Baleegh at KSAA-CAD 2024: Towards Enhancing Arabic Reverse Dictionaries

Mais Alheraki[*]
PNU
444010562@pnu.edu.sa

Souham Meshoul
PNU
sbmeshoul@pnu.edu.sa

Abstract

The domain of reverse dictionaries (RDs), while advancing in languages like English and Chinese, remains underdeveloped for 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 SOTA deep learning models and innovative data expansion 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 on the shared task metrics. 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.

1 Introduction

While reverse dictionaries have witnessed advancements in languages like English and Chinese e.g. WantWords (Qi et al., 2020), 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 First Arabic RD shared task launched in 2023 by King Salman Global Academy for Arabic Language (KSAA)[1] (Al-Matham et al., 2023), and the recent KSAA-CAD (Contemporary Arabic Dictionary) Shared Task for 2024[2].

Reverse dictionaries are a form of dictionaries where a description yields a set of words from the dictionary that semantically matches the description. 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 aligned with the text meaning. This enhances the efficiency of data mining and fosters the discovery of new insights (Chen and Zhao, 2022).

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.

Recent studies have explored various approaches to RD. In (Elbakry et al., 2023), the authors demonstrated success as the winning solution in the 2023 shared task (Al-Matham et al., 2023) using an ensemble of fine-tuned BERT models. Their results formed the SOTA baseline for this 2024 shared task. In (Qaddoumi, 2023), authors focused on enhancing Arabic word embeddings through a modified BERT model and data augmentation while in (Sibaee et al., 2023), authors utilized a SemiDecoder architecture with an SBERT encoder for effective word definition encoding.

In other languages, Authors of (Mane et al., 2022) proposed a unique approach using mT5 for Indian languages, while authors in (Ardoiz et al., 2022) emphasized the importance of high-quality lexicographic data for optimal RD model performance. For English, the authors in (Chen and Zhao, 2022) embedded 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.

In this paper, the use of a RD is cast as a supervised learning task and the main questions addressed are:

[1] https://arai.ksaa.gov.sa/sharedTask/
[2] https://arai.ksaa.gov.sa/sharedTask2024/

Proceedings of The Second Arabic Natural Language Processing Conference, pages 704–708
August 16, 2024 ©2024 Association for Computational Linguistics

1. What is the impact of enriching the definitions in the KSAA-CAD dataset with contextual examples on the performance of the proposed Arabic RD model?

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

We leverage the new KSAA-CAD dataset, which is shared exclusively with the participants of the 2024 RD Shared Task (Alshammari et al., 2024), comprising word-gloss pairs with corresponding target word embedding, alongside a SOTA baseline results. Our goal is to surpass or match the baseline performance. The methodology goes through 2 steps:

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 (Xue et al., 2021) will be fine-tuned on both the original and enriched KSAA-CAD dataset. Given the novelty of the T5 model and the scarcity of research on it within the Arabic RD literature, we set out to explore its capabilities in this specific domain.

The code utilized in this study has been made available on GitHub[3] to ensure the reproducibility of the experimental results.

2 Dataset

The KSAA-CAD dataset is an Arabic dictionary dataset containing **39,214 entries**, collected from various Arabic dictionaries.

	Train	Dev	Test
CA dictionary	31,372	3,921	3,922

Table 1: Statistics about the data sizes and splits

The dataset is split into `train`, `dev` and `test` sets, as seen in Table 1, with 3 features named: `word`, `gloss`, `pos`, and 3 target embeddings named: `electra`, `bertseg`, `bertmsa`. Table 2 demonstrates a sample entry from the dataset.

Sample word	Sample gloss
خصب	نامٍ، كثير العشب

Table 2: A sample data point from the dataset

2.1 Features engineering: enriching the dataset

As noticed while performing data exploration, the glosses are often short and formal descriptions written by expert linguists. Table 3 shows 2 words with 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 (Kilian et al., 1995). 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 would need to provide the model with more contextually relevant examples for each word-gloss pair.

The manual retrieval 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, therefore the need to find an automatic way to curate examples from publicly available Arabic datasets.

We used the Arabic Wikipedia Embeddings[4] 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 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 (Kamalloo et al., 2023).

To curate a number of examples for each word, firstly we append the definition to the word in a single input string, seprating them by a colon, then we embed the resulting string using *multilingual-22-12* model, which is the same model that was used to embed Wikipedia's text, then perform a vector search using cosine similarity 1, to look up the top

[3] `https://github.com/pr-Mais/ksaa-cad-2024`

[4] Cohere/wikipedia-22-12-ar-embeddings

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

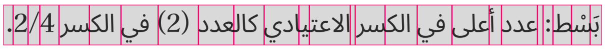

Figure 1: A sample sentence tokenized using the SentencePiece tokenizer

5 closest entries to the given word:gloss pairs. Finally, the enriched dataset will encompass a total of **156,860 examples**, distributed over 31,372 words in the training set, each word having 5 examples as a single feature.

$$CosineSimilarity = \frac{A \cdot B}{\|A\| \times \|B\|} \quad (1)$$

Table 3 demonstrates a data point with a retrieved **example**. The example may not directly contain the lemma كذاب, however the surrounding context establishes a clear semantic relationship with the word's meaning.

2.2 Data pre-processing

The KSAA-CAD dataset was prepared for training with two steps. Firstly, we created a new input by combining the glosses and their corresponding 1st example into a single input string, to enrich the training data. Secondly, each of the inputs (gloss only, and gloss+example) were tokenized using a pre-trained SentencePiece tokenizer. Figure 1 visualizes the result of tokenizing an Arabic sentence from the dataset.

3 System

Recent studies on RDs have utilized pre-trained transformers, as seen in literature, most of which focused on BERT-based transformers for Arabic. Consequently, the potential for exploring other architectures and pre-trained models for the Arabic language remains intact.

Our model architecture is based on **AraT5 V2**[6] (Nagoudi et al., 2021), a fine-tuned T5 transformer model for Arabic. The input is processed by the AraT5 V2 encoder, producing a hidden state matrix. A pooling layer 3 then converts this matrix into a fixed-length vector, which is subsequently fed into a linear layer for the final prediction. This architecture enables effective utilization of AraT5 V2's language understanding capabilities for RD tasks.

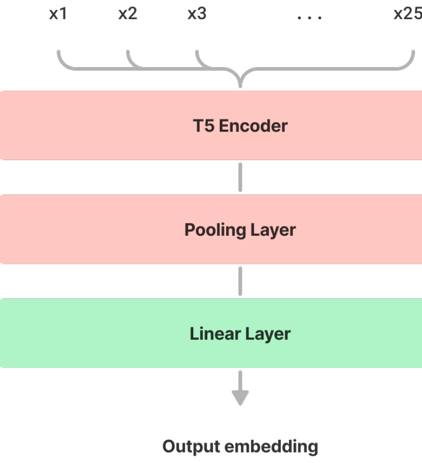

Figure 2: T5-based model architecture for Arabic RD. The input is a vector of size 256.

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

We train 2 models for 2 epochs with the same setup but different inputs, the number of epochs being low due to the over-fitting noticed when training for higher than 2 epochs. The first model is trained on only glosses, while the other on the glosses merged with example retrieved in step 2.1. Mean Square Error is used as a loss function, and Adam as an optimizer with a learning rate value of `3e-5`.

4 Results and Discussion

Table 4 presents the final performance across the shared task metrics on the development and test sets. Notably, incorporating example input alongside gloss definitions resulted in a marginal decrease in performance compared to utilizing gloss alone. This observation may be attributed to the automated example retrieval process, which could potentially introduce contextually irrelevant instances, thereby confounding the model's ability to differentiate between the true meaning of a word and its unrelated example. Furthermore, the finite availability of resources for example extraction might have led to instances where certain words were absent from the Wikipedia dataset, resulting in the retrieval of contextually inappropriate text.

Overall, our models did not surpass the baseline results across all embedding types. Nonetheless, we achieved competitive performance, particularly in predicting `bertseg` and `bertmsa` embeddings, which indicate the model is capable of aligning the words from both spaces to glosses, suggesting

[6] https://huggingface.co/UBC-NLP/AraT5-base-1024

Word	Gloss	Retrieved Example
كذاب	صيغة مبالغة من كذَبَ على: كثير الكذب	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

Table 3: A word with its gloss and newly added **examples** feature

Input	Embedding	MSE	Cosine	Rank
Gloss (ours)	Electra	0.2255 / 0.2257	0.5686 / 0.5678	0.1721 / 0.1781
	Bertmsa	0.3330 / 0.3299	0.7140 / 0.7168	0.3039 / 0.3021
	Bertseg	0.0776 / 0.0779	0.7752 / 0.7739	0.3518 / 0.3522
Gloss + example (ours)	Electra	0.2495 / 0.2495	0.5107 / 0.5095	0.3162 / 0.3213
	Bertmsa	0.3432 / 0.3425	0.7012 / 0.7022	0.4285 / 0.4356
	Bertseg	0.0805 / 0.0807	0.7657 / 0.7649	0.4512 / 0.4531
Baseline CamelBERT	Electra	0.1458 / 0.2459	0.7368 / 0.5065	0.0084 / 0.0334
	Bertmsa	0.2195 / 0.2195	0.8185 / 0.8185	0.0109 / 0.0110
	Bertseg	0.0555 / 0.0831	0.8436 / 0.7556	0.0126 / 0.0334
Baseline MARBERT	Electra	0.2436 / 0.2444	0.5132 / 0.5104	0.0335 / 0.0334
	Bertmsa	0.3495 / 0.3473	0.6949 / 0.6970	0.0335 / 0.0334
	Bertseg	0.0818 / 0.0816	0.7604 / 0.7610	0.0335 / 0.0334

Table 4: Results on the Dev/Test sets

AraT5 V2 has a similar vocabulary to the models which produced these embeddings.

While incorporating contextual examples did not improve the results on this task, we believe this is regarded to several reasons including the model single-layer architecture and shared task targets, which were obtained from the definitions only, causing the model to over-fit on the retrieved examples. Another reason may be attributed to the quality of the examples, which may not always provide meaningful context for the word:gloss pairs.

The model's performance could be enhanced through hyper parameter tuning, exploring advanced architectures, and potentially training some layers from the base encoder model.

5 Conclusion

Although not exceeding the baseline on all metrics, the initial results show promise for our approach. Further development could lead to exceeding baseline performance across all metrics. AraT5 V2 generally performed well on the given RD task, and further experimentation could unveil more of it potential.

Future work could involve several enhancements to the current model. One avenue for improvement is refining the data enrichment process, potentially by incorporating diverse resources beyond Wikipedia to enhance the model's knowledge base and adaptability. Additionally, exploring more sophisticated architectures and fine-tuning hyperparameters could further optimize the model's performance and accuracy. Finally, a promising direction would be to train the model's encoder-decoder architecture to directly predict word sequences instead of relying on target embeddings, potentially improving the generation of coherent and contextually relevant text.

References

Rawan Al-Matham, Waad Alshammari, Abdulrahman AlOsaimy, Sarah Alhumoud, Asma Wazrah, Afrah Altamimi, Halah Alharbi, and Abdullah Alaifi. 2023. Ksaa-rd shared task: Arabic reverse dictionary.

pages 450–460. Association for Computational Linguistics.

Waad Alshammari, Amal Almazrua, Asma Al Wazrah, Rawan Almatham, Muneera Alhoshan, Abdulrahman AlOsaimy, and Alfaifi Abdullah Altamimi, Afrah and. 2024. KSAA-CAD: Contemporary Arabic dictionary shared task. In *Proceedings of the 2nd Arabic Natural Language Processing Conference (Arabic-NLP), Part of the ACL 2024*. Association for Computational Linguistics.

Alfonso Ardoiz, Miguel Ortega-Martin, Óscar Garcia-Sierra, Jorge Álvarez, Ignacio Arranz, and Adrián Alonso. 2022. Mmg at semeval-2022 task 1: A reverse dictionary approach based on a review of the dataset from a lexicographic perspective. pages 68–74. Association for Computational Linguistics.

Pinzhen Chen and Zheng Zhao. 2022. A unified model for reverse dictionary and definition modelling. pages 8–13. Association for Computational Linguistics.

Ahmed Elbakry, Mohamed Gabr, Muhammad El-Nokrashy, and Badr AlKhamissi. 2023. Rosetta stone at ksaa-rd shared task: A hop from language modeling to word–definition alignment. pages 477–482. Association for Computational Linguistics.

Ehsan Kamalloo, Xinyu Zhang, Odunayo Ogundepo, Nandan Thakur, David Alfonso-Hermelo, Mehdi Rezagholizadeh, and Jimmy Lin. 2023. Evaluating embedding apis for information retrieval.

Anne Stallman Kilian, William E. Nagy, P. David Pearson, Richard C. Anderson, and Georgia Earnest Garcia. 1995. Learning vocabulary from context: effects of focusing attention on individual words during reading.

Sunil B Mane, Harshal Navneet Patil, Kanhaiya Balaji Madaswar, and Pranav Nitin Sadavarte. 2022. Wordalchemy: A transformer-based reverse dictionary. pages 1–5.

El Moatez Billah Nagoudi, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2021. Arat5: Text-to-text transformers for arabic language generation.

Abdelrahim Qaddoumi. 2023. Abed at ksaa-rd shared task: Enhancing arabic word embedding with modified bert multilingual. pages 472–476. Association for Computational Linguistics.

Fanchao Qi, Lei Zhang, Yanhui Yang, Zhiyuan Liu, and Maosong Sun. 2020. Wantwords: An open-source online reverse dictionary system. pages 175–181. Association for Computational Linguistics.

Serry Sibaee, Samar Ahmad, Ibrahim Khurfan, Vian Sabeeh, Ahmed Bahaaulddin, Hanan Belhaj, and Abdullah Alharbi. 2023. Qamosy at arabic reverse dictionary shared task: Semi decoder architecture for reverse dictionary with sbert encoder. pages 467–471. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. pages 483–498. Association for Computational Linguistics.