Cher at KSAA-CAD 2024: Compressing Words and Definitions into the Same Space for Arabic Reverse Dictionary

Pinzhen Chen[1] **Zheng Zhao**[1] **Shun Shao**[2]
[1]University of Edinburgh, [2]University of Cambridge
pinzhen.chen@ed.ac.uk

Abstract

We present Team Cher's submission to the ArabicNLP 2024 KSAA-CAD shared task on the reverse dictionary for Arabic—the retrieval of words using definitions as a query. Our approach is based on a multi-task learning framework that jointly learns reverse dictionary, definition generation, and reconstruction tasks. This work explores different tokenization strategies and compares retrieval performance for each embedding architecture. Evaluation using the KSAA-CAD benchmark demonstrates the effectiveness of our multi-task approach and provides insights into the reverse dictionary task for Arabic. It is worth highlighting that we achieve strong performance without using any external resources in addition to the provided training data.

1 Introduction

Reverse dictionary represents a unique and valuable technique enabling users to search for a word based on meaning or concepts (Zock and Bilac, 2004; Hill et al., 2016). This is an inverse process to the conventional use of dictionaries: looking up definitions given a word. Systems with the reverse dictionary capability hold significant potential for applications like writing assistance, information retrieval, and language education.[1] However, despite the growing interest in natural language processing (NLP) for the Arabic language, reverse dictionary capabilities remain relatively under-explored compared to their English counterparts.

The KSAA-CAD 2024 shared task (Alshammari et al., 2024), rooted from KSAA 2023 (Al-Matham et al., 2023) and CODWOE 2022 (Mickus et al., 2022) shared tasks, aims to advance research on Arabic semantics in two key areas: reverse dictionary and word sense disambiguation. We, Team Cher, participate in the reverse dictionary track, focusing on developing systems that can accurately convert human-readable definitions (glosses) to their corresponding words' embeddings. A highlight of our system is that it does not need to rely on external language resources, which are often modest in size for under-served languages. To aid reproducibility, our code is publicly available.[2]

Our approach reflects the philosophy of encoding information in different forms into the same representation space, a paradigm explored in representation learning for different modalities and languages (Ngiam et al., 2011; Chandar AP et al., 2014; Schwenk and Douze, 2017, inter alia). These approaches integrate various inputs via a single bottleneck representation layer and then reconstruct or generate suitable outputs, which is adopted by Bosc and Vincent (2018) to learn definition autoencoding in the field of word-definition modelling. This paper builds on our earlier research on word-definition semantic modelling (Chen and Zhao, 2022a,b), a multi-tasking network that learns reverse dictionary and definition generation as well as reconstruction tasks simultaneously.

Having experimented with static embeddings such as character-level autoencoding (`char`), skip-gram negative sampling (`sgns`, one variant of word2vec), and dynamic embeddings like `electra` (Mickus et al., 2022), we take this opportunity to extend our method to two new Arabic embedding sub-tracks: `bertmsa` from CAMeLBERT (Inoue et al., 2021) and `bertseg` from AraBERT (Antoun et al., 2020). We describe our methodologies and findings in developing an Arabic reverse dictionary system, hoping to contribute to the advancement of computational linguistics research for the

"Cher" means "beloved" or "precious" in French, reflecting our cherished bond with languages. It also echoes the Chinese idiom "三人行，必有我师" (in three companions, one is my teacher), implying that there is always something to learn from those around us.

[1]For example, https://www.onelook.com/thesaurus/

[2]https://github.com/PinzhenChen/unifiedRevdicDefmod

Proceedings of The Second Arabic Natural Language Processing Conference, pages 686–691

Arabic language. Further, all our experiments are data-constrained—without using extra resources like pre-trained embeddings—making our method language-agnostic and generalizable.

2 Task Setup

We provide a description of the dictionary and embedding data as well as the evaluation protocol provided by the shared task organisers in this section. The details can also be found available online.[3]

2.1 Dataset

The whole dataset is provided by the task organizers, comprising 31,372 training instances, 3,921 development instances, and 3,922 test instances, which builds on the previous year's edition. It consists of two core components: dictionary data and word embedding vectors. Three pre-trained language models are used to produce contextualized word embeddings from words taken from the dictionary data.

Dictionary data The dictionary data is derived from dictionaries of Contemporary Arabic Language (Omar et al., 2008; Namly, 2015). The selected dictionaries are based on lemmas rather than roots. Each data instance contains a word (lemma), the definition (gloss), and its part of speech (POS) derived from the dictionary data.

Embedding data The dataset comes with three types of contextualized word embeddings: AraELECTRA (Antoun et al., 2021), AraBERTv2 (Antoun et al., 2020), and CAMeLBERT-MSA (Inoue et al., 2021), referred to as `electra`, `bertseg`, and `bertmsa`, respectively. AraELECTRA has been pre-trained following the ELECTRA methodology (Clark et al., 2020), which trains a discriminator model upon the recovered masked tokens and substituted tokens. AraBERTv2 and CAMeLBERT-MSA are Arabic language models based on the BERT architecture (Devlin et al., 2019). Both models have been trained on contemporary Arabic data (El-khair, 2016) and the Arabic language used in 24 countries (Zeroual et al., 2019). Specifically, AraBERTv2 adopts the Farasa segmentation (Darwish and Mubarak, 2016).

We show an example from the development set for visualisation in Figure 1. It is worth noting that our systems are trained in a data-constrained condition. We only make use of the embeddings


```
{
"id": "ar.962714",
"word": "أَكْمَدَ",
"pos": "V",
"gloss": "غم وأمرض القلب",
"bertmsa": [0.22657, -0.32251, ...],
"bertseg": [0.13184, -0.20344, ...],
"electra": [0.22657, -0.32251, ...],
}
```


Figure 1: A sample data instance for visualisation.

and definition texts in the provided training set *solely*, without using explicit words or any external resources like text data or pre-trained language models. In other words, our models ingest only two fields as shown in Figure 1: `gloss` and an embedding, either `bertmsa` or `bertmsa` depending on the sub-track.

2.2 Evaluation metrics

This shared task uses the same metrics as CODWOE 2022 (Mickus et al., 2022). Reverse dictionary performance is evaluated by three metrics:

- Mean squared error (MSE) between reference and generated embeddings;
- Cosine similarity between reference and generated embeddings;
- Ranking score: percentage of test instances where the generated embedding has a higher cosine similarity with the reference than with other test instances.

The ranking of each participant's systems is carried out in a hierarchical manner. The main metric is the ranking score, which measures how accurately a model can rank its predictions compared to the ground truth. If multiple models perform similarly on the ranking score, MSE is then used as a secondary metric to further break ties. In cases where additional distinguishing factors are required, the cosine similarity serves as a tertiary metric. This multi-tiered approach aims to identify the model that delivers the best overall performance across various aspects of the task.

3 Methodology

3.1 Model architecture

Word-definition alignment Conventionally, reverse dictionary is approached using an encoder

[3] https://arai.ksaa.gov.sa/sharedTask2024/

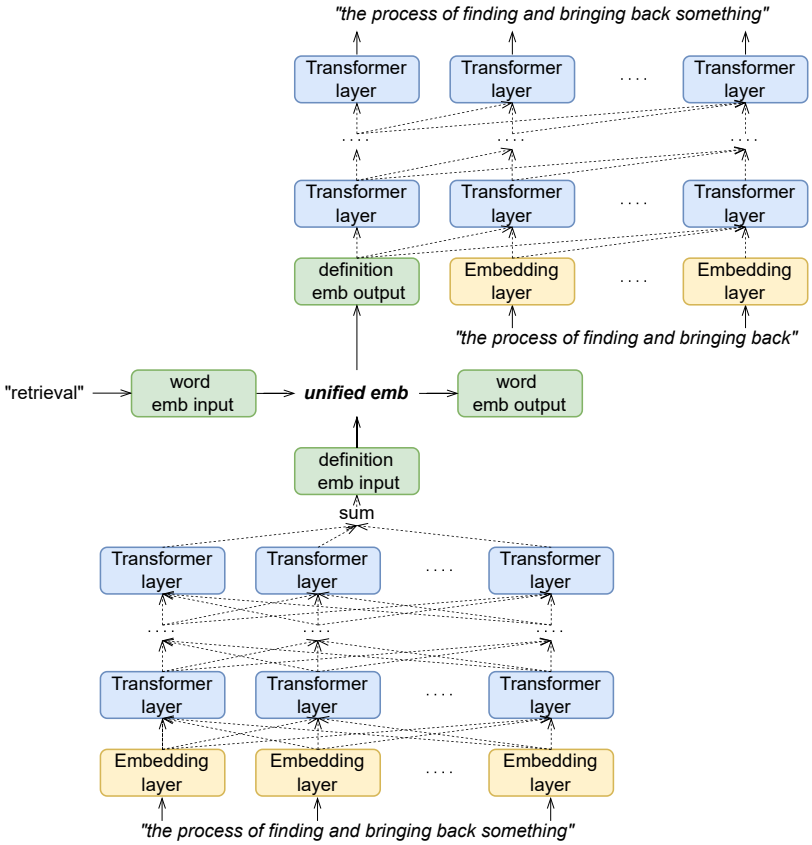


Figure 2: A detailed illustration of our model.

model which takes a sentence and returns a word embedding for downstream word retrieval (Hill et al., 2016; Thorat and Choudhari, 2016). Echoing the previous work on embedding different inputs into the same representation space, we argue that a word and its definition can be encoded in the same fashion to facilitate better representation learning. We use a model from our previous research (Chen and Zhao, 2022b) which jointly learns both reverse dictionary and definition generation connected using a shared embedding space. Referring to Figure 2, depending on the input and output combination, our model can be trained in four directions—word-to-word, word-to-definition, definition-to-definition, and definition-to-word—together with a vector similarity objective on the shared space representation.

What's new In this submission, we make three technical changes compared to the original model (Chen and Zhao, 2022a). We switch the activation function in the Transformer blocks from ReLU to GELU (Hendrycks and Gimpel, 2023). We also set the dimension of the shared space to be half of the input word embedding size or hidden size. This gives an information "bottlenecking" effect between the input and output ends.

In terms of a modification special to the Arabic language, we try out three segmentation approaches: whitespace tokenizer as in the original work, AraBERTv2 tokenizer (Antoun et al., 2020) based on Farasa (Darwish and Mubarak, 2016), and CAMeLBERT tokenizer (Inoue et al., 2021). In our experiments, we use the three different tokenizers to segment the definitions and pair them with the provided embeddings regardless of the model they are derived from.

4 Experiments and Results

4.1 Technical details

We set the shared space representation to have a size that is half of the input word embedding size. We use 1e-4 learning rate, 0.3 dropout probability, 4 transformer layers each with 4 heads. We use MSE for embedding losses (definition-to-word, word-to-word, shared space similarity), whereas for token generation (word-to-definition, definition-to-definition), we cross entropy with a label smoothing of 0.1. All models are trained with a budget of

100 epochs, and we also use early stopping, where the training stops if the definition-to-word loss does not improve within 8 consecutive epochs.

Finally, we experiment with ensembling, which averages the embedding predictions from different models. In ensemble 1, we average three models trained with definitions tokenized differently by the three tokenizers. In ensemble 2, we trained models with and without the embedding autoencoding and definition reconstruction task.

4.2 Evaluation results

We present our system results on the official test set evaluated by the automated scoring system from the shared task organizers. It is worth noting that we do not have access to the gold labels. Table 1 presents numbers for our models in the `bertmsa` track and Table 2 presents `bertseg` results. We observe that using the AraBERT (Farasa) tokenizer yields the best scores across all three metrics. This finding is consistent across two embedding architectures, yet the gap between different tokenization strategies is small. Interestingly, ensembling models trained with different tokenizers did not further improve upon single models.

Tokenizer	bertmsa		
	MSE	Cosine	Rank
whitespace	.0810	.7637	.4910
AraBERT (Farasa)	**.0800**	**.7671**	**.4834**
CAMeLBERT	.0819	.7618	.4991
ensemble 1	.0807	.7648	.4929
ensemble 2	.0807	.7648	.4931

Table 1: Test results for the `bertmsa` sub-track.

Tokenizer	bertseg		
	MSE	Cosine	Rank
whitespace	.3454	.6997	**.4913**
AraBERT (Farasa)	**.3446**	**.7012**	**.4913**
CAMeLBERT	.3483	.6978	.5001
ensemble 1	.3457	.6998	.4952
ensemble 2	.3458	.6998	.4951

Table 2: Test results for the `bertmsa` sub-track.

4.3 Visualisation of the shared space

In Appendix A Figures 3 and 4, we visualize the representations of words and definitions in the shared hidden space for `bertseg` and `bertmsa`, respectively. Since our model features a multi-task unified representation learning, we present plots from epoch 0 to epoch 10 with an interval of 2, to understand the tendency of representations and movements in the space. We find that regardless of how the words are embedded originally, whether `bertseg` or `bertmsa`, the word and definition clusters initially maintain a large distance. Through the training process, the two clusters move closer, potentially because the multi-task training encourages a word and its respective definition to have similar hidden representations.

5 Conclusion

We have presented a study of Arabic reverse dictionary using a joint word-definition modelling approach. We concentrated on the investigation of tokenizers and representation learning in the shared space for words and definitions. We hope our modelling approach can contribute to research on word and sentence semantics.

Ethical Considerations

We must note that none of the authors can speak Arabic, so we used Google Translate to inspect data. Some processing or modelling choices might not be optimal.

Acknowledgments

This work received support from UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding guarantee [grant number 10052546, HPLT] and the UKRI Centre for Doctoral Training in Natural Language Processing [EP/S022481/1].

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. In *Proceedings of ArabicNLP 2023*.

Waad Alshammari, Amal Almazrua, Asma Al Wazrah, Rawan Almatham, Muneera Alhoshan, Abdulrahman AlOsaimy, Afrah Altamimi, and Abdullah Alfaifi. 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.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021. AraELECTRA: Pre-training text discriminators for Arabic language understanding. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*.

Tom Bosc and Pascal Vincent. 2018. Auto-encoding dictionary definitions into consistent word embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Sarath Chandar AP, Stanislas Lauly, Hugo Larochelle, Mitesh Khapra, Balaraman Ravindran, Vikas C Raykar, and Amrita Saha. 2014. An autoencoder approach to learning bilingual word representations. *Advances in neural information processing systems*, 27.

Pinzhen Chen and Zheng Zhao. 2022a. Edinburgh at SemEval-2022 task 1: Jointly fishing for word embeddings and definitions. In *Proceedings of the 16th International Workshop on Semantic Evaluation*.

Pinzhen Chen and Zheng Zhao. 2022b. A unified model for reverse dictionary and definition modelling. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Kareem Darwish and Hamdy Mubarak. 2016. Farasa: A new fast and accurate Arabic word segmenter. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Ibrahim Abu El-khair. 2016. 1.5 billion words Arabic Corpus. *arXiv preprint*.

Dan Hendrycks and Kevin Gimpel. 2023. Gaussian error linear units (GELUs). *arXiv preprint*.

Felix Hill, Kyunghyun Cho, Anna Korhonen, and Yoshua Bengio. 2016. Learning to understand phrases by embedding the dictionary. *Transactions of the Association for Computational Linguistics*, 4:17–30.

Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in Arabic pre-trained language models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*.

Timothee Mickus, Kees Van Deemter, Mathieu Constant, and Denis Paperno. 2022. Semeval-2022 task 1: CODWOE – comparing dictionaries and word embeddings. In *Proceedings of the 16th International Workshop on Semantic Evaluation*.

Driss Namly. 2015. "Al wassit" LMF Arabic dictionary.

Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y. Ng. 2011. Multimodal deep learning. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*.

Ahmed Mukhtar Omar et al. 2008. Dictionary of contemporary arabic language. *With a help of Al-nnas team. Beirut, Lebanon: The world of books*, pages 75–6.

Holger Schwenk and Matthijs Douze. 2017. Learning joint multilingual sentence representations with neural machine translation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*.

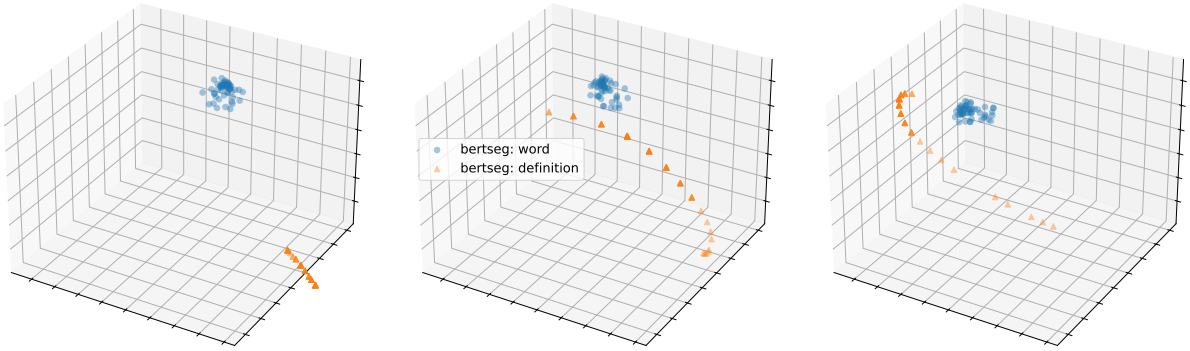
Sushrut Thorat and Varad Choudhari. 2016. Implementing a reverse dictionary, based on word definitions, using a node-graph architecture. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. The COLING 2016 Organizing Committee.

Imad Zeroual, Dirk Goldhahn, Thomas Eckart, and Abdelhak Lakhouaja. 2019. OSIAN: Open source international Arabic news corpus - preparation and integration into the CLARIN-infrastructure. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*.

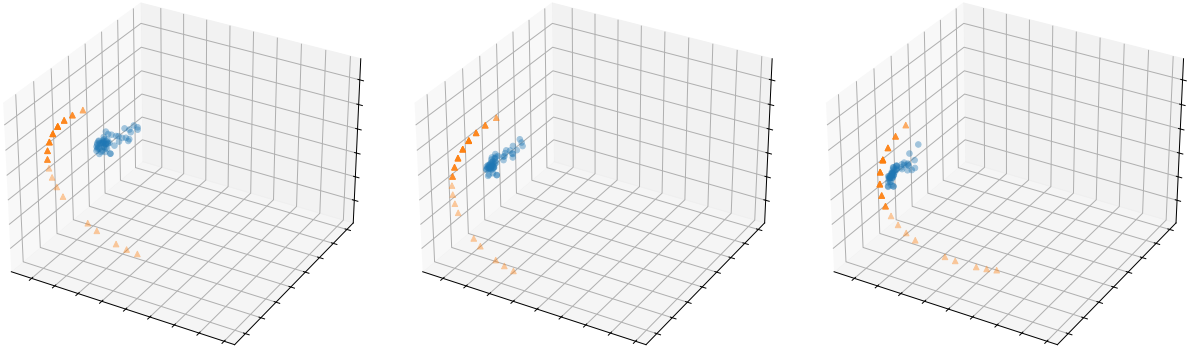
Michael Zock and Slaven Bilac. 2004. Word lookup on the basis of associations : from an idea to a roadmap. In *Proceedings of the Workshop on Enhancing and Using Electronic Dictionaries*.

A Visualisation of the shared space

The visualisations are presented in Figure 3 and Figure 4.

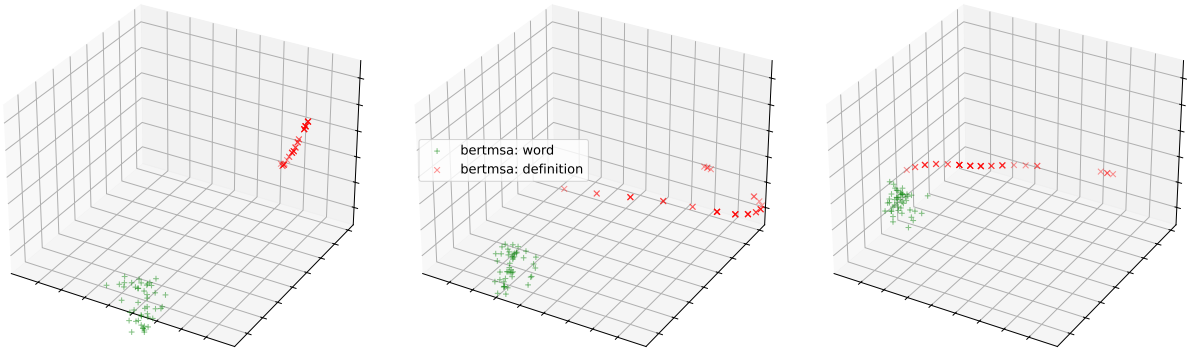


(a) epoch 0 (b) epoch 2 (c) epoch 4

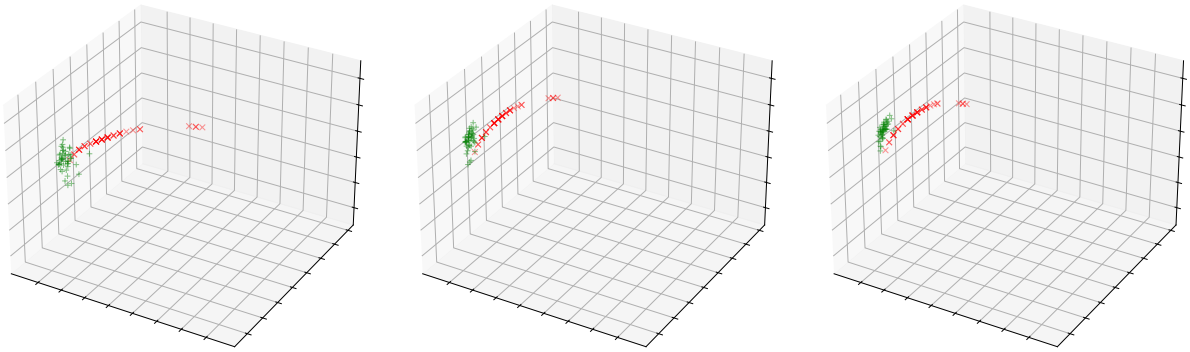


(d) epoch 6 (e) epoch 8 (f) epoch 10

Figure 3: Visualisation of word and definition representations in the *shared space* for the `bertseg` track.



(a) epoch 0 (b) epoch 2 (c) epoch 4



(d) epoch 6 (e) epoch 8 (f) epoch 10

Figure 4: Visualisation of word and definition representations in the *shared space* for the `bertmsa` track.