

Article

Arabic Lexical Substitution: AraLexSubD Dataset and AraLexSub Pipeline

Eman Naser-Karajah ^{1,*} and Nabil Arman ² ¹ Department of Computer Science, Al-Quds University, Abu Dis P144, Palestine² Department of Computer Science, Palestine Polytechnic University, Hebron P726, Palestine; narman@ppu.edu

* Correspondence: eman.naser@students.alquds.edu

Abstract: Lexical substitution aims to generate a list of equivalent substitutions (i.e., synonyms) to a sentence's target word or phrase while preserving the sentence's meaning to improve writing, enhance language understanding, improve natural language processing models, and handle ambiguity. This task has recently attracted much attention in many languages. Despite the richness of Arabic vocabulary, limited research has been performed on the lexical substitution task due to the lack of annotated data. To bridge this gap, we present the first Arabic lexical substitution benchmark dataset *AraLexSubD* for benchmarking lexical substitution 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 the first Arabic lexical substitution pipeline, *AraLexSub*, which uses the AraBERT pre-trained language model. The pipeline consists of several modules: substitute generation, substitute filtering, and candidate ranking. The filtering step shows its effectiveness by achieving an increase of 1.6 in the F1 score on the entire AraLexSubD dataset. Additionally, an error analysis of the experiment is reported. To our knowledge, this is the first study on Arabic lexical substitution.

Keywords: lexical substitution (LS); evaluation dataset; pre-trained language model BERT



Citation: Naser-Karajah, E.; Arman, N. Arabic Lexical Substitution: AraLexSubD Dataset and AraLexSub Pipeline. *Data* **2024**, *9*, 98. <https://doi.org/10.3390/data9080098>

Academic Editor: Jamal Jokar Arsanjani

Received: 21 March 2024

Revised: 13 June 2024

Accepted: 19 June 2024

Published: 30 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Lexical substitution (LS) is an essential task in natural language processing (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 [1,2]. Lexical substitution has two variants: substitute prediction and candidate ranking [3]. It is widely used in many NLP tasks like data augmentation, paraphrase generation, semantic text similarity, paraphrase generation, and word sense induction [4]. The LS task has been applied widely to English using different methods [5]. Little research has been conducted on Arabic and no evaluation dataset has been created on Arabic lexical substitution.

One of the main challenges in lexical substitution is that previous algorithms find the substitutions for the target words from lexical resources (like WordNet) and then rank them based on their contexts. Such algorithms have two typical challenges: (1) they should not ignore good substitute candidates, which can be suitable synonyms for the target word as 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 [6]. To address such challenges, we propose utilizing the BERT contextualized embedding model that can be used to extract synonyms in dynamic contexts [7].

This paper presents two-fold contributions:

- (1) **AraLexSubD, which is a benchmark dataset for evaluating the Arabic lexical substitution 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, ranked according to their semantic relatedness.
- (2) **AraLexSub, which is an Arabic lexical substitution pipeline.** The pipeline uses the pre-trained language model AraBERT to generate substitutions and rank them based on three features: the BERT prediction order (BERT probability), the BERT-based language model (loss), and BERT similarity. To our knowledge, this is the first attempt to build an Arabic LS pipeline. Our experimental results demonstrate encouraging results for Arabic LS.

This paper is organized as follows: Section 2 presents the related work, Section 3 describes the construction of the AraLexSubD dataset, and Section 4 presents the AraLexSub pipeline. In Section 5, we describe the evaluation of the AraLexSub pipeline. Section 6 presents the qualitative analysis of the AraLexSubD dataset construction and AraLexSub pipeline. Finally, in Section 7, the conclusion is presented.

2. Related Work

Lexical substitution 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, which is the first lexical substitution shared task, called SemeEaL-2007 task 10 [8]. 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 [9]. 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 [10] 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** [11] was constructed to mitigate this restriction by choosing 2474 sentences from the Manually Annotated Sub-Corpus (MASC) [12,13]; 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** [14], which builds on **CoInCo**, is a higher-coverage and higher-quality benchmark that treats lexical substitution 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 68,683 substitutions. **ALaSca** is another English large-scale lexical substitution dataset [15] that selects a set of target words to ALaSca, which 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 [16] that selects contexts, targets words from the TOEFL-11 dataset [17], and generates substitution using GPT-4. Following the annotation in [14], 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, which is a manual German lexical substitution dataset from GermEval 2015 [18] containing 2040 sentences from the German Wikipedia containing 153 unique target words. **EVALITA** is a manual Italian lexical substitution dataset from EVALITA 2009 [19] comprising 2310 sentences and 231 unique target words. **CHNLS** is a new automatic Chinese lexical substitution dataset [20] with 33,695 instances and 144,708 substitutes from three sources: News, Novel, and Wikipedia. The substitutions are generated using four unsupervised LS methods, and then human annotators judge the appropriateness of these substitutions or add new ones.

Regretfully, there is still a dearth of research on Arabic LS. As far as we know, there is currently no publicly available Arabic LS dataset to evaluate the ability of Arabic LS models.

3. AraLexSubD Dataset Construction

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. 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 [21]. The seven secondary columns are three for target words (root, lemma) and four for candidates (root, lemma, transformation), where the lemma and the root are found from Qabas [22]. 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.

3.1. 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; 80 target words were polysemous, and each of the 80 target words has 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 linguist annotators selected the target words of the general domain. In contrast, two linguist annotators, a human doctor and an economist, collaborated to ensure the accuracy of identifying and selecting common words (target words). These words could be utilized in both domains but within their respective contexts and maintained the specificity required for each domain.

The three linguist annotators select the general domain sentences from Arabic ontology [23] that cover a wide range of subjects, including poetry, the Quran, and essay sentences. Furthermore, they also write or select the sentences of the medical and finance domains from essays.

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 [24], Al-Muaser [25], and Al-Maani [26]. 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 expert annotators are used separately in this step to manually rank the substitutes for each sentence in the three domains based on the scoring guidelines and standards in [21]. These standards are used as annotation methodology to maintain consistency among linguists annotator scores. **The guidelines ranked 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 [27]. 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" [2]. 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, $T1 = C1 \ T2$. In other words, given two terms, T1 and T2, lexicalizing concepts C1 and C2, respectively, then T1 and T2 are considered synonyms iff $C1 = C2$ " [28].
- **Style:** How much of the use of the substitute is correct and robust in the sentence? For example, consider two substitutes (اسف, asfu, Sorry/اعتذر, aetadhar, apologize). Both substitutes have the same meaning, and both are frequently used, but (اسف, asfu, Sorry) present feelings, which makes the substitute (اسف, asfu, Sorry) more stylish.
- **Use (frequently used):** How often is a word used in this context? For example, consider two substitutes (جوال, jawaal, Mobile/خليوي, khilaywy, cellular). Which one is more useful? Both substitutes have the same meaning, but the خليوي(khilaywy, cellular) is rarely used in this context.

The scoring standards in [21], as shown in Figure 1, 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.

Categories	Score	Meaning
Same semantics	100	Same semantics, style, use
	90	Same semantics, style, less used
	80	Same semantics, style, rarely used
	70	Same semantics, style, not used
Close semantics	60	Close semantics, weak style, uncommon
	50	Close semantics, not exact purpose
Related/Different semantics	40	Semantically related
	30	Semantically related (somehow)
	20	Semantically related (somehow)
	10	Semantically very different

Figure 1. The fuzzy scoring scale–synonymy strength.

5. Merging all annotations’ ranking

The scoring annotations from the three annotators were merged by averaging them to achieve a more accurate, reliable, and balanced estimate. The average ranks of these substitutions are rearranged in ascending 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 nonsynonymous (see Figure 1). The final dataset instances in the AraLexSubD dataset are as presented in Table 1:

Table 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 2 presents the number of all PoS tags in each domain in the AraLexSubD dataset. The target words (nouns, verbs, and adjectives) are 317, 256, and 57, respectively. The finance and medical domains did not contain verbs, as we cannot find verbs in such specialized domains.

Table 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

3.2. The Ranking Experimental Setup Steps

Before describing the work, we conducted a training session and ranking tests. The purpose of the training session was to go over the experiment and emphasize the concept of synonymy. The three linguist annotators 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 the logic of thinking 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 annotator is given the sentences and suggested candidates for the three domains in a separate file in Google Sheets. The AraLexSubD dataset was completed after 3 months. The working hours were distributed as follows:

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

4. Lexical Substitution AraLexSub Pipeline

This section presents the steps followed in the AraLexSub pipeline, including substitute generation, filtering, and candidate ranking.

4.1. Substitute Generation

Given a sentence S and a target word w , the substitute generation generates a list of substitutions that can replace the word w . Recent LS approaches used the pre-trained language model BERT to generate the substitutes [6] as it proves its effectiveness in many NLP tasks. Our pipeline AraLexSub generates the substitutions using the AraBERT model, an Arabic bi-directional language model [29], which 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 [7]. 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 [30].

Our AraLexSub pipeline, which considers feeding AraBERT with sentence pair $(S, \hat{S})$ to generate the candidates, where S is the original sentence with the target word w and $\hat{S}$ is the same sentence after replacing the target word w in the sentence S with a [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

La yutheheru tahyozan le'ahaden be ayn ehe

[Do not show bias towards anyone in particular]

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

لا + يظهر + تحيزاً + لأحد + بعينه

The sentence pair ($S, \hat{S}$) is fed into AraBERT, as shown in Figure 2. AraBERT generates a substitute list for the [MASK] word (ayn, عين, particular).

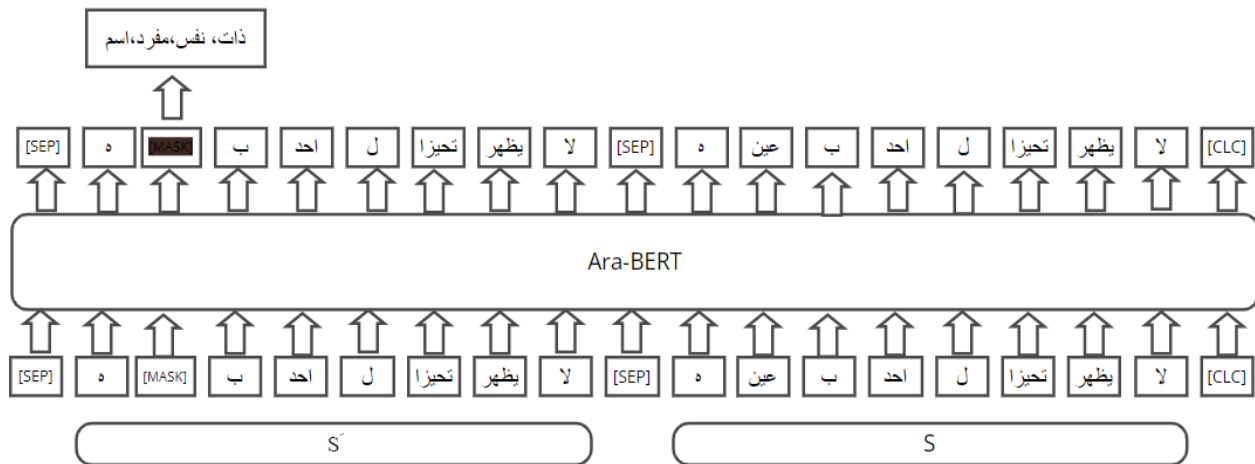


Figure 2. The substitution generation of AraLexSub for the target word prediction. The sentence is [لا يظهر تحيزاً لأحد بعينه/ Do not show bias towards anyone in particular] with the target word [عين/particular]. [MASK], [CLS], and [SEP] are Bert special symbols, where [MASK] is used to mask the word, [CLS] is added before each input instance, and [SEP] is a unique separator token.

AraBERT uses a Farasa tokenizer to tokenize the sentence into tokens before converting them into their appropriate embedding vector. During this step, we notice three main aspects, which are:

1. The AraBERT model is trained on a specific corpus, so the suggested substitutes will be limited to the training corpus, which means the unknown target word will not generate any substitutes.
2. The AraBERT tokenizer algorithm depends on the Word Piece algorithm. It splits the unknown tokens into subtokens by adding ## to the subtoken. For example, the word (يظهر, yazhar, 'show') will split the word into two subwords, [اي] and [ظهر##], which means that the subword [ظهر##] is a suffix followed by another subword. It is seen as an unknown token if there is no way to divide it into subtokens.
3. During the tokenization process, AraBERT showed another problem: the words with three letters and beginning with the letter waw "و". In our AraLexSubD dataset, four words appeared, as presented in Table 3. The segmenter segments them into three subwords: a conjunction, a preposition, and a pronoun. For example, for the word [ولي, Walla (v), go], the segmenter did not deal with the word [ولي, Walla(v), go] as a word; it is segmented into three different subwords [waw , و] as a conjunction, [le, ل] as a preposition, and [ee, ي] as a pronoun.

Table 4 presents the special case list of rules introduced for the PoS tags of the four words. The Farasa and Camel taggers utilize their PoS tag if the target word is not included in the particular list.

Table 3. PoS tags for the four words in the particular case list in the AraLexSubD dataset.

Word	PoS Tag
وَكَّرَ ، Waker ، den	Noun
وَلَّى ، Walla ، go	Verb
وَقَّرَ ، Waqra (v) ، revere/ وَقْرَ ، weqr (n) ، heavy	Verb/noun
وَلَّهَ ، Walh (n) ، infatuation/ وَلَّاهَ ، Walha(v) ، love	Noun/verb

Table 4. PoS tag rules for the four words in the particular case list.

Rule	PoS Tag
If the letter (ف،fa) is a prefix for the target word, then the target word is:	Verb
If the letter (ال،all) is a prefix for the target word, then the target word is:	Noun
If the PoS for the previous word is a verb, then the target word is:	Noun
If the pronoun (ي،ya) is connected to the target word, then it is:	Noun
If the pronoun (ني،nee/وا،oww) is connected to the target word, then it is:	Verb
Otherwise	Noun

4.2. Substitute Filtering

This section presents the filtering step for the generated substitutions. This step is essential to clean up the generated substitutions from unsuitable ones. The filtering step includes PoS filtering and postprocessing filtering, which are described below:

1. The PoS filter eliminates the generated substitutions that did not match the PoS of the target word. The PoS tag target word and the PoS tag substitutions must be the same: verb-verb, noun-noun, and adjective-adjective. In other words, if the target word has a verb PoS tag, then all of the suggested substitutions must have the verb PoS tag, and the same is true for noun and adjective target words.
2. The postprocessing filter cleans up the generated substitute list from the target word, its morphological derivative forms, the substitute morphological derivation, subwords with ##, subwords containing the + sign, and repeated words.

4.3. Candidate Ranking

After generating and filtering the substitutions, the filtered substitute candidates, $C = \{c_1, c_2, \dots, c_n\}$, where $c_i, i = 1 \dots n$, is the candidate and n is the number of substitute candidates, are ranked by three ranking features that capture an aspect of fitting the substitute to replace the target word. The three features used for ranking are described below:

1. The language modeling feature is a ranking feature (BERT loss) used to evaluate the fluency of substitutes for each sentence in our dataset. The AraBERT is used to calculate the sentence probability. AraBERT is an MLM, so we cannot estimate the sentence probability directly by AraBERT. Let w be the target word and $W = w_{-n} \dots w_{-1}, w, w_1 \dots w_n$, be the target word w context. The sentence probability is calculated by replacing the target word w with each substitution candidate set. 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.

2. The BERT prediction order is a ranking feature (BERT probability) that predicts the probability distribution of the words corresponding to the masked word. As our generation method is AraBERT, it depends on context when generating substitutions. 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.
3. BERT similarity 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 is.

4.4. Lexical Substitution Algorithm

The lexical substitution algorithm (AraLexSub) is presented in Algorithm 1. Each sentence S has one highlighted target word w of the type (noun, verb, adjectives). The sub-stitutions of the target word are generated using AraBERT and fed with the $(S, \hat{S})$ pair. Next, the generated substitutions are filtered using the PoS and postprocessing filters. Then, the filtered substitutions are ranked by the three ranking features: BERT loss, BERT probability, and BERT similarity features.

Algorithm 1: AraLexSub Algorithm

```

Sentence  $S$ 
target word  $w$ 
for each  $w \in S$ , do
    subs  $\leftarrow$  Substitution Generation ( $S, w$ ) by AraBERT by concatenating ( $S, \hat{S}$ )
    subs  $\leftarrow$  Substitution Filtering ( $S, w$ )
    for each filter, do
        PoS filter for generated subs
        return PoS-filtered subs
        postprocessing filter for the PoS-filtered subs
        return filtered subs
    end for
    rank_subs  $\leftarrow$  Substitute Ranking (subs)
    for each ranking feature  $f$ , do
        calculate feature scores
        scores  $\leftarrow$  scores ( $f$ )
    end for
    rank  $\leftarrow$  rank(scores)
    return rank_subs
end for

```

5. Evaluation

This section presents the quantitative evaluation procedure, metrics, and performance evaluation results of the AraLexSub pipeline.

5.1. Evaluation of Substitution Generation

The AraLexSub generates a set of substitutions by feeding AraBERT with $(S, \hat{S})$. There are no limits on the number of generated substitutes. The evaluation procedure contains automatic and manual evaluations over the three domains. Four metrics are used in the automatic evaluation to evaluate the performance of the substitution generation:

- Potential: The proportion of instances for which at least one of the generated substitutes is in the annotated AraLexSubD.
- Precision: The proportion of generated substitutes that are in the annotated AraLexSubD.
- Recall: The proportion of the annotated substitutes included in the generated substitutes.

- F1: The harmonic mean between Precision and Recall.

The generated substitutes by the AraBERT model have two advantages: (1) the AraBERT model is the only input needed, and no corpus is needed, and (2) the generated substitutions do not require morphological transformation.

The automatic evaluation compares the generated substitution with the annotated substitute in the AraLexSubD dataset, so it cannot provide a complete view of the generation method's feasibility. Automatic evaluation will exclude these correct substitutes that are not in the AraLexSubD dataset. However, manual evaluation includes all correct substitutions regardless of the annotated AraLexSubD.

The manual evaluation is calculated by counting the proportion of the correct substitutes in the generated substitutes, excluding the target word and its morphological forms, and ignoring the grammatical correctness (only meaning matters). The automatic and manual evaluation is computed across the three domains separately and the entire AraLexSubD, as presented in Table 5.

Table 5. Automatic and manual evaluation results for substitute generation over the three domains and the entire AraLexSubD.

Domain	Potential	Precision	Recall	F1	Manual Evaluation
General	57.24	26.07	24.59	25.31	51.01
Medical	73.75	32.25	38.27	35.01	56.05
Finance	60.00	21.08	26.89	23.63	54.15
Entire dataset	59.68	26.23	26.62	26.42	52.47

5.2. Evaluation of Substitution Filtering

The PoS filter and postprocessing filter clean up the generated substitutions from the original target word, its morphological form, substitutions' morphological derivation, and substitutions with different PoS tags. The automatic evaluation is applied using the evaluation metrics used in the substitute generation: Potential, Precision, Recall, and F1. The evaluation is computed across the three domains separately and the entire AraLexSubD, as presented in Table 6.

Table 6. Evaluation results for substitute filtering using PoS and postprocessing filters over the three domains and the entire AraLexSubD.

Domain	Filter Name	Potential	Precision	Recall	F1
General	Without filter	57.24	26.07	24.59	25.31
	PoS filter	51.49	28.79	21.35	24.52
	Postprocessing filter	56.17	32.24	24.01	27.52
	(PoS + Postprocessing) filters	50.64	34.05	21.01	25.99
Medical	Without filter	73.75	32.25	38.27	35.01
	PoS filter	71.25	39.25	35.79	37.44
	Postprocessing filter	73.75	41.75	38.03	39.80
	(PoS + Postprocessing) filters	71.25	47.08	35.80	40.67
Finance	Without filter	60.00	21.08	26.89	23.63
	PoS filter	57.50	23.15	24.48	23.80
	Postprocessing filter	60.00	26.75	26.88	26.81
	(PoS + Postprocessing) filters	57.50	29.17	24.48	26.62
Entire dataset	Without filter	59.68	26.23	26.62	26.42
	PoS filter	54.76	29.41	23.58	26.17
	Postprocessing filter	58.89	32.75	26.15	29.08
	(PoS + Postprocessing) filters	54.13	35.08	23.32	28.02

As we see in Table 4, despite the importance of PoS filtering, it decreases the potential and recall values due to ambiguity and incorrect PoS tagging from Camel and Farasah taggers, which removes correct substitutions. The postprocessing filter improves the precision and F1 values results for the three domains and the entire AraLexSubD dataset. 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 results in Table 6 show the importance of using both filters to clean up the generated synset, increasing Precision and F1 in the three domains and the entire AraLexSubD dataset.

5.3. Evaluation of Substitution Ranking

In the substitution ranking stage, the annotated candidates in the AraLexSubD dataset are provided, and the goal is to rerank them by the three ranking features: the BERT prediction order (BERT probability), BERT similarity, and language model (BERT loss). A ranking score is computed using each rank feature, which reranks the annotated substitutions according to the feature appropriateness. Two annotators evaluate the feature rank manually by comparing the annotated candidates rank in the AraLexSubD benchmark with each feature rank.

The results are presented in Table 7. The BERT prediction order (BERT probability) feature shows the best ranking score in the three domains compared to the annotated candidates rank in the AraLexSubD benchmark.

Table 7. Manual evaluation results for the ranking features for the three domains.

Domain	Rank Feature	Manual Evaluation
General	BERT loss	42.02
	BERT similarity	42.85
	BERT probability	51.89
Medical	BERT loss	45.31
	BERT similarity	43.85
	BERT probability	49.79
Finance	BERT loss	43.70
	BERT similarity	48.17
	BERT probability	50.19

BERT probability correctly ranks 51.89% of the annotated candidates in the general domain, 49.79% in the medical domain, and 50.19% in the AraLexSubD dataset.

6. Qualitative Analysis

This section discusses the qualitative analysis of the construction of the AraLexSubD dataset and the AraLexSub pipeline. For the AraLexSub pipeline, this work analyzes the errors of substitute generation and substitute filtering and defers the ranking errors to future research.

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, ayn al aetibar, 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, eyin] and lemma for the word [الاعتبار, al aetibar], 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 AraLexSub Pipeline

In this subsection, we analyze the AraLexSub pipeline to understand the source of the error in the substitute generation and substitute filtering steps. Some errors occur due to the following:

1. Difficulty: Some target words in the AraLexSubD dataset are challenging to understand. AraBERT cannot generate suitable candidates or has no suggestions, as it is trained on a specific corpus.
2. Ambiguity: The PoS tagger may wrongly tag some of the suggested substitutions due to the ambiguity in understanding the sentence or substitute.
3. Semantics: Misunderstanding the sentence's semantics can lead to incorrectly suggested substitutions.
4. Model architecture: The architecture of AraBERT removes the last character of the generated substitutions, such as the character ّ, or generates the antonyms of the target word.
5. Missing entries: Some suggested substitutes are correct but are not in the AraLexSubD dataset.

In our AraLexSub 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 are made during substitute generation, and errors 7 and 8 occur during substitute filtering. Many sentences are recorded for each error type. Table 8 presents the sentence count in AraLexSubD for each error. Table 8 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 of types 3 and 6.

Table 8. 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 المركزية هي**, Central muscles are the muscles that control the limbs of the body] with target word (المركزية/ Central), AraBERT generates no substitution except [وما، 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] **سيولة الدم من المؤشرات**, Blood fluidity is an essential vital indicator for diagnosing medical conditions] with target word (سيولة/ fluidity), AraBERT generates the substitutions (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] **يتم إزالة هامش من النسيج في**, 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 (قسم، جزء/ 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] **لكل شخص مستوى**, Every person has a certain standard of life] with target (حياة/ life), AraBERT generates ((إشارة، 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] **للمحل السياسي دراية بفنون**, The political analyst is familiar with the art of speaking] with target (دراية/ familiar), AraBERT generates (جهل، وعي، علم، إلمام/ familiarity, knowledge, awareness, ignorance) substitutions. The substitutions (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] **تحديد حجم الورم في حالة السرطان**, 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] قسمًا واحدًا 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.

7. Conclusions

This paper presents the manual construction of the first evaluation benchmark, the AraLexSubD dataset, for the Arabic lexical substitution task. AraLexSubD was built by eight native Arabic speakers and linguists (six linguist annotators, a doctor, and an economist). AraLexSubD contains 630 sentences with one target word and 2476 substitution candidates, semantically ranked, using the guidelines in [21]. The benchmark is split into three domains: general domains (470 sentences), finance domains (80 sentences), and medical domains (80 sentences). We propose an Arabic lexical substitution pipeline, AraLexSub, which contains three stages, generation, filtering, and ranking, using the AraLexSubD dataset. We believe the proposed AraLexSubD and AraLexSub 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. In the future, we will use different generation methods and ranking features.

Author Contributions: Conceptualization, N.A.; methodology, N.A. and E.N.-K.; software, E.N.-K.; validation, E.N.-K.; formal analysis, E.N.-K.; investigation, N.A. and E.N.-K.; resources, E.N.-K.; data curation, N.A. and E.N.-K.; writing—original draft preparation, E.N.-K.; writing—review and editing, N.A. and E.N.-K.; visualization, E.N.-K.; supervision, N.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The AraLexSubD dataset, constructed and presented in the article, is available on GitHub at <https://github.com/karajah2024/Arabic-Lexical-Substitution.git> (accessed on 1 March 2024).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. 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.
2. Seneviratne, S.; Daskalaki, E.; Lenskiy, A.; Suominen, H. CILex: An investigation of context information for lexical substitution methods. In Proceedings of the 29th International Conference on Computational Linguistics, Gyeongju, Republic of Korea, 12–17 October 2022; pp. 4124–4135.

3. 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.
4. Michalopoulos, G.; McKillop, I.; Wong, A.; Chen, H. 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, Dublin, Ireland, 22–27 May 2022; Volume 1, Long Papers. pp. 1226–1236.
5. Amrami, A.; Goldberg, Y. Word Sense Induction with Neural biLM and Symmetric Patterns. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; Association for Computational Linguistics: Stroudsburg, PA, USA, 2008.
6. Zhou, W.; Ge, T.; Xu, K.; Wei, F.; Zhou, M. BERT-based lexical substitution. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 3368–3373.
7. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. 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, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, (Long and Short Papers). pp. 4171–4186.
8. 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.
9. Sharoff, S. Open-source corpora: Using the net to fish for linguistic data. *Int. J. Corpus Linguist.* **2006**, *11*, 435–462. [CrossRef]
10. Biemann, C. Turk Bootstrap Word Sense Inventory (TWSI) 2.0. 2010. Available online: [https://aclweb.org/aclwiki/TWSI_Turk_bootstrap_Word_Sense_Inventory_2.0_\(Repository\)/](https://aclweb.org/aclwiki/TWSI_Turk_bootstrap_Word_Sense_Inventory_2.0_(Repository)/) (accessed on 1 March 2024).
11. 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.
12. 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.
13. Ide, N.; Baker, C.F.; 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.
14. Lee, M.; Donahue, C.; Jia, R.; Iyabor, A.; Liang, P. 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, Online, 6–11 June 2021; pp. 4362–4379.
15. 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.
16. Zhang, X.; Chen, Z.; Yu, Z. ProLex: A Benchmark for Language Proficiency-oriented Lexical Substitution. *arXiv* **2024**, arXiv:2401.11356.
17. Blanchard, D.; Tetreault, J.; Higgins, D.; Cahill, A.; Chodorow, M. TOEFL11: A corpus of non-native English. *ETS Res. Rep. Ser.* **2013**, *2013*, i-15. [CrossRef]
18. Hintz, G.; Biemann, C. Delexicalized supervised German lexical substitution. In Proceedings of the GermEval 2015, Essen, Germany, 29 September 2015; pp. 11–16.
19. Toral, A. The lexical substitution task at EVALITA 2009. In Proceedings of the EVALITA Workshop, 11th Congress of Italian Association for Artificial Intelligence, Reggio Emilia, Italy, 29 November 2010.
20. Qiang, J.; Liu, K.; Li, Y.; Li, Y.; Zhu, Y.; Yuan, Y.H.; Ouyang, X. Chinese Lexical Substitution: Dataset and Method. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Singapore, 6–10 December 2023; pp. 29–42.
21. Ghanem, S.; Jarrar, M.; Jarrar, R.; Bounhas, I. A Benchmark and Scoring Algorithm for Enriching Arabic Synonyms. In Proceedings of the 12th Global Wordnet Conference, San Sebastian, Spain, 23–27 January 2023; pp. 274–283.
22. Jarrar, M.; Hammouda, T.H. 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), Turin, Italy, 20–25 May 2024; pp. 13363–13370.
23. Available online: <https://ontology.birzeit.edu/> (accessed on 1 March 2024).
24. Available online: <https://shamela.ws/book/7028> (accessed on 1 March 2024).
25. Available online: <https://www.dohadictionary.org/bibliography> (accessed on 1 March 2024).
26. Available online: <https://www.almaany.com/> (accessed on 1 March 2024).
27. Miller, G.A.; Beckwith, R.; Fellbaum, C.; Gross, D.; Miller, K.J. Introduction to WordNet: An online lexical database. *Int. J. Lexicogr.* **1990**, *3*, 235–244. [CrossRef]
28. Jarrar, M. The Arabic ontology—An Arabic wordnet with ontologically clean content. *Appl. Ontol.* **2021**, *16*, 1–26. [CrossRef]

29. Antoun, W.; Baly, F.; Hajj, H. 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, Marseille, France, 11–16 May 2020; pp. 9–15.
30. Wu, X.; Lv, S.; Zang, L.; Han, J.; Hu, S. Conditional Bert contextual augmentation. In *Computational Science—ICCS 2019: 19th International Conference, Faro, Portugal, 12–14 June 2019, Proceedings, Part IV 19*; Springer International Publishing: Cham, Switzerland, 2019; pp. 84–95.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.