva, Switzerland: COLING, pp.2–9. [Accessed 23 November 2022]. Available from: https://aclanthology.org/W04-1602.

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. and McClosky, D. 2014. The Stanford CoreNLP Natural Language Processing Toolkit *In*: *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations* [Online]. Baltimore, Maryland: Association for Computational Linguistics, pp.55–60. [Accessed 23 November 2022]. Available from: https://aclanthology.org/P14-5010.

Martin, L., Fan, A., de la Clergerie, É., Bordes, A. and Sagot, B. 2022. MUSS: Multilingual Unsupervised Sentence Simplification by Mining Paraphrases *In*: *Proceedings of the 13th Conference on Language Resources and Evaluation* [Online]. Marseille, France: European Language Resources Association (ELRA), pp.1651–1664. [Accessed 7 July 2021]. Available from: http://arxiv.org/abs/2005.00352.

Martinc, M., Pollak, S. and Robnik-Šikonja, M. 2021. Supervised and Unsupervised Neural Approaches to Text Readability. *Computational Linguistics*. **47**(1), pp.141–179.

Marton, Y., Habash, N. and Rambow, O. 2013. Dependency Parsing of Modern Standard Arabic with Lexical and Inflectional Features. *Computational Linguistics*. **39**(1), pp.161–194.

Maskara, S. and Bhattacharyya, P. 2019. Recent works on Parallel Sentence Extraction from Comparable Corpora. , p.6.

Maurice, K. and Dickinson, G.J. 1990. *Rank Correlation Methods*. London: EdwardArnold.

Mc Laughlin, G.H. 1969. SMOG Grading-a New Readability Formula. *Journal of Reading*. **12**(8), pp.639–646.

McCarthy, D. and Navigli, R. 2007. SemEval-2007 Task 10: English Lexical Substitution Task *In*: *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)* [Online]. Prague, Czech Republic: Association for Computational Linguistics, pp.48–53. [Accessed 13 November 2022]. Available from: https://aclanthology.org/S07-1009.

McClure, G.M. 1987. Readability formulas: Useful or useless? *IEEE Transactions on Professional Communication*. **PC-30**(1), pp.12–15.

Meng, C., Chen, M., Mao, J. and Neville, J. 2020. ReadNet: A Hierarchical Transformer Framework for Web Article Readability Analysis *In*: J. M. Jose, E. Yilmaz, J. Magalhães, P. Castells, N. Ferro, M. J. Silva and F. Martins, eds. *Advances in Information Retrieval*. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp.33–49.

Michel, J.-B., Shen, Y., Aiden, A., Veres, A., Gray, M., Pickett, J., Hoiberg, D., Clancy, D., Norvig, P., Orwant, J., Pinker, S., Nowak, M. and Aiden, E. 2011.

Quantitative Analysis of Culture Using Millions of Digitized Books. *Science (New York, N.Y.)*. **331**, pp.176–82.

Mikolov, T., Chen, K., Corrado, G. and Dean, J. 2013. Efficient Estimation of Word Representations in Vector Space.

Mohammadi, H. and Khasteh, S.H. 2019. Text as Environment: A Deep Reinforcement Learning Text Readability Assessment Model. *arXiv:1912.05957 [cs]*.

Mukherjee, N., Patra, B.G., Das, D. and Bandyopadhyay, S. 2016. JU_NLP at SemEval-2016 Task 11: Identifying Complex Words in a Sentence *In*: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)* [Online]. San Diego, California: Association for Computational Linguistics, pp.986–990. [Accessed 20 November 2021]. Available from: https://aclanthology.org/S16-1152.

Naeem, N. n.d. al-Baheth al-Arabi | Aldaad Arabic Culture and Language Resources. [Accessed 13 October 2021]. Available from: https://resources.aldaad.org/resources/al-baheth-al-arabi/.

Narayan, S. and Gardent, C. 2014. Hybrid Simplification using Deep Semantics and Machine Translation *In*: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* [Online]. Baltimore, Maryland: Association for Computational Linguistics, pp.435–445. [Accessed 6 November 2021]. Available from: https://aclanthology.org/P14-1041.

Narayan, S. and Gardent, C. 2015. Unsupervised Sentence Simplification Using Deep Semantics. *arXiv:1507.08452 [cs]*.

Narayan, S., Gardent, C., Cohen, S.B. and Shimorina, A. 2017. Split and Rephrase. *arXiv:1707.06971 [cs]*.

Nassiri, N., Lakhouaja, A. and Cavalli-Sforza, V. 2018a. Arabic Readability Assessment for Foreign Language Learners *In*: M. Silberztein, F. Atigui, E. Kornyshova, E. Métais and F. Meziane, eds. *Natural Language Processing and Information Systems*. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp.480–488.

Nassiri, N., Lakhouaja, A. and Cavalli-Sforza, V. 2018b. Modern Standard Arabic Readability Prediction *In*: A. Lachkar, K. Bouzoubaa, A. Mazroui, A. Hamdani and A. Lekhouaja, eds. *Arabic Language Processing: From Theory to Practice*. Communications in Computer and Information Science. Springer International Publishing, pp.120–133.

Nassiri, N., Lakhouaja, A. and Cavalli-Sforza, V. 2018c. Modern Standard Arabic Readability Prediction *In*: A. Lachkar, K. Bouzoubaa, A. Mazroui, A. Hamdani and A. Lekhouaja, eds. *Arabic Language Processing: From Theory* *to Practice*. Communications in Computer and Information Science. Springer International Publishing, pp.120–133.

Nat, G. 2016. Sensible at SemEval-2016 Task 11: Neural Nonsense Mangled in Ensemble Mess *In*: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)* [Online]. San Diego, California: Association for Computational Linguistics, pp.963–968. [Accessed 25 April 2020]. Available from: https://www.aclweb.org/anthology/S16-1148.

Niklaus, C., Cetto, M., Freitas, A. and Handschuh, S. 2021. Context-Preserving Text Simplification. *arXiv:2105.11178 [cs]*.

Nisioi, S., Štajner, S., Ponzetto, S.P. and Dinu, L.P. 2017. Exploring Neural Text Simplification Models *In*: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* [Online]. Vancouver, Canada: Association for Computational Linguistics, pp.85–91. [Accessed 25 April 2020]. Available from: https://www.aclweb.org/anthology/P17-2014.

Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S. and Marsi, E. 2005. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*. **13**(2), p.95.

Nivre, J., de Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajič, J., Manning, C.D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R. and Zeman, D. 2016. Universal Dependencies v1: A Multilingual Treebank Collection *In*: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)* [Online]. Portorož, Slovenia: European Language Resources Association (ELRA), pp.1659–1666. [Accessed 23 November 2022]. Available from: https://aclanthology.org/L16-1262.

Östling, R. and Tiedemann, J. 2016. Efficient Word Alignment with Markov Chain Monte Carlo. *The Prague Bulletin of Mathematical Linguistics*. **106**(1), pp.125–146.

Paetzold, G. and Specia, L. 2016a. Benchmarking Lexical Simplification Systems *In*: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)* [Online]. Portorož, Slovenia: European Language Resources Association (ELRA), pp.3074–3080. [Accessed 26 April 2020]. Available from: https://www.aclweb.org/anthology/L16-1491.

Paetzold, G. and Specia, L. 2015. LEXenstein: A Framework for Lexical Simplification. , pp.85–90.

Paetzold, G. and Specia, L. 2016b. SemEval 2016 Task 11: Complex Word Identification *In*: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)* [Online]. San Diego, California: Association for Computational Linguistics, pp.560–569. [Accessed 25

April 2020]. Available from: https://www.aclweb.org/anthology/S16-1085.

Paetzold, G. and Specia, L. 2016c. Understanding the Lexical Simplification Needs of Non-Native Speakers of English *In*: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers* [Online]. Osaka, Japan: The COLING 2016 Organizing Committee, pp.717–727. [Accessed 25 April 2020]. Available from: https://www.aclweb.org/anthology/C16-1069.

Paetzold, G. and Specia, L. 2016d. Unsupervised Lexical Simplification for Non-Native Speakers. *Proceedings of the AAAI Conference on Artificial Intelligence*. **30**(1).

Paetzold, G. and Specia, L. 2016e. Unsupervised Lexical Simplification for Non-Native Speakers *In*:, p.7.

Paetzold, G.H. and Specia, L. 2017. A Survey on Lexical Simplification. *Journal of Artificial Intelligence Research*. **60**, pp.549–593.

Paetzold, G.H. and Specia, L. 2016. Vicinity-Driven Paragraph and Sentence Alignment for Comparable Corpora. *arXiv:1612.04113 [cs]*.

Papineni, K., Roukos, S., Ward, T. and Zhu, W.-J. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation *In*: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* [Online]. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, pp.311–318. [Accessed 6 November 2021]. Available from: https://aclanthology.org/P02-1040.

Pasha, A., Al-Badrashiny, M., Diab, M., El Kholy, A., Eskander, R., Habash, N., Pooleery, M., Rambow, O. and Roth, R. 2014. MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic *In*: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)* [Online]. Reykjavik, Iceland: European Language Resources Association (ELRA), pp.1094–1101. [Accessed 13 October 2021]. Available from: http://www.lrec-conf.org/proceedings/lrec2014/pdf/593_Paper.pdf.

Pennington, J., Socher, R. and Manning, C. 2014. Glove: Global Vectors for Word Representation *In*: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp.1532–1543.

Pera, M.S. and Ng, Y.-K. 2012. BReK12: a book recommender for K-12 users *In*: *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval - SIGIR '12* [Online]. Portland, Oregon, USA: ACM Press, p.1037. [Accessed 21 February 2022]. Available from: http://dl.acm.org/citation.cfm?doid=2348283.2348457.

Petersen, S.E. and Ostendorf, M. 2007. Text Simplification for Language Learners: A Corpus Analysis *In*: *In Proceedings of Workshop on Speech and Language Technology for Education*.

Pilán, I., Vajjala, S. and Volodina, E. 2016. A Readable Read: Automatic Assessment of Language Learning Materials based on Linguistic Complexity. *arXiv:1603.08868 [cs]*.

Pitler, E. and Nenkova, A. 2008. Revisiting Readability: A Unified Framework for Predicting Text Quality *In*: *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing* [Online]. Honolulu, Hawaii: Association for Computational Linguistics, pp.186–195. [Accessed 19 February 2022]. Available from: https://aclanthology.org/D08-1020.

Qiang, J. and Wu, X. 2021. Unsupervised Statistical Text Simplification. *IEEE Transactions on Knowledge and Data Engineering*. **33**(4), pp.1802–1806.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W. and Liu, P.J. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv:1910.10683 [cs, stat]*.

Řehůřek, R. and Sojka, P. 2010. Software Framework for Topic Modelling with Large Corpora *In*:, pp.45–50.

Richards, J.C. 1974. Word Lists: Problems and Prospects. *RELC Journal*. **5**(2), pp.69–84.

Rios, M., Aziz, W. and Specia, L. 2011. TINE: A Metric to Assess MT Adequacy *In*: *Proceedings of the Sixth Workshop on Statistical Machine Translation* [Online]. Edinburgh, Scotland: Association for Computational Linguistics, pp.116–122. [Accessed 6 November 2021]. Available from: https://aclanthology.org/W11-2112.

Rogers, A., Kovaleva, O. and Rumshisky, A. 2020. A Primer in BERTology: What we know about how BERT works. *arXiv:2002.12327 [cs]*.

Saddiki, H., Bouzoubaa, K. and Cavalli-Sforza, V. 2015. Text readability for Arabic as a foreign language *In*: *2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)*., pp.1–8.

Saddiki, H., Habash, N., Cavalli-Sforza, V. and Al Khalil, M. 2018. Feature Optimization for Predicting Readability of Arabic L1 and L2 *In*: *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*. Melbourne, Australia: Association for Computational Linguistics, pp.20–29.

Safaya, A., Abdullatif, M. and Yuret, D. 2020. KUISAIL at SemEval-2020 Task 12: BERT-CNN for Offensive Speech Identification in Social Media *In*: *Proceedings of the Fourteenth Workshop on Semantic Evaluation* [Online]. Barcelona (online): International Committee for Computational

Linguistics, pp.2054–2059. [Accessed 28 January 2021]. Available from: https://www.aclweb.org/anthology/2020.semeval-1.271.

Saggion, H. 2017. *Automatic Text Simplification*. Morgan & Claypool Publishers.

Saggion, H., Štajner, S., Bott, S., Mille, S., Rello, L. and Drndarevic, B. 2015. Making It Simplext: Implementation and Evaluation of a Text Simplification System for Spanish. *ACM Transactions on Accessible Computing*. **6**(4), pp.1–36.

Sato, S., Matsuyoshi, S. and Kondoh, Y. 2008. Automatic Assessment of Japanese Text Readability Based on a Textbook Corpus. , p.7.

Scarton, C., Oliveira, M., Candido Jr., A., Gasperin, C. and Aluísio, S. 2010. SIMPLIFICA: a tool for authoring simplified texts in Brazilian Portuguese guided by readability assessments *In*: *Proceedings of the NAACL HLT 2010 Demonstration Session* [Online]. Los Angeles, California: Association for Computational Linguistics, pp.41–44. [Accessed 13 October 2021]. Available from: https://aclanthology.org/N10-2011.

Scarton, C., Paetzold, G. and Specia, L. 2018. Text Simplification from Professionally Produced Corpora *In*: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)* [Online]. Miyazaki, Japan: European Language Resources Association (ELRA). [Accessed 11 October 2021]. Available from: https://aclanthology.org/L18-1553.

Schriver, K. 1990. Evaluating Text Quality: The Continuum From Text-Focused to Reader-Focused Methods. *Professional Communication, IEEE Transactions on*. **32**, pp.238–255.

Schriver, K.A. 1989. Evaluating text quality: the continuum from text-focused to reader-focused methods. *IEEE Transactions on Professional Communication*. **32**(4), pp.238–255.

Schumacher, E., Eskenazi, M., Frishkoff, G. and Collins-Thompson, K. 2016. Predicting the Relative Difficulty of Single Sentences With and Without Surrounding Context *In*: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* [Online]. Austin, Texas: Association for Computational Linguistics, pp.1871–1881. [Accessed 20 June 2020]. Available from: https://www.aclweb.org/anthology/D16-1192.

Schwarm, S. and Ostendorf, M. 2005. Reading Level Assessment Using Support Vector Machines and Statistical Language Models *In*: *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL’05)* [Online]. Ann Arbor, Michigan: Association for Computational Linguistics, pp.523–530. [Accessed 5 March 2022]. Available from: https://aclanthology.org/P05-1065.

Senter, R.J. and Smith, E.A. 1967. *AUTOMATED READABILITY INDEX* [Online]. CINCINNATI UNIV OH. [Accessed 28 February 2022]. Available from: https://apps.dtic.mil/sti/citations/AD0667273.

Shahrour, A., Khalifa, S., Taji, D. and Habash, N. 2016. CamelParser: A system for Arabic Syntactic Analysis and Morphological Disambiguation *In*: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations* [Online]. Osaka, Japan: The COLING 2016 Organizing Committee, pp.228–232. [Accessed 22 November 2019]. Available from: https://www.aclweb.org/anthology/C16-2048.

Shardlow, M. 2013a. A Comparison of Techniques to Automatically Identify Complex Words. *In*: *51st Annual Meeting of the Association for Computational Linguistics Proceedings of the Student Research Workshop* [Online]. Sofia, Bulgaria: Association for Computational Linguistics, pp.103–109. [Accessed 27 April 2020]. Available from: https://www.aclweb.org/anthology/P13-3015.

Shardlow, M. 2014. A Survey of Automated Text Simplification. *International Journal of Advanced Computer Science and Applications*. **4**(1).

Shardlow, M. 2013b. The CW Corpus: A New Resource for Evaluating the Identification of Complex Words *In*: *Proceedings of the Second Workshop on Predicting and Improving Text Readability for Target Reader Populations* [Online]. Sofia, Bulgaria: Association for Computational Linguistics, pp.69–77. [Accessed 13 November 2021]. Available from: https://aclanthology.org/W13-2908.

Shardlow, M. and Alva-Manchego, F. 2022. Simple TICO-19: A Dataset for Joint Translation and Simplification of COVID-19 Texts *In*: *Proceedings of the 13th Conference on Language Resources and Evaluation*. Marseille, France: European Language Resources Association (ELRA), pp.3093–3102.

Sharoff, S. 2021. Genre annotation for the Web: Text-external and text-internal perspectives. *Register Studies*. **3**(1), pp.1–32.

Sharoff, S. 2006. Open-source Corpora: Using the net to fish for linguistic data. *International Journal of Corpus Linguistics*. **11**, pp.435–462.

Sharoff, S., Umanskaya, E. and Wilson, J. 2014. *A Frequency Dictionary of Contemporary Russian Core Vocabulary for Learners*. Routledge.

Sheehan, K.M. 2017. Validating Automated Measures of Text Complexity. *Educational Measurement: Issues and Practice*. **36**(4), pp.35–43.

Sheehan, K.M., Kostin, I., Napolitano, D. and Flor, M. 2015. The TextEvaluator Tool. *The Elementary School Journal*.

Shen, W., Williams, J., Marius, T. and Salesky, E. 2013. *A Language-Independent Approach to Automatic Text Difficulty Assessment for Second-Language*

Learners: [Online]. Fort Belvoir, VA: Defense Technical Information Center. [Accessed 20 February 2022]. Available from: http://www.dtic.mil/docs/citations/ADA595522.

Siddharthan, A. 2002. An Architecture for a Text Simplification System *In*: *Proceedings of the Language Engineering Conference (LEC'02)*. LEC '02. USA: IEEE Computer Society, p.64.

Siddharthan, A. 2004. Syntactic Simplification and Text Cohesion.

Sikka, P. and Mago, V. 2020. A Survey on Text Simplification. *arXiv:2008.08612 [cs]*.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L. and Makhoul, J. 2006. A Study of Translation Edit Rate with Targeted Human Annotation *In*: *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers* [Online]. Cambridge, Massachusetts, USA: Association for Machine Translation in the Americas, pp.223–231. [Accessed 6 November 2021]. Available from: https://aclanthology.org/2006.amta-papers.25.

Snover, M., Madnani, N., Dorr, B. and Schwartz, R. 2009. Fluency, Adequacy, or HTER? Exploring Different Human Judgments with a Tunable MT Metric *In*: *Proceedings of the Fourth Workshop on Statistical Machine Translation* [Online]. Athens, Greece: Association for Computational Linguistics, pp.259–268. [Accessed 6 November 2021]. Available from: https://aclanthology.org/W09-0441.

Soliman, R. 2018. The Implementation of the Common European Framework of Reference for the Teaching and Learning of Arabic as a Second Language in Higher Education *In*: K. M. Wahba, L. England and Z. A. Taha, eds. *Handbook for Arabic Language Teaching Professionals in the 21st Century*. New York: Routledge, pp.118–137.

Soricut, R. and Marcu, D. 2006. Discourse Generation Using Utility-Trained Coherence Models *In*: *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions* [Online]. Sydney, Australia: Association for Computational Linguistics, pp.803–810. [Accessed 5 March 2022]. Available from: https://aclanthology.org/P06-2103.

S.P, S., Kumar M, A. and K P, S. 2016. AmritaCEN at SemEval-2016 Task 11: Complex Word Identification using Word Embedding *In*: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)* [Online]. San Diego, California: Association for Computational Linguistics, pp.1022–1027. [Accessed 20 November 2021]. Available from: https://aclanthology.org/S16-1159.

Specia, L. 2010. Translating from Complex to Simplified Sentences *In*: T. A. S. Pardo, A. Branco, A. Klautau, R. Vieira and V. L. S. de Lima, eds.

Computational Processing of the Portuguese Language. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp.30–39.

Specia, L., Jauhar, S. and Mihalcea, R. 2012. *SemEval-2012 task 1: English Lexical Simplification*.

Štajner, S. and Glavaš, G. 2017. Leveraging event-based semantics for automated text simplification. *Expert Systems with Applications: An International Journal*. **82**(C), pp.383–395.

Štajner, S., Mitkov, R. and Saggion, H. 2014. One Step Closer to Automatic Evaluation of Text Simplification Systems *In*: *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)* [Online]. Gothenburg, Sweden: Association for Computational Linguistics, pp.1–10. [Accessed 6 November 2021]. Available from: https://aclanthology.org/W14-1201.

Štajner, S. and Nisioi, S. 2018. A Detailed Evaluation of Neural Sequence-to-Sequence Models for In-domain and Cross-domain Text Simplification *In*: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)* [Online]. Miyazaki, Japan: European Language Resources Association (ELRA). [Accessed 19 February 2022]. Available from: https://aclanthology.org/L18-1479.

Štajner, S., Saggion, H. and Ponzetto, S.P. 2019. Improving lexical coverage of text simplification systems for Spanish. *Expert Systems with Applications*. **118**, pp.80–91.

Straka, M. and Straková, J. 2017. Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe *In*: *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies* [Online]. Vancouver, Canada: Association for Computational Linguistics, pp.88–99. [Accessed 23 November 2022]. Available from: http://aclweb.org/anthology/K17-3009.

Stymne, S., Tiedemann, J., Hardmeier, C. and Nivre, J. 2013. Statistical Machine Translation with Readability Constraints *In*: *Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013)* [Online]. Oslo, Norway: Linköping University Electronic Press, Sweden, pp.375–386. [Accessed 12 April 2022]. Available from: https://aclanthology.org/W13-5634.

Sulem, E., Abend, O. and Rappoport, A. 2018a. BLEU is Not Suitable for the Evaluation of Text Simplification *In*: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* [Online]. Brussels, Belgium: Association for Computational Linguistics, pp.738–744. [Accessed 7 July 2021]. Available from: http://aclweb.org/anthology/D18-1081.

Sulem, E., Abend, O. and Rappoport, A. 2018b. Semantic Structural Evaluation for Text Simplification *In*: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* [Online]. Stroudsburg, PA, USA, pp.685–696. [Accessed 11 October 2021]. Available from: http://arxiv.org/abs/1810.05022.

Sulem, E., Abend, O. and Rappoport, A. 2018c. Simple and Effective Text Simplification Using Semantic and Neural Methods *In*: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* [Online]. Melbourne, Australia: Association for Computational Linguistics, pp.162–173. [Accessed 30 April 2021]. Available from: http://aclweb.org/anthology/P18-1016.

Sun, H. and Zhou, M. 2012. Joint Learning of a Dual SMT System for Paraphrase Generation *In*: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* [Online]. Jeju Island, Korea: Association for Computational Linguistics, pp.38–42. [Accessed 8 November 2021]. Available from: https://aclanthology.org/P12-2008.

Surya, S., Mishra, A., Laha, A., Jain, P. and Sankaranarayanan, K. 2019. Unsupervised Neural Text Simplification *In*: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* [Online]. Florence, Italy: Association for Computational Linguistics, pp.2058–2068. [Accessed 8 July 2021]. Available from: https://www.aclweb.org/anthology/P19-1198.

Tamimi, A.K.A., Jaradat, M., Al-Jarrah, N. and Ghanem, S. 2014. AARI: automatic arabic readability index. *Int. Arab J. Inf. Technol.* **11**, pp.370–378.

Thibadeau, R., Just, M.A. and Carpenter, P.A. 1982. A Model of the Time Course and Content of Reading*. *Cognitive Science*. **6**(2), pp.157–203.

Thorndike, E.L. 1921. The teacher’s word book. , p.140.

Tonelli, S., Palmero Aprosio, A. and Saltori, F. 2016. SIMPITIKI: a Simplification corpus for Italian *In*: A. Corazza, S. Montemagni and G. Semeraro, eds. *Proceedings of the Third Italian Conference on Computational Linguistics CLiC-it 2016* [Online]. Accademia University Press, pp.291–296. [Accessed 23 April 2020]. Available from: http://books.openedition.org/aaccademia/1855.

Toutanova, K., Brockett, C., Tran, K.M. and Amershi, S. 2016. A Dataset and Evaluation Metrics for Abstractive Compression of Sentences and Short Paragraphs *In*: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* [Online]. Austin, Texas: Association for Computational Linguistics, pp.340–350. [Accessed 8 November 2021]. Available from: https://aclanthology.org/D16-1033.

Tschirner, E., Bärenfänger, O. and Wisniewski, K. 2015. *Assessing Evidence of validity of the ACTFL CEFR Reading Profiencey TEST (lpt and rpt) usinga standard-setting approach*. Alexandria, VA: Leipzig: Institute for Test Research and Test Development.

Vaidya, P. 2014. Decoding in Statistical Machine Translation Using Moses And Cygwin on Windows. *International Journal of Engineering Research*. **3**(2), p.5.

Vajjala, S. 2021. Trends, Limitations and Open Challenges in Automatic Readability Assessment Research. *arXiv:2105.00973 [cs]*.

Vajjala, S. and Lõo, K. 2014. Automatic CEFR Level Prediction for Estonian Learner Text *In*: *Proceedings of the third workshop on NLP for computer-assisted language learning*. Uppsala, Sweden: LiU Electronic Press, pp.113–127.

Vajjala, S. and Lucic, I. 2019. On Understanding the Relation between Expert Annotations of Text Readability and Target Reader Comprehension *In*: *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications* [Online]. Florence, Italy: Association for Computational Linguistics, pp.349–359. [Accessed 21 February 2022]. Available from: https://aclanthology.org/W19-4437.

Vajjala, S. and Lučić, I. 2018. OneStopEnglish corpus: A new corpus for automatic readability assessment and text simplification *In*: *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications* [Online]. New Orleans, Louisiana: Association for Computational Linguistics, pp.297–304. [Accessed 19 February 2022]. Available from: https://aclanthology.org/W18-0535.

Vajjala, S. and Meurers, D. 2014a. Exploring Measures of "Readability" for Spoken Language: Analyzing linguistic features of subtitles to identify age-specific TV programs *In*: *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)* [Online]. Gothenburg, Sweden: Association for Computational Linguistics, pp.21–29. [Accessed 19 February 2022]. Available from: https://aclanthology.org/W14-1203.

Vajjala, S. and Meurers, D. 2012. On Improving the Accuracy of Readability Classification using Insights from Second Language Acquisition *In*: *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP* [Online]. Montréal, Canada: Association for Computational Linguistics, pp.163–173. [Accessed 19 February 2022]. Available from: https://aclanthology.org/W12-2019.

Vajjala, S. and Meurers, D. 2013. On The Applicability of Readability Models to Web Texts *In*: *Proceedings of the Second Workshop on Predicting and Improving Text Readability for Target Reader Populations* [Online]. Sofia, Bulgaria: Association for Computational Linguistics, pp.59–68. [Accessed 19 February 2022]. Available from: https://aclanthology.org/W13-2907.

Vajjala, S. and Meurers, D. 2014b. Readability Assessment for Text Simplification: From Analyzing Documents to Identifying Sentential Simplifications. *ITL-International Journal of Applied Linguistics*. **165(2)**, pp.194–222.

Vajjala, S., Meurers, D., Eitel, A. and Scheiter, K. 2016. Towards grounding computational linguistic approaches to readability: Modeling reader-text interaction for easy and difficult texts *In*: *Proceedings of the Workshop on Computational Linguistics for Linguistic Complexity (CL4LC)* [Online]. Osaka, Japan: The COLING 2016 Organizing Committee, pp.38–48. [Accessed 21 February 2022]. Available from: https://aclanthology.org/W16-4105.

Valencia, S.W., Wixson, K.K. and Pearson, P.D. 2014. Putting text complexity in context: Refocusing on comprehension of complex text. *The Elementary School Journal*. **115(2)**, pp.270–289.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. and Polosukhin, I. 2017. Attention Is All You Need. *arXiv:1706.03762 [cs]*.

Vickrey, D. and Koller, D. 2008. Sentence Simplification for Semantic Role Labeling *In*: *Proceedings of ACL-08: HLT* [Online]. Columbus, Ohio: Association for Computational Linguistics, pp.344–352. [Accessed 2 February 2022]. Available from: https://aclanthology.org/P08-1040.

Vogel, M. and Washburne, C. 1928. An Objective Method of Determining Grade Placement of Children's Reading Material. *The Elementary School Journal*. **28**(5), pp.373–381.

Vossen, P. (ed.). 1998. *In*: *EuroWordNet: a multilingual database with lexical semantic networks for European Languages*. Kluwer Academic Publishers.

Vu, T., Hu, B., Munkhdalai, T. and Yu, H. 2018. Sentence Simplification with Memory-Augmented Neural Networks *In*: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)* [Online]. New Orleans, Louisiana: Association for Computational Linguistics, pp.79–85. [Accessed 28 April 2020]. Available from: https://www.aclweb.org/anthology/N18-2013.

Wang, T., Chen, P., Amaral, K. and Qiang, J. 2016. An Experimental Study of LSTM Encoder-Decoder Model for Text Simplification. *arXiv:1609.03663 [cs]*.

Wang, T., Chen, P., Rochford, J. and Qiang, J. 2016. Text Simplification Using Neural Machine Translation.

Watanabe, W., Júnior, A.C., Uzêda, V.R. de, Fortes, R., Pardo, T. and Aluísio, S. 2009. Facilita: reading assistance for low-literacy readers *In*: *SIGDOC '09*.

Wikimedia 2017. Wiktionary. [Accessed 20 November 2021]. Available from: https://www.wiktionary.org/.

Wilkens, R., Zilio, L., Cordeiro, S.R., Paula, F., Ramisch, C., Idiart, M. and Villavicencio, A. 2017. LexSubNC: A Dataset of Lexical Substitution for Nominal Compounds *In*: *IWCS 2017 — 12th International Conference on Computational Semantics — Short papers* [Online]. [Accessed 13 November 2022]. Available from: https://aclanthology.org/W17-6941.

Woodsend, K. and Lapata, M. 2011. Learning to Simplify Sentences with Quasi-Synchronous Grammar and Integer Programming *In*: *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing* [Online]. Edinburgh, Scotland, UK.: Association for Computational Linguistics, pp.409–420. [Accessed 6 November 2021]. Available from: https://aclanthology.org/D11-1038.

Wróbel, K. 2016. PLUJAGH at SemEval-2016 Task 11: Simple System for Complex Word Identification *In*: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)* [Online]. San Diego, California: Association for Computational Linguistics, pp.953–957. [Accessed 13 November 2021]. Available from: https://aclanthology.org/S16-1146.

Wubben, S., van den Bosch, A. and Krahmer, E. 2012. Sentence Simplification by Monolingual Machine Translation *In*: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* [Online]. Jeju Island, Korea: Association for Computational Linguistics, pp.1015–1024. [Accessed 23 April 2020]. Available from: https://www.aclweb.org/anthology/P12-1107.

Xia, M., Kochmar, E. and Briscoe, T. 2016. Text Readability Assessment for Second Language Learners. *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications.*, pp.12–22.

Xu, W., Callison-Burch, C. and Napoles, C. 2015. Problems in Current Text Simplification Research: New Data Can Help. *Transactions of the Association for Computational Linguistics*. **3**, pp.283–297.

Xu, W., Napoles, C., Pavlick, E., Chen, Q. and Callison-Burch, C. 2016. Optimizing Statistical Machine Translation for Text Simplification. *Transactions of the Association for Computational Linguistics*. **4**, pp.401–415.

Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A. and Raffel, C. 2021. mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer *In*: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* [Online]. Online: Association for Computational Linguistics, pp.483–498. [Accessed 28 September 2021]. Available from: https://aclanthology.org/2021.naacl-main.41.

Yimam, S.M., Biemann, C., Malmasi, S., Paetzold, G., Specia, L., Štajner, S., Tack, A. and Zampieri, M. 2018. A Report on the Complex Word Identification Shared Task 2018 *In*: *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications* [Online]. New Orleans, Louisiana: Association for Computational Linguistics, pp.66–78. [Accessed 25 April 2020]. Available from: https://www.aclweb.org/anthology/W18-0507.

Zamanian, M. and Heydari, P. 2012. Readability of Texts: State of the Art. *Theory and Practice in Language Studies*. **Vol. 2**, pp.43–53.

Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q. and Artzi, Y. 2020. BERTScore: Evaluating Text Generation with BERT. *arXiv:1904.09675 [cs]*.

Zhang, X. and Lapata, M. 2017. Sentence Simplification with Deep Reinforcement Learning *In*: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* [Online]. Copenhagen, Denmark: Association for Computational Linguistics, pp.584–594. [Accessed 28 April 2020]. Available from: https://www.aclweb.org/anthology/D17-1062.

Zhang, Y., Ye, Z., Feng, Y., Zhao, D. and Yan, R. 2017. A Constrained Sequence-to-Sequence Neural Model for Sentence Simplification. *arXiv:1704.02312 [cs]*.

Zhu, Z., Bernhard, D. and Gurevych, I. 2010. A Monolingual Tree-based Translation Model for Sentence Simplification *In*: *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)* [Online]. Beijing, China: Coling 2010 Organizing Committee, pp.1353–1361. [Accessed 13 October 2021]. Available from: https://aclanthology.org/C10-1152.

Appendix A

Dataset Two: complex- simple parallel corpus snapshot

Snapshot of the parallel sentences extracted from the novel “Saqq Al-Bambuu” (Al-Sanousi, 2013), compared to the authorised simplified version for students of Arabic as a second language (Familiar and Assaf, 2016)

Complex (Al-Sanousi, 2013)	Simple (Familiar and Assaf, 2016)
المترجم إبراهيم سلام، يعمل في حقل الترجمة.	المترجم ابراهيم سلام، يعمل في الترجمة.
يجيد إلى جانب اللغة الفلبينية كلا من الغتين العربية والإنكليزية.	يجيد، بالإضافة إلى اللغة الفلبينية اللغتين العربية والإنكليزية.
ولد في مندناو لعائلة مسلمة جنوب الفلبين.	ولد في مندناو، لعائلة مسلمة، جنوب الفلبين.
تلقى هناك دروسا في العربية لدى معهد الدراسات الإسلامية في مانيلا، وحصل على منحة دراسية من قبل اللجنة الوطنية الكويتة للتربية والعلوم والثقافة ليتلقى تعليمه في المعهد الديني في الكويت.	هناك درس العربية في معهد الدراسات الإسلامية في مانيلا وحصل على منحة دراسية ليدرس في المعهد الديني في الكويت.
التحق بجامعة الكويت، كلية الآداب، متخرجا فيها حاصلا على ليسانس لغة عربية.	التحق بجامعة الكويت، كلية الآداب وحصل منها على ليسانس لغة عربية.
يعمل حاليا بوظيفة مترجم في سفارة جمهورية الفلبين لدى الكويت.	يعمل حاليا بوظيفة مترجم في سفارة جمهورية الفلبين في الكويت.
أقام دورات وبرامج في اللغة العريبة والثقافة الإسلامية للمهتدين الجدد في المركز الكوتي الفلبيني الثقافي.	أقام برامج في اللغة العربية والثقافة الإسلامية في المركز الكويتي الفلبيني الثقافي.
عمل، ولا يزال، على ترجمة الأخبار التي تخص الجالية الفلبينية، المنشورة في الصحف الكويتية، وإعادة نشرها في الصحف الفلبيية ك: bulletin Newspaper manila Philippine star Philippine daily inquirer.	عمل، وما زال، على ترجمة أخبار الجالية الفلبينية المنشورة في الصحف الكويتية وإعادة نشرها في الصحف الفلبينية ك:Newspaper Manila Bulletin، Philippine star، و Philippine Daily Inquirer.
كلمة المترجم ترجمتي لهذه الأوراق لا تعني بالضرورة موافقتي على كل ما جاء فيها.	كلمة المترجم ترجمتي لهذه الأوراق لا تعني موافقتي على كل ما جاء فيها.
مهمتي هنا، وإن كنت أشغل حيزا بشخصيتي الحقيقية، في هذا العمل، لا تتعدى تحويل كلمات النص من اللغة الفلبينية إلى اللغة العربية بناء على طلب الكاتب.	مهمتي فقط تحويل كلمات النص من اللغة الفلبينية إلى اللغة العربية.
لكل لغة خصوصيتها، ولأن اللغة جزء من ثقافة الشعوب، والثقافات وإن تشابهت فيما بينها فلابد أن يتفرد بعضها بما يميزه عن بعضها الآخر لهذا وجدتني أمام الكثر من المفردات الفلبينة التي ليس لها مرادف دقيق في العربية.	لكل لغة خصوصيتها، ولأن اللغة جزه من ثقافة الشعوب، وجدتني أمام الكثير من المفردات الفلبينية التي ليس لها ترجمة في العربية.

خصوصا تلك المفردات المحلية أو الشعبية التي لا توجد في الثقافات الأخرى.	خصوصا تلك المفردات الغارقة بالمحلية أو الشعبية التي لا توجد في الثقافات الأخرى.
ورغم معرفتي وعشقي للعربية لغة القرآن الكريم، فكانت ترجمة بعض المفردات صعبةفأسأل الله أن أكون قد نجحت في ذلك.	ورغم اتقاني وعشقي العربية لغة القرآن الكريم: فقد وجدتني في مأزق أمام تلك المفردات، ما جعلني أتصرف في كثير من العبارات الواردة في هذا النص بشكل يكاد يطابق المعنى الحرفي لها، وأسأل الله أن أكون قد وفقت في ذلك.
في البرتغالية كتب بنفس الحروف، ولكنه ينطق جوزيه.	وفي البرتغالية بالحروف ذاتها يكتب، ولكنه ينطق جوزيه.