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 (`https://www.pnu.edu.sa/ar/Deanship/PostGraduate/Documents/Ł%20Łł%20Ł%20%20Ł%20Ł.pdf`) and I confirm that ethical issues have been considered, evaluated and appropriately addressed in this research.

Signed:

Candidate Name: <u>Mais Alheraki</u>

ID Number: <u>444010562</u>

Date: <u>2 May 2024</u>

List of Figures

List of Tables

List of abbreviations

RD: Reversed Dictionary

NLP: Natural Language Processing

NLG: Natural Language Generation

NLU: Natural Language Understanding

BERT: Bidirectional Encoder Representations from Transformers

Seq2Seq: Sequence to Sequence

Abstract

The domain of reverse dictionaries, while advancing in languages like English and Chinese, remains significantly underdeveloped for less resourced languages, such as Arabic. This study attempts to explore a data-driven approach to enhance word retrieval processes in Arabic reverse dictionaries. The research focuses on the ArabicNLP 2024 Shared Task, which provides a 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 for word embedding prediction, and evaluating the results using Mean Squared Error and Cosine similarity. The study aims to contribute to the advancement of Arabic reverse dictionaries.

1 Introduction

While reversed dictionaries have witnessed advancements in languages like English and Chinese (e.g., WantWords [11]), their development for Arabic remains significantly underdeveloped. This gap is particularly concerning for a language with a rich linguistic heritage and widespread use. Despite this, robust and technologically advanced tools for reverse lexical searches in Arabic are scarce, with the only ongoing effort being the 1st Arabic Reverse Dictionary shared task launched in 2023 by the King Salman Global Academy for Arabic Language [1].

1.1 Content of study

This research aims to address the limitations of existing Arabic reverse dictionary models by proposing a data-driven approach that leverages deep learning techniques to enhance word retrieval processes. We focus on the ArabicNLP 2024 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 employs state-of-the-art deep learning models and innovative data augmentation techniques to improve performance.

1.2 Problem Statement & Motivation

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

However, existing Arabic reverse dictionary models are often limited in scope and functionality, frequently relying on traditional dictionary structures that fail to capture the nuances and complexities of the language. This research is motivated by the desire to enrich this field by exploring new methodologies that can significantly enhance the performance of Arabic reverse dictionaries.

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 terms. 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.

---

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

صيغة مبالغة من كذَبَ/ كذَّبَ على: كثير الكذب

Pre-trained SentencePeice Tokenizer

5334, 41789, 445, 548, 48385, 445, 556, 259, 212898, 508, 1050, 508, 275, 259, 212898, 508, 1050, ...
Input IDs of the tokenized text, max length is 256

Fine-tuned Model — **Predicted Embedding** — Cosine Similarity

Closest Target Embedding

كذّاب
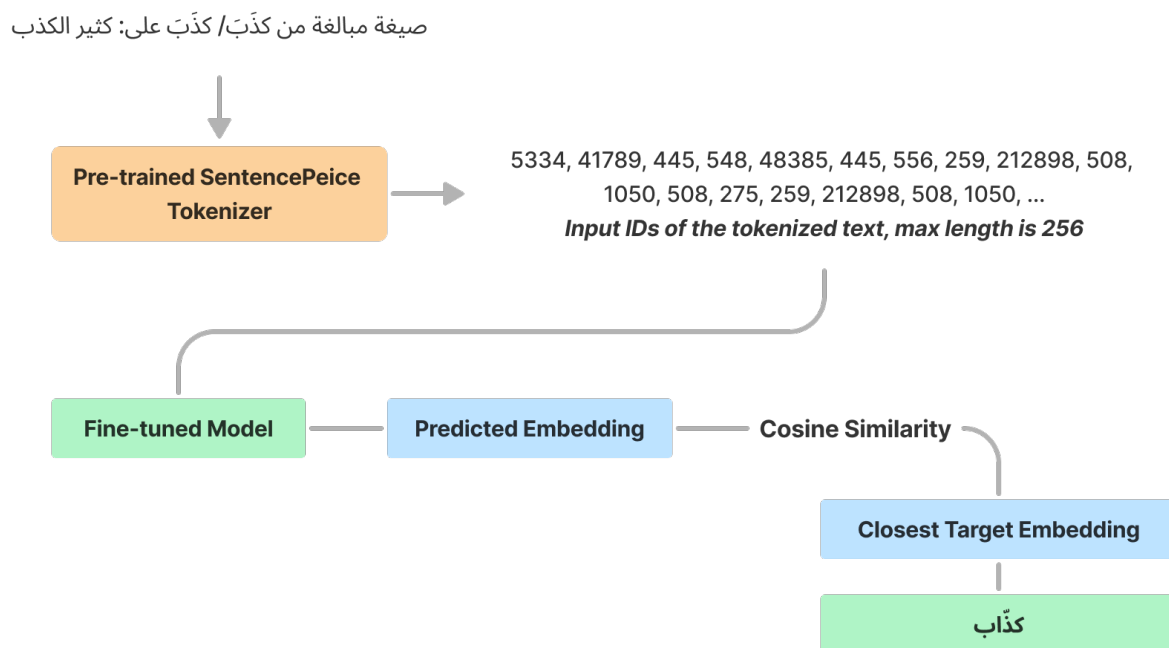
Figure 1: *The workflow of a reverse dictionary*

1.4 Proposed solution

This research leverages the new ArabicNLP 2024 Shared Task dataset, comprising 39,214 word-gloss pairs with corresponding target word embeddings. Our goal is to surpass the baseline performance by employing cutting-edge deep learning models and innovative data augmentation techniques. Figure 1 illustrates the proposed solution's workflow.

The proposed solution 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. **T5 Model Integration:** A T5 model will be trained to predict word embeddings from the enriched dataset.

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, and reverse dictionaries.

2. Section 3 reviews related work in the field of reverse dictionaries, highlighting recent advancements and research gaps.

3. Section 4 outlines the methodology and proposed solution, detailing the dataset, data expansion techniques, model architecture, and evaluation metrics.

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

2   Background material

This chapter lays the groundwork for understanding the subsequent sections of this dissertation by exploring the fundamentals of natural language processing and deep learning. 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.

Natural Language Processing (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 [21].

2.2   Reverse Dictionary and Data Science

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.

Reversed dictionaries are a form of dictionaries where a description yields a set of words from the dictionary that best matches the description. Traditionally used as tools for linguistic exploration, reversed dictionaries have evolved to play a significant role in data science. Initially conceived to assist with language retrieval challenges, they now offer innovative solutions in data analysis and processing. A prime use-case in data science is their application in data exploration and analysis, where reverse dictionaries facilitate the identification of relevant features within complex datasets by generating key terms or phrases. This enhances the efficiency of data mining and fosters the discovery of new insights [15].

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. 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.

2.3 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 tasks, including NLP.

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.

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 the pre-trained model, the 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.

2.3.1 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 recurrent neural networks (RNNs) or transformers:

- Encoder RNN/Transformer: This network processes the input sequence and encodes it into a fixed-length vector representation capturing the semantic meaning.
- Decoder RNN/Transformer: 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.

Recent advancements in Seq2Seq models include attention mechanisms, which allow the decoder to selectively focus on relevant parts of the encoded input during the generation process.

2.3.2 Text embeddings and tokenization

Text embeddings are a crucial component in NLP tasks. They represent words or sub-word units (like characters) 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.

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. The choice of tokenization strategy can significantly impact the performance of deep learning models in NLP tasks.

2.3.3 SentencePiece tokenizer

SentencePiece is an unsupervised text tokenizer, used by the T5 models for text data processing [6]. 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.

2.3.4 Transformer Architecture: A Paradigm Shift

The Transformer architecture, introduced by Vaswani et al. in 2017 in the famous paper "Attention is All You Need" [4], 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

2.3.5 BERT vs. T5: Pre-trained Transformer Models

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 [5] 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 [9], 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.

2.4   ArabicNLP 2024 Shared Task

The KSAA-CAD Shared Task[2] is a part of the ArabicNLP [3] conference for 2024, and it aims to advance the field of Arabic natural language processing by providing a platform for researchers to collaborate and innovate. This task specifically targets the development of models capable of predicting words from their definitions in Arabic. Participants are provided with a dataset containing word-gloss pairs and corresponding word embeddings, challenging them to build models that can accurately predict the target words based on the provided glosses.

---

[2]The new task has been launched in May this year, more information can be found here: arai.ksaa.gov.sa/sharedTask
[3]arabicnlp2024.sigarab.org

3   Related work

Understanding the meaning behind words, even within the same language, presents a significant challenge for machines. Monolingual reversed 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 [19]. (2023) as part of the ArabicNLP 2023 Shared Task, where they demonstrate an approach to Arabic reverse dictionary 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 [22], a method is introduced to enhance Arabic word embeddings using a modified BERT Multilingual model with data augmentation, targeting improvements in Arabic reverse dictionary 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. [23] 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 extensive research in the area of RDs. Mane et al. [16] 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.

Ardoiz et al. [14], in the SemEval RD task, studied the significance of high-quality lexicographic data in the efficiency of reversed 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 in SemEval RD task, evaluates Transformer-based models enhanced with LST and BiLSTM layers for reverse dictionary 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 [15] 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 reverse dictionary 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 reversed dictionary, Siddique et al. 2022 [17] 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 [12] 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 [10] presents a cross-lingual, multi-channel reverse dictionary 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 [8] and Hedderich et al. 2019 [7] 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?

Table 1: *Summary of recent research on Reverse Dictionaries*

Author	Year	Language	Dataset	Methodology	Results
Elbakry et al. [19]	2023	Arabic, English	KSAA Shared Task 2023 RD dataset	Ensemble of fine-tuned BERT models	Improved results with ensemble strategy, winner of the 2023 Shared Task
Qaddoumi [22]	2023	Arabic	KSAA Shared Task 2023 RD dataset	Modified BERT Multilingual model with data augmentation	High ranking in shared task
Sibaee et al. [23]	2023	Arabic	KSAA Shared Task 2023 RD dataset	SemiDecoder architecture with SBERT encoder	High ranking in shared task
Mane et al. [16]	2022	Indian	Hindi and Marathi WordNet, and English dictionary definition by Hill et al. [3]	mT5 pretrained model	Focus on translating and understanding user inputs
Ardoiz et al. [14]	2022	English	Data from SemEval 2022 Task [a]	Sentence-transformer model	Importance of high-quality lexicographic data
Tran et al.[18]	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. [15]	2022	English	Reverse dictionary	Transformer-based architectures for joint optimization	Superior performance in reverse dictionary tasks
Siddique [17]	2022	English	Reverse dictionary	Transformer-based model for adjective phrases	Outperforms online reverse dictionaries
Yan et al. [12]	2020	English	Reverse dictionary	BERT and mBERT for monolingual and cross-lingual tasks	Effective word prediction from descriptions
Zhang et al. [10]	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. [8]	2019	English	Reverse dictionary	Multi-sense embeddings for multi-sense words	Improved performance on English RD
Hedderich et al. [7]	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

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

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.

4 Methodology and proposed solution

Most recent studies on reversed dictionaries have utilized pretrained models, as seen in the literature. Moreover, all studies on Arabic reversed dictionaries used BERT and its variations. Consequently, the potential for exploring other architectures and pretrained models remain 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 expand the dataset with relevant context, and finally the model architecture and evaluation results.

Briefly:

1. The Shared Task 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 pairs in the dataset. 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 T5 model 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

Table 2: *Dataset splits*

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

The dataset used is an Arabic dictionary containing 39,214 entries, splitted into train, validation and test sets, as seen in Table 2, with 6 features named: word, gloss, pos, electra, bertseg, bertmsa, described in Table 3. The dataset was provided to the participants in the Shared Task and is not publicly available.

Table 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, . . .]

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 last 3 features represent the embeddings, each embedding corresponds to the representation of the word in a high dimensional space using a set of pretrained models, employing AraELECTRA (Antoun et al., 2021), AraBERTv2 (Antoun et al., 2020), and camelBERT-MSA (Inoue et al., 2021), respectively referred to as electra, bertseg, and bertmsa.

5. In the previous table, 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   Data expansion

As noticed while performing data exploration, the glosses are usually short and formal descriptions written by expert linguists. In Table 5, we notice that some glosses are short and concise, making its usage unclear, and results in a vague understanding of the word.

Table 4: *Words with short glosses*

Word	Gloss
رقع	ضربه بها
قعد	حبسه عنه

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. Therefore, and inspired by our ability to perceive and understand new vocabulary from context [1], we propose that in order to enhance the

model's ability to learn words from user queries, the model should be trained on usage examples that put the words into context.

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.

The datThe dataset of choice is the Arabic wikipedia embeddings 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 (or embedding), alongside other metadata. Text embeddings in the dataset are achieved through CohereAI's multilingual-22-12 semantic embeddings model, trained for multilingual comprehension encompassing 101 languages including Arabic. This closed-source model is accessible via Cohere's API (Kamalloo2023). aset of choice is the Arabic wikipedia embeddings 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 (or embedding), alongside other metadata. Text embeddings in the dataset are achieved through CohereAI's multilingual-22-12 semantic embeddings model, trained for multilingual comprehension encompassing 101 languages including Arabic. This closed-source model is accessible via Cohere's API [20].

To curate a number of examples for each word, we use a semantic similarity approach by embedding the word and gloss using multilingual-22-12 model, and running a vector search using cosine similarity, to look for the top 5 closest entries from Wikipedia to the given dictionary gloss. As observed in Table 5, the example may not directly contain the dictionary entry, but the surrounding context establishes a clear semantic relationship with the word's meaning.

Table 5: A word with its gloss and 2 examples from Wikipedia

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

4.3 Modeling

This section details the modeling approach employed to construct an Arabic RD retrieval system. 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 through a retrieval framework that leverages word

embeddings and cosine similarity measures for efficient retrieval of words based on its glosses. The pretrained mT5 model is used, which is a transformer model built on sequence to sequence language modeling approach.

4.3.1 Data preprocessing

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 the tokenization step before training.

Data preprocessing can be summarized into two main steps:

1. **Text Augmentation:** The dataset is augmented by merging the gloss and its corresponding top two examples into a single text string that serves as the input for the model. This step enriches the training data and provides the model with additional context.

2. **Tokenization:** The augmented text is tokenized using the mT5 tokenizer. This tokenizer leverages SentencePiece, a subword-level tokenization algorithm trained on the original corpora used to train the mT5 model.

Figure 2 visualizes the result of tokenizing an Arabic sentence from the dataset, using mT5 pre-trained tokenizer.

بَسْط: عدد أعلى في الكسر الاعتيادي كالعدد (2) في الكسر 2/ 3.

Figure 2: A sample sentence tokenized using the mT5 tokenizer

4.3.2 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 3 illustrates the model architecture, which consists of an encoder, a pooling layer, and a linear layer.

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.
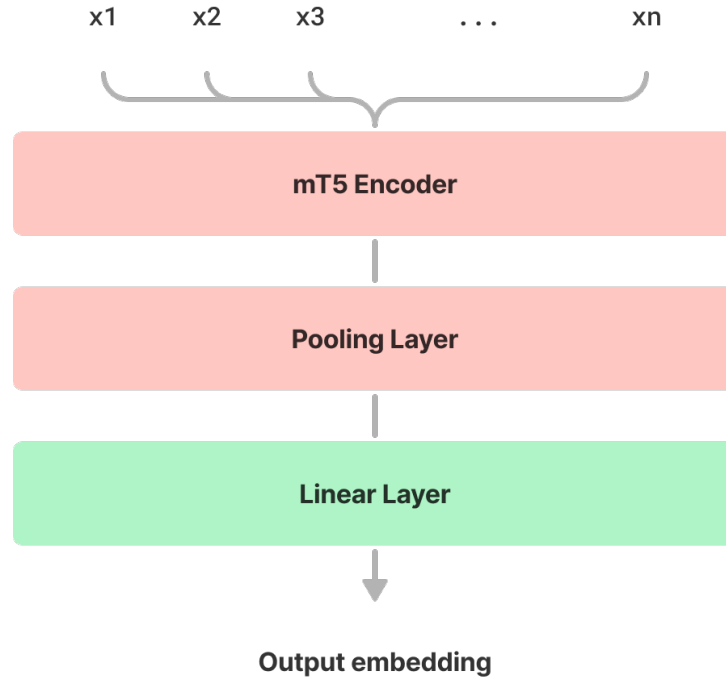
Figure 3: A sample sentence tokenized using the mT5 tokenizer

The last hidden state of the mT5 encoder is not a vector, rather a matrix, and adapt it to output an embedding of a specific shape, a pooling layer will first transform the hidden state matrix to a vector of a fixed (the mT5 encoder output shape). This layer can be represented by the following equation. 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 target embedding sizes: 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 task. Therefore, the model's objective is to predict target word embeddings instead of the words themselves.

Finally, the pre-trained models employed for the encoder part are mT5 Base and AraT5 V2, 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 [13].

5 Data analysis and results

5.1 Environment setup

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

For example generation, we utilized Postgres engine for SQL with the pgvector extension. This extension enables efficient storage and querying of vectors based on cosine similarity. 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 and training of pre-trained models with the transformer architecture.

5.2 Data expansion analysis

To generate an example for a single entry in the dataset, we first merge the word and its gloss. In Figure 4, the word is بَشط 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 space, which is achievable using cosine similarity, such that the closest embedding to the input is the example we're looking for.
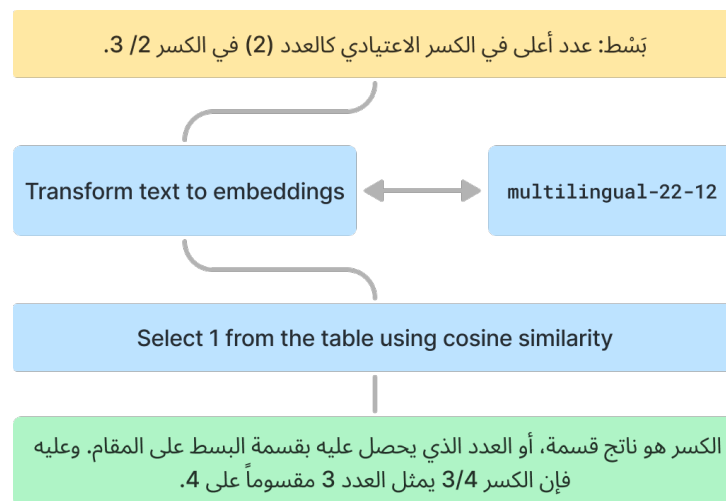

Figure 4: The process of generating a single example for a word:gloss pair

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

The training is done on the training set, with Mean Square Error loss function and an Adam optimizer with a learning rate value of 3e-5, which has been obtained via trial and error. Training has been done for 1 epoch, as with more epochs we noticed an overfitting over the training data, which is expected given that we're only training a single linear layer. Figure 5 shows the full training pipeline with target variations.

The only difference between the models is the target size, which is 256 for electra, 768 for bertseg, and 768 for bertmsa. The model is trained on the training set and evaluated on the validation set using cosine similarity and Mean Squared Error (MSE) as evaluation metrics.

The only trainable layer in the 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, as it is pre-trained and does not require further training.

5.4 Evaluation results

This subsection presents the final evaluation results of our model 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. 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 6 summarizes the final results of our models compared to the baseline models.

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

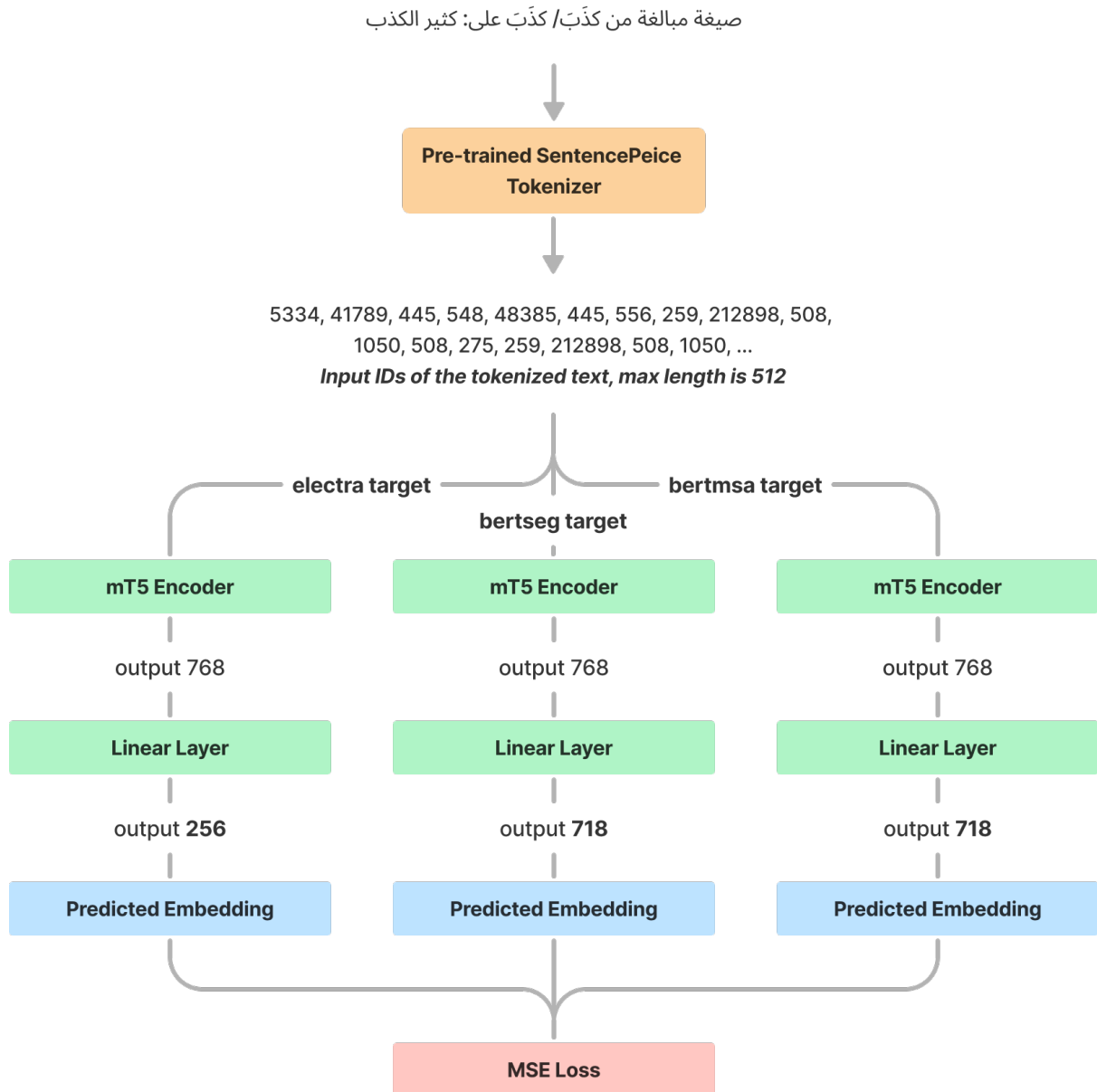$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{3}$$

صيغة مبالغة من كذَّبَ/ كذَبَ على: كثير الكذب

Pre-trained SentencePeice Tokenizer

5334, 41789, 445, 548, 48385, 445, 556, 259, 212898, 508,
1050, 508, 275, 259, 212898, 508, 1050, ...
Input IDs of the tokenized text, max length is 512

electra target   bertmsa target

bertseg target

| **mT5 Encoder** | **mT5 Encoder** | **mT5 Encoder** |

output 768   output 768   output 768

| **Linear Layer** | **Linear Layer** | **Linear Layer** |

output **256**   output **718**   output **718**

| **Predicted Embedding** | **Predicted Embedding** | **Predicted Embedding** |

MSE Loss

Figure 5: *The training pipeline with target variations*

Table 6: Results of the model on the validation set

Model	Embedding	Cosine Similarity	MSE
mT5-base (Ours)	electra	0.5107	0.2498
	bertseg	0.7657	0.0806
	bertmsa	0.7012	0.3434
AraT5v2-base (Ours)	electra	0.5152	0.2459
	bertseg	0.7656	0.0789
	bertmsa	0.6965	0.3484
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

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 averages across the validation set.

Our mT5-base model trained on bertseg embeddings achieves the highest cosine similarity score (0.7657) among our models, marginally exceeding the MARBERT baseline (0.7603) by 0.0054. However, it falls short of the CamelBERT baseline across all embedding types, with the largest gap observed using electra embeddings (e.g. 0.8185 vs. 0.5107).

Table 7: Predictions for a sample word:gloss pair

Input	Top retrieved predictions	Ground truth
منسوب إلى تاريخ ، له أهمية كبيرة	زراعي	تاريخي
جَرَّعَهُ الماءَ: سَقَاه إياه	شِرْب	جَرَّع

Table 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 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. The mT5 model 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. **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.

3. **Fine-tuning:** Fine-tuning the model on a larger dataset or domain-specific data could improve its performance on the task by adapting the model to the specific characteristics of the data.

4. **Transfer Learning:** Leveraging transfer learning from models pre-trained on similar tasks or domains could provide a head start for training the model on the reverse dictionary task.

5. **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.

Conclusion

This report presents a comprehensive overview of the Arabic Reverse Dictionary task, detailing the dataset, methodology, and results of our model. 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.

The evaluation results on the validation set demonstrate the model's potential, with the mT5-base model trained on bertseg embeddings achieving the highest cosine similarity score among our models. However, the model falls short of the CamelBERT baseline across all embedding types, indicating room for improvement. Possible avenues for enhancing the model's performance include hyperparameter tuning, advanced architectures, fine-tuning, transfer learning, and human evaluation.

Despite the limitations, this domain is still of high importance in the field of Arabic NLP. Further research and experimentation are needed to address the challenges and limitations identified in this work, including a focus on data quality, model architecture, and evaluation metrics.

This work contributes to the growing body of research on Arabic reverse dictionaries, providing insights into the challenges and opportunities in this domain. By exploring novel approaches and methodologies, we aim to advance the state-of-the-art in Arabic reverse dictionary tasks and facilitate the development of more accurate and efficient word embedding retrieval systems.

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] 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.

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

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

[21] 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.

[22] 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.

[23] 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.