Deanship of Graduate Studies

Al-Quds University

Towards Automated Arabic Synonyms Extraction

Eman Abed Al-Kareem Mousa Naser

PhD Dissertation

Jerusalem-Palestine

1446/2025

Towards Automated Arabic Synonyms Extraction

Prepared by:

Eman Abed Al-Kareem Mousa Naser

PhD: Information Technology Engineering
Al-Quds University-Palestine

Supervisor: Prof.Nabil Arman

This dissertation proposal was submitted in Partial Fulfillment of the Requirements for the PhD In Engineering of Information Technology Joint PhD Program Al-Quds, AAUP, and PPU
Deanship of Scientific Research and Graduate Studies

1446/2025

Al-Quds University
Deanship of Graduate Studies
Information Technology Engineering

Dissertation Approval

Towards Automated Arabic Synonyms Extraction

Prepared by : Eman Abed Al-Kareem Mousa Naser
Registration No: No.21912541

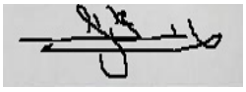
Supervisor : Prof.Nabil Arman

PhD dissertation submitted and accepted. Data 11/ 1 / 2025
The names and signatures of the examining committee members are as follows:

Head of the Committee, Prof. Nabil Arman Signatures 

Interior Examiner Dr. Diya AbuZeina Signatures 

External Examiner Dr. Ahmad Al-Taani Signatures 

External Examiner Dr. Hassan Al-Tarawneh Signatures 

Jerusalem-Palestine

1446/2025

Dedication

I lovingly dedicate my success to my dear husband, Ahmad, whose unwavering support and encouragement have been the base of this journey. Your patience, understanding, and belief in me are my source of strength.

I dedicate this success with love to my wonderful children, Layth and Jannah. Thank you for your encouragement and patience.

With all my love, I dedicate this success to my mom, Teacher Najwah, and dad, Teacher Abed Al-Kareem. Thank you for your support and belief in me.

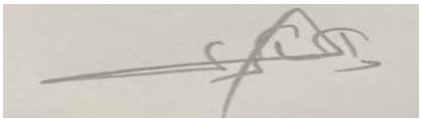
I also dedicate this achievement to my brothers, Dr. Musa and Dr. Mohammad, as well as sisters Sana, Bayan, and Dua. Thanks for Your encouragement

With deep love, I dedicate this success to the soul of my beloved sister, Dr. Rawan. I wish that you had been with me this day.

Thank you all for being my foundation and my inspiration.

Declaration

I certify that this thesis submitted for the degree of PhD is the result of my own research, except where otherwise acknowledged, and that this study (or any part of the same) has not been submitted for a higher degree to any other university or Institution.

Signed 

Eman Abed Al-Kareem Mousa Naser

Data: 11 /01/2025

Acknowledgments

First and foremost, I express my deepest gratitude to the Almighty Allah, the Most Gracious and Most Merciful. His boundless grace, blessings, and guidance have illuminated every step of my journey, providing me with strength and perseverance.

I want to thank my honorable supervisor, Professor Nabil Arman, for his support while working on my PhD thesis.I want to express my sincere gratitude to the program committee, which opened this unique specialization in our country and provided us with this valuable opportunity.

I am deeply grateful to all the teaching staff for the valuable information that was a significant reason for our establishment to embark on the doctoral thesis journey.

I would also like to thank Al-Quds University, represented by the Dean of Scientific Research, for providing the publication fees for scientific research for the doctoral degree.

Lastly, my heartfelt gratitude goes to my husband. His unwavering support, encouragement, and understanding have been my strength. Through his patience and belief in my abilities, he has empowered me to overcome challenges and pursue my aspirations with renewed vigor and confidence.

Abstract

Synonyms extraction gains special attention as synonyms are essential in improving Natural Language Processing (NLP) application performance. The Lexical Substitution (LS) task is utilized for Synonym extraction, which generates a set of equivalent substitutions (i.e., synonyms) to the target word or phrase in a sentence that saves the sentence's meaning. This task can enhance writing, language understanding, and NLP models and address ambiguity. Recently, LS has attracted much attention in many languages. Despite the richness of Arabic vocabulary, limited research has been performed on the LS task due to the lack of annotated data. To bridge this gap, we present the first Arabic LS benchmark dataset, AraLexSubD for benchmarking LS pipelines. AraLexSubD is manually built by eight native Arabic speakers and linguists (six linguist annotators, a doctor, and an economist) who annotate the 630 sentences. AraLexSubD covers three domains: general, finance, and medical. It encompasses 2476 substitution candidates ranked according to their semantic relatedness.

We also present an Arabic LS pipeline, AraLexSubPro, which offers different techniques for generating, selecting, and ranking substitutions. To make a thorough comparison, AraLexSubPro uses four different methods as baselines to generate substitute candidates for the target words: a synonym dictionary-based approach using Arabic Word Net (AWN), a pre-trained language model-based approach (AraBERT), AraBERT dropout (partial masking), and a hybrid approach between AraBERT and AWN. The results showed that the hybrid approach achieved the best results compared to the other approaches. The generated substitutions are filtered and then ranked based on six high-quality features to compare thoroughly: word similarity, word frequency, BERT prediction order (BERT probability), BERT-based language model (Loss), BERT similarity, and the BERTscore. The substitutions are then reranked based on our AraLexSubPro ranker. Additionally, an error analysis of the experiment is reported.

To evaluate the AraLexSubPro pipeline, we use our first benchmark dataset for the Arabic LS task AraLexSubD dataset, which can automatically evaluate the Arabic LS systems. To our knowledge, this is the first study on Arabic lexical substitution. The results were encouraging and fundamental for Arabic LS research. To speed up research on this field, we have put the AraLexSubD data on GitHub at the following link:
https://github.com/karajah2024/Arabic-Lexical-Substitution.git

List of Abbreviations

Abbreviation	Description
AlaSca	Automatically large-scale
AraBERT	Arabic Bidirectional Encoder Representations Transformer
AraLexSubD	Arabic Lexical Substitution Dataset
AraLexSubPro	Arabic Lexical Substitution Pipeline
AWN	Arabic WordNet
BERT	Bidirectional Encoder Representation Transformer
CHNLS	Chinese Lexical Substitution
CILex	Context Information for Lexical
CoInCo	Concepts in Context
CQC	Cyclic and Quasi-Cyclic
GENESIS	Generating Substitutes in Contexts
GPT-4	Generative pre-trained transformer 4
IDF	Inverse Document Frequency
LexSubCon	Lexical Substitution based on Contextual embedding
LS	Lexical Substitution
MSA	Modern Standard Arabic
NADIA	News Articles Dataset in Arabic
NLP	Natural Language Processing
PoS	Part of Speech
ProLex	Proficiency Lexical Substitution
PWN	Princeton WordNet
QAWN	Quranic AWN
RDF	Resource Description Framework
RMS	Root Mean Square
SWORD	Stanford Word Substitution
TWSI	Turk Word Sense Inventory
VSM	Vector Space Model
WN	

List of Figures

2.1	Example of a translation graph [6]	7
2.2	Cycles and quasi-cycles construction [17]	7
3.1	An example of a sentence in the general dataset	22
3.2	The same sentence with a candidate scored by one of the linguists	22
3.3	The fuzzy scoring scale–synonymy strength.	23
4.1	Arabic Lexical Substitution pipeline (AraLexSubPro)	28
4.2	The substitution generation of AraLexSub for the target word prediction	29
5.1	The chart of the F1 scores for the AraLexSubPro generation methods	36
5.2	The chart of the F1 scores of the filtring evaluation results for the AWN	37
5.3	The chart of the F1 scores of the filtring results for AraBERT	38
5.4	The chart of the F1 scores of the filtring results for AraBERT dropout	39
5.5	The chart of the F1 scores of the filtring results for Hybrid	40

List of Tables

2.1	A comparison between the different synonyms extraction approaches	14
2.2	A comparison between the LS Datasets	19
3.1	The final dataset instances are in the AraLexSubD dataset	24
3.2	PoS tags in the AraLexSubD dataset	24
3.3	The Root Mean Squared Error (RMSE) between the scores of each linguist and the average scores of all linguists	25
4.1	The medical and financial documents from the NADIA Dataset	32
5.1	Automatic evaluation results for generation approaches for the three domains	35
5.2	Filtering evaluation results for AWN generation method	37
5.3	Filtering evaluation results for the AraBERT	37
5.4	Filtering evaluation results for AraBERT dropout	38
5.5	Filtering evaluation results for the Hybrid	93
5.6	Manual evaluation results for the AraLexSubPro ranking features	40
5.7	Manual evaluation results for the BERTscore ranking feature	41
6.1	The count of each Error type results over the three domains	45

Table of Contents

Declaration I
Acknowledgments II
Abstract III
List of Abbreviations IV
List of Figures V
List of Tables VI
Chapter One: Introduction 1
1. Introduction 1
1.1 Problem Statement 2
1.2 Objectives of The Study 2
1.3 Research Questions 3
1.4 Significance of the Study 3
1.5 Thesis Layout 4
Chapter Two: Background and Literature Review 5
2.1 Background 5
2.2 Literature Work 6
2.3 Research Gap 20
Chapter Three: AraLexSubD Dataset Construction 21
3.1 Introduction 21
3.2 AraLexSubD Construction Steps 22
3.3 Ranking Experimental Setup Steps 24
3.4 Linguists Agreement Evaluation 25
Chapter Four: Methodology 26
4.1 AraLexSubPro Pipeline Methodology 26
4.2 Subistiution Generation 28
4.3 Substitution Filtering 30
4.4 Substitution Ranking 31
4.5 AraLexSubPro Algorithm 33
Chapter Five: Results and Discussions 34
5.1 Introduction 34
5.2 Evaluation of Substitution Generation 35
5.3 Evaluation of Substitution Filtering 36
5.4 Evaluation of Substitution Ranking 40
5.5 Summary 41

Chapter Six: Qualitative Analysis 42

6.1. Analysis of AraLexSubD Dataset 42
6.2. Analysis of AraLexSubPro Pipeline 42
6.2.1 The analysis of substitution generation results: 43
6.2.2 The Analysis of Substitution Filtrating Results 44
6.2.3 The Analysis of Substitution Ranking Results 44
6.3 AraLexSubPro Pipeline Error Types 45
Chapter Seven: Conclusion and Future Work 47
7.1 Conclusion 47
7.2 Recommendations and Future Work 48
References 49
الملخص 54

Chapter One

Introduction

1. Introduction

NLP helps computers learn and use languages as humans. Synonyms are essential in many NLP application areas [1]. In Information retrieval, synonyms can expand queries and retrieve richer results. It could also be beneficial in automatic text summarization, employed to identify repetitive information to avoid redundant summaries. In language generation, synonyms are used to create more varied texts [2]. Two words are synonymous if they can be interchanged in the same sentence without changing its meaning, such as {arrange, organize}.

> Synonyms are defined in [3] as "*two expressions are synonymous if the substitution of one for the other never changes the truth value of a sentence in which the substitution is made*." A more formal definition of synonymy in ontology engineering is "*a formal equivalence relation (i.e., reflexive, symmetric, and transitive)." Thus, "Two terms are synonyms iff they have the exact same concept (i.e., refer, intentionally, to the same set of instances). Thus, T1 =Ci T2. In other words, given two terms, T1 and T2, lexicalizing concepts C1 and C2, respectively, then T1 and T2 are considered to be synonyms iff C1 = C2*" [4].

Two main techniques are used for automatically extracting synonyms: translation graphs and deep learning techniques [5]. Synonyms can be automatically extracted from different resources, such as bilingual dictionaries, WordNets (WN), and carpus. These synonyms can be extracted for a general language or domain-specific field, such as a medical field [6]. Deep learning techniques can generate synonyms using various tasks, such as word embedding (e.g., Word2Vec), contextualized embedding (e.g., BERT), and LS.

This thesis focuses on the Arabic LS task for generating synonyms as there is a lack of advanced research for the Arabic language using deep learning techniques; few efficient approaches exist. Most of the efficient existing approaches are for the English language. This thesis is the first attempt to build an Arabic LS pipeline consisting of substitution generation, filtering, and candidate ranking. Our experimental results demonstrate encouraging results for Arabic LS.

1.1 Problem Statement

In order to enable LS in Arabic NLP applications, this research attempts to examine current synonym extraction techniques utilizing LS task for possible adaptation to produce synonyms in the Arabic language automatically. It is especially crucial in light of the difficulties of speaking Arabic, including its ambiguity, linguistic intricacy, and dialectal variances.

The main challenges are:

1. Developing or improving current methods to adapt them for Arabic synonym extraction is vital and challenging for Arabic NLP applications. There has been little advanced research on Arabic synonym extraction, although Arabic is one of the six official UN languages and is the language spoken by approximately 330 million people. The few effective methods currently in use mainly concentrate on the English language.
2. There is no clear evaluation dataset for Arabic, making validating theoretical assumptions and performing extracted resources challenging. Metrics like Precision, Recall, and F-measure require an ideal set for evaluation, which is difficult to establish for Arabic due to the lack of high-quality training data and the absence of a gold standard, typically provided by human experts.

1.2 Objectives of The Study

The thesis has three contributions focused on extracting synonyms using the Arabic LS task:

1. It surveys the most relevant works on extracting synonyms, focusing on the extraction techniques and their evaluation methods and datasets. We cluster these works into four groups: extracting synonyms using translation graphs, extracting synonyms using discovering new transition pairs, constructing new WordNets approaches by exploring synonym graphs, and synonym extraction using word embedding and language models [9].

   Paper name: Current Trends and Approaches in Synonyms Extraction: Potential Adaptation to Arabic [9].
   Conference name: International Conference on Information Technology (ICIT).

2. AraLexSubD: a benchmark dataset for evaluating the Arabic LS methods. The dataset is built by eight native Arabic language linguists who annotated 630 sentences, which are divided into three domains: the general domain (470 sentences), the finance domain (80 sentences), and the medical domain (80 sentences). The target words have 2476 substitution candidates, which we ranked according to their semantic relatedness [16].
   Despite the richness of Arabic vocabulary, limited research has been performed on the LS task due to the lack of annotated data. We build the first Arabic LS benchmark dataset to bridge this gap.

   Paper name: Arabic Lexical Substitution: AraLexSubD Dataset and AraLexSub Pipeline [16].
   Journal name: Data (MDPI) Q2, impact factor: (2.6), SJR 0.5 and Citescore:4.3, indexed within Scopus, ESCI (Web of Science).

3. AraLexSubPro: an Arabic LS pipeline that offers different techniques for generating, filtering, and ranking substitutes:

- To make a thorough comparison, AraLexSubPro uses four different methods as baselines to generate substitute candidates for the target words: a synonym dictionary-based approach (AWN), a pre-trained language model-based approach (AraBERT), AraBERT dropout (partial masking), and a hybrid approach between AraBERT and AWN.
- The generated substitutions are filtered using PoS, Post-processing, and semantic filters, which remove the inappropriate substitutions from the generated substitutions.
- The ranking is based on six high-quality features to compare thoroughly: word similarity, word frequency, BERT prediction order (BERT probability), BERT-based language model (Loss), BERT similarity, then reranking based on our AraLexSubPro ranker.
- The sixth ranking feature, the BERTscore, was examined for the particular domain dataset (finance and medical domains) to capture how IDF (Inverse Document Frequency), which differs in each domain, affects the candidate ranking. Since each domain has its own set of specific synonyms, they are ordered according to their IDF in that particular.

Paper name: Toward Automated Arabic Synonyms Extraction using Arabic Lexical Substitution [66].
Journal name: IEEE Access, Q1, Impact Factor: (3.4), SJR 0.960 and CiteScore:9.8, indexed within Scopus, Web of Science (Clarivate Analytics).

1.3 Research Questions

This research will attempt to answer the following questions:

1. Considering its unique linguistic characteristics, how can LS techniques be effectively applied to extract synonyms automatically for the Arabic language?
2. Can developing a new LS dataset, AraLexSubD, improve the accuracy and reliability of automated synonym extraction using LS in Arabic?
3. How can the hybrid approach combining the AWN and AraBERT approaches improve the accuracy and contextual relevance of automated synonym extraction using LS in Arabic?
4. Does applying the filtering step to lexical substitution generation output improve the quality of synonym extraction in Arabic?
5. What ranking features can be incorporated into the LS process to prioritize synonyms most appropriate to the context in Arabic?

1.4 Significance of the Study

This study significantly contributes to the NLP field, using the LS task for synonym extraction or substitution generation. By addressing the challenges posed by the unique characteristics of the Arabic Language and the lack of evaluation datasets, this thesis introduces the AraLexSubD benchmark dataset to evaluate the Arabic LS methods. Additionally, the proposed AraLexSubPro pipeline with innovation techniques, including traditional methods, BERT, hybrid models that generate substitutions, and six ranking features. The research provides a robust foundation for the Arabic LS task. The pipeline addresses the limitations of traditional methods. Our experimental results demonstrate encouraging results for Arabic LS.

1.5 Thesis Layout

The remainder of this thesis is organized as follows: Chapter 2 introduces the related work of the Lexical substitution. Chapter 3 describes the construction of AraLexSubD, chapter 4 describes the AraLexSubPro Pipeline, and Chapter 5 presents the experiment in terms of results and discussion. Chapter 6 presents the qualitative analysis. Finally, concluding remarks are emphasized, and recommendations for future work are formalized in chapter 7.

Chapter Two

Background and Literature Review

2.1 Background

The study of extracting synonyms has gained increasing importance due to technological development, which increases the need for accurate and correct context synonyms for various NLP applications, which remains limited in Arabic. Arabic is challenging due to its richness, morphological complexity, and ambiguity, making it challenging to build Arabic NLP applications.

We define synonyms as: "*Two expressions are synonymous if the substitution of one for the other never changes the truth value of a sentence in which the substitution is made*" [9].

One of the known synonymy databases for English is the Princeton WordNet (PWN) [3], built manually at Princeton University as a network of lexical concepts. Sets of synonyms (called Synsets) in WordNet are synonyms connected by semantic relations such as hypernyms and meronyms. Although many WordNets were built in other languages, following the same way PWN was built, most of them are small as the manual construction of WordNet is time-consuming and expensive.

Deep learning techniques are increasingly used in NLP to extract Arabic synonyms automatically and in a more precise way. Extracting synonyms accurately and contextually is vital for various applications such as machine translation, information retrieval, and text summarization, which can avoid redundancy and improve the understanding of language models.

Synonyms extracted from statistical language models, like Word2Vec embeddings, can be defined as "closely related" words [7]. Such language models capture context similarity between words by converting the words into vectors. Words are considered similar if their vectors are close to each other, which is typically measured using cosine similarity. Word embeddings are static models (i.e., they do not represent the meanings in different contexts).

Contextualized embedding models, like BERT, have recently been used to extract synonyms, showing and proving their power for generating synsets in dynamic contexts. These models use LS to replace the target word with the most suitable synonyms without changing the sentence's meaning [8].

LS is an essential task in NLP applications, which aims to replace a word in a sentence with suitable candidates (e.g., synonyms) as long as the sentence's meaning is maintained [9,10]. LS has two variants: substitute generation and candidate ranking [11]. It is widely used in many NLP tasks like data augmentation, paraphrase generation, semantic text similarity, and word sense induction [11,12]. The LS task has been applied widely to English using different methods [13]. Little research has been conducted on Arabic; no evaluation dataset has been created on Arabic lexical substitution.

One of the main challenges in LS is that previous algorithms find the substitutions for the target words from lexical resources (like WordNet) and then rank them based on their contexts [33]. Such algorithms have two typical challenges: (1) they should not ignore good substitute candidates because they are not included in the lexical resources, and (2) they should preserve the sentence's meaning as it contains all the meanings of the word and cannot scan the exact meaning that is suitable for the sentence meaning. To address such challenges, we propose utilizing the BERT contextualized embedding model that can be used to extract synonyms in dynamic contexts [8].

2.2 Literature Work

This section surveys the most relevant works on extracting synonyms, focusing on the extraction techniques and their evaluation methods and datasets. We perform the clustering of these works into four groups: extracting synonyms using translation graphs, extracting synonyms using discovering new transition pairs, constructing new WordNets approaches by exploring synonymy graphs, and synonym extraction using word embedding and language models [9].

2.2.1 Extracting Synonyms Using Translation Graphs

Some researchers proposed to build synonyms using translation graphs, which can be language-dependent or independent. The idea is to take bilingual dictionaries as input to build a translation graph between two words in a given language, find the translations for it in the other language, and then extract the paths that present synonyms. The key idea here is how to extract these paths. The accuracy of the generated synonyms depends on the structure and the accuracy of the input dictionaries.

A recent approach [6] suggests converting a bilingual dictionary into an undirected translation graph. The approach has two steps. The first step is to find all possible candidate synonyms from cyclic paths. The propagation of the translation stops when it reaches the root word (i.e., cycle), when no translations are found, or when it reaches the maximum number of specified levels. Figure 2.1 presents an example of a translation graph for the words (الغابة). ´If a path was found (**a1**→ e1→ a2→ e2 → a3 → e3 → **a1**), then all words in the cycles are converted into sets of bilingual synonyms such as {a1, a2, a3}:= {e1, e2, e3}. The second step is consolidating the candidate synsets with the exact translation by taking the union between them. To evaluate this approach, the authors converted the bilingual synsets found in the Arabic Word Net (AWN) into a flat bilingual dictionary and then rebuilt the AWN again using their algorithm. The accuracy is measured by cosine similarity between the original AWN and the generated AWN. The accuracy reached 82%. The algorithm results depend on the accuracy of the bilingual input dictionaries.

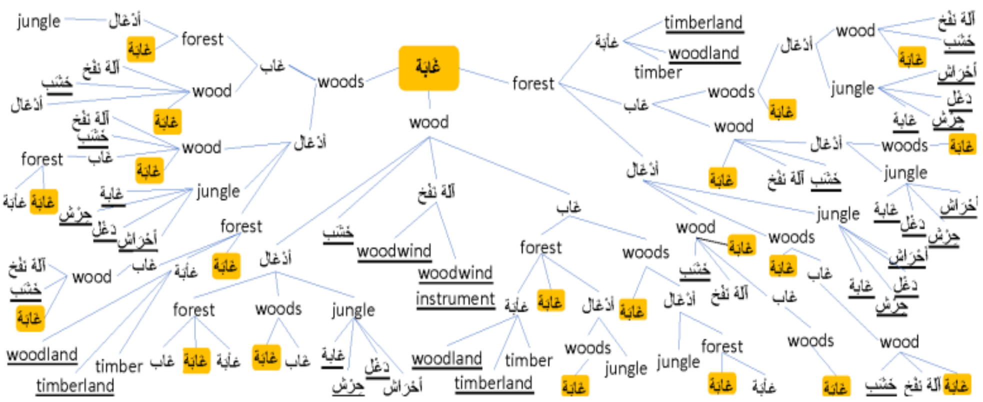


Figure 2.1: Example of a Translation Graph [6].

A similar approach enriched an English-Italian thesaurus's quality by discovering missing synonyms [17]. The authors claimed that their approach could be generalized and used for other languages, and it could also improve the quality of the user dictionary. They reach up to 80% accuracy at the sixth level. Their approach takes each word in the thesaurus and finds all corresponding translations. A directed translation graph Cyclic and Quasi-Cyclic (CQC) are constructed as illustrated in Figure 2.2. Depth First Search discovers the paths using the scoring function, which weighs paths based on path length. The shorter path takes higher weight, and then the paths are ranked. The algorithm is evaluated using the TOEFL dataset, and its results are compared with those of other algorithms used in the same datasets.

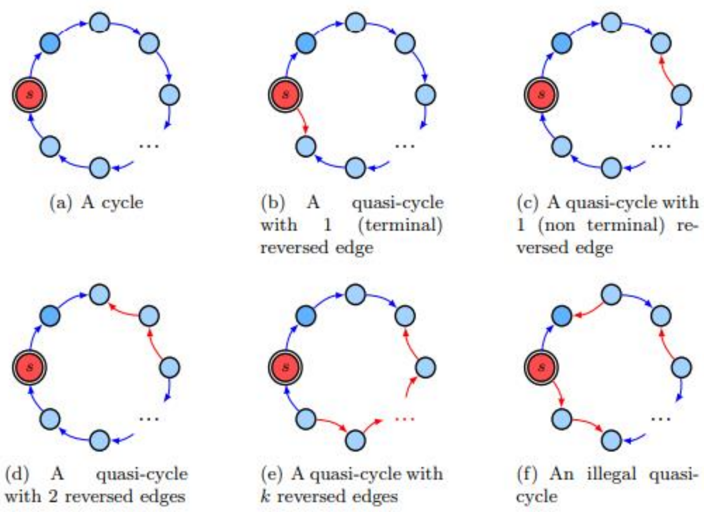


Figure 2.2: Cycles and Quasi-Cycles Construction [17].

The main difference between the two approaches, [6] and [17], is how the paths are extracted. No Quasi-Cycles are used in [6], and more importantly, a consolidation phase is introduced instead of using path weight as in [17]. Nevertheless, both approaches build translation graphs from bilingual dictionaries to extract synonyms. Although this method's precision is promising, their results depend on the accuracy of input dictionaries.

2.2.2 Discovering New Transition Pairs

Other researchers proposed to find new translation pairs using translation graphs, a task related to extracting synonyms. The discovered pairs offer new translation pairs between

languages that do not have bilingual dictionaries for translation between them. It can also enrich the existing bilingual dictionaries with newly discovered translation pairs.

A recent approach in [18] presents three algorithms for automatically discovering new translation pairs between languages based on bilingual input dictionaries. The cycle-based algorithm is used to build the translation graphs between the selected bilingual dictionaries in Apertium dictionaries to find translations between Portuguese, French, and English with a cycle length of at least 4. The path-based algorithm assigns weight to each path in the translation graph-based translation pairs number in each path. Paths with a short length and higher frequency take lower weights. The multi-way neural machine translation generates the target pairs' translations. They used parallel corpora for the selected languages (Spanish, Italian, French, Portuguese, Romanian, and English) as the translation pairs are English-Spanish, French-Romanian, and Italian Portuguese. They add the output for cycle and path algorithms to generate the target translation. They evaluate their algorithm by randomly extracting translation pairs and comparing them with manual pair translations for the same selected pairs. Their approaches show low recall and an acceptable precision (25-75%).

Another approach was proposed in [19] to discover new translation pairs from Apertium RDF Graph, containing 22 Apertium bilingual dictionaries. The translation graph is built by building context for each word. The context is computed by finding all translations for each word of three levels and then calculating cycles occurring in context. They restricted the process to level 7. The dense cycle is calculated to find the confidence score to accept word translation. Two experiments were used to evaluate this approach. The first is done by removing the English-Spanish Spanish–English from the Apertium RDF graph, then extracting it again using their algorithm to compare the two results. The algorithm was able to extract 72% of the original translations. The second experiment is done by finding a new English-French translation, which does not exist in the Apertium RDF graph. The algorithm extracted 73% of the translations in the converted wiktionary English-French file compared to the converted wiktionary English-French file.

In these approaches, extracted new translations depend on the number of translations for the target language. If higher, a higher recall can be expected. In addition, wrong translation may occur due to polysemy. The problem can be reduced by using many languages to generate correct new translation pairs. For example, an OTIC algorithm was built in [20] to detect the wrong translation when creating a bilingual dictionary using another language. The approach is extended in [21] to develop multilingual lexicons from bilingual lexicons.

2.2.3 Constructing New WordNets Approaches

This section presents related approaches to constructing WordNets automatically by focusing on synonymy extraction. These approaches use different linguistic resources such as dictionaries, wiktionaries, corpora, and existing WordNets by translating or mapping the new WordNet synsets to the existing WordNets [6].

> One of the most critical parts of the automatic construction of synonyms is evaluating the accuracy, precision, and recall, as there is no standard evaluation methodology and dataset. Lexical databases or word embedding-based approaches are currently being evaluated by comparing with existing WordNets or manually evaluating the generated synsets [22].

In [23], the authors proposed constructing a translation graph from multiple Wiktionaries to create word pairs with the same meaning for at least one synset. Each synset would have a complete subgraph when the translations between all terms with the same meaning are constructed in the undirected graph.

The clustering algorithm is used to extract the synsets. The input WordNet maps the discovered synsets by clusters to PWN. The Greedy algorithm extends the synsets with two metrics, belongingness and coverage, which assign scores to word synsets pairs. The input WordNet is used to initialize the synsets and expand them. The supervised binary classification is used to classify the new words included in synsets. The translation graph is linked with the existing WordNet for the selected language to induce new synsets using a clustering algorithm. The approach is evaluated by building Slovenian, Persian, German, and Russian WordNets from scratch and comparing the results with original WordNets. The accuracy varies from 20% up to 88%.

Generating synonyms using machine learning algorithms such as Greedy algorithms [23] has parameters and features that could affect algorithm results, such as choosing the threshold values. It must be set using a separate validation set from the test set. The generated synsets will be smaller but more accurate for high threshold values than WordNet synsets. It can reach above 90% accuracy. A lower threshold value generates larger synsets. However, it should be followed by a filtration step using a more detailed monolingual corpus.

A language-dependent approach was presented in [2]. This approach uses three kinds of input data: bilingual English-Chinese corpus, Chinese monolingual corpus, and Chinese monolingual dictionaries, which are used to extract synonyms automatically and separately for each input, and then results for each input are combined. For the monolingual dictionary, the hubs and authorities are used and considered as word features, then a feature vector for each word is constructed. The synonyms are extracted from the Chinese monolingual dictionary using feature vectors to measure the cosine similarity between two words. In the bilingual carpus, the synonyms are extracted using the word translation using the English-Chinese bilingual dictionary, then assigned a translation probability based on the English–Chinese corpus. The Chinese monolingual corpus finds the words in the same contexts by finding their named attributes dependency triples. For each word in the carpus, they find the relation of the word with the next ones (word1, relation type, word2). If these attributes are the same, they are considered synonyms. The results of the three algorithms are combined using a binary classifier. This approach is evaluated by comparing the compound synonyms from the tree algorithms to the compound WordNet and thesaurus synonyms.

WordNet construction from monolingual corpora uses considerable corpora resources in machine translation or Word Embedding. As they provided a vast vocabulary and context information, these approaches recall is usually higher than multilingual graph-based approaches. Using more than one input type to enrich the graph is not always needed to get good results. Building larger translation graphs by integrating other bilingual dictionaries may improve the results.

Another effort in [24] is to generate new WordNet synsets using existing WordNets, machine translations, and bilingual dictionaries. The authors considered their approach as language-independent. Each word in the PWN synset is translated to the target language using machine translation. Then, they check for irrelevant translations because of polysemy using intermediate WordNet or bilingual dictionary of the target language. A rank method for the translated synsets is then applied. The higher rank implies a word belonging to the new WordNets corresponding to the target language. This approach was evaluated manually by volunteers who use WordNet's language as their mother tongue, and their judgment is given a rating of quality from 1 (Poor) to 5 (Excellent). They randomly select 500 synsets and compare the volunteer judgment. They achieved an average score of 3.78/5.00 or 75.60%.

Other related approaches include the use of senses in Wiktionary [25], finding missing edges and consolidating similar synsets [26], finding semantic relations between concepts using wordnets and Wikipedia [27], building a graph of relationships between nodes using

bilingual dictionaries, monolingual corpora, mono/multilingual Thesauri, and Wiktionary [28].

2.2.4 Synonym Extraction Using Word Embedding and Language Models

This section presents approaches to generate synonyms or WordNets automatically using Word Embedding and language models like BERT. These approaches can use different linguistic resources like wiktionaries and corpora and can be considered general approaches, but they cannot be used directly to find synonyms. They must be filtered to the semantic similarity only.

2.2.4.1 Synonym Extraction Using A Word Embedding

Recently, neural network models have been used to extract synonyms using a Word Embedding model from corpora. The authors in [7] presented an approach for extracting Arabic synsets. It can be considered a general approach with a two-step algorithm. In the first step, words in the NewsCrawl 2014 corpus are tagged with POS. Then, Word Embedding is constructed by converting every word into a vector presentation. Only words with a frequency of 25 are considered. The skip-gram model is used to group similar words —the cosine similarity is then used to cluster the words into adjectives, nouns, adverbs, and verbs. In the second step, the approach is evaluated by the Simlex-99 dataset, a forward neural network, and a Word Embedding list from the first step to discover the synonyms. They achieve 76% accuracy.

Another approach [29] uses is word embedding to enhance WordNets. This approach is an extension of the approach presented in [24]. After a translation graph for the target language is constructed from PWN, Word Embeddings are used to improve the quality of the translation step's synsets. Irrelevant words were removed in the synsets by calculating the cosine similarity between the words in the synset. A synset similarity threshold value is used to accept or dismiss the word from the synset. The same is done to validate the synsets relations. The cosine similarity is calculated between all synset pairs. When the values are above the threshold, they are maintained or discarded. They evaluated their approach by generating AWN using the Wattan-2004 corpus. Then, they create the Word Embedding using a continuous bag of words (CBOW). They generate 60% of the manual AWN with a precision of 78.4%.

A recent approach is proposed in [30] to construct WordNets using Word Embeddings and machine translations. A translation of a target word using a bilingual dictionary or machine translation is extracted and then used to find nominee synsets from the PWN. After that, a score for each synset is assigned to each synset. The resulting synsets are ranked by computing cosine similarity with Word Embeddings. The experiment was done for nouns, adjectives, verbs, and adverbs, and their results were promising. They achieved a high level of accuracy as they got 90-94% on precision. The main challenge is that the resources of all language pairs are required, which is not easy to find.

Another unsupervised learning methodology is discussed in [31]. This approach consists of two steps. The first step is to capture the best settings for finding synonym relations by training the Word Embedding using Word2Vec, CBOW, and substitution generation models by two corpora, the KSUCCA and Gigaword. The preprocessing step includes tokenization, diacritic, numbers, English letter removals, and normalization. The next step is to filter the list of similar emending to catch synonyms using the synoExtactor pipeline. This pipeline finds the most similar words by calculating the cosine similarity for preselected words from the two corpora. The high score implies the same context, which contains synonyms and

other relations, such as anatomy. Secondly, they apply three filters. The first is the Lemmatization filter, which removes inflections. They create a dictionary of lemmas for each lemma, adding all words with the same lemma from carpus. The second filter is the collection filter that uses the collocation dictionary generated from corpora. The stop and collection words that have appeared less than five times were removed. The last filter is the POS filter, which keeps the words with the same POS. Synonyms always have the same POS. The approach is evaluated by comparing the generated synsets with the Ama'any Arabic thesauri; a manual evaluation by two linguistics is also done. The SynoExtractor reached 60% precision for the KSUCCA corpora and 74% precision for the Gigaword corpora.

Another approach in [32] developed an automatic extraction model of synonyms to create Quranic AWN (QAWN) using three resources: the boundary annotated Qur'an, lexical recourses for collecting derived words from the Qur'an, and traditional dictionaries. In the first step, they present the Holy Qur'an using the Vector Space Model (VSM). In the second step, they used traditional Arabic dictionaries to extract the meaning of the Quran words. They used term frequency and inverse document frequency in the vector space model and computed cosine similarities between Quranic words depending on textual definitions extracted from traditional Arabic dictionaries. The words with the highest similarity were clustered to form a synset. They evaluated their approach using an information retrieval system. They reached 34.13% recall and very low precision as clustering words could not generate actual synsets as they contained other word relations.

In Word Embedding approaches, supervised techniques help generate or filter more accurate synsets when comparing unsupervised techniques. Supervised methods need labeled data to nominate synonyms from other relations, not to extract relations. While the unsupervised techniques did not need labeled data, they can use any raw carpus using the clustering techniques. Still, another approach is needed to filter synonyms from other relations.

2.2.4.2 Synonym Extraction Using Language Models

This section presents related approaches and Datasets for the LS problem. A novel approach is presented in [33]. It uses the BERT-based LS model inspired by [8]. The model extracts the relevant synsets within the same contexts and extracts the most suitable synonyms for the target word without changing the sentence's meaning by measuring the contextual representation similarity. The novel approach uses partial drop masking for embedding the target word by setting it to zero, then predicting the target word based on its position in the sentence by calculating the validation score. By ranking the validation scores, the higher values present the target word's synonyms.

It extracts the most suitable synonyms for the target word without changing the sentence's meaning. This approach is evaluated using two benchmark datasets, the SemEval 2007 dataset (LS07) and the CoInCo dataset (LS14), using five metrics (best, best-mode, out-of-ten, out-of-ten-mode, and Precision @1). The metrics that present the best predictions' quality are evaluated using the best, best mode, and Precision @1. While the coverage of the synsets is evaluated using out-of-ten and out-of-ten-mode.

The results show that their embedding dropout to BERT performs better than the previous works on the same two benchmark datasets. They reached 51.1% and 56.3% in Precision@1 for the LS07 and LS14 datasets, respectively. This approach permits the partial BERT to have balanced consideration for the target word's semantics, which will generate only the relevant synsets for the sentence meaning.

BERT is a Word Embedding approach that can either mask the embedding of the target word or not. In masking the embedding of the target word case, the BERT generates synonyms

for the target word, which do not fit even in the same context. The second case, which does not mask the embedding of the target word, will generate the same word as a synonym.
The approach presented in [33] makes partial masking for the target Word Embedding, then predicts the target word based on its position in the sentence by calculating the validations score. This approach extracts only the most suitable synonyms for the target word without changing the sentence's meaning, proven by their precision results exceeding the older approaches' precision values.
The main difference between the BERT approach and the partial masking BERT approach discussed in [33] is that the BERT approach may ignore suitable synonyms for the target word. Besides, they do not consider the sentence's meaning. In contrast, the partial BERT has balanced consideration for the target word's semantics, which will generate only the relevant synsets for the sentence meaning.
GENESIS, a generative approach for LS that generates context substitutions, is presented in [34]. The paper presents a LS method and produces large-scale silver data that can be used to improve the performance of LS systems further. GENESIS has three stages: substitution generation, filtering, and ranking. The substitution generation step utilizes the BART seq2seq model, fine-tuned by concatenating CoInCo and TWSI datasets. The filtering step removes the target word, different PoS tag words, and words not part of the vocabulary. All generated substitutions of the target word must be in WordNet-v3.0, including the hyponymy, hypernymy, similar-to, or see-also relations of the target word. The ranking step is based on contextual similarity by computing the cosine similarity of the target word vector with the substitution vector.
GENESIS uses standard evaluation metrics (best, best-mode, oot,oot-mode, precision @1, precision @3, and recall @10) to assess the quality of generated substitutions alongside human judgment for qualitative analysis. It achieved state-of-the-art results on several metrics, including 20.6 for best, 33.2 for precision @3, and 39.5 for recall @10. In qualitative assessments, annotators deemed 79% of GENESIS-generated substitutes suitable, indicating high-quality output.
Silver data generation: The model generates silver data by fine-tuning on a gold dataset, SemCor, a manually annotated corpus where instances are sense-tagged according to the WordNet sense inventory, and it is used in English Word Sense Disambiguation (WSD). SemCor generates a list of substitutions for each input instance using the fine-tuned model and keeps only the substitutes whose similarity to the target is higher than a threshold.
GENESIS performance depends on the quality and size of training data, and the lack of large-scale, annotated corpora for LS is a challenge for supervised techniques. Besides, the evaluation metrics depend on fixed vocabulary, which may not capture the quality of substitutions outside this vocabulary.
Although contextual word embedding models consider the provided context, they cannot adequately explain how a substitute will impact the sentence's overall meaning. Another LS approach is CILex, which uses contextual sentence embeddings to enhance prediction accuracy [35]. This approach captures the effect of replacing the target word with the suggested substitution over the overall meaning of a sentence. CILex has three stages: substitution generation, filtering, and ranking.
The substitution generation step utilizes contextual sentence embeddings from XLNet RoBERTa and BERT. The filtering step is based on the lemma of the target word and POS tag, and multi-words were removed from the list. The ranking step is based on four features: the XLNet model, sentence similarity, WordNet-based similarity, and validation scores, which are computed and combined to assess the quality of the substitutes.
CILex uses standard evaluation metrics (precision @1, precision @3, and recall @10) to assess the accuracy of the ranked substitutions. Experiments on the LS07 and CoInCo

datasets showed that CILex achieved successful predictions in 76.8% and 77.99% of the cases, respectively. It achieved superior results on the LS07 and CoInCo datasets by approximately 4% and 6.75%, respectively. It achieved 22.59 for precision @1 on LS07, 54.65 on CoInCo, 39.43 for precision @3 on LS07, and 53.92 on CoInCo. Recall @10 reached 54.65 on LS07 and 53.57 on CoInCo.

CILex effectively integrates contextual sentence embeddings and additional context information, leading to more semantically consistent substitutes. Although the substitution generation was enhanced, the aspect of candidate ranking did not exhibit the same improvements. This approach had no successful predictions based on the gold substitutes.

Another novel LS framework, LexSubCon, integrating contextual embeddings with external lexical knowledge, is presented in [36]. LexSubCon has two stages: substitution generation and ranking. The substitution generation step uses the Mix-Up embedding strategy, which utilizes the BERT model and WordNet, combining the target words embedding with the average embedding of probable synonyms from both WordNet and BERT for improved accuracy.

The ranking step is based on two features: Proposal Score and Gloss-Sentence Similarity. The Proposal Score uses the BERT to compute it for each candidate word based on contextual embeddings. The Gloss-Sentence Similarity measures the cosine similarity between the gloss-sentence embeddings of the target word and each candidate word to evaluate semantic relevance.

LexSubCon evaluation results outperformed the state-of-the-art models over LS07 and CoInCo datasets by 2%. For the LS07 dataset, LexSubCon achieved a 51.7% Precision@1. For the CoInCo dataset, LexSubCon achieved a best score of 14.0%. This approach effectively integrates contextual information with structured lexical knowledge using the Mix-Up strategy, ensuring meaningful substitutions, instead of the partial masking (dropout) used in [33], which uses only BERT and randomly sets the target word embedding vector positions to zero based on the dropout ratio.

Another approach to the Chinese language is presented in [37], which presents the Chinese LS method and dataset CHNLS. The method has two stages: substitution generation and ranking. An ensemble method utilizes four techniques in the substitution generation: Dictionary-based, Embedding-based, BERT-based, and Paraphraser-based. Each method generates 15 substitutes for a target word, combined using an ensemble approach to select the top candidates. The effectiveness of the generated substitutes is ranked using BARTScore and BERTScore metrics.

This approach uses five evaluation metrics (best, best-mode, out-of-ten, out-of-ten-mode, and Pre@1). Experiments showed that the ensemble method achieved successful high-quality 90.5% precision and coverage of 97% in substitutes on the CHNLS dataset. The ensemble method has multiple LS techniques, leading to a broader range of substitutions and increased diversity. Despite the automated generation of substitution reducing the workload on human annotators, the dataset's quality still relies on the expertise and availability of human annotators.

Table 2.1 presents a comparison of the different approaches in chronological order. The comparison is based on the input resource, the methodology used, the evaluation methods used to validate the approach results, and the approach accuracy. Some of the four group approaches are language-dependent, which is done for a particular language and cannot extract synonyms for other languages. While most approaches are language-independent, they can be used as a general methodology to extract synonyms in other languages.

Table 2.1-A: A comparison Between The Different Synonyms Extraction Approaches

Num	Approaches	Paper name	Year	Methodology	Language	Input	Evaluation	Accuracy
1	**Extracting Synonymy using translation graphs**	Cycling in graphs to semantically enrich and enhance a bilingual dictionary [17]	2012	Builds a directed translation graph (cyclic and quasi-cyclic)	A general approach	1. English-Italian 2. Bilingual dictionary	Use the TOFEL dataset and compares their results with other algorithms used in the same datasets.	80%
		Extracting Synonyms from Bilingual Dictionaries [6]	2021	1. Builds a translation graph 2. Synsets consolidation	A general approach	Arabic- English Bilingual dictionary	Rebuild the AWN again using their algorithm	82%
2	**Discovering new transition pairs**	Leveraging RDF graphs for crossing multiple bilingual dictionaries [19].	2016	Aperteum RDF Graph	A general approach	22- Aperteum RDF Graph 1. English- Spanish 2. English - France	1. Extract English-Spanish, Spanish– English using their algorithm. 2. Find a new English-French translation, which does not exist in the Apertium RDF graph	English-Spanish 72% English – France 73%
		Leveraging Knowledge Graphs with Neural Machine Translation for Automatic Multilingual Dictionary Generation [18]	2019	1. Graph-based. 2. Path-based 3. Multi-way neural machine translation	It can't be general – language-dependent	1. Bilingual dictionaries English - Portuguese 2. Carpus	Choose randomly extracted translation pairs and comparing them with manual pair translations for the same selected pairs	25-75%
3	**Constructing new WordNets**	Optimizing synonym extraction using monolingual and bilingual resources [2]	2003	1. The synonyms extracted from Chinese monolingual dictionaries 2. Word translation using Bilingual carpus (English –Chinese) 3. Chinese monolingual corpus to find words in the same context	It can't be general – language-dependent	1. Bilingual carpus (English – Chinese) 2. Chinese monolingual corpus 3. Chinese monolingual dictionaries	Compare the compound synonyms from the tree algorithms to the compound WordNet and thesaurus synonyms.	low accuracy
		Automatically constructing WordNet synsets [24]	2014	1. Machine translation 2. Ranking translated synsets	A general approach	1. Existed WordNet 2. Bilingual dictionaries	Randomly select 500 synsets and compare the volunteer judgment	75%
		Synset expansion on translation graph for automatic WordNet construction [8]	2019	1. Clustering algorithm 2. Greedy algorithm 3. Supervised learning	A general approach	1. Wiktionary 2. Existed WordNet	Built Slovenian, Persian, German, and Russian WordNets from scratch, then compare the results with original WordNets of these languages	88%

Table 2.1-B: A comparison Between The Different Synonyms Extraction Approaches

Num	Approaches	Paper name	Year	Methodology	Language	Input	Evaluation	Accuracy
4	**Extracting Synoynoms using Word embeddings**	Enhancing automatic WordNet construction using word embeddings [29]	2016	Word Embedding's	A general approach	1. WordNet 2. Textual corpora	Generate Arabic WordNet AWN using Wattan 2004 corpora. Then they create the word Embedding using a continuous bag of words (CBOW).	78%
		Towards an automatic extraction of synonyms for Quranic Arabic WordNet [32]	2016	Use the term frequency and inverse document frequency Tf-idf in the vector space model (VSM)	It can't be general	1. The Holy Qur'an, 2. lexical recourses for the Qur'an 3. Traditional dictionaries	An information retrieval system	34.13%
		Automated WordNet construction using word embeddings. [30]	2017	1. Unsupervised learning 2. Machine translations 3. Word embeddings.	A general approach	1. Existed WordNet PWN 2. French and Russian	Build 200 French and Russian test WordNets using Google translator as a machine translator, word embedding dataset in [17].	90-94%
		BERT-based Lexical Substitution [33]	2019	Partially mask BERT	A general approach	LS07 trial	1. SemEval 2007 (LS07). 2. CoinCo (LS14)	51.1 % for LS07 56.3 % for LS14
		Extracting Word Synonyms from Text using Neural Approaches [7]	2020	Two-step approach using NN to: 1. Build word embedding list 2. Use word embeddingt to train NN	A general approach	1. Textual corpora 2. Sim Lex-999 lexicon	Use Simlex-99 dataset, a forward neural network, and a word embedding list from the first step to discover the synonyms.	76%
		A Novel Pipeline for Arabic Synonym Extraction Using Word2Vec Word Embeddings [31]	2021	Unsupervised with 2 step approach : 1. Build word embedding list from two corpora. 2. SynoExtractor pipeline to filter synsets	A general approach	1. KSUCCA 2. Gigaword	1. Compare the generated synsets with the Ama'any Arabic thesauri. 2. A manual evaluation by two linguistics	60% for KSUCCA 74% for Gigaword
5.		GENESIS: A Generative Approach to Substitutes in Context [34]	2021	It has three stages: substitution generation, filtering, and ranking. 1. substitution generation step utilizes the BART model, fine-tuned by concatenating 2. The filtering step removes the target word, different PoS tag words, 3. The ranking step is based the cosine similarity	A general approach	BART seq2seq model	Concantenate the CoInCo and TWSI to fine tune BART	20.6 for best, 33.2 for precision@3, 39.5 for recall@10 79% of the proposed replacements are suitable substitutes

Table 2.1-C: A comparison Between The Different Synonyms Extraction Approaches

6.		CILex: An Investigation of Context Information for Lexical Substitution Methods [35]	2022	CILex has three stages: substitution generation, filtering, and ranking. 1. The substitution generation step utilizes contextual sentence embeddings from XLNet RoBERTa and BERT. 2. The filtering step is based on the lemma of the target word and POS tag 3. The ranking step is based on four features: the XLNet model, sentence similarity, WordNet-based similarity, and validation scores,	A general approach	XLNet, RoBERTa, and BERT.	the LS07 and CoInCo datasets	22.59 for precision @1 on LS07, 54.65 on CoInCo 39.43 for precision @3 on LS07 and 53.92 on CoInCo. Recall @10 reached 54.65 on LS07 and 53.57 on CoInCo. successful predictions of 76.8% in LS07 and 77.99% on CoInCo
7.		LexSubCon: Integrating Knowledge from Lexical Resources into Contextual Embeddings for Lexical Substitution [36]	2022	LexSubCon has two stages: substitution generation and ranking. 1. The substitution generation step uses the Mix-Up embedding strategy using BERT model and WordNet,. 2. The ranking step used Proposal Score and Gloss-Sentence Similarity.	A general approach	BERT model and WordNet	the LS07 and CoInCo datasets	LexSubCon outperformed the state-of-the-art evaluation results over LS07 and CoInCo datasets by 2%. LS07 dataset, achieved a 51.7% Precision@1 CoInCo dataset, achieved a best score of 14.0%.
8.		Chinese Lexical Substitution: Dataset and Method [37]	2023	The method has two stages: substitution generation and ranking. 1. Substitution generation: Dictionary-based, Embedding-based, BERT-based, and Paraphraser-based. 2. Ranked using BARTScore and BERTScore metrics.	A general approach	Dictionary, Embedding-based, BERT, and Paraphraser	CHNLS dataset	ensemble method achieved successful high-quality 90.5% precision and coverage of 97% in substitutes on the CHNLS dataset.

LS datasets can be divided into manual and automatic construction. The existing LS datasets are primarily for English. Each instance in the LS datasets comprises a sentence, a target word, and suggested substitutions.

Manual construction of English LS datasets includes SemeEaL, the first LS shared task, called SemeEaL-2007 task 10 [38]. The dataset in this shared task consists of 201 manually selected target words, which are polysemous, and 10 different sentences for each target result in 2010 sentences. Each sentence has one target word from the English Internet Corpus [39]. Five annotators then suggest three substitutes for all 201 target words from their memory, resulting in 12,300 labels and four substitutes for each target.

Afterward, the Turk bootstrap Word Sense Inventory (TWSI) dataset [40] was the first attempt to construct a large-scale English dataset by choosing 25K sentences from Wikipedia with 1012 distinct nouns annotated through Amazon Mechanical Turk. This dataset deals with polysemous target words, which means different substitutes for the various contexts of the same target word, as TWSI deals with only noun target words.

CoInCo [41] was constructed to mitigate this restriction by choosing 2474 sentences from the Manually Annotated Sub-Corpus (MASC) [42, 43]. All the words in the sentences were target words. Six annotators classified the target words into suitable and unsubstitutable, resulting in 3874 distinct words with defined part-of-speech tags. Each annotator was asked to suggest 5 substitutions, resulting in 167,446 labels and 7.2 substitutions for each target word.

Automatic construction of English LS datasets includes the Stanford word substitution benchmark SWORDS [44], which builds on CoInCo, is a higher-coverage and higher-quality benchmark that treats LS as a classification problem by asking humans to judge the appropriateness of given substitutes and not to suggest them. The dataset contains 1132 sentences with 1132 target words and 68683 substitutions.

ALaSca is another large English LS dataset [45] that selects a set of target words for ALaSca and collects sentences containing target words. ALaSca then clusters words based on context, considering the target word polysemy. ALaSca then provides possible substitutes depending on the context of the target word. The dataset contains 3442 target words, 34,755 sentences, and 50.24 substitutes for each target word.

ProLex is a novel benchmark [46] that selects contexts, targets words from the TOEFL-11 dataset [47], and generates substitution using GPT-4. Following the annotation in [44], which gives the annotators a context, a target word, and a candidate substitute to judge whether the substitution is appropriate for the target word, the dataset contains 680 sentences with 680 different target words and 2.9 acceptable substitutes on average.

LS dataset construction of other languages includes GermEval, a manual German LS dataset from GermEval 2015 [48] containing 2040 sentences from the German Wikipedia containing 153 unique target words. EVALITA is a manual Italian LS dataset from EVALITA 2009 [49] comprising 2310 sentences and 231 unique target words.

The Chinese LS dataset, CHNLS, consists of 33,695 instances and 144,708 substitutes. The sentences with target words are extracted from News, Wikipedia, articles, and Novels. Then, for each target word, an ensemble of four methods, Dictionary-based, Embedding-based, BERT-based, and Paraphraser-based, are used to generate 15 substitutions. Then, the results of each substitution method are concatenated. Finally, the human annotators select the appropriate substitutions. The final substitutions set are the substitutions that have been marked at least once by annotators, or they can add new ones.

In conclusion, there remains a significant gap in research on Arabic LS. To the best of our knowledge, no publicly available Arabic LS dataset currently exists to evaluate the capabilities of Arabic LS models.

Table 2.2 presents a comparison of the different LS Datasets in chronological order. The comparison is based on language, the way of construction, the number of sentences, the number of target words, the number of substitutions, and the dataset characteristics.

Table 2.2: A comparison between the LS Dataset

Num	Dataset nam	languag e	Paper name	Year	Constructio n	# of sentences	# of target word	# of substitutions	Dataset Characteristics
1.	SemEval-2007 (Task 10)	English	English lexical substitution task [38]	2007	Manual	2010	201	12300	*First shared LS task *Contains polysemous *Done by 5 annotators *Aims to evaluate the ability to replace target words with suitable alternatives in context without predefined sense inventories. *It reflects word sense disambiguation (WSD).
2.	EVALITA 2009	Italian	The lexical substitution task at EVALITA 2009 [49]	2009	Manual	2310	231	not specified	* three annotators *Aims to improve Word Sense Disambiguation (WSD) by allowing unsupervised approaches.
3.	TWSI	English	Turk Bootstrap Word Sense Inventory (TWSI) [40]	2010	Manual	25,000	1012	not specified	* Large-scale dataset, *Contains polysemous
4.	CoInCo	English	What Substitutes Tell Us Analysis of an "All-Words" Lexical Substitution Corpus [41]	2014	Manual	2,474	3,874	167,336	* first large-scale English in terms of lemma coverage
5.	GermEval 2015	German	Delexicalized supervised German lexical substitution [48]	2015	Manual	2,040	153	not specified	* First published dataset * Allows the evaluation of WSD * Each noun has 10 annotated sentences *Each adjective has 10 annotated sentences *Each verb has 20 annotated sentences.
6.	SWORDS	English	Swords: A Benchmark for Lexical Substitution with Improved Data Coverage and Quality [44]	2021	Automatic	1,132	1,132	68,683	* Deals with LS as a classification problem
7.	AlaSca	English	ALaSca: An Automated approach for Large-Scale Lexical Substitution [45]	2022	Automatic	34,755	34,755	50.24 per word	*Context clustering for polysemy
8.	ProLex	English	ProLex: A Benchmark for Language Proficiency-oriented Lexical Substitution [46]	2023	Automatic	680	680	2.9 per word	* Substitutions are generated using GPT-4
9.	CHNLS	Chinese	Chinese Lexical Substitution: Dataset and Method [37]	2023	Automatic	33,695	Not specified	33,695	* First published dataset * Substitutes are selected from News, Novels, Wikipedia

2.3 Research Gap

Arabic is the official tongue of more than 330 million people in 22 countries and one of the six official languages of the United Nations [15]. Despite that, little research has been done on Arabic, as the main challenge is the lack of human annotation. At the same time, the LS task has been applied widely to the English language employing various algorithms, such as [33], and only one for the Arabic language [16] due to the challenges and the lack of evaluation datasets.

This thesis examines current synonym extraction techniques utilizing the LS task in NLP for other languages. It then investigates potential adaptation to automatically generate synonyms for the Arabic language, highlighting the difficulties of Arabic, including its ambiguity and linguistic intricacy, complex morphology, high context sensitivity for sentence meanings, polysemous meaning, the lack of annotated datasets and lexical resources, and ensuring annotation consistency by diverse contexts and multiple annotators.

The main challenges are:

1. Developing or improving current methods to adapt them for Arabic synonym extraction is vital and challenging for Arabic NLP applications. There has been limited advanced research on Arabic synonym extraction regarding LS. The few existing efficient approaches primarily focus on the English language.
2. There is no evaluation dataset for Arabic, making validating theoretical assumptions and performing extracted resources challenging.

Synonym extraction or substitute generation, crucial for NLP applications, is accomplished using the LS task in this work. Due to the lack of evaluation datasets, we developed the first annotated benchmarking dataset, AraLexSubD, to assess Arabic LS techniques. We also constructed the first Arabic LS pipeline, AraLexSubPro, which assessed hybrid models, BERT, and conventional methods for generating substitutions and then ranked them according to six rank features. For the Arabic LS challenge, the research offers a solid basis and promising results for Arabic LS in our experiment.

By focusing on the challenges of the Arabic language, our thesis seeks to advance the Arabic NLP field. This study is the first attempt to build an Arabic LS dataset and pipeline. It contributes valuable resources for future research and development. Our experimental results demonstrate encouraging results for Arabic LS.

We believe the proposed AraLexSubD and AraLexSubPro can accelerate future research on this task. Despite the initial positive results of this challenging task, the substitution generation method and substitution ranking feature can affect the performance.

Chapter Three

AraLexSubD Dataset Construction

3.1 Introduction

Building an Arabic LS dataset is challenging due to the language's complexity and richness. Arabic words have a complex morphology, with numerous grammatical forms that rely on grammatical rules and structures, making generating accurate and contextually relevant substitutions for Arabic words difficult. Additionally, many Arabic words are polysemous with multiple meanings that depend on context, which adds another layer of complexity to creating an Arabic LS dataset. The AraLexSubD dataset is used to assess the AraLexSub steps in the pipeline.

Eight Arabic native linguist annotators have been involved in constructing the AraLexSubD dataset. Six linguist annotators are top students who graduated recently with high distinction from the Department of Linguistics and Translation, have a high level of proficiency, and follow the annotation guidelines consistently. The other two annotators are a human doctor and an economist. The eight native Arabic-speaker linguists annotate the 630 sentences in the AraLexSubD dataset. The annotators then split the sentences into the general domain (470 sentences), the finance domain (80 sentences), and the medical domain (80 sentences). The PoS tags of the target words are 317 nouns, 256 verbs, and 57 adjectives.

For each target word, several sentences were created, considering its multiple meanings (polysemy). The reason behind this approach is that the definition of a word is influenced by the context in which it is used. By presenting different sentences, each employing the target word in a distinct context, then exploring the meanings associated with the word becomes possible. The number of polysemous target words with more than one sentence reached approximately 80 in the general domain and 80 in particular domains.

The AraLexSubD dataset contains five primary and five secondary columns. The five primary columns are the sentences with one target word, the target word, the target word PoS tag in the sentence, the possible candidate words (synonyms) for each target word, and the ranking order for the candidate words using the scoring guidelines in [50]. The five secondary columns are two for target words (root, lemma) and three for candidates (root, lemma, transformation), where the lemma and the root for the target word and the candidates

are retrieved from Qabas [51]. Qabas is a morphological part of Arabic ontology. A transformation column is added to add the necessary transformation characters to the candidate based on the target word morphological form.
Figure 3.1 shows an example of a sentence in the general dataset. Figure 3.2 shows the same sentence in Figure 3.1 with a candidate's score by one of the linguists based on the guidelines.

sentence	Original	Original lemma id	Original lemma	Original lemma root	Original PoS
لا يظهر تحيزاً لأحد **بعينه.**	عين	202001305	عَيْنٌ | ع	عين	noun

Figure 3.1: An example of a sentence in the general dataset.

Substitute lemma	Substitute lemma root	Substitute tranformation	ماذا تم الإضافة على transformation	same meaning / style/ frequant
ذَاتٌ	ذات	بذاته	اضافة حرف الجر ب + حرف ه	نفس الدلالة والاسلوب والشيوع 100
اِسْمٌ	سمو	باسمه		نفس الدلالة، الأسلوب مقبول، شائعة الى حد ما 70
نَفْسٌ	نفس	بنفسه		نفس الدلالة والاسلوب والشيوع 100
مُفْرَدٌ	فرد	بمفرده		نفس الدلالة، الأسلوب ضعيف ، غير شائعة 60

Figure 3.2: The same sentence in Figure 3.3 with a candidate scored by one of the linguists.

3.2 AraLexSubD Construction Steps

This section presents the steps that we followed to construct AraLexSubD:

1. Determining Domains

The AraLexSubD dataset has three domains:

- General domain: The general domain comprises 470 sentences with 390 distinct target words. Eighty target words were polysemous, and each of the 80 target words had at least two sentences with different contexts.
- Medical domain: The medical domain comprises 80 sentences focused on a medical field with 80 target words.
- Finance domain: The finance domain comprises 80 sentences focused on finance, with 80 target words.

In other words, for every 80 target words, polysemy was applied to two contexts, one in medicine and the other in finance. Three linguist annotators selected the contexts.

2. Extracting Target Words and Sentences

Three linguists chose the general domain's target words, and two annotators (a medical doctor and an economist) worked together to guarantee that the common words (target words) were accurately identified and chosen.
The general domain sentences extracted from the Arabic lexicographic database [52] cover many subjects, including poetry, the Quran, and essay sentences. Furthermore, they also write or select the sentences of the medical and finance domains from essays. The selected sentences are in Modern Standard Arabic (MSA) and are used in many contexts.

3. Providing Substitutes

The three linguist annotators who have chosen the general domain target words and have selected or written the sentences for the three domains are used to determine the possible substitutes for each target word in the three domains.

The chosen substitutes should be synonyms for target words and not alter the sentence's sense. The likely candidates are selected from the Arabic ontology and Qabas, as well as Arabic dictionaries such as Al-Waseet [53], Al-Muaser [54], and Al-Maani [55]. When one of the linguist annotators could not think of or find a suitable candidate, the linguist put no entries. Then, the candidates from the three linguist annotators for each sentence in the AraLexSubD dataset were merged.

4. Ranking Substitute Guidelines

Three linguists were asked to manually rank the substitutes for each sentence in the three domains based on the scoring guidelines introduced in [50]. These guidelines are used as annotation methodology to maintain consistency among linguist's scores. The guidelines suggest to rank the substitutes mainly based on semantics, style, and use, as follows:

- Same semantics (synonyms): The candidates should share the exact meaning of the target word and not should not alter the sentence's meaning [3]. We define synonyms as: "*Two expressions are synonymous if the substitution of one for the other never changes the truth value of a sentence in which the substitution is made*" [9]. A more formal definition of synonymy in ontology engineering is "*a formal equivalence relation (i.e., reflexive, symmetric, and transitive)." Thus, "Two terms are synonyms iff they have the same concept (i.e., refer, intentionally, to the same set of instances). Thus, T1 =Ci T2. In other words, given two terms, T1 and T2, lexicalizing concepts C1 and C2, respectively, then T1 and T2 are considered synonyms iff C1 = C2*" [4].
- Style: How much of the use of the substitute is correct and robust in the sentence? For example, consider two substitutes (اسف, Sorry /اعتذر, apologize). Both substitutes have the same meaning, and both are frequently used, but (اسف, Sorry) present feelings, which makes the substitute (اسف, Sorry) more stylish.
- Use (frequently used): How often is a word used in this context? For example, consider two substitutes (جوال, Mobile /خليوي, cellular). Which one is more useful? Both substitutes have the same meaning, but the (خليوي, cellular) is rarely used in this context.

The scoring standards in [50], as shown in Figure 3.5, are fuzzy scales from 0 to 100, representing the strength of the synonymy relation. The strength is 100, which means the same semantics, style, and use. The scoring schema has three categories: a score from 60 to 100 means that the substitutes have the same meaning, a score from 50 to 60 is close in meaning, and below 50 means different semantics.

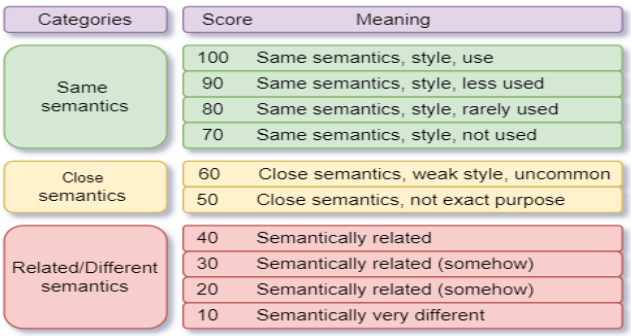


Figure 3.3: The fuzzy scoring scale–synonymy strength.

5. Merging all Annotations' Ranking

The scoring annotations from the three annotators were merged by averaging them to reflect a more balanced estimate. The average ranks of these substitutions are rearranged in descending order to obtain the final ranking for each instance. One annotator checks the final ranks and removes the inappropriate substitutes whose score is under 60% as non-synonymous. Table 3.1 presents some statistics about the AraLexSubD dataset.

Table 3.1: The Final Dataset Instances are in the Aralexsubd Dataset.

Domains	# of Target Words	# of Candidates	# of Candidates < 60%
Finance	80	254	9
Medical	80	282	22
General	470	1940	29
Entire dataset	630	2476	60

Table 3.2 presents the number of target words in each domain categorized per each PoS tag. The target words (nouns, verbs, and adjectives) are 318, 256, and 57, respectively. The finance and medical domains did not contain verbs, as we cannot easily find polysemous verbs in such specialized domains.

Table 3.2: PoS tags in the AraLexSubD dataset.

Domains	# Noun	Verb	#Adjective
Finance	63	0	17
Medical	70	0	10
General	185	256	29
Entire dataset	317	256	57

The AraLexSubD dataset, constructed and presented in the chapter, is available on GitHub at https://github.com/karajah2024/Arabic-Lexical-Substitution.git.

3.3 Ranking Experimental Setup Steps

Before starting the annotation process, we conducted a training session and ranking tests. The purpose of the training session was to emphasize the notion of synonymy. The three linguists who carried out the ranking phase took part in three ranking tests.

The training session explained the ranking guidelines to determine if the suggested candidates in each sentence are synonyms for the target word by substituting the candidates and determining if it alters the sentence's meaning. This training session emphasized that linguists have the same understanding and agreement as much as possible, making their ranking consistent.

Three tests were performed by giving each linguist 10, 20, and 50 sentences to try alone. Then, the results for each test are discussed jointly to compare their works and identify the gaps and inconsistencies. After the first test is performed and jointly discussed, the next test is conducted, and so on. After that, each linguist is given the sentences and suggested candidates for the three domains in a separate file in Google Sheets.

The AraLexSubD dataset was completed after 4 months. The working hours were distributed as follows:

1. Determining the domains, extracting target words and sentences, and providing substitutions took about one month.
2. Scoring 630 sentences with 2476 candidates (synonyms) took 25 working hours for each annotator linguist over one month.
3. The scoring annotations from the three annotators were merged into one score by averaging them, which took about 2 working hours.
4. The annotator who removed the inappropriate substitutes whose score is under 60 took about 1 working hour.

5. Adding the lemma and the root for each target word and its candidates from Qabas took about 25 working hours over one month.

3.4 Linguists Agreement Evaluation

In order to assess the quality of the AraLexSubD dataset annotations and measure the agreements/disagreements among the three linguists, the Root Mean Squared Error (RMSE) was computed among their scores, as presented in Table 3.5. A smaller RMSE value indicates higher agreement and consistency among linguists, while a more significant RMSE value suggests more significant divergence in their annotations.

The RMSE was computed pairwise among the scores given by three linguists (L1, L2, L3) to understand how much they agreed or deviated from one another. RMSE was also calculated between each linguist's scores and the average score to assess how far an individual linguist's annotations were from the overall consensus.

The RMSE values of the linguists (L1, L2, L3) ranged between 0.17 - 0.19 for pairwise agreement among linguists, suggesting moderate agreement. The RMSE between each linguist's scores and their average scores ranged between 0.11 and 0.13, indicating a relatively minor deviation.

The RMSE does not measure the accuracy of the linguist's scores. It highlights how the deviation between them. The results showed that linguists L1 and L2 have the closest RMSE to their average scores, while Linguists L1 and L3 have the highest RMSE to their average scores.

Table 3.3: The Root Mean Squared Error (RMSE) between the scores of each linguist and the average scores of all linguists.

	L1 (RMSE)	L2 (RMSE)	L3 (RMSE)	Avg (RMSE)
L1		0.17	0.19	0.12
L2	0.17		0.18	0.11
L3	0.19	0.18		0.13
Avg	0.12	0.11	0.13	

Chapter Four

Methodology

This chapter presents the methodology followed in this research, designed to address the research objectives and answer the research questions systematically. The study involves several interconnected phases encompassing dataset construction, model development, and evaluation. This chapter focused on developing the AraLexSubPro pipeline, which integrates traditional, contextual, and hybrid approaches for synonym extraction [66].

4.1 AraLexSubPro Pipeline Methodology

This chapter presents an LS pipeline, AraLexSubPro, which provides different techniques for generating, selecting, and ranking substitution. To make a thorough comparison, AraLexSubPro uses four different methods as baselines to generate substitution candidates for the target words: a synonym dictionary approach (AWN), a pre-trained language model approach (AraBERT), AraBERT dropout approach(partial masking), and a hybrid approach using AraBERT and AWN.
The generated substitutions are filtered and then ranked based on **six** high-quality features to compare thoroughly: word similarity, word frequency, BERT prediction order (BERT probability), BERT-based language model (Loss), BERT similarity, and the BERTscore. The substitutions are then reranked based on our AraLexSubPro ranker. The AraLexSubPro pipeline was evaluated using the first Arabic LS benchmark, the AraLexSubD dataset. This thesis presents the first comprehensive study on the Arabic LS task. The pipeline was implemented in Python using Google Colab, a platform that offers powerful server resources for efficiently processing and running the developed model.

This section has three contributions:

1. It proposes four different generation baselines for the Arabic LS task using AraLexSubPro.
2. It ranks the candidates based on six high-quality features.
3. It examines the six-ranking feature, the BERTscore, for the particular domain dataset (finance and medical domains) to capture how IDF (Inverse Document Frequency), which differs in each domain, affects the candidate ranking. Since each domain has its own set of specific synonyms, they are ordered based on their IDF in that particular domain.

The AraLexSubPro pipeline has three steps: substitution generation, filtering, and ranking. The structure of our AraLexSubPro pipeline is shown in Figure 4.6.

- Substitution Generation aims to generate substitution for the target words. We present four different methods for substitution generation: a synonym dictionary approach (AWN), an AraBERT approach, an AraBERT dropout approach (partial masking), and a hybrid approach using AraBERT and AWN.
- Substitution Filtering removes the inappropriate substitutions from the generated substitutions. The filtering stage includes the PoS filter, a semantic filter, and a post-processing filter.
- Substitution Ranking ranks the candidate substitutions that suit the target word context and preserve the sentence's meaning. We utilize six ranking features: word similarity, word frequency, BERT prediction order (BERT probability), BERT-based language model (Loss), BERT similarity, and the BERTscore.

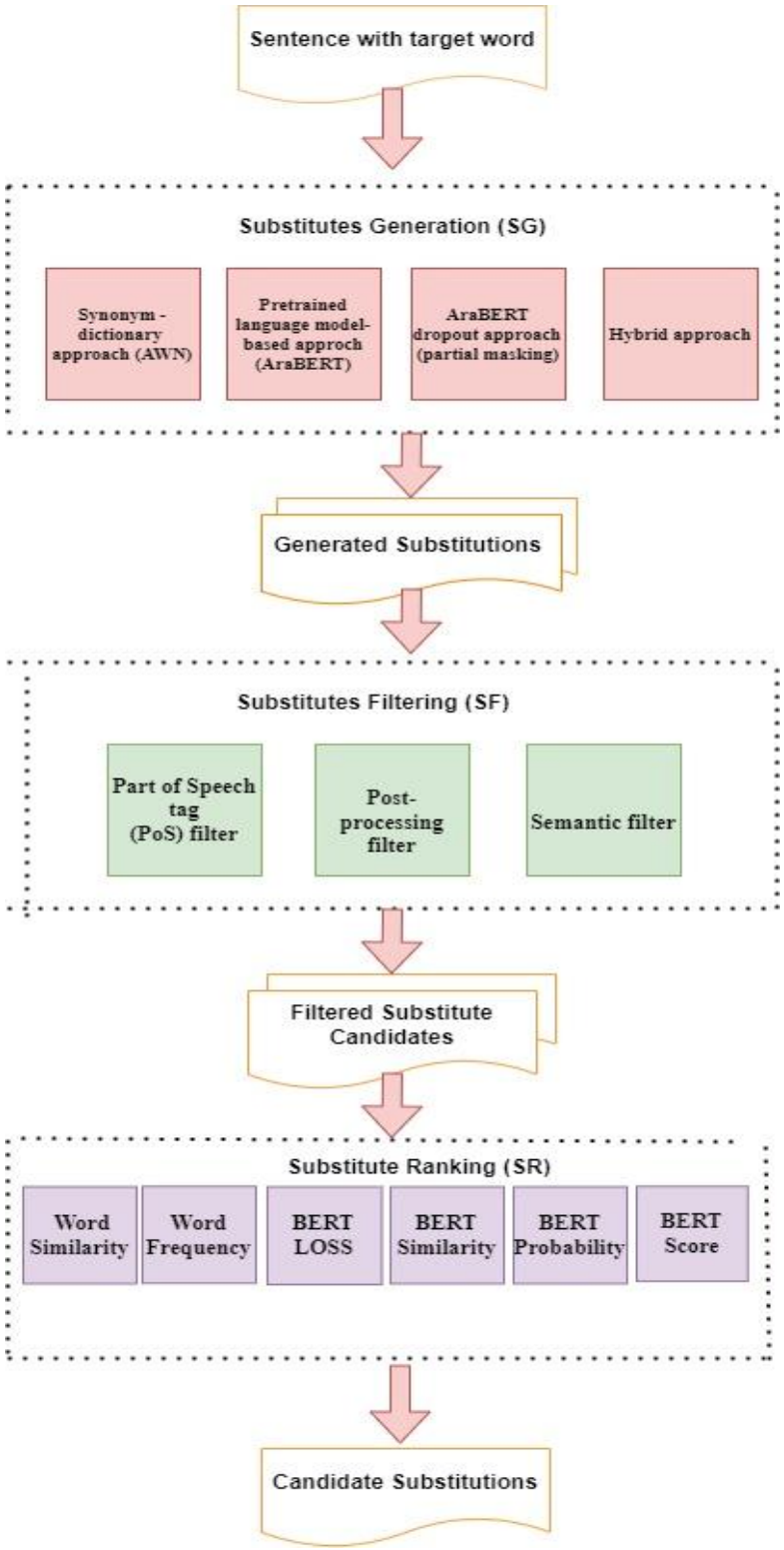


Figure 4.1: Arabic Lexical Substitution pipeline (AraLexSubPro).

4.2 Subistiution Generation

Suppose sentence **S** with target word w. The substitution generation generates substitutions that preserve the sentence's meaning. To make a thorough comparison, AraLexSubPro uses four different approaches as baselines to generate Substitution candidates for the target words:

1. **Synonym dictionary approach (AWN)**

Most synonym extraction (generation) algorithms use dictionaries, thesaurus [9], and the PWN [3, 6]. This step generates substitutions using AWN [56]. The generated substitutions do not consider the target word context in the sentence, so all the meanings (polysemous) with the same PoS of the target word appear as suggested substitutions. The advantage of this method is that it is simple and easy to implement. However, AWN (10,000 distinct synsets) is smaller than PWN (117,000 distinct synsets). The manual construction of WordNet is expensive and time-consuming [9].

2. **An AraBERT approach**

This step generates substitutions using the AraBERT language model [57], an Arabic bi-directional language model that applies BERT-masked language modeling.

BERT is a deep bi-directional model and self-supervised method based on the encoder in the transformer architecture. The transformer provides more structured memory, which handles long-term dependencies in the text. BERT can be trained with masked language modeling (MLM) and next-sentence prediction (NSP). MLM predicts the next word in a sequence given its left and right context, while NSP checks if the second sentence in the pair that is given is the subsequent sentence in the original text [8]. BERT achieves the NSP task by prepending every sentence with a particular classification token [CLS] and a unique separator token [SEP] combined with the sentences. During training, BERT applies the masked language modeling task by replacing random words with unique tokens [MASK]. The bi-directional nature of the BERT model allows candidate generation depending on the whole sentence context [58].

The AraBERT is fed with (S, Ś) sentence pair to generate the substitutions [14], where S is the sentence with its target word w, and Ś is the same sentence after masking its target word w with [MASK] symbol. AraBERT tokenizes the sentence into tokens before converting them into their embedding vector. For example, consider this sentence S in the general domain of AraLexSubD with target word w (بعينه):

لا يظهر تحيزاً لأحد بعينه

[Do not show bias towards anyone in particular]

The sentence لا يظهر تحيزاً لأحد بعينه is segmented as follows:

لا+ يظهر+ تحيزاً+ ل+أحد ب+عين+ه

The sentence pair (S, Ś) is fed into AraBERT, as shown in Figure 4.7. AraBERT generates a substitute list for the [MASK] word (عين, particular).

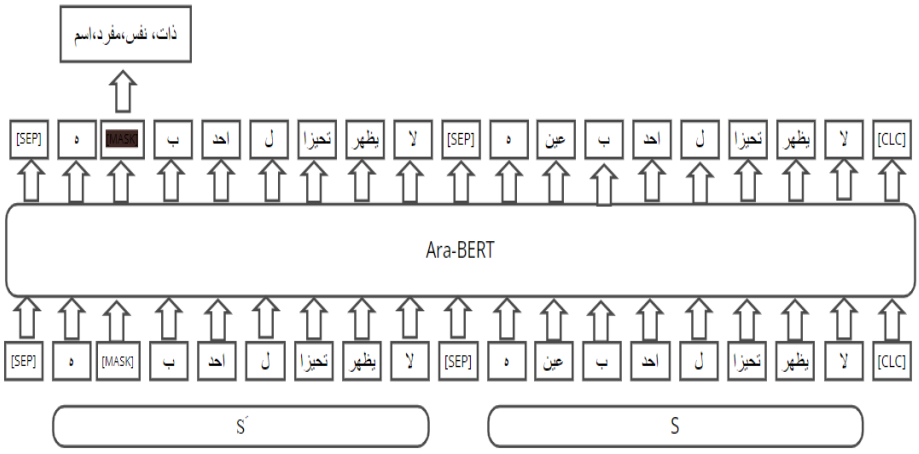


Figure 4.2: The substitution generation of AraLexSubPro for the target word prediction. The sentence is [لا يظهر تحيزاً لأحد بعينه/ Do not show bias towards anyone in particular] with the target word [عين/ particular]. [MASK], [CLS], and [SEP] are BERT special symbols, where [MASK] is used to mask the word, [CLS] is added before each input instance, and [SEP] is a unique separator token.

3. An AraBERT Dropout (Partial masking) approach

The dropout method [33] was proposed for English. It partially drops out the target word's embedding, which helps the BERT consider its semantics and contexts to generate substitute candidates at the target word position. The candidates were then validated based on the global contextual representation of the sentence. This is the first attempt to apply the dropout in [33] to Arabic. In this step, AraBERT generates substitutions, feeding the AraBERT as in [33] with a single sentence, and the target word embedding vector is partially dropped out.

4. A hybrid approach using AraBERT and AWN

The hybrid approach, presented in Algorithm 1, combines the AWN and AraBERT methods. Our hybrid approach Jaccard similarity is computed using the AWN synsets and the AraBERT substitutions, where the result is between 0-1. For example, suppose that the generated synset from the synonym dictionary (AWN) is two synsets: A:= { A1, A2, A3} where A1-A3 are the candidate substitutions of A and B:= { B1, B2, B3, B4, B5, B6, B7, B8} where B1-B8 are the candidate substitutions of B. The pre-trained language model (AraBERT) generated substitutions are C:= {C1, C2, C3, C4, C5} where C1-C5 are the candidate substitutions of C. Jaccard value is computed between (A, C), and (B, C). The two synsets with the highest Jaccard value are concatenated into one synset. Suppose the highest Jaccard value is between (A, C), then the two synsets A and C are concatenated. If the Jaccard value is zero, then the AraBERT synset is the hybrid synset.

Algorithm 1: **Hybrid approach using AraBERT and AWN**

```
Sentence   S
Target word   w
AraBERT substitutions  AraBERT_Sub
AWN synsets        AWN_Sys
Hybrid substitutions   Hybrid_Sub
 Hybrid_Sub = [ ]
Jaccard Similarity      Jaccard_sim
Jaccard_sim ←0
For each w ∈S, do
 AraBERT_Sub ←Substitution Generation(S,w) by AraBERT
 AWN_Sys ←  synset retrieved (w) from AWN.
      For each AWN_Sys AND AraBERT_Sub do
         Compute Jaccard_sim  between the AWN_sys  AND  AraBERT_Sub.
           Identify the highest Jaccard_sim
               If  Jaccard_sim >0 then
                     Jaccard_sim ← Jaccard_sim
                   Hybrid_Sub =  Concatenate AWN_Sys AND AraBERT_Sub
               else
                     Hybrid_Sub =   AraBERT_Sub
               end If
          end For
       return Hybrid_Sub
end For
```

4.3 Substitution Filtering

The filtering stage of the generated substitutions is shown in this section. This step is necessary to remove the unsuitable generated substitutions. The step includes a PoS filter, Post-processing filter, and semantic filter, which are described below:

1. The PoS filter filters the PoS tag of the generated substitutions from the AraBERT and AraBERT dropout and the hybrid approach. Filtered based on PoS consistency with the target words using Camel and Farasah taggers.

2. The post-processing filter filters the substitutions of AraBERT, AraBERT dropout, and hybrid approaches by removing the target word and its morphological derivatives, the morphological derivation of the generated substitutions, and symbols. The base of comparison is the lemma of the target word and the lemma of the generated substitute.
3. The semantic filter filters the generated substitutions of the synonym dictionary approach (AWN) using AraBERT to measure the semantic similarity between the contextual embeddings of the original sentence with the target word and the contextual embeddings of the original sentence after exchanging the target word with each generated substitution. Then, the cosine similarity between them is calculated. The words that have the highest similarity values are considered synonyms.

4.4 Substitution Ranking

Substitution candidate ranking based on the ranking features has two paths: (1) either ranking the generated substitutions, which are generated by the LS generation method or the filtered generated substitutions, and (2) reranking the annotated substitutions, which are ranked and given in the dataset [35]. In our ranking methodology, we followed the second path.

Six high-quality ranking features: word frequency, word similarity, BERT prediction order, BERT similarity, language model, and BERTscore. Each feature captures an aspect of fitting the substitute to replace the target word. A ranking score is computed for each feature. The six high-quality features used for ranking are described below:

1. The word frequency feature is used in many LS approaches, such as [59,60], using the Zipf scale that uses the SUBTLEX lists proposed by Marc Brysbaert [61]. The Zipf word frequency aims to return the word frequency offered by Marc Brysbaert's logarithmic scale, which provides the frequency of a word in over 40 languages, including Arabic, collected from five huge corpus domains: Wikipedia, Subtitles, News, Web text, and Twitter [62]. The Arabic language is morphology-rich, so word frequency plays a role in Arabic language processing. The more frequently words are, the more familiar they are to readers.
2. The word similarity feature is a ranking feature used to capture the semantic similarity between the target word and the suggested substitutions. The vector representation is obtained using word embedding models. The ARAVEC model [63] is used, and the cosine similarity between the target word and each substitution is computed. A higher similarity value means semantically similar appropriate alternatives with a higher ranking value [37].
3. The language modeling feature is a ranking feature (BERT loss) [14] used to evaluate the fluency of substitutes for each sentence in our dataset. The AraBERT is used to calculate the sentence probability. Let w be the target word and $W = w_{-n} \ldots w_{-1}, w, w_1 \ldots, w_n$, be the target word context. The sentence probability is calculated by replacing the target word w with each substitution candidate. Then, mask one word of W in the sentence around the target word position from back and front, then feed it into AraBERT to calculate the cross-entropy loss of the masked word. This step is repeated for each word in the sentence. The substitute candidates will be ranked based on the average loss of W. The lower loss is a good substitute for the original target word. A context with a symmetric window size equal to five is used around the target words.
4. The BERT prediction order feature is a ranking feature (BERT probability) that predicts the probability distribution of the words corresponding to the masked word. This feature includes information about the context and the target word. The substitute with a higher probability is more relevant to the original target word.
5. BERT similarity feature is a ranking feature that depends on BERT for the contextual representation of the original sentence and the contextual representation of the sentence

after replacing the target word with one of the generated substitution lists. It calculates the cosine similarity between these sentences. This feature measures how much the generated candidates preserve meaning. The substitute with a higher similarity is more relevant to the original target word.

6. BERTscore feature is a ranking feature that needs to find the sentence pairs $(S, S\hat{})$ where S is the reference sentence (the original sentence with its target word). $S\hat{}$ is the candidate sentence (the original sentence after exchanging the target word with each substitution). Each token contextual embedding in the sentence pairs is found using AraBERT. Then, the cosine similarity between each token vector in S and each vector in $S\hat{}$ is computed and weighted with inverse document frequency scores. BERTScore strongly correlates with human judgments and is widely used to evaluate text generation tasks [64].

Computing the BERTscore needs four steps:

1. Contextual Embeddings: Reference and candidate sentence pairs (S, Sˆ) are represented using contextual embeddings based on surrounding words, computed using AraBERT.
2. Cosine Similarity: The similarity between contextual embeddings of each token vector in reference and candidate sentences is measured using cosine similarity.
3. Importance Weighting: Rare word importance is considered using Inverse Document Frequency (IDF), which measures how important a term is across all documents in the corpus. It is calculated by taking the logarithm of the total number of documents in the corpus divided by the number of documents in which the term appears. The IDF is computed using the NADIA Dataset [65]. NADIA, which contains a large corpus parsed for medical and financial domain documents, is then tokenized to find IDF, as our AraLexSubD dataset contains medical and financial domains.
4. Token Matching for Precision and Recall: Each token in the candidate sentence is matched to the most similar token in the reference sentence, and vice versa, to compute Recall and Precision, which are then combined to calculate the F1 score.

The primary purpose of using the BERTscore is to capture the efficiency of this method for ranking in the Arabic LS task. Choosing the two domains (finance and medical domains) is to capture how the ranking of the candidates is altered by IDF (Inverse Document Frequency), which is different in each domain. Each domain has its own set of specific synonyms ordered based on their IDF in the particular domain. To benefit from the IDF, we use the BERTscore.

Suppose a writer searches for a word candidate. The BERT model generates substitutions and ranks them based on how BERT orders them. However, we went beyond the BERTscore and reranked the substitutions based on the word domain IDF. This means the substitutions are prioritized and arranged according to their relevance in the given domain. For instance, the word "ball" may have different candidates for someone involved in sports than someone studying geography.

For example, suppose that BERT ranks the specific domain candidates as {A, B, C, D, E, F}. The BERT model order comes from the trained BERT model corpus, while the BERTscore reranks them as { C, B, A, D, F, E}. We use the NADIA Dataset [65] to calculate IDF for the specific domain corpus (finance and medical corpus).

NADIA Dataset is parsed for Medical and financial domains. The domains are tokenized to find IDF. Table 4.3 presents the specifications of the medical and financial documents.

Table 4.3: The Medical and Financial Documents from the NADIA Dataset.

Medical & financial domains in NADIA Dataset	Medical domain	Finance domain
num of document	11,376 doc's	45,457 doc's
mean document length	427 words	480 words
total words in each domain	4,861,746 words	21,845,119 words
unique words in each domain	41,654 words	53,796 words

4.5 AraLexSubPro Algorithm

The LS algorithm AraLexSubPro is presented in Algorithm 2. For each sentence *S*, one target word of the types (noun, verb, adjective) is used to generate, filter, and rank the substitutions. One of the above four substitution generation methods is chosen to generate the substitutes for the target word. Afterward, filtration is done to clean up the generated substitutions, and then various rankings for each generated candidate using each feature are computed. Then, a new ranking score (Ranker) is computed by averaging the highest three ranking feature values.

Algorithm 2: AraLexSubPro Algorithm

```
for each target word w ∈S, do
    subs←Substitution Generation(S,w)
    subs←Substitution Filtering (S,w)
        for each filter ( the generation method), do
            filter the subs for each w
        end for
    subs←Substitute Ranking features(subs)
     score←0
          for each ranking feature f, do
             calculate feature score
               score←score (f)
          end for
     rank←rank(scores)
    avg_rank average ← (the highest rank value features)
 end for
```

Chapter Five

Results and Discussions

5.1 Introduction

This chapter describes the evaluation process, metrics, and results. The AraLexSubPro evaluation includes substitution generation, filtering, and ranking to validate the effectiveness of AraLexSubPro [66] using the evaluation dataset AraLexSubD [16].
To compare the performance of the different methods of generation and filtering in AraLexSubPro, we used the following evaluation metrics over the three domains:

- Potential: The proportion of instances for which at least one of the generated substitutions is in the annotated AraLexSubD.
- Precision: The proportion of generated substitutions in the annotated AraLexSubD, as shown in equation 5.1. This metric evaluates the quality of the generated substitutions.

$$Precision = \frac{|\{\text{Generated substitutions} \cap \text{Annotated substitutions}\}|}{|\{\text{Generated substitutions}\}|} \quad (5.1)$$

- Recall: The proportion of the annotated substitution s is included in the generated substitutions, as shown in equation 5.2. This metric assesses the coverage of the generated substitutions.

$$Recall = \frac{|\{\text{Generated substitutions} \cap \text{Annotated substitutions}\}|}{|\{\text{Annotated substitutions}\}|} \quad (5.2)$$

- F1: The harmonic mean between the precision and the recall, as shown in equation 5.3.

$$F1 = 2.\ \frac{\text{Precision}.\ Recall}{\text{Precision} + Recall} \quad (5.3)$$

The performance of different ranking features is compared using a manual evaluation metric in the three domains.

- The manual evaluation is calculated by calculating the fraction of the correct feature rank of candidates compared to the rank of candidates in the annotated AraLexSubD benchmark.

The evaluations were made based on the lemma (soft evaluation). All of the generated substitutions are lemmatized before comparison, which means the form of the word is not a basis for comparison. The generated substitutions may correct substitutions that have wrong transformations.

5.2 Evaluation of Substitution Generation

In the substitute generation stage, AraLexSubPro generates a set of potential substitutes by four approaches: a synonym dictionary-based approach (AWN), AraBERT, AraBERT dropout, and a hybrid approach between AraBERT and AWN. There are no limits on the number of generated substitutes. Table 5.1 presents the evaluation results and metrics of the AraLexSubPro generation methods over the three AraLexSubD domains and the entire AraLexSubD dataset.

Table 5.1: Automatic Evaluation Results For Generation Approaches for the Three Domains.

Domain	Generation method	Potential	Precision	Recall	F1
General	AWN	29.36	6.11	18.32	9.16
	AraBERT	58.11	29.87	25.18	27.32
	AraBERT dropout	30.43	9.63	10.26	9.94
	a hybrid	**59.3**	**30.90**	**26.62**	**28.60**
Medical	AWN	27.5	6.48	12.16	8.45
	AraBERT	73.90	32.37	39.15	35.44
	AraBERT dropout	38.76	12.57	18.13	14.84
	a hybrid	**74.22**	**33.56**	**40.25**	**36.60**
Finance	AWN	26.25	8.45	13.06	10.26
	AraBERT	62.5	21.20	28.57	24.34
	AraBERT dropout	33.75	10.54	14.57	12.23
	a hybrid	**63.51**	**22.24**	**29.69**	**25. 43**
EntireDataset	AWN	28.73	6.45	16.87	9.33
	AraBERT	62.23	27.25	30.08	26.85
	AraBERT dropout	33.01	10.32	12.07	11.13
	a hybrid	**64.11**	**25.35**	**31.51**	**28.10**

Figure 5.1 presents the bar chart of the F1 scores for the AraLexSubPro generation methods over the three AraLexSubD domains and the entire AraLexSubD dataset.

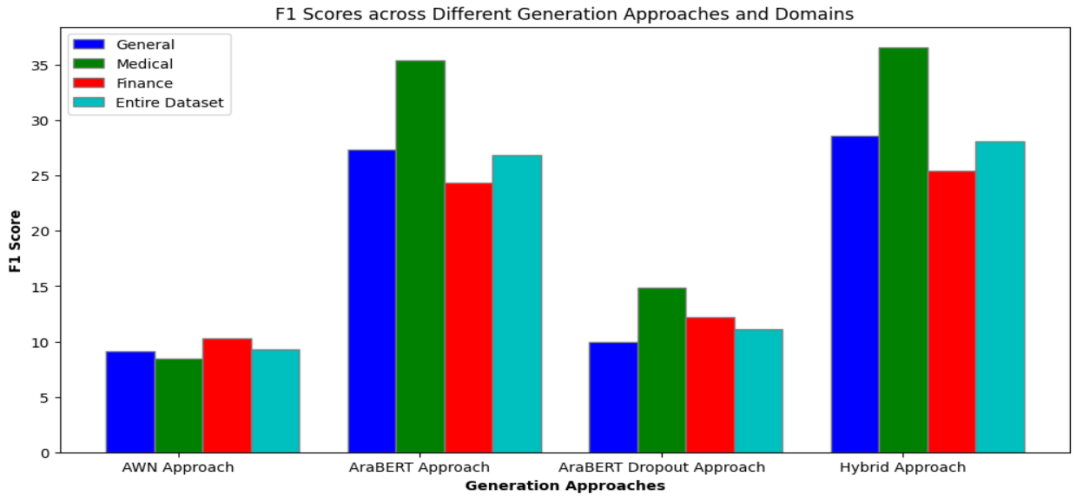


Figure 5.1: The chart of the F1 scores for the AraLexSubPro generation methods.

By comparing the F1 results for each generation method, we note that the ranking of the methods according to the best is as follows: Hybrid, AraBERT, AraBERT dropout, and AWN, respectively. The AraBERT and Hybrid generation methods are more effective than the AraBERT dropout and AWN methods.

AWN has the lowest precision value because no sense of disambiguation is carried out in synonym generation, which generates dozens of substitutions. The low recall value is due to the correct generated substitutes not being in the AraSublexD dataset or nothing being generated because of a significant limitation in their coverage (10,000 synsets).

The AraBERT method offers impressive results that only consider the target word context to generate substitutions without requiring input from a corpus or linguistic database as its trained model. AraBERT has a good balance of precision and recall values and a high potential, which means that AraBERT can generate at least one of the annotated substitutes in the AraLexSubD, the low recall value is due to the correct generated substitutes not being in the AraLexSubD dataset.

AraBERT Dropout, based on a single sentence, has worse results than AraBERT, which shows the strength of feeding the pre-trained model with sentence pairs (S, Ś) in lexical substitutions, it generates a semantically related substitution but not a similar of the target word meaning which cause a low precision and recall values.

The hybrid method, which combines the advantages of AraBERT and AWN, shows its power by offering the highest and best results in potential, precision, recall, and F1. However, sometimes, it is close to AraBERT due to AWN's significant limitation in their coverage. Many target words are not in AWN, which causes no intersection between the AWN and AraBERT synsets, which leads the Jaccard similarity value to be equal to zero.

5.3 Evaluation of Substitution Filtering

AraLexSubPro filters the substitutions generated by four approaches in the substitute filtering stage. The filtering stage includes three filters: the PoS filter, the Post-processing filter, and the Semantic filter. The evaluation is computed across the three domains separately, as well as the entire AraLexSubD dataset for each generation method.

Table 5.8 presents the filtring evaluation results for the AWN generation approach. We noticed the impressive results for the semantic filter, which increases precision and F1 in domains and the entire AraLexSubD dataset.

Table 5.2: Filtering Evaluation Results for AWN Generation Method.

Domain	Generation method	Potential	Precision	Recall	F1
General	Without filter	29.36	6.11	18.32	9.16
	Semantic filter	29.36	**7.09**	18.11	**10.20**
Medical	Without filter	27.50	6.48	12.16	8.45
	Semantic filter	27.50	**8.74**	12.01	**10.12**
Finance	Without filter	26.25	8.45	13.06	10.26
	Semantic filter	26.25	**8.71**	13.06	**10.45**
EntireDataset	Without filter	28.73	6.45	16.87	9.33
	Semantic filter	28.73	**10.50**	16.50	**12.83**

Figure 5.9 presents the bar chart of the F1 scores of the filtring evaluation results for the AWN generation approach, comparing results with and without the semantic filter over the three AraLexSubD domains and the entire AraLexSubD dataset.

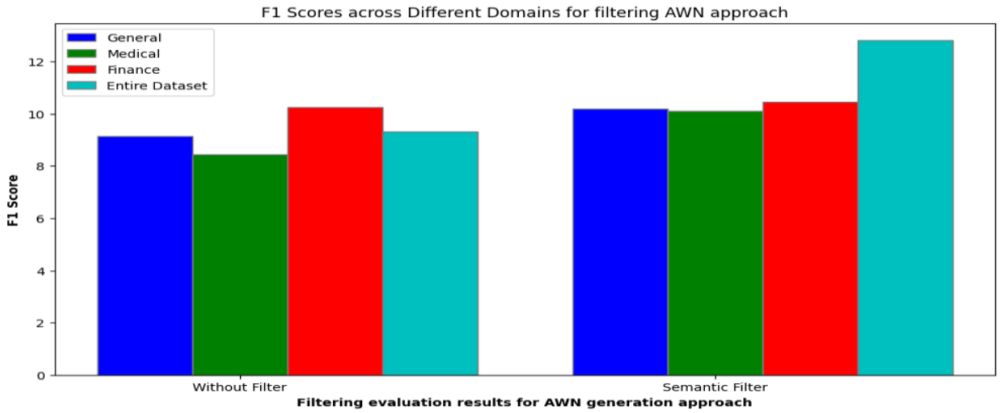


Figure 5.2: The chart of the F1 scores of the filtring evaluation results for the AWN.

Table 5.3 presents the filtering evaluation results for the AraBERT generation approach. The results show the importance of PoS and post-processing filters in cleaning up the generated substitutions. We noticed significant results when applying both filters, increasing precision and F1 in different domains and the entire AraLexSubD dataset.

Table 5.3: Filtering Evaluation Results for the AraBERT.

Domain	Filter name	Potential	Precision	Recall	F1
General	Without filter	58.11	29.87	25.18	27.32
	PoS filter	52.50	30.50	22.40	25.83
	Postprocessing filter	57.61	34.61	25.22	29.18
	(PoS+Postprocessing)Filters	51.42	36.32	22.42	**27.73**
Medical	Without filter	73.90	32.37	39.15	35.44
	PoS filter	71.82	39.60	36.40	37.93
	Postprocessing filter	73.91	42.50	39.10	40.73
	(PoS+Postprocessing)Filters	71.42	48.20	36.91	**41.81**
Finance	Without filter	62.50	21.20	28.57	24.34
	PoS filter	59.55	23.50	26.56	24.94
	Postprocessing filter	62.53	27.21	28.45	27.81
	(PoS+Postprocessing)Filters	59.62	29.61	26.35	**27.89**
EntireDataset	Without filter	62.23	27.25	30.08	26.85
	PoS filter	57.56	30.12	27.56	28.78
	Postprocessing filter	61.44	33.55	30.41	31.90
	(PoS+Postprocessing)Filters	57.66	36.48	27.44	**31.32**

Figure 5.3 presents the bar chart of the F1 scores of the filtring evaluation results for the AraBERT generation approach over the three AraLexSubD domains and the entire AraLexSubD dataset.

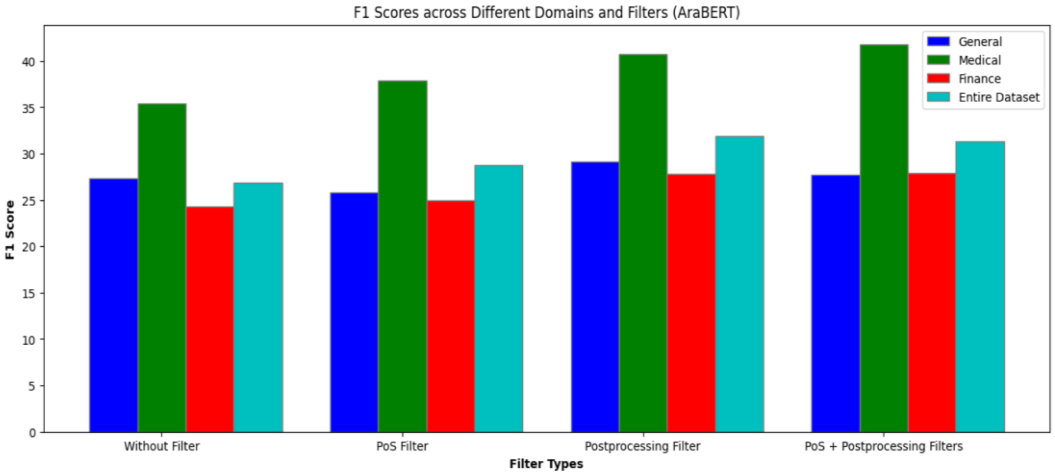


Figure 5.3: The Chart of the F1 Scores of the Filtring Results for AraBERT.

As presented in Table 5.9, despite the importance of PoS filtering, it decreases the potential and recall values due to ambiguity, leading to wrong PoS tagging from Camel and Farasah taggers, which removes correct substitutions, explaining the decrease in potential value.
The post-processing filter improves the precision and F1 values results for the three domains and the entire AralexSubD dataset.
Combining PoS and post-processing filters increases the precision but decreases the recall. Applying the PoS and Post-processing filters in the medical domain shows significant improvement. However, combining both filters increases the precision and F1 score, improving the substitutions' quality.
The results of the filtering evaluation of the AraBERT dropout approach are presented in Table 5.10. The results for applying each filter alone show that PoS filtering decreases the potential and recall, while the post-processing filter improves precision and F1. In contrast, applying both filters produced noteworthy effects, enhancing precision and F1 for the three domains and overall on the AraLexSubD dataset.

Table 5.4: Filtering Evaluation Results for AraBERT Dropout.

Domain	Filter name	Potential	Precision	Recall	F1
General	Without filter	30.43	9.63	10.26	9.94
	PoS filter	26.59	11.21	8.86	9.90
	Postprocessing filter	29.36	11.10	10.05	10.55
	(PoS+Postprocessing)Filters	26.17	**11.52**	8.79	**9.97**
Medical	Without filter	38.76	12.57	18.13	14.85
	PoS filter	37.5	15.10	16.56	15.80
	Postprocessing filter	38.75	14.89	17.39	16.04
	(PoS+Postprocessing)Filters	37.5	**16.93**	16.56	**16.75**
Finance	Without filter	33.75	10.54	14.57	12.23
	PoS filter	31.25	11.21	11.97	11.58
	Postprocessing filter	33.75	12.04	14.26	13.06
	(PoS+Postprocessing)Filters	31.25	**13.25**	11.98	**12.60**
EntireDataset	Without filter	33.01	10.32	12.07	11.13
	PoS filter	29.52	12.01	10.60	11.26
	Postprocessing filter	32.70	12.20	11.98	12.09
	(PoS+Postprocessing)Filters	29.36	**12.72**	10.55	**11.54**

Figure 5.11 presents the bar chart of the F1 scores of the filtring evaluation results for the AraBERT dropout generation approach over the three AraLexSubD domains and the entire AraLexSubD dataset.

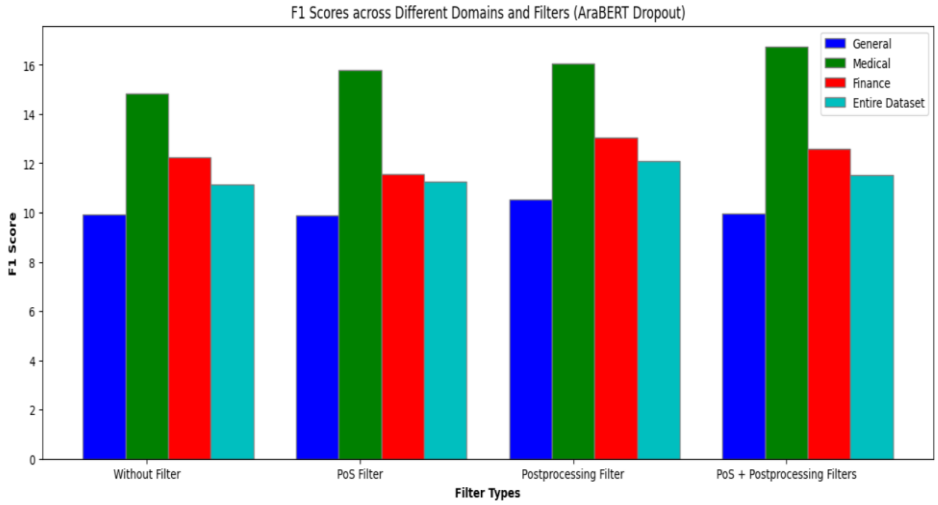


Figure 5.4: The Chart of the F1 Scores of the Filtring Results for AraBERT Dropout.

Table 5.5 presents the filtring evaluation results of the Hybrid approach. When both filters were used, we saw noteworthy outcomes in precision and F1, which increased in the three domains and the entire AraLexSubD dataset, and recall increased in the general domain. While applying each filter alone, PoS filtering decreases the potential and recall in medical, finance, and the entire AraLexSubD dataset while increasing the recall in the general domain. The post-processing filter improves precision and F1 in the three domains, and the entire AraLexSubD dataset.

Table 5.5: Filtering Evaluation Results for the Hybrid.

Domain	Filter name	Potential	Precision	Recall	F1
General	Without filter	60.3	32.90	26.62	29.43
	PoS filter	56.95	31.06	27.58	29.22
	Postprocessing filter	58.01	35.24	28.18	31.32
	(PoS+Postprocessing)Filters	57.32	38.99	27.29	32.11
Medical	Without filter	74.22	33.56	40.25	36.60
	PoS filter	72.5	40.63	37.82	39.17
	Postprocessing filter	73.75	43.10	39.59	41.27
	(PoS+Postprocessing)Filters	72.5	49.73	37.51	42.76
Finance	Without filter	63.51	22.24	31.69	26.14
	PoS filter	59.75	23.63	28.81	25.96
	Postprocessing filter	64.51	28.16	30.57	29.32
	(PoS+Postprocessing)Filters	59.76	30.52	28.49	29.47
EntireDataset	Without filter	64.11	28.35	33.51	30.71
	PoS filter	58.41	31.94	28.25	29.98
	Postprocessing filter	61.74	34.50	31.61	32.99
	(PoS+Postprocessing)Filters	57.94	38.47	28.99	33.06

Figure 5.12 presents the bar chart of the F1 scores of the filtring evaluation results for the Hybrid generation approach over the three AraLexSubD domains and the entire AraLexSubD dataset.

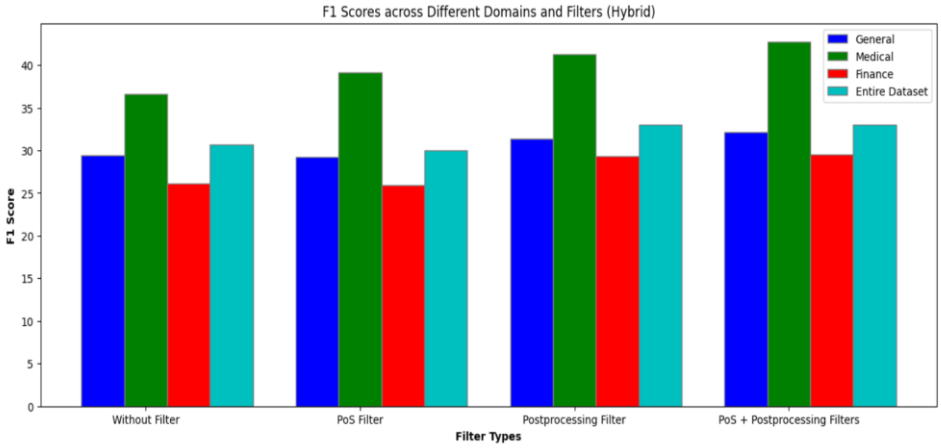


Figure 5.5: The Chart of the F1 Scores of the Filtring Results for Hybrid.

The filtered hybrid method shows its power in increasing the F1 comparable to the filtered AraBERT,the filtered hybrid F1 value is higher in the entire AraLexSubD dataset by 1.74, 4.38 in the general domain, 0.95 in the medical domain, and 1.58 in the finance domain.

5.4 Evaluation of Substitution Ranking

AraLexSubPro ranks the substitutions by six high-quality candidate ranking features: word similarity, word frequency, BERT prediction order (BERT probability), BERT-based language model (Loss), BERT similarity, and the BERTscore.
In the AraLexSubPro ranking stage, the annotated candidates are received from the AraLexSubD benchmark. Then, the ranking score is computed using each rank feature, which orders the annotated substitutions according to their appropriateness. The evaluation is computed across the three domains and the entire AraLexSubD dataset, as presented in Table 5.6.
Two annotators evaluate the feature rank manually by comparing the annotated candidates' rank in the AraLexSubD benchmark with each feature rank.

Table 5.6: Manual Evaluation Results for the AraLexSubPro Ranking Features.

Domain	Rank feature	Manual evaluation
General	Word similarity	44.33
	Word Frequency	49.68
	BERT LOSS	42.02
	BERT similarity	42.85
	BERT probability	51.89
	AraLexSubPro ranker	**52.85**
Medical	Word similarity	47.50
	Word Frequency	46.55
	BERT LOSS	45.31
	BERT similarity	43.85
	BERT probability	49.79
	AraLexSubPro ranker	**49.93**
Finance	Word similarity	51.40
	Word Frequency	49.92
	BERT LOSS	43.70
	BERT similarity	48.17
	BERT probability	50.19
	AraLexSubPro ranker	**52.50**
All Dataset	Word similarity	45.56
	Word Frequency	49.24
	BERT probability	50.76
	AraLexSubPro ranker	**51.48**

As presented in 5.6, the best three feature results are Word similarity, Word Frequency, and BERT probability. An AraLexSubPro ranker employs a linear combination of these three features by assigning equal weight to them. Hence, the resulting aggregate is the average of the candidate score in each feature. Combining these three features enhances LS ranking to identify suitable substitutions that fit the context and ensure grammatical coherence and semantic faithfulness.
The effectiveness of the ranker is presented in Table 5.12. The ranker ranks correctly the ranked annotated substitution in the AraLexSubD by 52.85% in the general domain, 49.93% in the medical domain, 52.5% in the finance domain, and 51.48% for the whole AraLexSubD dataset. Compared to the BERT probability feature, the ranker enhances the result by 0.96% in the general domain, 0.14% in the medical domain, 2.3% in the finance domain, and 0.72% for the whole AraLexSubD dataset.
The sixth ranking feature, the BERTscore, is applied to the finance and medical domains to capture and examine how IDF affects the candidate ranking. Since each domain has its own set of specific synonyms, they are ordered based on their IDF in that particular. The results are presented in Table 5.13.

Table 5.7: Manual Evaluation Results for the BERT Score Ranking Feature for the Finance and Medical Domains.

BERTscore	Medical domain			Finance domain		
Domain	Precision	Recall	F1	Precision	Recall	F1
Medical domain	**57.99**	**58.02**	**58.01**	43.05	42.19	42.62
Finance domain	37.66	39.72	38.67	**54.66**	**57.76**	**56.17**

The results show the importance of IDF in computing the BERTscore in medical and finance domains. It achieves the best rank score value compared to the other features in Table 5.7. The best rank result in the medical domain was the AraLexSubPro Ranker, which achieved 49.93, while BERTscore achieved 58.01 in F1. Also, the best rank result in the finance domain was the AraLexSubPro Ranker, which achieved 52.5, while the BERTscore reached 56.17 in F1.

5.5 Summary

This thesis presents the Arabic LS pipeline, AraLexSubPro, which contains three stages: generation, filtering, and ranking. It is evaluated using the AraLexSubD dataset.
We proposed four different methods to generate the substitutions and introduced six high-quality ranking features. Experiment results have shown that the hybrid method achieved the best generation results. Moreover, the AraLexSubPro ranker achieved the best ranking results, while the BERTscore achieved the best ranking results in the medical and finance domains.
The proposed AraLexSubD and AraLexSubPro can accelerate future research on this task. We noticed that the substitution generation method and substitution ranking feature can affect the performance of the LS pipeline.

Chapter Six

Qualitative Analysis

This chapter discusses the qualitative analysis of the construction of the AraLexSubD dataset and the AraLexSubPro pipeline.

6.1. Analysis of AraLexSubD Dataset

The manual dataset construction process is time-consuming and labor-intensive. Human annotators must consider appropriate substitutes, considering various linguistic and contextual factors. The task's complexity makes annotating many instances within a realistic budget and timetable challenging.
Additionally, providing annotators with all synonymous suggestions to the annotated target words is impossible, leading to missing entries that can affect the evaluation results.
During the construction of the AraLexSubD dataset, which annotates target words, sentences, and suitable substitutes, finding the roots and lemmas of the target word and substitutions was sometimes challenging for three main reasons:

1. Some target words and annotated substitutions have two roots in the three domains.
2. Some target words and annotated substitutions do not have a lemma on Qabas. The annotator found the word lemma and then added it to Qabas.
3. For the target words that are phrases of two words, such as [عين الاعتبار, in consideration], no lemma is found for these phrases in Qabas, but the lemma for each word alone is found. In the case of the lemma for the word [عين] and lemma for the word [الاعتبار], if the target word contains two words, then the lemma for the first word is the phrase lemma.

During the construction of the particular domains, medical and finance domains, ranking the suggested substitutions was sometimes challenging for two main reasons:

1. Lack of clarity of synonyms because they are specialized words and not general ones.
2. The options presented in the rank list are accurate.

6.2. Analysis of AraLexSubPro Pipeline

This section presents the qualitative analysis of the AraLexSubPro pipeline regarding substitute generation, filtering, and ranking to understand the source of the error. As Arabic is a morphologically complex and ambiguous language, some errors occur due to the following:

1. Difficulty: Some target words in the AraLexSubD dataset are Difficult to understand. It leads to generating wrong substitutions or even having no suggestions.
2. Ambiguity: Some target words in the AraLexSubD dataset are ambiguous, so Camel and Farasah taggers may wrongly tag the target word, leading to the wrong generation of substitutions.

3. Semantics: A wrong understanding of the sentence's semantics generates neutral or semantically related substitutions.
4. Model architecture: The architecture of the AraBERT model removed the last character of the generated substitutions, sometimes generating the target word's antonyms or neutral substitutions. Some approaches are suitable for specific languages but not other languages, such as partial masking, which is built for English but unsuitable for Arabic.
5. Coverage limitations in AWN, which is limited to 10,00 synsets besides the AraBERT model, are trained on a specific corpus, so the suggested substitutes will be limited to the training corpus.
6. Missing entries in the AraLexSubD dataset: some generated substitutions are correct but not annotated in the AraLexSubD dataset.

6.2.1 The analysis of substitution generation results:

The synonym dictionary-based approach is simple, easily understood, and deployed in different languages. As the task is an Arabic LS task, The AWN is used. The main problem with the AWN generation method is that it generates invalid candidates in all cases. Sometimes, the number of generated candidates is enormous, and most are wrong, as no sense of disambiguation is carried out. Sometimes, nothing is generated as AWN has only 10,000 synsets, which is very small compared to English WordNet.

Besides, AWN generates candidates depending on the PoS tag of the target word (the target word itself or its lemma). Sometimes, Camel and Farasah taggers can not determine the PoS tag correctly due to ambiguity, such as the target word [عين], which is found in two sentences with different meanings in the AraLexSubD dataset:

لا يظهر تحيزاً لأحد بعينه

[Do not show bias towards anyone in particular]

عينه بمنصب هام

[Appoint him to an important position]

In the first sentence, the target word [عين] is a noun, while in the second sentence, it is a verb. Camel tagged the target word [عين] in both sentences as nouns, bringing the two sentences to the same synsets, which are wrong.

The pre-trained language model-based approach (AraBERT) is a trained model that considers the target word context without requiring input from a corpus or linguistic database.

The generation evaluation results are affected by four factors: (1) Arabic is a complex language, and each word has many morphological forms, making LS challenging. (2) Arabic is a rich language. Many synonyms exist for each target word, and the annotators may potentially not annotate all candidates for the target words in AraLexSubD. (3) The AraBERT model is trained on a particular corpus, and the substitute generation is restricted to the training corpus. No substitute for the unknown target word will be generated. (4) The ARABERT model sometimes generates neutral substitutions. They are neither wrong nor correct but compromise the sentence's meaning. In comparison, the manual evaluation shows its importance by considering all correct substitutions

AraBERT Dropout (Partial masking) is based on feeding AraBERT with a single sentence that partially masks the target word. The dropout percentage is set to 0.3, and the alpha parameter is to 0.1, as in [9]. The evaluation results are worse than AraBERT, which shows the strength of feeding the pre-trained model with sentence pairs (S, Ś) in lexical substitutions. This approach should be adapted to generate more suitable synonyms in Arabic automatically.

The hybrid approach between AraBERT and AWN: The power of the hybrid technique is demonstrated by combining the benefits of both AraBERT and AW, which achieves the best generation results. The main challenge was owing to AWN's severe coverage limitations; because many of the target words are absent from AWN, there is no intersection of the AWN and AraBERT synsets, resulting in a zero Jaccard similarity value, which makes the results occasionally close to AraBERT.
The power of computing the Jaccard similarity instead of intersection as the generated substitutions from AWN and AraBERTcan be varied in length. If we consider taking the intersection instead of Jaccard, even if the intersection value between A∩ C and B∩C is high, many words that do not fit the context and the meaning will concatenate in the same synset.
Considering the results and nature of the four-generation methods on the AraLexSubD benchmark, combining different methods can create better substitution generators, as seen in the hybrid method.

6.2.2 The Analysis of Substitution Filtrating Results

The semantic filter, which AraBERT uses to compute the cosine similarity between the original sentence with the target word and the original sentence after replacing the target word with each generated substitution, shows its power, which increases F1 in domains and the entire AraLexSubD dataset.
Despite the importance of PoS filtering, due to ambiguity for some target words, Camel and Farasah taggers cannot correctly determine the PoS tag for the target words, which removed correct substitutions.
For example, in [عينه في منصب هام, He appointed him to an important position], the target word [عين, appointed] could be a verb or noun. However, it is a verb in this sentence, and the PoS tagger tags [عين, appointed] as a noun, which causes the PoS filter to delete all candidates with a verb tagging, explaining the decrease in potential value.
The post-processing filter improves the F1 results over the three domains and the whole AralexSubD dataset. However, it eliminates words that consist of two letters since it cannot distinguish between a subword and a complete word.
Applying both PoS and post-processing filters cleans up the generated substitutions, improving the F1 results in domains and the entire AraLexSubD dataset.

6.2.3 The Analysis of Substitution Ranking Results

The ranking features rank the candidates based on their weights, and the higher weights should be ranked higher. Our proposed ranker, AraLexSubPro Ranker, has competitive results and has reported better results than the other candidate ranking features.
The BERTscore ranking feature replaces the target word in the original sentence with each substitute candidate, producing an updated sentence. It uses the AraBERT model to measure the similarity between the original and updated sentences. The BERTScore ranking feature shows its effectiveness since it quantifies the extent to which each substitute has preserved the meaning of the original sentence. It directly tackles the main objective of lexical substitution: maintaining the sense of the original sentence while substituting a word. BERTscore achieved the best ranking results in the medical and finance domains.
Our AraLexSubPro Ranker and the BERTscore feature demonstrate promising results in capturing contextual information and ranking semantically substitutes, as they rank well-suited alternatives initially ranked lower in the other rank features ascend to higher positions.

6.3 AraLexSubPro Pipeline Error Types

In our AraLexSubPro pipeline, eight types of errors were identified:

1. No candidate substitutions are generated (Difficulty).
2. None of the generated substitutions are synonyms (Difficulty).
3. Part of the generated substitutions are synonyms (Semantics).
4. The generated substitutions are affected by the architecture of generational methods such as AraBERT (Model architecture).
5. The generated substitutions are neutral; they are neither wrong nor correct but compromise the sentence's meaning (Semantics).
6. The generated substitutions are synonyms but are not in the AraLexSubD dataset (Missing entries).
7. Due to ambiguity, the PoS tagger Farassa cannot correctly determine the PoS tag for the generated substitution (Ambiguity).
8. The filtering step removes the words of two letters as it cannot determine if it is a complete word or a subword.

Type 1, 2, 3, 4, 5, and 6 errors occurred during substitute generation, and errors 7 and 8 occurred during substitute filtering. Many sentences are recorded for each error type. Table 6.14 presents the sentence count in AraLexSubD for each error. Table 6.14 also shows that AraBERT makes the fewest errors of types 1, 2, 4, and 5. However, it can be noticed that AraBERT makes many errors in types 3 and 6.

Table 6.1. The Count of Each Error Type Results Over the Three Domains.

Domain\Error Type	1	2	3	4	5	6	7	8
General (470 sentences)	15	48	366	30	32	195	20	45
Medical (80 sentences)	3	3	72	3	5	11	4	5
Finance (80 sentences)	2	4	74	9	4	24	4	6
Entire dataset (630 sentences)	20	55	512	42	41	230	28	56

Below are examples that are chosen randomly as an example of each error type in substitute generation and substitute filtering errors:

- The error of type 1: In the medical domain, for the sentence [العضلات المركزية هي العضلات التي تتحكم في أطراف الجسم, Central muscles are the muscles that control the limbs of the body] with target word المركزية), Central), AraBERT generates no substitution except [ما، هذه و+، التي,] which are removed by the postprocessing filter.
- The error of type 2: In the medical domain, for the sentence [سيولة الدم من المؤشرات الحيوية المهمة لتشخيص الحالات الطبية, Blood fluidity is an essential vital indicator for diagnosing medical conditions] with target word (سيولة/ fluidity), AraBERT generates the substitutions (ندر، فصيل/rare, faction), and none have the same semantics of the target word and fit the context.
- The error of type 3: In the medical domain, for the sentence [يتم إزالة هامش من النسيج في العملية الجراحية, A margin of tissue is removed in the surgical procedure] with target word هامش), margin), AraBERT generates the substitutions (جزء، كيس، حيز، قسم / part, bag, space, section), a part of the generated substitutions are synonyms, which are (جزء,قسم/part, section) and are included in the AraLexSubD dataset.
- The error of type 4:
  - Missing the character ة: In the general domain, for the sentence [لكل شخص مستوى حياة معين, Every person has a certain standard of life] with target (حياة/life), AraBERT generates (إشار ، حال) substitutions, which are correct, and included in the AraLexSubD dataset, but with missing character ة at the end of the generated candidates, they should be (إشارة، حالة/Signal, condition).

- Antonyms: In the general domain, for the sentence [للمحل السياسي دراية بفنون القول, The political analyst is familiar with the art of speaking] with target (دراية, familiar), AraBERT generates (إلمام، علم، وعي، جهل/familiarity, knowledge, awareness, ignorance) substitutions. The substitutions (إلمام، علم، وعي) are correct and in the AraLexSubD dataset except for the جهل substitution, which is an antonym for the target word (دراية/familiar).

- The error of type 5: In the medical domain, for the sentence [تحديد حجم الورم في حالة السرطان يعتبر أمرا حاسما للتشخيص وتحديد خطة العلاج, Determining the size of the tumor in the case of cancer is crucial for diagnosis and identification] with target word (حجم, size), AraBERT generates the substitutes (مكان، نوع/Place, type), which are neutral, they are not wrong nor correct but compromise the sentence's meaning. The words (مكان، نوع) are suitable to the sentence but do not fit the original sentence context.
- The error of type 6: In the general domain, for the sentence [لكل شخص مستوى حياة معين, Every person has a certain standard of life] with target (حياة, life), AraBERT generates (عيش تفكير، عمر، معيشية، ذكاء) substitutions that are suitable, but the sentence itself contains more than one option. Still, the option was specified in the AraLexSubD dataset as معيشة.
- The error of type 7: In the medical domain, for the sentence [كل ساعة تجرى 7 عمليات جراحية, Every hour 7 surgeries are performed] with target word (عمليات/operations), AraBERT generates the substitute (تدخل / intervene), which is semantically correct and in the AraLexSubD dataset, but when applying the PoS filter, which removes the unmatched PoS tags in the filtering step, the substitute (تدخل) is removed as the tagger tags it as a verb, not a noun.
- The error of type 8: In the medical domain, for the sentence [صداع نصفي يصيب عادة قسما واحدا من الرأس, Migraines usually affect one part of the head] with target word (رأس/head), AraBERT generates the substitute (مخ), which is semantically related and not in the AraLexSubD dataset, but when applying the postprocessing filter, which removes the subwords, the substitute (مخ) is removed as the filter understands it as a subword containing two letters.

Chapter Seven

Conclusion and Future Work

7.1 Conclusion

This thesis introduces three contributions. Firstly, we clustered the most relevant works on extracting synonyms, focusing on the extraction techniques and their evaluation methods and datasets into four groups: extracting synonyms using translation graphs, extracting synonyms using discovering new transition pairs, constructing new WordNets approaches by exploring synonymy graphs, and synonym extraction using word embedding and language models.

Secondly, we built the AraLexSubD dataset, the first benchmark for evaluating Arabic lexical substitution, created with the expertise of eight native Arabic speakers, including linguists, a doctor, and an economist. The dataset includes 630 sentences and 2,476 substitution candidates and is divided into general, finance, and medical domains. The AraLexSubD dataset, constructed and presented in the thesis, is available on GitHub at https://github.com/karajah2024/Arabic-Lexical-Substitution.git.

Thirdly, we also developed AraLexSubPro, a pipeline for Arabic LS that operates in three stages: generation, filtering, and ranking. To generate possible substitutes, we used four methods: a synonym dictionary (AWN), a pre-trained language model (AraBERT), a modified AraBERT with partial masking, and a hybrid approach combining AraBERT and AWN. The generated substitutions were filtered using the PoS, Post-processing, and Semantic filters. This step was necessary to clean up the generated substitutions. The ranking step used six key features: word similarity, word frequency, BERT prediction order (BERT probability), BERT-based language model (Loss), BERT similarity, and the BERTscore. The substitutions are then reranked based on our AraLexSubPro ranker. The AraLexSubPro pipeline was evaluated using the first Arabic LS benchmark, the AraLexSubD dataset.

Experiment results indicate that traditional lexical resources are insufficient for accurate synonym extraction in Arabic. AWN is very small and does not contain all the Arabic words. On the other hand, AraBERT showed promising results in capturing the context of the target word, while the hybrid between AWN and AraBERT achieved the best results. Moreover, the AraLexSubPro ranker achieved the best ranking results, while the BERTscore achieved the best ranking results in the medical and finance domains.

This research answered the thesis questions for the Arabic LS for the automated Arabic synonym extraction, addressing the unique linguistic characteristics of the Arabic language. Arabic LS pipeline demonstrated its effectiveness in extracting synonyms automatically.

A significant contribution of this work was the AraLexSubD dataset, which improved the accuracy and reliability of automated synonym extraction in Arabic by providing high-quality, diverse annotations.

Additionally, the hybrid approach combining the Arabic WordNet (AWN) and AraBERT shows substantial accuracy and contextual relevance improvements, leveraging both lexical and contextual knowledge.
The filtering step showed its importance by cleaning up the substitution generation and enhancing the quality of extracted synonyms. Furthermore, incorporating ranking features allowed the prioritization of synonyms most appropriate to the given context.
This thesis presents the first comprehensive study on the Arabic LS task. The AraLexSubD dataset and AraLexSubPro pipeline are significant steps forward in Arabic LS research. While the initial results are promising, we recognize that choosing generation methods and ranking features plays a crucial role in the pipeline's performance. We plan to explore different approaches to refine these aspects further. We are confident that AraLexSubD and AraLexSubPro will be valuable for future research in this challenging yet vital Arabic natural language processing area.

7.2 Recommendations and Future Work

In the future, the presented work in this thesis could be expanded in several steps. We plan to expand the AraLexSubD dataset to be larger by adding more diverse examples. We also plan to build an automatic evaluation dataset to overcome the manual construction limitations.
We also intend to explore other substitution generation methods and ranking features to improve the Arabic LS task. Besides, we plan to fine-tune the AraBERT model to better deal with Modern Standard Arabic (MSA) characteristics such as poems and various Arabic dialects. We also intend to improve the partial masking approach, making it more suited to the complexities of the Arabic language.

References

1. Darwish, K., Habash, N., Abbas, M., Al-Khalifa, H., Al-Natsheh, H., Bouamor, H., Bouzoubaa, K., Cavalli-Sforza, V., El-Beltagy, S., El-Hajj, W., Jarrar, M., Mubarak, H. (2021). A Panoramic Survey of Natural Language Processing in the Arab World. Communications of the ACM, April 2021, Vol. 64 No. 4, Pages 72-81.
2. H.WU and M. ZHOU (2003, July). Optimizing synonym extraction using monolingual and bilingual resources. In Proceedings of the second international workshop on paraphrasing (pp. 72-79).
3. G. A. Miller, R. Beckwith, C. Fellbaum, D.Gross, Katherine and J. Miller, (1990) Introduction to WordNet: An On-line Lexical Database, International Journal of Lexicography, Volume 3, Issue 4, Winter 1990, Pages 235–244.
4. M. Jarrar (2019). The Arabic ontology–an Arabic WordNet with ontologically clean content. Applied Ontology, (Preprint), 1-26.
5. Al-Hajj, M., Jarrar, M., (2021). LU-BZU at SemEval-2021 Task 2: Word2Vec and Lemma2Vec performance in Arabic Word-inContext disambiguation. In Proceedings – the 11th Workshop on Semantic Evaluation (SemEval2021).
6. M. Jarrar, E. Karajah, M. Khalifa and K.Shaalan. (2020) Extracting Synonyms from Bilingual Dictionaries. Proceedings of the 11th Global WordNet Conference.
7. N. Mohammed, (2020). Extracting word synonyms from text using neural approaches. Int. Arab J. Inf. Technol., 17(1), 45-51.
8. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019, June). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (pp. 4171-4186).
9. Naser-Karajah, E., Arman, N., Jarrar, M. Current trends and approaches in synonyms extraction: Potential adaptation to Arabic. In Proceedings of the 2021 International Conference on Information Technology (ICIT), Amman, Jordan, 14–15 July 2021, IEEE: Piscataway, NJ, USA, 2021, pp. 428–434.
10. Qiang, J., Liu, K., Li, Y., Yuan, Y., & Zhu, Y. (2023, July). ParaLS: Lexical Substitution via Pretrained Paraphraser. In The 61st Annual Meeting Of The Association For Computational Linguistics.
11. Melamud, O., Levy, O., Dagan, I. A simple Word Embedding model for lexical substitution. In Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, Denver, CO, USA, 31 May–5 June 2015, pp. 1–7.
12. Wang, C., Mao, S., Ge, T., Wu, W., Wang, X., Xia, Y., ... & Zhao, D. (2023, July). Smart Word Suggestions for Writing Assistance. In Findings of the Association for Computational Linguistics: ACL 2023 (pp. 11212-11225).
13. Amrami, A., & Goldberg, Y. (2018). Word Sense Induction with Neural biLM and Symmetric Patterns. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics.
14. Qiang, J., Lu, X., Li, Y., Yuan, Y., & Wu, X. (2021). Chinese lexical simplification. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29, 1819-1828.
15. Farghaly, A., & Shaalan, K. (2009). Arabic natural language processing: Challenges and solutions. ACM Transactions on Asian Language Information Processing (TALIP), 8(4), 1-22.

16. Naser-Karajah, E., & Arman, N. (2024). Arabic Lexical Substitution: AraLexSubD Dataset and AraLexSub Pipeline. Data, 9(8), 98.
17. T. Flati, R. Navigli (2012). The CQC algorithm: Cycling in graphs to semantically enrich and enhance a bilingual dictionary. Journal of Artificial Intelligence Research, 43, 135-171.
18. D. Torregrosa, M. Arcan, S. Ahmadi and J. P. McCrae (2019). Tiad 2019 shared task: Leveraging knowledge graphs with neural machine translation for automatic multilingual dictionary generation. Translation Inference Across Dictionaries.
19. M. Villegas, M. Melero, J.Gracia and N. Bel. (2016, May). Leveraging RDF graphs for crossing multiple bilingual dictionaries. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16) (pp. 868-876).
20. K. Tanaka and K. Umemura (1994). Construction of a bilingual dictionary intermediated by a third language. In COLING, pages 297–303.
21. L. T. Lim, B. Ranaivo-Malançon, and E. K. Tang (2011). Low cost construction of a multilingual lexicon from bilingual lists. Polibits, 43:45–51,(2011).
22. S.Neale. (2018).A survey on automatically-constructed WordNets and their evaluation: Lexical and word embedding-based approaches. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation.
23. G.Ercan and F.Haziyevb (2019). Synset expansion on translation graph for automatic WordNet construction. Information Processing & Management, 56(1), 130-150.
24. K. N.Lam, F.Al Tarouti and J. Kalita (2014, June). Automatically constructing WordNet synsets. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (pp. 106-111).
25. Ustalov, D., Panchenko, A., & Biemann, C. (2017, July). Watset: Automatic Induction of Synsets from a Graph of Synonyms. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 1579-1590).
26. D.Ustalov, M. Chernoskutov, C.Biemann, and A.Panchenko (2017). Fighting with the sparsity of synonymy dictionaries for automatic synset induction. In International Conference on Analysis of Images, Social Networks and Texts,(pp. 94-105). Springer, Cham.
27. R.Navigli, and S.P.Ponzetto (2012). BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network.Artificial intelligence 193: 217-250.
28. I.Gurevych, J.Eckle-Kohler, S.Hartmann, M.Matuschek, C.M. Meyer, and C. Wirth (2012). UBY-a large-scale unified lexicalsemantic resource based on LMF. In Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, pp. 580-590.
29. F. Al Tarouti and J. Kalita (2016, June). Enhancing automatic WordNet construction using word embeddings. In Proceedings of the Workshop on Multilingual and Cross-lingual Methods in NLP (pp. 30-34).
30. M. Khodak, A. Risteski, C. Fellbaum and S. Arora (2017, April). Automated WordNet construction using word embeddings. In Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their Applications (pp. 12-23).
31. Al-Matham, R. N., & Al-Khalifa, H. S. (2021). Synoextractor: a novel pipeline for Arabic synonym extraction using Word2Vec word embeddings. Complexity, 2021(1), 6627434.
32. M.AlMaayah, M.Sawalha, and M. A. Abushariah. (2016). Towards an automatic extraction of synonyms for Quranic Arabic WordNet. International Journal of Speech Technology 19, no. 2 (2016): 177-189.

33. W.Zhou, T.Ge, K.Xu, F.Wei, and M. Zhou (2019). "BERT-based lexical substitution." In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 3368-3373.
34. Lacerra, C., Tripodi, R., & Navigli, R. (2021, November). Genesis: A generative approach to substitutes in context. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (pp. 10810-10823).
35. Seneviratne, S., Daskalaki, E., Lenskiy, A., & Suominen, H. (2022, October). CILex: An investigation of context information for lexical substitution methods. In Proceedings of the 29th International Conference on Computational Linguistics (pp. 4124-4135).
36. Michalopoulos, G., McKillop, I., Wong, A., & Chen, H. (2022, May). LexSubCon: Integrating Knowledge from Lexical Resources into Contextual Embeddings for Lexical Substitution. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 1226-1236).
37. Qiang, J., Liu, K., Li, Y., Li, Y., Zhu, Y., Yuan, Y. H., ... & Ouyang, X. (2023, December). Chinese Lexical Substitution: Dataset and Method. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (pp. 29-42).
38. Mc Carthy, D., Navigli, R. Semeval-2007 task 10: English lexical substitution task. In Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007), Prague, Czech Republic, 23–24 June 2007, pp. 48–53.
39. Sharoff, S. Open-source corpora: Using the net to fish for linguistic data. Int. J. Corpus Linguist. 2006, 11, 435–462.
40. Biemann, C. (2010). Turk Bootstrap Word Sense Inventory (TWSI) 2.0.
41. Kremer, G., Erk, K., Padó, S., Thater, S. What substitutes tell us—Analysis of an "all-words" lexical substitution corpus. In Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, Gothenburg, Sweden, 26–30 April 2014, pp. 540–549.
42. Ide, N., Baker, C., Fellbaum, C., Fillmore, C., Passonneau, R. MASC: The manually annotated sub-corpus of American English. In Proceedings of the 6th International Conference on Language Resources and Evaluation, LREC 2008, Marrakech, Morocco, 28–30 May 2008, European Language Resources Association (ELRA): Paris, France, 2008, pp. 2455–2460
43. Ide, N., Baker, CF, Fellbaum, C., Passonneau, R.J. The manually annotated sub-corpus: A community resource for and by the people. In Proceedings of the ACL 2010 Conference Short Papers, Uppsala, Sweden, 11–16 July 2010, pp. 68–73.
44. Lee, M., Donahue, C., Jia, R., Iyabor, A., & Liang, P. (2021, June). Swords: A Benchmark for Lexical Substitution with Improved Data Coverage and Quality. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 4362-4379).
45. Lacerra, C., Pasini, T., Tripodi, R., Navigli, R. ALaSca: An Automated approach for Large-Scale Lexical Substitution. In Proceedings of the 30th International Joint Conference on Artificial Intelligence, IJCAI 2021, Montreal, QC, Canada, 21–26 August 2021, pp. 3836–3842.
46. Zhang, X., Chen, Z., & Yu, Z. (2024). ProLex: A Benchmark for Language Proficiency-oriented Lexical Substitution. arXiv preprint arXiv:2401.11356.
47. Blanchard, D. (2013). TOEFL11: A Corpus of Non-native English. Educational Testing Service.
48. Hintz, G., & Biemann, C. (2015). Delexicalized supervised German lexical substitution. Proceedings of GermEval, 11-16.

49. Toral, A. (2009). The lexical substitution task at EVALITA 2009. In Proceedings of EVALITA Workshop, 11th Congress of Italian Association for Artificial Intelligence, Reggio Emilia, Italy.
50. Ghanem, S., Jarrar, M., Jarrar, R., & Bounhas, I. (2023, January). A Benchmark and Scoring Algorithm for Enriching Arabic Synonyms. In Proceedings of the 12th Global Wordnet Conference (pp. 274-283).
51. Jarrar, M., & Hammouda, T. H. (2024, May). Qabas: An Open-Source Arabic Lexicographic Database. In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024) (pp. 13363-13370).
52. Birzeit University. *Ontology Platform - Birzeit University*. Available online: https://ontology.birzeit.edu/ (accessed on 1 March 2024)
53. Shamela. Available online: https://shamela.ws/book/7028 (accessed on 1 March 2024).
54. DOHA Dictionary. Available online: https://www.dohadictionary.org/bibliography (accessed on 1 March 2024).
55. Almaany. Available online: https://www.almaany.com/ (accessed on 1 March 2024)
56. Elkateb, S., Black, W., Vossen, P., Farwell, D., Pease A., & Fellbaum, C. (2006). Arabic WordNet and the Challenges of Arabic. In Proceedings – Arabic NLP/MT Conference (pp. 665-670).
57. Antoun, W., Baly, F., & Hajj, H. (2020, May). AraBERT: Transformer-based Model for Arabic Language Understanding. In Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection (pp. 9-15).
58. Wu, X., Lv, S., Zang, L., Han, J., & Hu, S. (2019). Conditional bert contextual augmentation. In Computational Science–ICCS 2019: 19th International Conference, Faro, Portugal, June 12–14, 2019, Proceedings, Part IV 19 (pp. 84-95). Springer International Publishing.
59. Szarvas, G., Busa-Fekete, R., & Hüllermeier, E. (2013, October). Learning to rank lexical substitutions. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (pp. 1926-1932).
60. Szarvas, G., Biemann, C., & Gurevych, I. (2013, June). Supervised all-word lexical substitution using delexicalized features. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 1131-1141).
61. Beyond Kucera, M. (2009). 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(4), 977-90.
62. Wordfreq. Available online: https://pypi.org/project/ wordfreq/(accessed on 8 Jan 2024).
63. Soliman, A. B., Eissa, K., & El-Beltagy, S. R. (2017). Aravec: A set of arabic word embedding models for use in arabic nlp. Procedia Computer Science, 117, 256-265
64. Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2019). Bertscore: Evaluating text generation with bert. arXiv preprint arXiv:1904.09675.
65. R. Al-Debsi, A. Elnagar, and O. Einea. Nadia: News articles dataset in Arabic for multi-label text categorization. Mendeley Data, V2, 2019, doi: 10.17632/hhrb7phdyx.2.
66. Naser-Karajah, E., & Arman, N. (2024). Towards Automated Arabic Synonyms Extraction using Arabic Lexical Substitution. IEEE Access.
****
67. Shardlow, M. (2014). A survey of automated text simplification. International Journal of Advanced Computer Science and Applications, 4(1), 58-70.

68. Giuliano, C., Gliozzo, A., & Strapparava, C. (2007, June). Fbk-irst: Lexical substitution task exploiting domain and syntagmatic coherence. In Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007) (pp. 145-148).
69. Martinez, D., Kim, S. N., & Baldwin, T. (2007, June). MELB-MKB: Lexical substitution system based on relatives in context. In Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007) (pp. 237-240).
70. Erk, K., & Padó, S. (2010, July). Exemplar-based models for word meaning in context. In Proceedings of the acl 2010 conference short papers (pp. 92-97).
71. Thater, S., Fürstenau, H., & Pinkal, M. (2011, November). Word meaning in context: A simple and effective vector model. In Proceedings of 5th International Joint Conference on Natural Language Processing (pp. 1134-1143).
72. Miller, G. A. (1995). WordNet: a lexical database for English. Communications of the ACM, 38(11), 39-41.
73. Hassan, S., Csomai, A., Banea, C., Sinha, R., & Mihalcea, R. (2007, June). Unt: Subfinder: Combining knowledge sources for automatic lexical substitution. In Proceedings of the fourth international workshop on semantic evaluations (SemEval-2007) (pp. 410-413).
74. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems, 26.
75. Melamud, O., Goldberger, J., & Dagan, I. (2016, August). context2vec: Learning generic context embedding with bidirectional lstm. In Proceedings of the 20th SIGNLL conference on computational natural language learning (pp. 51-61).
76. Sefara, T. J., & Mokgonyane, T. B. (2021). Practical approach on implementation of Wordnets for South African languages.
77. Boukhatem, N. (2014). The Arabic natural language processing: Introduction and challenges. International Journal of English Language & Translation Studies, 2(3), 106-112.
78. S. Arora, Y Li, Y. Liang, T. Ma and A. Risteski (2016). A latent variable model approach to pmi-based word embeddings. Transactions of the Association for Computational Linguistics, 4, 385-399.(2016)
79. McCarthy, D., & Navigli, R. (2009). The English lexical substitution task. Language resources and evaluation, 43, 139-159.
80. Soderland, S., Etzioni, O., Weld, D. S., Skinner, M., & Bilmes, J. (2009, August). Compiling a massive, multilingual dictionary via probabilistic inference. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (pp. 262-270).
81. Helou, M. A., Palmonari, M., & Jarrar, M. (2016). Effectiveness of automatic translations for cross-lingual ontology mapping. Journal of Artificial Intelligence Research, 55, 165-208.
82. Arefyev, N., Sheludko, B., Podolskiy, A., & Panchenko, A. (2020). A comparative study of lexical substitution approaches based on neural language models. arXiv preprint arXiv:2006.00031.

نحو استخراج المرادفات العربية آليا

اعداد: ايمان عبد الكريم موسى نصر

اشراف : الأستاذ الدكتور نبيل عرمان

الملخص

حظي استخلاص المرادفات اهتماماً خاصاً نظراً لأهمية وضرورة المرادفات في تطوير أداء تطبيقات معالجة اللغة الطبيعية. وطُوِّرت مهمة الاستبدال المعجمي لاستخراج المرادفات والتي تهدف إلى انشاء قائمة من المرادفات لكلمة أو عبارة مستهدفة مع الحفاظ على المعنى الأصلي للجملة؛ وذلك لتحسين الكتابة و زيادة فهم اللغة و تعزيز آداء نماذج معالجة اللغة الطبيعية و التعامل مع الغموض اللغوي. كما تلقت هذه المهمة اهتماماً واسعاً في عدة لغات. وبالرغم من ثراء مفردات اللغة العربية إلا أن الأبحاث في هذه المهمة كانت محدودة نظراً لعدم توفر قاعدة بيانات موسمة. وبذلك نقدم لكم أول قاعدة بيانات موسمة للاستبدال اللغوي في اللغة العربية AraLexSubD. وأُعدت AraLexSubD يدوياً من قبل ثمانية من اللغويين والناطقين الأصليين باللغة العربية (ستة موسمين لغويين، ودكتور، واقتصادي) الذين قاموا بتوسيم 630 جملة. كما شملت AraLexSubD ثلاثة مجالات: المجال العام و المالي والطبي. وتضمنت 2476 كلمة بديلة محتملة مصنفة بناءً على ارتباطها الدلالي.

كما نوفر أيضًا نهج للاستبدال المعجمي باللغة العربية، AraLexSubPro، الذي يتضمن عدة تقنيات لتوليد البدائل واختيارها وترتيبها. ولإجراء مقارنة شاملة، يعتمد AraLexSubPro على أربع طرق مختلفة كنقاط مرجعية لتوليد مرشحي البدائل للكلمات المستهدفة: نهج يعتمد على قاموس المرادفات(AWN) ، ونهج يعتمد على نموذج لغة مدرب مسبقًا(AraBERT) ، إخفاء جزئي AraBERT، ونهج هجين يجمع بين AraBERT وAWN . يتم تصفية البدائل المولدة وترتيبها بناءً على ستة معايير عالية الجودة، بما في ذلك تشابه الكلمات، وتكرارها، (BERT Loss) ,(BERT probability) , (BERTscore) و(BERT similarity) وبعد ذلك، تتم إعادة ترتيب البدائل استنادًا إلى مصنف AraLexSubPro . بالإضافة إلى ذلك، نقدم تحليلًا للأخطاء التي ظهرت خلال التجربة.

ولتقييم آداء منهج AraLexSubPro استخدمنا أول مجموعة بيانات معيارية للاستبدال اللغوي باللغة العربية AraLexSubD، الذي يمكنه تقييم أنظمة الاستبدال اللغوي في اللغة العربية تلقائياً. وحسب معرفتنا هذه أول دراسة حول الاستبدال اللغوي في اللغة العربية. كما كانت النتائج مشجعة وأساسية لأبحاث الاستبدال اللغوي في اللغة العربية. و تتوفر AraLexSubD في هذا الرابط لتسريع البحث في هذا الموضوع.

لتقييم أداء نهج AraLexSubPro، نستخدم أول مجموعة بيانات معيارية للاستبدال المعجمي باللغة العربيةAraLexSubD ، التي يمكنها تقييم أنظمة الاستبدال المعجمي العربي تلقائيًا. حسب معرفتنا،

هذه هي الدراسة الأولى حول الاستبدال المعجمي في اللغة العربية، حيث ان النتائج مشجعة وأساسية للابحاث في هذا المجال. لتسريع الأبحاث، وضعنا البيانات AraLexSubDعلى منصة GitHub على الرابط التالي:

https://github.com/karajah2024/Arabic-Lexical-Substitution.git