College of Computer and Information Sciences

Research Project Report

Master of Science in Computing (Data Science)

A Deep Learning Approach to Arabic Reverse Dictionary

Student Name	Mais Alharaki
Student ID	444010562
Submission Date	2 May 2024

Second Semester 2023–2024

Contents

Declaration

I hereby certify that:

- This material, which I now submit for assessment on the programme of study leading to the award of MSc Computing (Data Science) is entirely my own work, and that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

- Due acknowledgement has been given in the bibliography and references to ALL sources be they printed, electronic or personal.

- Unless this dissertation has been confirmed as confidential, I agree to an entire electronic copy or sections of the dissertation to be available to allow future students the opportunity to see examples of past dissertations.

- I agree to my dissertation being submitted to a plagiarism detection service, where it will be stored in a database and compared against work submitted from this or any other Department or from other institutions using the service. In the event of the service detecting a high degree of similarity between content within the service this will be reported back to my supervisor and program leader, who may decide to undertake further investigation that may ultimately lead to disciplinary actions, should instances of plagiarism be detected.

- I have read the PNU Policy Statement on Ethics in Research and I confirm that ethical issues have been considered, evaluated and appropriately addressed in this research.

Signed:

Candidate Name: Mais Alheraki

ID Number: 444010562

Date: 2 May 2024

List of Figures

List of Tables

List of abbreviations

RD: Reverse Dictionary

CAD: Contemporary Arabic Dictionary

SOTA: State of the Art

NLP: Natural Language Processing

NLG: Natural Language Generation

NLU: Natural Language Understanding

BERT: Bidirectional Encoder Representations from Transformers

Seq2Seq: Sequence to Sequence

LR: Learning Rate

Abstract

The domain of reverse dictionaries, while advancing in languages like English and Chinese, remains underdeveloped for less Arabic. This study attempts to explore a data-driven approach to enhance word retrieval processes in Arabic RDs. The research focuses on the ArabicNLP 2024 Shared Task, named KSAA-CAD, which provides a dictionary dataset of 39,214 word-gloss pairs, each with a corresponding target word embedding. The proposed solution aims to surpass the baseline performance by employing state-of-the-art deep learning models and innovative data augmentation techniques. The methodology involves enriching the dataset with contextually relevant examples, training a T5 model to align the words to their glosses in the space, and evaluating the results using Mean Squared Error and Cosine similarity. We find that our model is closely aligned with the baseline performance on bertseg and bertmsa targets, however does not perform well on electra target, suggesting the need for further exploration of data augmentation techniques and model architectures to enhance Arabic RDs. The findings of this study contribute to the ongoing research in Arabic NLP and reverse dictionaries, providing insights into the challenges and opportunities in this domain.

1 Introduction

While reverse dictionaries have witnessed advancements in languages like English and Chinese (e.g. WantWords [17]), they remain less developed and explored for Arabic. This gap is particularly concerning for a language with a rich linguistic heritage and widespread use. The only ongoing effort in this domain is the 1st Arabic RD shared task launched in 2023 by King Salman Global Academy for Arabic Language (KSAA)[1].

1.1 Content of study

This research aims to explore new methods for Arabic RD, by proposing a data-driven approach that leverages deep learning to enhance word retrieval processes. We focus on the KSAA-CAD Shared Task dataset, which provides a rich collection of approximately 39k word-gloss pairs extracted from various Arabic dictionaries, each associated with a corresponding target word embedding. The proposed solution explores the performance of state-of-the-art deep learning models combined with data augmentation techniques.

1.2 Problem statement & motivation

The Arabic language, with its intricate morphology and diverse dialects, presents unique challenges for Natural Language Processing (NLP) tasks. RDs are crucial tools for language learners, translators, and researchers, enabling them to identify words based on their meanings or descriptions.

However, the research and available tools within the Arabic RD domain are limited, with few studies exploring new methodologies or leveraging advanced deep learning models, which motivates the need for more research in this area.

The main research questions addressed in this study are:

1. What is the impact of enriching the definitions from dictionary data with examples and expanding the dataset on performance?

2. Are there any pre-trained architectures other than BERT that have similar good

[1] `https://arai.ksaa.gov.sa/sharedTask/`

performance on Arabic?

1.3 Aim and objectives

This project aims to enhance word retrieval processes in Arabic by leveraging data-driven techniques to establish a more intuitive and accurate connection between conceptual descriptions and corresponding dictionary words. This approach is specifically designed to improve the language learning experience by utilizing sophisticated data analytics to map natural language inputs to lexical outputs. Focusing on Arabic, a less explored language in this domain, presents a unique opportunity to contribute significantly to the field of data science by extending its applications to new linguistic territories.

To achieve this aim, the following objectives are set:

1. Explore the KSAA-CAD dataset and understand the task requirements.
2. Enrich the dataset with contextually relevant examples to enhance the model's understanding of word meanings.
3. Fine-tune a pre-trained T5 model on the original and enriched dataset.
4. Evaluate the model's performance using Mean Squared Error and Cosine similarity metrics.
5. Compare the results with the baseline performance and discuss the findings.
6. Submit a system design paper to the ArabicNLP 2024 conference based on the results.

1.4 Proposed solution

We leverage the new KSAA-CAD dataset, which is shared exculsively with the participants of the 2024 RD Shared Task (more in section 2.5), comprising 39,214 word-gloss pairs with corresponding target word embeddings, alongside a SOTA baseline results[2]. Our goal is to surpass the baseline performance by employing

[2] `https://github.com/ksaa-nlp/KSAA-CAD#baseline-results`

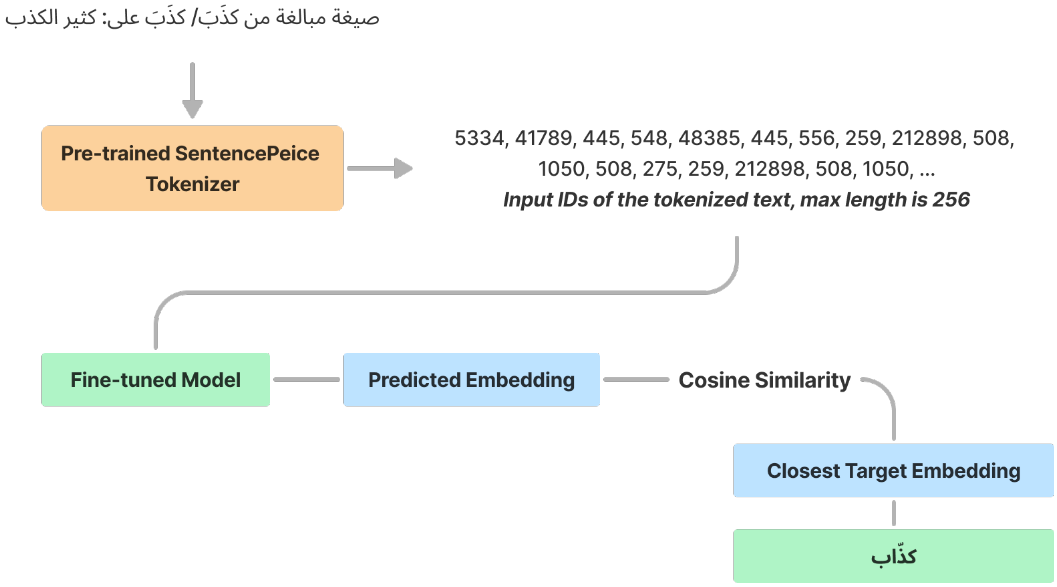


Figure 1.1: *The workflow of the arabic RD system*

cutting-edge deep learning models and innovative data augmentation techniques. Figure 1.1 illustrates the Arabic RD system workflow. Our methodology is twofold:

1. **Data Enrichment:** The dataset will be enriched with contextually relevant examples to enhance the model's understanding of word meanings. This will be achieved by leveraging a large Arabic text corpus to curate examples for each word-gloss pair.

2. **Pre-trained T5 Model Adaption:** A pre-trained T5 model [21] will be fine-tuned on both the original and enriched KSAA-CAD dataset. We explore T5 for it being more recent and less explored in the Arabic RD literature.

1.5 Structure of the report

The upcoming sections are structured as follows:

1. Section 2 provides background material on natural language processing, deep learning, reverse dictionaries, and more.

2. Section 3 reviews related work in the field of RDs, highlighting recent advancements and methodologies.

3. Section 4 outlines our methodology, detailing the dataset, data enrichment techniques, and model architecture.

4. Section 5 presents evaluation metrics, results of the experiments and discusses the findings.

2 Background material

This chapter lays the groundwork for understanding the subsequent sections of this report by exploring the fundamentals of natural language processing, deep learning, text embeddings, transformers, and more. These disciplines are instrumental to the overall research presented here.

2.1 Natural Language Processing

Language is the way we communicate and exchange information, it's composed of symbols, rules and repetitive expressions. Natural Language Processing or NLP is a field where AI and linguistics cross together. It's interested in enabling machines to understand and model language, therefore enabling more natural communication between humans and machines.

NLP encompasses two major subfields: Natural Language Understanding (NLU) and Natural Language Generation (NLG). NLU focuses on enabling computers to comprehend the meaning of words, phrases, and expressions within human language. In contrast, NLG concerns the process of generating meaningful phrases and paragraphs, essentially allowing machines to "write" human-like text [29].

2.2 Deep learning

Deep learning is a subfield of machine learning inspired by the structure and function of the human brain. It utilizes artificial neural networks (ANNs) with multiple hidden layers, enabling the extraction of complex patterns from large amounts of data. This capability has revolutionized various fields, including computer vision, natural language processing (NLP), and speech recognition.

Though the idea can be traced back many years, deep learning only recently has proved to outperform traditional ML algorithms in many areas, including NLP. In 2016, [7] introduced the Transformer architecture, which has since become the foundation for many state-of-the-art NLP models, such as BERT, GPT-3, and T5.

However, training deep learning models from scratch often requires significant computational resources and large amounts of labeled data. This can be a barrier for tasks with limited data availability. Here's where transfer learning comes in as a powerful technique.

2.2.1 Transfer learning

Transfer learning is an approach that leverages knowledge gained from a pre-trained deep learning model on a source task and applies it to a new, related target task. By transferring the learned weights and features from a pre-trained model, a target model can achieve good performance even with limited training data. This is particularly beneficial for NLP tasks, where obtaining large amounts of labeled data can be expensive and time-consuming.

A popular example of transfer learning models borrowed from the computer vision field is the ImageNet [5] dataset, where models trained on it can be used for other image classification tasks, naming VGG [3], ResNet [4], and Inception [3].

Models trained on large corupses of text are known as "language models", these models learn to assign probabilities to tokens (words or characters) given their occurance in the training corpus. Pre-trained language models can be used as a starting point, or as sometimes referred to as **checkpoints**, to a lot of downstream NLP tasks such as sentiment analysis, text classification, and topic modeling, instead of starting from scratch, which saves a vast amount of time and computational resources, given that a model have already learned the patterns from the original data.

Neural networks allowed for the development of more complex language models, where models can analyzed huge amounts of text data and learn the patterns and relationships between words, using various architectures such as RNNs, LSTMs, and Transformers.

2.2.2 Sequence to Sequence approach for language modeling

The sequence to sequence (Seq2Seq) approach is a powerful deep learning architecture specifically designed for tasks that involve processing and generating sequences. In NLP, this translates to tasks like machine translation, where an input sequence in one language is transformed into an output sequence in another language. Seq2Seq models typically consist of two deep neural networks:

- Encoder: This network processes the input sequence and encodes it into a fixed-length vector representation capturing the semantic meaning.
- Decoder: This network utilizes the encoded representation from the encoder and generates the output sequence one element at a time, conditioned on the previously generated elements.

2.2.3 Tokenization and text embeddings

Tokenization is the process of breaking down text data into smaller units suitable for processing by deep learning models. These units can be words, characters, or even sub-word units like morphemes, which stands for a logical unit of a language that cannot be further divided (e.g. use, able, forming *usable*). The choice of tokenization strategy can significantly impact the performance of deep learning models in NLP tasks.

Text embeddings represent the tokens as numerical vectors in a high-dimensional space, where words with similar meanings tend to be positioned closer together. Popular techniques for text embedding include Word2Vec and GloVe, which learn these representations by analyzing large text corpora, meaning a large set of texts (usually in electronic format) which is considered to be representative of a language.

In the case of deep learning models, text embeddings are often learned as part of the training process, and is often the result of the encoder part of a language model, where the input text is transformed into a fixed-length vector representation.

2.2.4 SentencePiece tokenizer

SentencePiece is an unsupervised text tokenizer, used by the T5 models for text data processing [9]. Unlike traditional word-based tokenization, SentencePiece utilizes subword units, which are smaller linguistic components like prefixes, suffixes, and morphemes. This approach offers several advantages. Firstly, it allows T5 to effectively handle OOV words by combining subword units to represent them. Secondly, SentencePiece is language-independent, enabling T5 to process text in diverse languages with a single tokenizer.

During tokenization, SentencePiece analyzes the training corpus to identify frequently occurring subword units and builds a vocabulary. When processing text, it segments the input into these subword units and assigns them unique token IDs, which the T5 model uses for its internal operations. This subword-level representation provides T5 with a more granular understanding of the text, enhancing its ability to perform various NLP tasks effectively.

2.2.5 Transformer architecture: a paradigm shift

The Transformer architecture, introduced by Vaswani et al [7] in 2017 in the famous paper “Attention is All You Need”, has become a dominant force in NLP due to its ability to efficiently capture long-range dependencies within sequences. Unlike RNNs, which process sequences sequentially, Transformers rely solely on attention mechanisms. These mechanisms allow each element in the input sequence to attend to (focus on) other elements, enabling the model to understand the context of each word in relation to the entire sequence. This parallel processing approach facilitates faster training compared to RNNs and is particularly effective for tasks requiring long-range dependency modeling, such as machine translation and text summarization.

2.2.6 Pre-trained transformer models: BERT and T5

While the core Transformer architecture provides a powerful foundation, further advancements have led to the development of specialized pre-trained models like BERT and T5. These models leverage the strengths of Transformers and are pre-trained on massive amounts of unlabeled text data, allowing them to learn general contextual representations of language.

- **BERT (Bidirectional Encoder Representations from Transformers)**: Introduced by Google AI [8] in 2018, BERT is a pre-trained Transformer model that excels at understanding the context of words in a sentence and their relationships. It can be fine-tuned for various NLP tasks like question answering and sentiment analysis. However, BERT requires fine-tuning for specific tasks, which can be computationally expensive.

- **T5 (Text-to-Text Transfer Transformer)**: Introduced also by Google AI in 2019 [12], T5 utilizes a text-to-text format for all NLP tasks. It employs a single encoder-decoder architecture and learns to transform the input sequence into the desired output sequence. This approach makes T5 versatile, allowing it to handle a wide range of tasks by simply changing the format of the input and desired output. T5 often requires less fine-tuning compared to BERT, making it quicker to deploy for various tasks. However, it might not achieve the same level of deep contextual understanding as BERT in tasks where this is crucial.

2.3 Reverse dictionaries

Dictionaries in their conventional form map words to a set of meanings or definitions, often combining them with some examples on how the words are used in context. Dictionaries are the foundation of various NLP tasks, serving as lexical resources, where they help in understanding information such as word meanings, parts of speech, and relationships between words. Tasks such as stemming and lemmatization also rely on dictionaries to return words to their base form. However, traditional dictionaries are unidirectional, providing word meanings based on the input word.

Reverse dictionaries are a form of dictionaries where a description yields a set of words from the dictionary that semantically matches the description. Traditionally used as tools for linguistic exploration, reverse dictionaries have evolved to play a significant role in data science. A prime use-case is their application in data exploration and analysis, where reverse dictionaries facilitate the identification of relevant features within complex textual datasets by generating key terms or phrases. This enhances the efficiency of data mining and fosters the discovery of new insights [23].

In machine learning, reverse dictionaries aid in feature engineering, enriching model inputs with nuanced context. This utility extends to automated metadata generation for effective data cataloging and management. For instance, they map specific terms (e.g., "machine learning") to broader categories (e.g., "computer science"). By analyzing the corpus for terms frequently appearing alongside these categories in the reverse dictionary, we can identify relevant metadata tags. This approach surpasses simple word frequency, utilizing relationships between terms for a richer description of the corpus content.

Additionally, they enhance content curation and recommendation systems, offering more precise content descriptors and improving recommendation relevance.

Furthermore, reverse dictionaries streamline text summarization and topic modeling [2], assisting in distilling essential information from large text volumes. They also play a crucial role in improving chatbot and customer service automation by accurately interpreting user queries and intents.

2.4 Embedding-based retrieval

Embedding-based retrieval is a technique that leverages text embeddings to retrieve relevant information from a dataset. By representing text as numerical vectors in a high-dimensional space, embedding-based retrieval models can efficiently search for semantically similar words or phrases. This approach is particularly useful for reverse dictionaries, where the goal is to look for the closest words to a query based on their descriptions or meanings (semantic relationship to the query) [16].

2.5 ArabicNLP 2024 Shared Task

The KSAA-CAD Shared Task[3] is a part of the ArabicNLP[4] conference for 2024, it aims towards developing an RD system capable of predicting words from their definitions in Arabic.

Participants are provided with a dataset containing 39k instances, and consisting of word-gloss pairs and corresponding word embeddings.

We particpated in this task, and the proposed solution is based on the dataset provided by the organizers. The aim is to outperform the baseline, and enhance the dataset with a new feature. Our solution will be evaluated through the shared task, and a corresponding system design paper (based on this report) will be submitted to the conference.

[3] Task has been launched in May 2024, more here: arai.ksaa.gov.sa/sharedTask

[4] arabicnlp2024.sigarab.org

3 Related work

Understanding the meaning behind words, even within the same language, presents a significant challenge for machines. Monolingual reverse dictionaries address this directly, aiming to identify a target word based on its definition in the same language. This task is particularly crucial for languages like Arabic, with its rich vocabulary and unique cultural nuances. However, research in this area remains less explored compared to other languages with rich resources.

3.1 Previous studies

One notable recent contribution is the work presented by ElBakry et al [27]. (2023) as part of the ArabicNLP 2023 Shared Task, where they demonstrate an approach to Arabic RD tasks, successfully handling both Arabic and English definition inputs. It utilizes an ensemble of fine-tuned BERT models, specifically CamelBERT-MSA and MARBERTv2, to predict word embeddings from provided definitions. By leveraging an ensemble strategy, the authors achieved improved results compared to single models, highlighting the benefits of this approach.

On the same task another attempt by Qaddoumi [30], a method is introduced to enhance Arabic word embeddings using a modified BERT Multilingual model with data augmentation, targeting improvements in Arabic RD tasks. By customizing BERT for Arabic and employing data augmentation strategies, the study achieves significant enhancements in semantic accuracy. However, it suggests further exploration into the effects of data augmentation and the need for expanded datasets.

Building on this, Sibaee et al. [31] presently employs a SemiDecoder architecture combined with an SBERT encoder. This methodology excels in encoding word definitions into vectors using SBERT, followed by training with the SemiDecoder model. The approach leverages SBERT's proficiency in capturing semantic similarity and the SemiDecoder's training efficiency, leading to a high ranking in the shared task.

Other languages received more research in the area of RDs. Mane et al. [24] proposed a unique approach to reverse dictionaries with mT5, aiming at Indian languages support, where mT5 was employed for its ability to understand and generate language across multiple languages. It contrasts with BERT's Masked Language Modeling, focusing instead on translating and understanding user inputs to produce accurate word predictions. The results showed that mT5 outperformed BERT-based models in the RD task for both Indian languages and an English baseline.

Ardoiz et al. [22], in the SemEval RD task, studied the significance of high-quality lexicographic data in the efficiency of reverse dictionaries models. They suggest that refining the dataset by incorporating high-quality lexicographic data could significantly impact the task's outcomes, emphasizing the need for a robust dataset for optimal model performance. Their methodology involved a sentence-transformer model named "distiluse-base-multilingual-cased-v2", which was trained to make the definition embeddings as similar as possible as the word gloss.

On the other hand, Tran et al. 2022 [26] in SemEval RD task, evaluates Transformer-based models enhanced with LST and BiLSTM layers for RD across five languages, named English, Italian, Spanish, French and Russian, showcasing partial improvements over the CODWOE (COmparing Dictionaries and WOrd Embeddings) competition's baseline. It explores monolingual, multilingual, and zero-shot cross-lingual settings, providing insights into the viability of cross-lingual methodologies.

Chen et al. 2022 [23] took a different approach on the English language by embedding both the definitions and words into the same shared space using transformer-based architectures to optimize the model across both tasks simultaneously. The model demonstrated superior performance in RD tasks, achieving high accuracy and consistency over previous methods. For definition modeling, while showing improvements, the results suggest areas for future enhancement, particularly in generating higher-quality definitions as indicated by human evaluations and BLEU scores.

Covering a specific instance of the English RD, Siddique et al. 2022 [25] focused on adjective phrases in Precisiated Natural Language, as mentioned that adjectives count for a large amount of content in natural language, hence highlighting the importance of a better representation for it. The proposed transformer-based model was reported to outperform the Onelook.com and WantWords online reverse dictionaries.

Following similar approaches in literature, Yan et al. 2020 [18] incorporated BERT and mBERT into the RD task for both monolingual and cross-lingual contexts. The authors propose a method that enables effective word prediction from descriptions without needing parallel corpora for cross-lingual tasks. This approach addresses challenges such as data sparsity, polysemy, and the alignment of cross-lingual word embeddings. The methodology involved modifying the input sequence to include masked tokens that BERT or mBERT would predict, converting these predictions into word scores, and using these scores to rank the possible target words.

Zhang et al. 2019 [13] presents a cross-lingual, multi-channel RD model, addressing the variability of input queries and targeting both high and low-frequency words, showing state-of-the-art performance across English and Chinese datasets. The model combines a sentence encoder with multiple characteristic predictors (POS, morpheme, word category, sememe) to enhance word retrieval from descriptions. Experiments demonstrate significant improvements over conventional methods and commercial systems, particularly for human-written descriptions, while suggesting the model's adaptability to diverse linguistic features and robustness in handling variable inputs.

Finally, covering monolingual English RD, Pilehvar et al. 2019 [11] and Hedderich et al. 2019 [10] emphasized on the importance of representing multi-sense words using different embeddings. Both methodologies address the limitations of single-sense embeddings by allowing for distinct representations of a word's different meanings, demonstrating substantial improvements in performance on the English language.

3.2 Research gap

Reading into the topic from literature proposed some questions:

1. What is the impact of enriching the definitions from dictionary data with examples and expanding the dataset on performance?
2. Are there any pre-trained architectures other than BERT that have similar good performance on Arabic?

Table 3.1: *Summary of recent research on Reverse Dictionaries*

Author	**Year**	**Language**	**Dataset**	**Methodology**	**Results**
Elbakry et al. [27]	2023	Arabic, English	KSAA ST 2023 RD	Ensemble of BERT models	1st rank in the 2023 ST
Qaddoumi [30]	2023	Arabic	KSAA ST 2023 RD	Modified mBERT model with data augmentation	2nd rank in the 2023 ST
Sibaee et al. [31]	2023	Arabic	KSAA ST 2023 RD	SemiDecoder architecture with SBERT encoder	3rd rank in the 2023 ST
Mane et al. [24]	2022	Indian	Hindi & Marathi WordNet, English dictionary by [6]	T5/mT5 based models	T5/mT5 outperformed BERT-based models
Ardoiz et al. [22]	2022	English	Data from SemEval 2022 Task[a]	Sentence-transformer model	Importance of high-quality lexicographic data
Tran et al.[26]	2022	English, Italian, Spanish, French, Russian	SemEval RD task 2023	Transformer-based models with LST and BiLSTM layers	Partial improvements over baseline
Chen et al. [23]	2022	English	Reverse dictionary	Transformer-based architectures for joint optimization	Superior performance in reverse dictionary tasks
Siddique [25]	2022	English	Reverse dictionary	Transformer-based model for adjective phrases	Outperforms online reverse dictionaries
Yan et al. [18]	2020	English	Reverse dictionary	BERT and mBERT for monolingual and cross-lingual tasks	Effective word prediction from descriptions
Zhang et al. [13]	2019	English, Chinese	Reverse dictionary	Cross-lingual, multi-channel model with sentence encoder	State-of-the-art performance across English and Chinese datasets
Pilehvar et al. [11]	2019	English	Reverse dictionary	Multi-sense embeddings for multi-sense words	Improved performance on English RD

[a] https://github.com/TimotheeMickus/codwoe/blob/main/data/README.md

Moreover, Arabic RDs are not well discovered and researched for the Arabic language, which is evident by the fact that only few articles have explored it very recently in the literature. Our contribution in this domain will explore and attempt to answer both questions presented earlier.

At the end of this report, we will present the results of our experiments and discuss the findings, to determine wether the research question have been fullfilled, and if not, what are the limitations and future work that can be done to improve the results.

4 Methodology and proposed solution

Most recent studies on reverse dictionaries have utilized pretrained models, as seen in the literature. Moreover, all studies on Arabic reverse dictionaries used BERT and its variations. Consequently, the potential for exploring other architectures and pre-trained models for the Arabic language remains intact.

In this section, we go more in depth into the proposed solution, starting with understanding the dataset and task at hand, the methods used to enrich the dataset with relevant context, and finally the model architecture and evaluation results.

Briefly:

1. The KSAA-CAD dataset contains 39k instances with its splits ready for experimentation.
2. The first part of the pipeline concerns generating contextually close examples for each word:gloss pair in the KSAA-CAD dataset. In other words, creating a new feature that puts the lexical word into an example use. The goal is to enrich the context of each word's meaning in the dictionary.
3. The inputs are then tokenized and encoded using SentencePiece tokenizer to prepare for training.
4. A pre-trained T5 model [21] is trained to predict the word embeddings. The architecture leverages sequence to sequence language modeling, an architecture that hasn't been explored for Arabic RD in the literature.
5. Finally, results are evaluated using Mean Squared Error and Cosine similarity, and compared with the baseline.

4.1 Data description

The KSAA-CAD dataset is an Arabic dictionary dataset containing **39,214 entries**, cleaned and ready for research, collected from various Arabic dictionaries. It isn't available publicly, and have been obtained as a result of participating in the ArabicNLP

Shared Task for Reverse Dictionaries, as explained previously in section 2.5. We quote the dataset description from the Shared Task organizers:

> The first of these is the "Contemporary Arabic Language Dictionary" by Ahmed Mokhtar Omar (Omar, 2008), a resource previously utilized in the first iteration KSAA-RD. The second is the newly released dictionary of the Arabic contemporary language "Mu'jam Arriyadh" (Altamimi et al., 2023). The third is the "Al Wassit LMF Arabic Dictionary" (Namly, 2015). These dictionaries comprise words, commonly referred to as lemmas, and these may come with glosses, part of speech (POS), and examples.

Table 4.2: *Dataset splits as provided by the Shared Task organizers*

Train	**Validation**	**Test**
31,372	3,921	3,921

The dataset is already split into train, validation and test sets, as seen in Table 4.2, with 3 features named: word, gloss, pos, and 3 targets named: electra, bertseg, bertmsa. Table 4.3 demonstrates a sample entry from the dataset.

1. A **word** is an entry in the dictionary, or a lemma.

2. A **gloss** is the definition or meaning for this word based on its part of speech.

3. A **pos** stands for Part-of-Speech, which is a grammatical tag assigned to words in NLP, and might include one of the following: noun, verb, adjective.

4. The final 3 values represent our targets, each of them corresponding to a representation of the word in an independent multi-dimensional space using a set of different pretrained models, employing AraELECTRA [15], AraBERTv2 [14], and camelBERT-MSA [19], respectively referred to as **electra**, **bertseg**, and **bertmsa**.

Table 4.3: *Dataset description*

Feature	**Value**
word	عين
gloss	عضو الإبصار في الكائن الحي
pos	n
electra	[0.4, 0.3, . . .]
bertseg	[0.7, 2.9, . . .]
bertmsa	[0.8, 1.4, . . .]

In Table 4.3, the word عين is a noun which means "The organ of vision in a living organism", its part-of-speech is "Noun", and is represented with 3 different word embeddings.

4.2 Features engineering: enriching the dataset

As noticed while performing data exploration, the glosses are usually short and formal descriptions written by expert linguists. Table 4.5 shows 2 words short and concise glosses, making its usage unclear, and might result in a vague understanding of the word.

Average users are unlikely to provide such precise descriptions, on the contrary, user queries might lack any key words that could identify the target word or set of words. Humans exhibit remarkable facility in acquiring new vocabulary from context early in childhood [1]. Therefore, and inspired by that capability, we propose that in order to enhance the model's ability to align words and descriptions from user queries, we need to provide the model with more contextually relevant examples for each word-gloss pair.

The manual curation of contextually relevant examples is a laborious and resource-demanding task, which, given the short time frame of this experiment, is not a feasible solution, hence the need to find an automatic way to curate examples from publicly available Arabic datasets.

We chose the Arabic wikipedia embeddings dataset from the Embedding Archives project by CohereAI, which contains 3.1 million entries from Wikipedia, each entry containing a text, and the embedding of that text, alongside other metadata. Text embeddings in

Table 4.4: *Words with short glosses*

Word	Gloss
رقع	ضربه بها
قعد	حبسه عنه

the dataset are achieved through CohereAI's *multilingual-22-12*[5] semantic embeddings model, trained for multilingual comprehension encompassing 101 languages including Arabic. This closed-source model is accessible via Cohere's API [28].

To curate a number of new examples for each word in our dataset, we use a semantic similarity approach by embedding the word and gloss using *multilingual-22-12* model, which is the same model that was used to embed Wikipedia text, then perform a vector similarity search using cosine similarity, to look for the top 5 closest entries from Wikipedia to the given word:gloss pairs. Figure 4.2 illustrates the process of generating examples for a *word:gloss* pair.

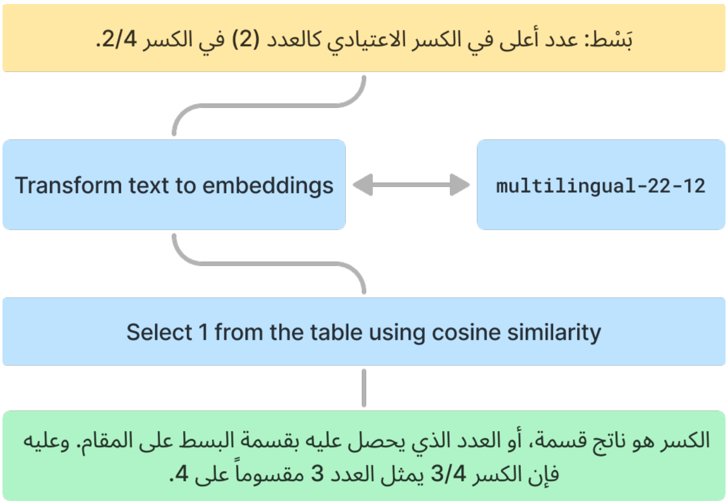

بَسْط: عدد أعلى في الكسر الاعتيادي كالعدد (2) في الكسر 2/4.
Transform text to embeddings
multilingual-22-12
Select 1 from the table using cosine similarity
الكسر هو ناتج قسمة، أو العدد الذي يحصل عليه بقسمة البسط على المقام. وعليه
فإن الكسر 3/4 يمثل العدد 3 مقسوماً على 4.

Figure 4.2: *The process of generating a single example for a word:gloss pair*

Table 4.5 demonstrates an instance of the dataset with the new **example** feature, and as can be noticed, the feature may not directly contain the dictionary lemma كذاب, but the surrounding context establishes a clear semantic relationship with the word's meaning.

[5] https://cohere.com/blog/multilingual

Table 4.5: *A word with its gloss and newly added* ***examples*** *feature*

Word	Gloss	Examples
كذاب	صيغة مبالغة من كذَبَ على: كثير الكذب	وردت لفظ الكذب ومشتقاتها في القرآن الكريم في مواضع متعددة وبصيغ متعددة.، ووردت بعدد (٢٥١) موضعًا، على (٦) أوجه وهو أسوء أنواع الجهل، وهو الاِعْتِقَادُ الجَازِمُ بِمَا لاَ يَتَفقُ مَعَ الحَقِيقَةِ، إِذْ يَعْتَقِدُ الْمَرْءُ عَارِفاً عِلْماً وَهُوَ عَكْسُ ذَلِكَ. وهو تعبيرٌ أُطلِقَ على من لا يسلِمْ بجهله، ويدَعَى ما لا يعلم

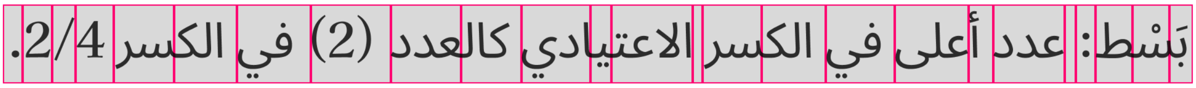
4.2.1 Data pre-processing

For machines to understand textual data, it must be preprocessed to represent it numerically. Fortunately, the dataset at hand is mostly clean and ready to be used, requiring only a **tokenization** step before training.

Data pre-processing pipeline can be summarized into two main steps:

1. **Combining the gloss with the examples:** The dataset is augmented by merging the gloss and its corresponding top two examples, into a single text string that represents the input to the model. This step enriches the training data and provides the model with additional context.
2. **Tokenization:** The input is then tokenized using the SentencePiece tokenizer, a pre-trained subword-level tokenization algorithm. This tokenizer is specifically trained for the mT5 model and is capable of handling multiple languages, including Arabic.

Figure 4.3 visualizes the result of tokenizing an Arabic sentence from the dataset, using



بَسْط: عدد أعلى في الكسر الاعتيادي كالعدد (2) في الكسر 2/4.

Figure 4.3: A sample sentence tokenized using the mT5 tokenizer

our pre-trained tokenizer.

4.3 Modeling

This section goes through the modeling pipeline and the proposed architecture for the Arabic RD. The primary objective is to surpass the performance of the 2024 RD Shared Task baseline on the cosine similarity metric.

We plan to achieve this objective by utilizing a retrieval framework that leverages 3 different word embeddings and cosine similarity measures for efficient retrieval of words based on its glosses. The end goal is to achieve a high alignment between the glosses and the words.

The methodolgy leverages transfer learning, where a pre-trained language model model is fine-tuned on the dataset to closely align glosses to words in the same multi-dimensional space. The words are already represented using the electra, bertseg, and bertmsa embeddings, which are considered as 3 independent targets in 3 different spaces.

The model architecture is based on the mT5 model [21], a variant of the T5 model [12], which is a text-to-text transformer model. The mT5 model is pre-trained on a large corpus of multilingual text data, making it suitable for handling diverse languages, including Arabic.

4.3.1 Model architecture

The proposed model architecture leverages a pre-trained mT5 model as its foundation, particularly the encoder part of the model. The last hidden layer of the mT5 encoder contains the embedding information needed to represent the input sentence in the target words' space. Figure 4.4 illustrates the model architecture, which consists of an encoder, a pooling layer, and a linear layer.

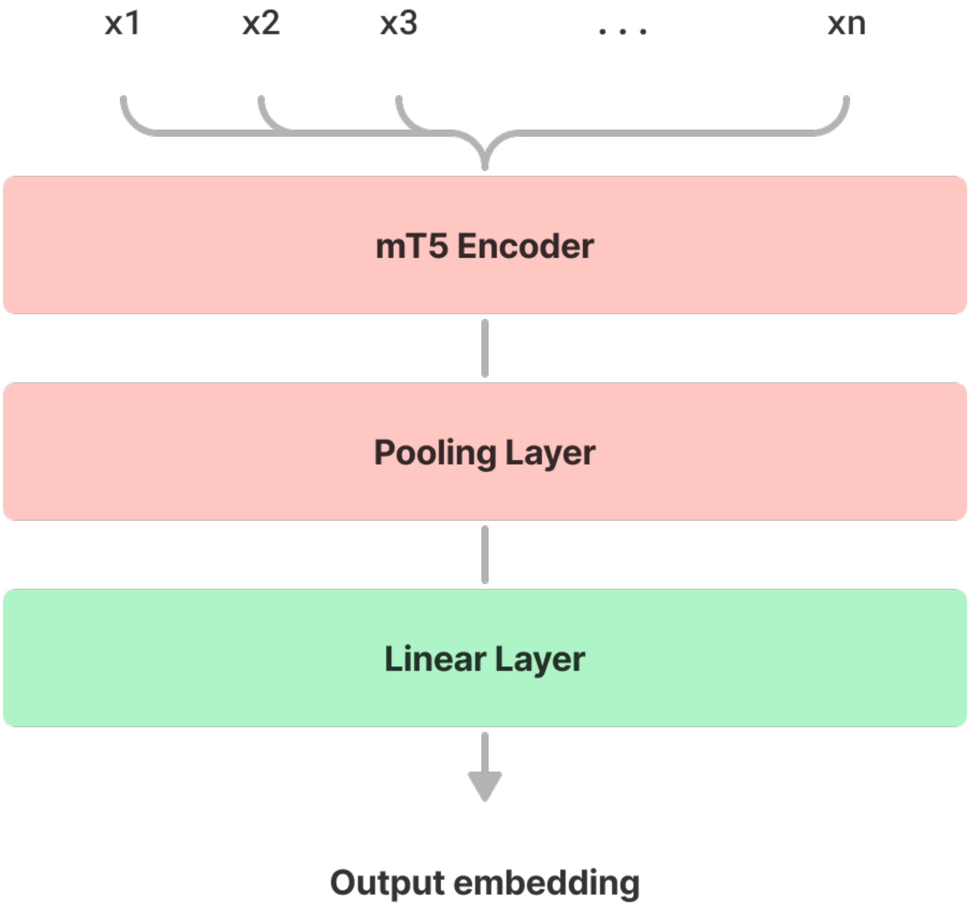


Figure 4.4: *A sample sentence tokenized using the mT5 tokenizer*

The input is a vector of size 256, which is the tokenized and encoded input sentence. The input is then passed through the mT5 encoder, which outputs a matrix of shape (batch size, sequence length, hidden size). The last hidden state of the encoder is then passed through a pooling layer to transform it into a vector of a fixed length of 718. Finally, the output of the pooling layer is passed through a linear layer that transforms it into the desired target shape, which is 256 for electra, 768 for bertseg, and 768 for bertmsa.

The last hidden state of the mT5 encoder is not a vector, rather a matrix, thus we cannot use it directly as an input to the linear layer. To solve the problem, we introduced a pooling layer that transforms the hidden state matrix to a vector of a fixed length. This layer can be represented by equation 1. This equation calculates a weighted sum of the last hidden states O where the weights are determined by the attention mask A, normalized by the total weight (or count of non-zero entries in A for each sequence).

$$pool = \frac{\sum_{j,k}(O_{ijk} \cdot A_{ijk})}{\sum_{j} A_{ij}} \tag{1}$$

The final layer is a linear layer that takes the output from the pooling layer, and transforms it into the desired target shape. The dataset includes three targets with different shapes: 256 for electra, 768 for bertseg, and 768 for bertmsa. To accommodate these distinct shapes, the final layer of the model is adapted with two variations based on the target size, resulting in three different models.

The task at hand is not a direct sequence to sequence problem, rather an information retrieval problem. Therefore, the model's objective is to predict target words embeddings instead of the words themselves.

Finally, the pre-trained models employed for the encoder part are **mT5 Base**[6] and **AraT5 V2**[7], both of which are variations of the T5 model. AraT5 V2 is a fine-tuned version of the mT5 base model on diverse Arabic data, making it more specialized for Arabic and suitable for our task [20].

[6] https://huggingface.co/google/mt5-base

[7] https://huggingface.co/UBC-NLP/AraT5-base

5 Data analysis and results

5.1 Environment setup

This subsection details the software environment and hardware specifications used for feature engineering, model development, and experimentation. Primarily, the technical stack leverages Python and Google Colab for development.

For feature engineering, we utilized Postgres engine for SQL with the pgvector extension. This extension enables efficient storage of vector type data, and allow fast and efficient querying based on cosine similarity and other similarity measures. Additionally, the HNSW (Hierarchical Navigable Small World) algorithm is used for indexing, facilitating fast and accurate nearest neighbor searches within the high-dimensional vector space.

Model development was conducted within a Google Colab environment equipped with a single A100 GPU and 12 GB RAM. The chosen framework for this task was PyTorch, which is specifically designed for deep learning. Additionally, few other libraries were employed, notably the Hugging Face Transformers for loading pre-trained models with transformer architecture.

5.2 Feature engneering analysis

To generate a new feature representing a contextual example for a single entry in the dataset, we first merge the word and its gloss. As seen in Figure 4.2, the word is بَسْط and what comes after it is the gloss, which forms the input sentence. Next, the input is transformed into an embedding using the *multilingual-22-12* model API, which results in a vector of size 768, this ensures we're projecting the input into the Wikipedia dataset space. The next step is to find the closest texts to our input in the shared space, which is achievable using cosine similarity.

The process is repeated for each word:gloss pair in the dataset, generating a new feature named **examples**. The new feature is a list of strings, each string represents a contextually relevant example for the word:gloss pair. Table 4.5 showcased an instance of the dataset with a newly added **examples** feature.

While enriching the glosses with additional context demonstrates promise, it is not without limitations. Random sampling of the results reveals instances where the retrieved entries exhibit a weaker semantic connection to the dictionary entry. Consequently, human intervention remains necessary for data labeling and refinement to ensure better quality, potentially augmented by incorporating additional data sources to enrich the retrieval process.

5.3 Training

We used the training set which contains around 31k entries, to train a number of models. The Mean Square Error loss function is used to calculate the reconstruction loss, and Adam optimizer with a learning rate value of 3e-5 to optimize the weigths in the backward step, the LR has been obtained via trial and error.

The archeticture and foundation model is T5 [21], however there's various checkpoints fine-tuned from the pre-trained original mT5 model. Among these checkpoints, we chose **mT5-base** (pre-trained and published by Google and original authors) and **AraT5v2** [20] (fine-tuned from mT5-base on a large Arabic corpus). Subsequent experiments were conducted on these two models, with the aim of comparing their performance on the task.

Briefly, the following experiments were conducted:

1. **Glosses only as input**:
   (a) **mT5-base** fine-tuned to predict electra embeddings on the original training set.
   (b) **mT5-base** fine-tuned to predict bertseg embeddings on the original training set.
   (c) **mT5-base** fine-tuned to predict bertmsa embeddings on the original training set.
   (d) **AraT5v2** fine-tuned to predict electra embeddings with target size 256.
   (e) **AraT5v2** fine-tuned to predict bertseg embeddings with target size 768.
   (f) **AraT5v2** fine-tuned to predict bertmsa embeddings with target size 768.

2. **Glosses merged with 2 examples as input**:
    (a) **mT5-base** fine-tuned to predict electra embeddings on the augmented training set.
    (b) **mT5-base** fine-tuned to predict bertseg embeddings on the augmented training set.
    (c) **mT5-base** fine-tuned to predict bertmsa embeddings on the augmented training set.
    (d) **AraT5v2** fine-tuned to predict electra embeddings with target size 256.
    (e) **AraT5v2** fine-tuned to predict bertseg embeddings with target size 768.
    (f) **AraT5v2** fine-tuned to predict bertmsa embeddings with target size 768.

We will refer to the first set of experiments as **first experiment**, and the second set as **second experiment**.

The training setup is the same for all experiments, with the only difference being the target size, the input data and the number of epochs. Figure 5.5 shows the full training pipeline with target variations.

Based on the archeticture seen in Figure 4.4, the only trainable layer in our model is the linear layer, which is responsible for transforming the output of the pooling layer into the desired target shape. The rest of the model is frozen.

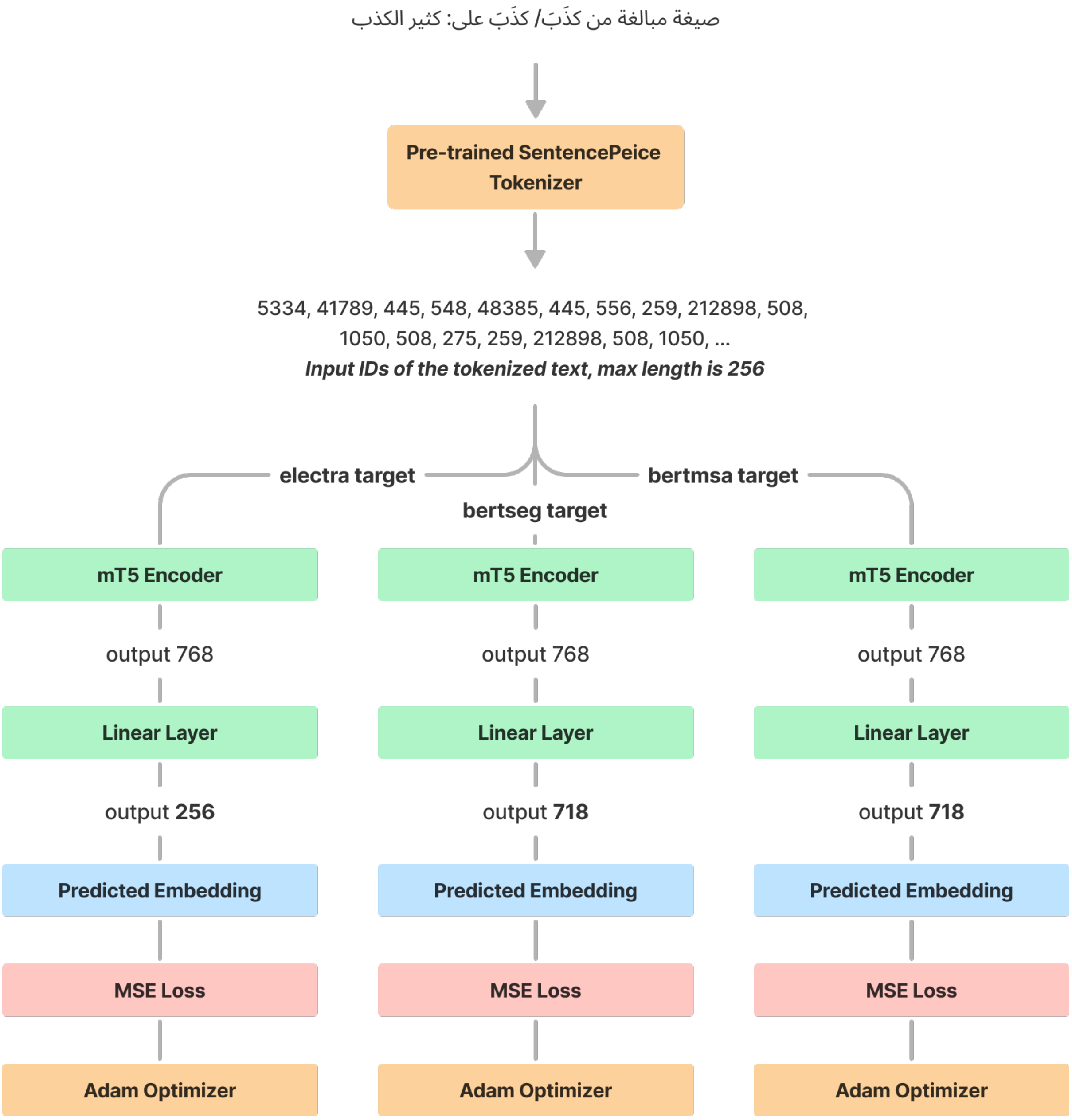


Figure 5.5: *The training pipeline with target variations*

5.4 Evaluation results

This subsection presents the final evaluation results of our experiments on the validation set. As the shared task withholds target values for the test set, its performance remains unevaluated. Once scores are submitted, official results will be announced alongside the winning solutions (those who outperform the baseline). Nonetheless, the validation set performance provides valuable insights into the model's potential.

Cosine similarity 2 and MSE 3 are employed as evaluation metrics, measuring the semantic similarity between embeddings and the reconstruction error, respectively. Table 5.6 summarizes the final results of our experiments compared to the baseline models. Moreover, Figure 5.6 visualizes the results for better comparison.

$$CosineSimilarity = \frac{A \cdot B}{\|A\| \times \|B\|} \tag{2}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3}$$

For cosine similarity, higher values (closer to 1) indicate better performance in capturing semantic relationships between the generated embeddings and the original word embedding. Lower MSE values indicate better reconstruction accuracy, meaning the model can more faithfully reproduce the original word embedding from the generated embeddings. The results presented are averaged across the validation set.

Table 5.6: *Results of the experiments on the validation set compared to the baseline. Numbers marked in bold indicate the best performance on the given target among the 3 targets.*

Model	**Target**	**Cosine Similarity**	**MSE**
mT5-base (Ours) with examples	electra	0.5107	0.2498
	bertseg	0.7657	0.0806
	bertmsa	0.7012	0.3434
mT5-base (Ours)	electra	0.5614	0.2274
	bertseg	0.7763	0.0770
	bertmsa	0.7012	0.3434
AraT5v2-base (Ours) with examples	electra	0.5152	0.2459
	bertseg	0.7656	0.0789
	bertmsa	0.6965	0.3484
AraT5v2-base (Ours)	electra	0.5686	0.2255
	bertseg	0.7752	0.0776
	bertmsa	0.7140	0.3330
CamelBERT (Baseline)	electra	**0.8185***	0.2195
	bertseg	**0.8436***	0.0555
	bertmsa	0.5113	0.2428
MARBERT (Baseline)	electra	0.6948	0.5016
	bertseg	0.7603	0.0818
	bertmsa	**0.7368***	0.1457

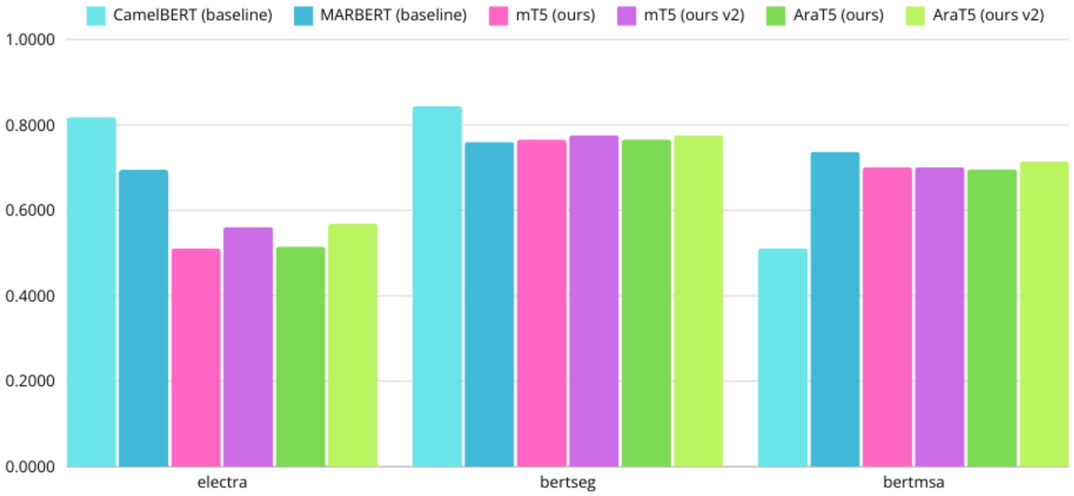


Figure 5.6: *Comparison of model performance on the validation set,* ***ours v2*** *stands for the models trained without examples.*

Overall, our models did not surpass the baseline models across all embedding types. Nonetheless, both mT5-base and AraT5v2-base models achieved competitive performance, particularly in predicting bertseg and bertmsa embeddings. The results indicate that the models are capable of capturing semantic relationships between words and glosses, as evidenced by the cosine similarity scores.

Notably, the results of the second experiment demonstrated worse performance compared to the first experiment. This discrepancy may be attributed to the quality of the augmented examples, which may not always provide meaningful context for the word:gloss pairs. Further refinement of the augmentation process and data quality may lead to improved performance. Another possible explanation is the targets provided by the shared task organizers, which was obtained from the definitions only, causing the model to overfit on the augmented examples.

Table 5.7: Predictions for a sample word:gloss pair

Input	Top k=1 prediction	Ground truth
منسوب إلى تاريخ ، له أهمية كبيرة	زراعي	تاريخي
جَرَّعَهُ الماءَ: سَقَاه إياه	شِرْب	جَرَّع

Table 5.7 showcases the model's predictions for two sample word:gloss pairs, highlighting the top retrieved predictions and the ground truth. It can be noticed that the model's predictions are not always accurate, with the first example predicting زراعي instead of the correct تاريخي, and the second example predicting شِرْب instead of the correct جَرَّع, though it can be seen that the predictions have some syntactic and semantic similarity to the ground truth, indicating that the model is learning some meaningful relationships between the words.

Possible explanations for the performance gap include the model's architecture, training duration, and the quality of the augmented examples. We may require further fine-tuning or additional training epochs to achieve optimal performance. Moreover, the quality of the augmented examples may impact the model's ability to learn meaningful word embeddings, necessitating further refinement.

To further enhance the model's performance, we propose the following:

1. **Hyperparameter tuning:** optimizing hyperparameters like learning rate, batch size, and optimizer settings could potentially improve performance.

2. **Better resources for data enrichment:** refining the process of generating contextually relevant examples to ensure higher quality and more meaningful context for the word:gloss pairs.

3. **Advanced architectures:** exploring more sophisticated architectures like larger pre-trained models (e.g., mT5-large) or incorporating techniques like selective training of specific model components could potentially lead to significant performance gains.

4. **Human evaluation:** conducting a human evaluation of the model's predictions to assess the quality of the generated word embeddings and identify areas where the model excels or falls short.

6 Conclusion

This report presents a comprehensive overview of the Arabic RD task, detailing the dataset, methodology, and results of our experiments. The proposed solution leverages a pre-trained mT5 model for word embedding retrieval, with the goal of surpassing the baseline performance on the cosine similarity metric. The model is trained on augmented examples generated from the Arabic Wikipedia dataset, aiming to enrich the glosses with additional context for improved performance.

While we did not outperform the baseline on all metrics, we achieved promising results. AraT5 generally performed slightly better than mT5, suggesting the effectiveness of fine-tuning on a specialized Arabic model.

While incorporating contextual examples did not imrove the results on this task, we belive this is regarded to several reasons including the model single-layer archeticture and shared task targets. The model's performance could be enhanced through hyperparameter tuning, exploring advanced architectures, and potentially using transfer learning or human evaluation for further refinement.

Our contributions include:

1. Exploring the potential of mT5 archeticture for Arabic RD tasks, which have not been extensively studied in the literature.
2. Introducing a novel approach to enriching glosses with contextually relevant examples, aiming to improve the model's performance and building an enhanced RD dataset.
3. Conducting a comprehensive evaluation of the proposed solution, providing insights into the model's strengths and limitations.

Future work will focus on:

1. Refining the data enrichment process, possibly integrating new resources other than Wikipedia.

2. Exploring more advanced architectures and hyperparameter tuning to enhance the model's performance.

3. Conducting human evaluations to assess the models performance and identify areas for improvement.

Finally, we believe that the proposed solution has the potential to significantly enhance the performance of Arabic RD models, and we hope that our findings will inspire further research in this domain.

References

[1] A. S. Kilian, W. E. Nagy, P. D. Pearson, R. C. Anderson, and G. E. Garcia, *Learning vocabulary from context: Effects of focusing attention on individual words during reading*, 1995. [Online]. Available: `https://core.ac.uk/display/4826065`.

[2] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proceedings of the National Academy of Sciences*, vol. 101, pp. 5228–5235, 2004. DOI: `10.1073/pnas.0307752101`. [Online]. Available: `https://www.pnas.org/doi/abs/10.1073/pnas.0307752101`.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Sep. 2014. [Online]. Available: `http://arxiv.org/abs/1409.1556`.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Dec. 2015. [Online]. Available: `http://arxiv.org/abs/1512.03385`.

[5] O. Russakovsky, J. Deng, H. Su, *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015. DOI: `10.1007/s11263-015-0816-y`.

[6] F. Hill, K. Cho, A. Korhonen, and Y. Bengio, "Learning to understand phrases by embedding the dictionary," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 17–30, Dec. 2016. DOI: `10.1162/TACL_A_00080`. [Online]. Available: `https://aclanthology.org/Q16-1002`.

[7] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," Jun. 2017. [Online]. Available: `http://arxiv.org/abs/1706.03762`.

[8] J. Devlin, M.-W. Chang, K. Lee, K. T. Google, and A. I. Language, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018. [Online]. Available: `https://github.com/tensorflow/tensor2tensor`.

[9] T. Kudo and J. Richardson, "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," Aug. 2018. [Online]. Available: `http://arxiv.org/abs/1808.06226`.

[10] M. A. Hedderich, A. Yates, D. Klakow, and G. de Melo, "Using multi-sense vector embeddings for reverse dictionaries," *IWCS 2019 - Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pp. 247–258, 2019. DOI: 10.18653/V1/W19-0421.

[11] M. T. Pilehvar, "On the importance of distinguishing word meaning representations: A case study on reverse dictionary mapping," J. Burstein, C. Doran, and T. Solorio, Eds., Association for Computational Linguistics, Jun. 2019, pp. 2151–2156. DOI: 10.18653/v1/N19-1222. [Online]. Available: https://aclanthology.org/N19-1222.

[12] C. Raffel, N. Shazeer, A. Roberts, *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," Oct. 2019. [Online]. Available: http://arxiv.org/abs/1910.10683.

[13] L. Zhang, F. Qi, Z. Liu, Y. Wang, Q. Liu, and M. Sun, "Multi-channel reverse dictionary model," Dec. 2019. [Online]. Available: http://arxiv.org/abs/1912.08441.

[14] W. Antoun, F. Baly, and H. Hajj, "Arabert: Transformer-based model for arabic language understanding," Feb. 2020. [Online]. Available: http://arxiv.org/abs/2003.00104.

[15] W. Antoun, F. Baly, and H. Hajj, "Araelectra: Pre-training text discriminators for arabic language understanding," Dec. 2020. [Online]. Available: http://arxiv.org/abs/2012.15516.

[16] J. T. Huang, A. Sharma, S. Sun, *et al.*, "Embedding-based retrieval in facebook search," Association for Computing Machinery, Aug. 2020, pp. 2553–2561, ISBN: 9781450379984. DOI: 10.1145/3394486.3403305.

[17] F. Qi, L. Zhang, Y. Yang, Z. Liu, and M. Sun, "Wantwords: An open-source online reverse dictionary system," Q. Liu and D. Schlangen, Eds., Association for Computational Linguistics, Oct. 2020, pp. 175–181. DOI: 10.18653/v1/2020.emnlp-demos.23. [Online]. Available: https://aclanthology.org/2020.emnlp-demos.23.

[18] H. Yan, X. Li, X. Qiu, and B. Deng, "Bert for monolingual and cross-lingual reverse dictionary," T. Cohn, Y. He, and Y. Liu, Eds., Association for Computational Linguistics, Nov. 2020, pp. 4329–4338. DOI: 10.18653/v1/2020.findings-emnlp.388. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.388.

[19] G. Inoue, B. Alhafni, N. Baimukan, H. Bouamor, and N. Habash, "The interplay of variant, size, and task type in Arabic pre-trained language models," in *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, Kyiv, Ukraine (Online): Association for Computational Linguistics, Apr. 2021.

[20] E. M. B. Nagoudi, A. Elmadany, and M. Abdul-Mageed, "Arat5: Text-to-text transformers for arabic language generation," Aug. 2021. [Online]. Available: http://arxiv.org/abs/2109.12068.

[21] L. Xue, N. Constant, A. Roberts, *et al.*, "Mt5: A massively multilingual pre-trained text-to-text transformer," K. Toutanova, A. Rumshisky, L. Zettlemoyer, *et al.*, Eds., Association for Computational Linguistics, Jun. 2021, pp. 483–498. DOI: 10.18653/v1/2021.naacl-main.41. [Online]. Available: https://aclanthology.org/2021.naacl-main.41.

[22] A. Ardoiz, M. Ortega-Martin, Ó. Garcia-Sierra, J. Álvarez, I. Arranz, and A. Alonso, "Mmg at semeval-2022 task 1: A reverse dictionary approach based on a review of the dataset from a lexicographic perspective," G. Emerson, N. Schluter, G. Stanovsky, *et al.*, Eds., Association for Computational Linguistics, Jul. 2022, pp. 68–74. DOI: 10.18653/v1/2022.semeval-1.7. [Online]. Available: https://aclanthology.org/2022.semeval-1.7.

[23] P. Chen and Z. Zhao, "A unified model for reverse dictionary and definition modelling," Y. He, H. Ji, S. Li, Y. Liu, and C.-H. Chang, Eds., Association for Computational Linguistics, Nov. 2022, pp. 8–13. [Online]. Available: https://aclanthology.org/2022.aacl-short.2.

[24] S. B. Mane, H. N. Patil, K. B. Madaswar, and P. N. Sadavarte, "Wordalchemy: A transformer-based reverse dictionary," 2022, pp. 1–5. DOI: 10.1109/CONIT55038.2022.9848383. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9848383.

[25] B. Siddique and M. M. Beg, “Adjective phrases in pnl and its application to reverse dictionary,” *IEEE Access*, vol. 10, pp. 28 385–28 396, 2022, ISSN: 21693536. DOI: 10.1109/ACCESS.2022.3158011.

[26] T. H. H. Tran, M. Martinc, M. Purver, and S. Pollak, “Jsi at semeval-2022 task 1: Codwoe - reverse dictionary: Monolingual and cross-lingual approaches,” G. Emerson, N. Schluter, G. Stanovsky, *et al.*, Eds., Association for Computational Linguistics, Jul. 2022, pp. 101–106. DOI: 10.18653/v1/2022.semeval-1.12. [Online]. Available: https://aclanthology.org/2022.semeval-1.12.

[27] A. Elbakry, M. Gabr, M. ElNokrashy, and B. AlKhamissi, “Rosetta stone at ksaa-rd shared task: A hop from language modeling to word–definition alignment,” H. Sawaf, S. El-Beltagy, W. Zaghouani, *et al.*, Eds., Association for Computational Linguistics, Dec. 2023, pp. 477–482. DOI: 10.18653/v1/2023.arabicnlp-1.43. [Online]. Available: https://aclanthology.org/2023.arabicnlp-1.43.

[28] E. Kamalloo, X. Zhang, O. Ogundepo, *et al.*, “Evaluating embedding apis for information retrieval,” May 2023. [Online]. Available: http://arxiv.org/abs/2305.06300.

[29] D. Khurana, A. Koli, K. Khatter, and S. Singh, “Natural language processing: State of the art, current trends and challenges,” *Multimedia Tools and Applications*, vol. 82, pp. 3713–3744, 3 Jan. 2023, ISSN: 15737721. DOI: 10.1007/s11042-022-13428-4.

[30] A. Qaddoumi, “Abed at ksaa-rd shared task: Enhancing arabic word embedding with modified bert multilingual,” H. Sawaf, S. El-Beltagy, W. Zaghouani, *et al.*, Eds., Association for Computational Linguistics, Dec. 2023, pp. 472–476. DOI: 10.18653/v1/2023.arabicnlp-1.42. [Online]. Available: https://aclanthology.org/2023.arabicnlp-1.42.

[31] S. Sibaee, S. Ahmad, I. Khurfan, *et al.*, “Qamosy at arabic reverse dictionary shared task: Semi decoder architecture for reverse dictionary with sbert encoder,” H. Sawaf, S. El-Beltagy, W. Zaghouani, *et al.*, Eds., Association for Computational Linguistics, Dec. 2023, pp. 467–471. DOI:

10.18653/v1/2023.arabicnlp-1.41. [Online]. Available: https://aclanthology.org/2023.arabicnlp-1.41.