Can Dual Encoders do it all?

Amrita Nair
Universität des Saarlandes

Abstract

Dual encoders have been used in a variety of applications like question answering tasks, information retrieval/extraction, entity linking and so on. Dual encoders seem like a robust architecture but as with most systems, they have advantages, disadvantages and shortcomings. This report attempts to provide a glimpse at the potential and different applications of dual encoders while keeping in mind their pitfalls too. This report is inspired by Luan et al. (2021) and their use of dual encoders as first stage retrievers.

1 Introduction

The dual encoder architecture has been used across the field of information retrieval. They have been used as retrievers, (Luan et al., 2021), as re-rankers, (Menon et al., 2022), for entity linking, (Gillick et al., 2019), (Wu et al., 2019), for question-answering tasks (Dong et al., 2022), (Wang et al., 2022) and so on. In the last couple of years there has been an increasing body of research in leveraging the advantages such as the ease of working with large-scale information retrieval (Luan et al., 2021) as well as the disadvantages like the limitation of fixed length encodings in retrieving large documents (Luan et al., 2021). This report attempts to provide an introduction to the potential and applications of dual encoders.

2 What are Dual Encoders?

Though the idea and concept of an entity similar to dual encoders has existed for a couple of years under the term Siamese networks, Bromley et al. (1993), the term 'Dual Encoders' was first formally defined by Gillick et al. (2018).

Siamese networks are models where two entities are encoded by two copies of the same network. A more elaborate description of Siamese Networks is provided by Chicco (2020) where they describe the network as a combination of two identical feed forward neural networks which each generate an output that is compared using a similarity metric like cosine similarity to predict whether the two entities are similar or not. The architecture provided in figure 1 is referenced from Chicco (2020).

It shows two feed forward neural networks, whose inputs are two entities that are to be compared. The similarity measure most used is cosine similarity.

Siamese networks have been generally applied in situations where a comparison has to be made, for example, in image analysis to recognise fingerprints from images by Baldi and Chauvin 1993, for face verification by Chopra et al. and so on.

Dual encoders, as defined by Gillick et al. (2018) are models in which a pair of items are encoded in a shared space. During training, candidate items are encoded by the candidate encoder to a d-dimensional(potentially lower dimensional space) and at inference, a query encoder encodes the query to a vector space after which candidate items are determined using similarity metrics or nearest distance measures like approximate nearest neighbor search.

According to Dong et al. (2022), dual encoders are preferred due to the fact that the embedding index of the encoders can grow dynamically with new information which is not the case with generative networks which need to be retrained with new data. Additionally, this makes them easy to productionize.

Dual encoders can have different types of architectures, as mentioned in Dong et al. (2022). They can be broadly categorised under Siamese Dual Encoders(SDE) and Asymmetric Dual Encoders(ADE).

In asymmetric dual encoders, two separate encoders are used to encode the two entities that are to be compared. These encoders may share some or no parameters. In the Dense Passage Re-

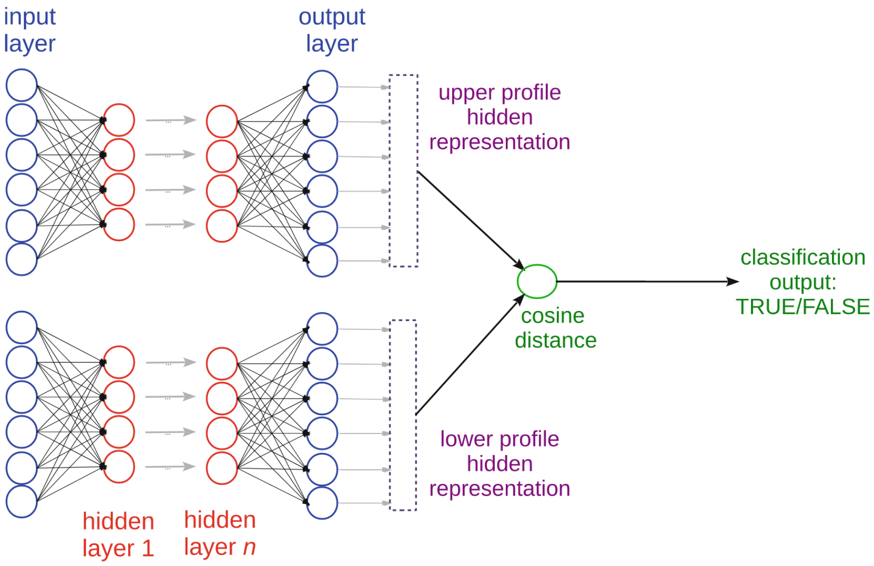

input layer
output layer
upper profile hidden representation
classification output: TRUE/FALSE
cosine distance
lower profile hidden representation
hidden layer 1
hidden layer *n*

Figure 1: Chicco (2020)

treiver(DPR) model introduced by Karpukhin et al. (2020), given a collection of passages, the aim is to retrieve top-k passages for a given query, which is a standard open domain question answering situation. At inference time, a different encoder is used to encode the query to a d-dimensional vector space, after which the top-k passages similar to the query are retrieved.

Similarly, Lee et al. (2020), also tackles the problem of open domain question answering, in terms of phrase retrieval. The architecture here too includes a separate phrase encoder and question encoder.

On the other hand, in siamese dual encoders, the parameters are shared between the two encoders. An example would be ST5 model introduced by Ni et al. (2021b) where the architecture consists of two shared weight transformer modules that are used to encode the inputs.

The prowess of dual encoders are not just in retrieval, but can also be seen in ranking. Cross-attention models, which learn a joint embedding for the query and the document do perform better for ranking but dual encoder models can be improved with a sufficiently large encoder size as is stated in Menon et al. (2022). The paper also shows empirically that the gap between the performance of dual encoders and cross-attention models is due to dual encoders over fitting to the training set.

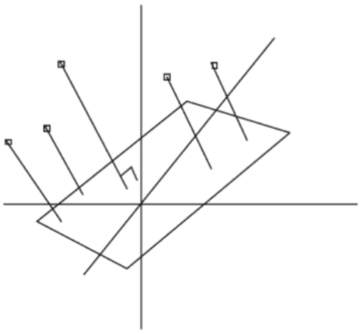


Figure 2: Vempala (2004)

2.1 Random Projection

Random projection, Vempala (2004) is a topic of interest in this report due to the fact that dual encoders compress the input to a lower dimensional space, for which, random projection can be used as is done in Luan et al. (2021).

The Johnson-Lindenstrauss lemma, Johnson (1984), Pyrcz (2019) state that *points in a high-dimensional space can be linearly embedded in a space of lower dimensionality in such a way that distances between the points are preserved.*

A consequence of the lemma is that projecting a space to a lower dimension does not depend on the initial number of parameters, it only depends on the number of data points, the original dimensionality and the acceptable error limit.

Hence, a random matrix which is of shape *m x p* where *m* is the original number of features and *p* is

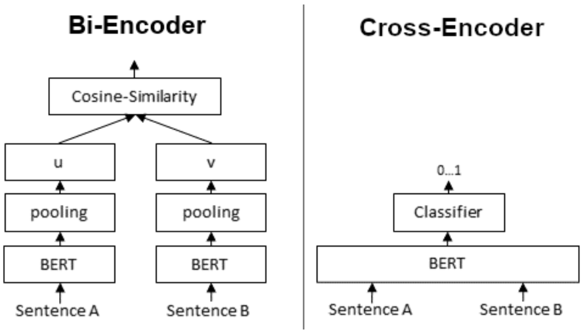


Figure 3: Reimers and Gurevych (2019)

the lower dimension by the original is multiplied by the *n x m* feature matrix and a scalar to give a feature matrix of lower dimensionality.

$$Y_{n \times p} = \frac{1}{\sqrt{p}} X_{n \times m} R_{m \times p} \tag{1}$$

The scaling factor above $(\frac{1}{\sqrt{p}})$ is chosen to account for the impact on pairwise distances of working in the lower dimensional space.

The random matrix can be filled with either **Gaussian entries** or **Rademacher entries**. The values for the former are drawn from the Gaussian distribution, Wikipedia (2023a) while for the latter, the values are drawn from the Rademacher distribution, Wikipedia (2023b). This process is known as random projection, loosely defined as *the process of mapping high-dimensional matrices to a lower-dimension by a random matrix.*

2.2 Cross-encoders

A succinct definition is provided by Reimers and Gurevych (2019). In a **cross-encoder**, the two entities to be compared are passed simultaneously to the transformer network. An output value between 0 and 1 to indicates the similarity within the pair. No sentence embedding is produced in this case.

In the literature, bi-encoders are mentioned now and again. It is unclear whether bi-encoders are the same as dual encoders given that the architecture seems similar too. Assuming they are, the difference between **a bi-encoder/dual-encoder** and a cross-encoder is that, for bi-encoders, the input is passed independently to the transformer. Hence, two sentence embeddings are produced. These sentence embeddings are then compared using cosine similarity.

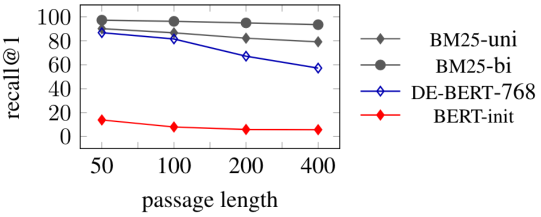


Figure 4: Luan et al. (2021)

3 Dual Encoders vs Sparse Retrievers

The traditional model of information retrieval involved spare retrievers that encoded the document into a sparse vector whose dimensions were the same as the length of the vocabulary, *v* of the corpus. The query vector would be mapped to the same dimension and models like TF-IDF would be used to choose the documents similar to the query. In contrast, dual encoders encode the documents to a dense representation **k**, where $k \ll v$

At first glance it seems that dual encoders should perform better than the sparse methods since they encode context but this is not the case as demonstrated by Luan et al. (2021). Figure 4 shows the recall@1 for a passage retrieval task. DE-BERT is a BERT based dual encoder, and the rest of the architectures are appropriately named. BM25 performs better than BERT and even the dual encoder when it comes to longer passages.

Sparse retrieval models perform better than dual encoders when it comes to **precise term overlap**. The capability of a model to detect precise term overlap has been termed as fidelity by Luan et al. (2021) and this term is said to be a tractable proxy of capacity. The paper further conducts a theoretical as well as an empirical investigation to confirm that there are some limitations in the capacity of of fixed length encodings to support retrieval of longer documents.

The theoretical investigation uses random projection, Vempala (2004) to compress the documents to a denser representation. Rademacher embeddings are used to fill the random matrix which raises the question as to why Gaussian embeddings weren't used exclusively since using Rademacher embeddings seems to be uncommon in the literature.

An interesting lemma provided in the paper upper bounds the pairwise error probability for a given k. The corollary provides a convenient view on the situation.

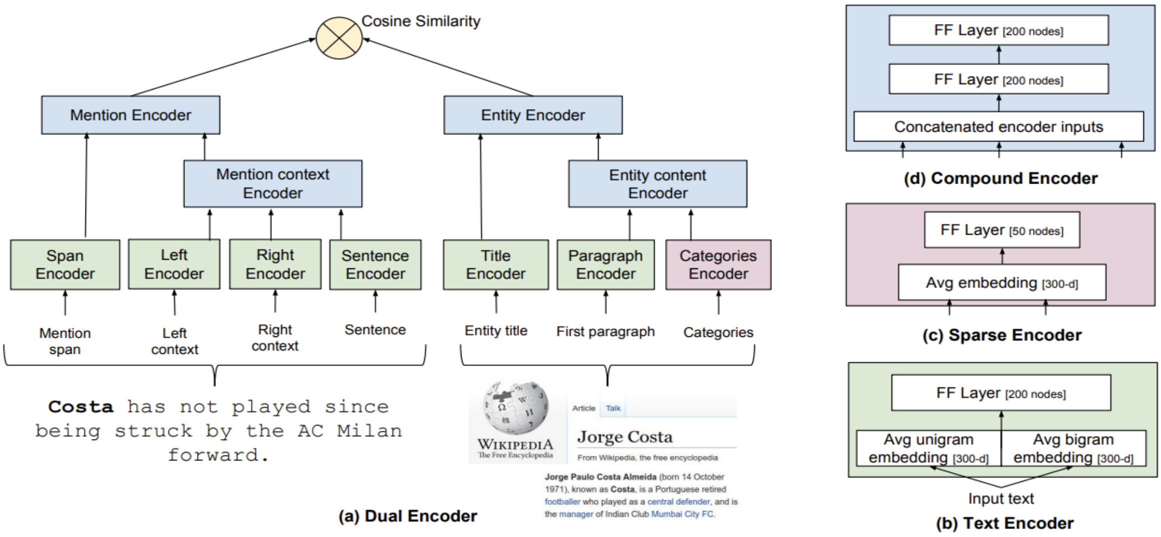


Figure 5: Gillick et al. (2019)

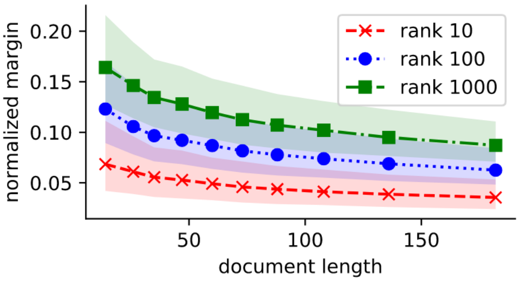


Figure 6: Luan et al. (2021)

Given vectors q(query), d_1(document 1), d_2(document 2), such that the normalized margin (geometrically, how much better is a document compared to its competitors), ϵ, is greater than zero. If A is a random matrix with Gaussian or Rademacher entries, such that if $k > 12\epsilon^{-2}\ln(\frac{4}{\beta})$, then

$$P(\langle Aq, Ad_1\rangle \leq \langle Aq, Ad_2\rangle) \leq \beta \qquad (2)$$

where $\langle Aq, Ad_i\rangle$ is the cosine similarity between q and d_i and β is the error probability that $\langle Aq, Ad_1\rangle \leq \langle Aq, Ad_2\rangle$

Another interesting result is the conclusion that, *the probability of returning d_1, which is the document that is the most similar to a query among all given documents, is bounded by a function of the embedding size(after random projection) k and normalized margin.*

A consequence of this result is that, to achieve recall@1, *for a given (q, d_1, D) triple, where D is the total set of documents, with probability $\geq 1-\beta$, the value of k should be set to*

$$k \geq \frac{2}{\frac{\epsilon^2}{2} - \frac{\epsilon^3}{3}} \ln \frac{4(|D| - r_0 + 1)}{\beta} \qquad (3)$$

where ϵ is the smallest normalized margin.

3.1 Compare normalized margins and document length

An empirical investigation was conducted using TF-IDF and BM-25 models. The TREC-CAR Dietz and Craswell dataset is separated by document length and for each query the normalized margins between the document with the best score and every other document in its group is calculated. The focus is on the 10th, 100th and 1000th smallest normalised margins. The results are presented in figure 6.

The figure shows that **normalized margins decrease, as the document length increases**.

Luan et al. (2021) also introduces a model named multi-vector encoding that can combine the dense representation feature of dual encoders which help computing semantic similarity with the ability to maintain fidelity with respect to sparse vector representation models. The introduction of this model was deemed necessary by the authors since sparse models are inadequate for detecting contexts and attentional architectures are impractical for large scale retrieval.

Overall, theoretical and empirical techniques were used to characterize the fidelity of fixed-length dual encoders, focusing on the role of document length. Based on these observations, hybrid models were proposed that yield strong performance while maintaining scalability.

The capability of dual encoders is not limited due to the fixed length encoding of the vectors. Ni et al. (2021a) suggest that the reason that dual encoders do not generalize to other domains for retrieval tasks is because of the bottleneck of the embedding size which can be fixed by increasing the size of the dual encoder model. This idea is followed with an empirical study that compares the Generalizable T5-based dense Retrievers (GTR) model introduced by the paper against the BEIR zero-shot retrieval benchmark, especially for out-of-domain generalization.

4 Dual Encoders for Entity Linking

The task of entity resolution consists of matching entities from a knowledge base to a "mention", which are certain spans of text in a document. Most architectures for solving this task involve a two-step approach. The first step retrieves candidate entities and the second step selects the most likely candidate.

Each entity would have "aliases" in the knowledge base which are just other possible ways of referring to the entity. However, these entity tables are not very efficient since they cannot store all aliases for every entity. At a certain point, there will be "cut-offs" which would leave out some aliases for an entity. Also, the problem of ambiguity remains. For a given mention, there can be uncertainty regarding the appropriate entity. The context of the mention gives a clue as to what the category of an ambiguous entity could be. Furthermore, for low-resource domains, it would be difficult to find/construct alias tables.

A better approach would be to map both the entity and mentions to a common vector space. This approach has been explored by Gillick et al. (2019). Figure 5 shows the architecture of a dual encoder model, where the mention side encoder combines information about the mention span and mention context. The entity side encoder combines the entity related information together too.

The architecture follows the typical dual encoder model where two networks are used to separately encode the entity and mentions. The authors mention that there is no interaction between the two networks so it can be assumed that this system follows an asymmetric dual encoder architecture.

The *compound encoder* mentioned in the architecture adds useful sub-structure to each network. The architecture shows a layer which concatenates encoder inputs and two feed-forward layers. The *text encoder* is used for text input while the *sparse encoder* is used for sparse ID input and all text encoders share a common set of embeddings. More information about the architecture can be found in section 4.1 of Gillick et al. (2019)

Cosine similarity is used to calculate similarity between the two representations. The retrieval results presented show a very high recall@100 for the model introduced by the paper(DEER) in comparison to the other models like BM25.

An alternative approach to the one suggested in Gillick et al. (2019) would be to use the retrieval space only to generate candidate entities and then re-rank them by using a cross-attention encoder over the target mention and each of the candidate entities. This method was suggested by Agarwal and Bikel (2020).

The reason provided by the authors of Agarwal and Bikel (2020) for using cross-attentional architectures to re-rank is simple. The dual encoder model is good for learning representations for both the entity and the mention in a vector space. However, to disambiguate between entities, further context/information on the entity and mention side is needed.

An example given in the paper illustrates this point perfectly. Consider that the mention in the text is *Asia Cup* and the candidate entities are `2018 Asia Cup` and `2016 Asia Cup`. Now, to disambiguate between these two entities, looking at the contexts of the mention and the entity would help. If the mention has a year in its context, it would be easy to identify the entity. However, if the year is not present, but the location is present, given that there is information on the location of the `2018 Asia Cup` and `2016 Asia Cup`, this situation too can be solved. A cross-attention model allows for the use of such detailed information/features about the mention and the entity. Paraphrasing from the paper, *""Cross-attention gives the opportunity to choose relevant context selectively depending in the specific mention and entity in question and the*

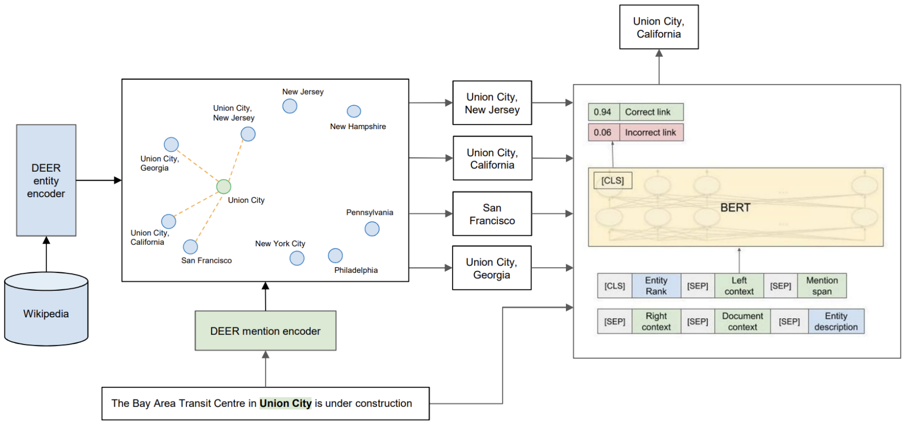
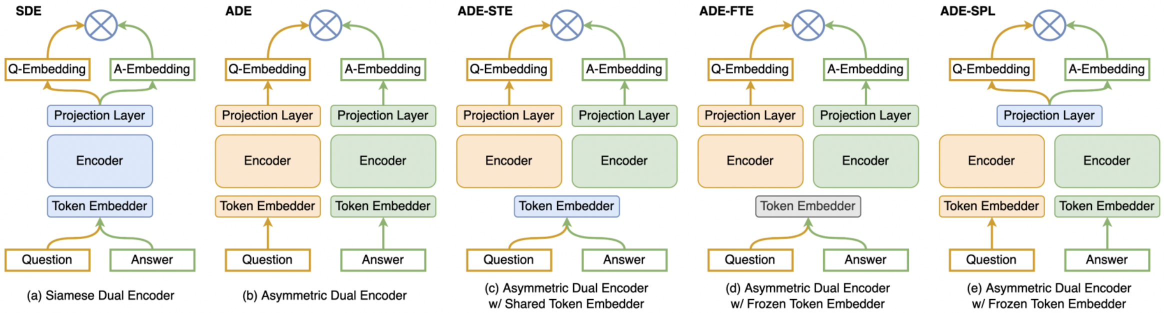


Figure 7: Agarwal and Bikel (2020)



(a) Siamese Dual Encoder

(b) Asymmetric Dual Encoder

(c) Asymmetric Dual Encoder w/ Shared Token Embedder

(d) Asymmetric Dual Encoder w/ Frozen Token Embedder

(e) Asymmetric Dual Encoder w/ Frozen Token Embedder

Figure 8: Dong et al. (2022)

available features.

The candidate generator follows the same architecture as that of the dual encoder model in Gillick et al. (2019). The entity encoder encodes the entities, the mention encoder encodes the mentions and the nearest neighbors are selected using cosine similarity. The 100 retrieved entities are then used for ranking.

The ranking problem is treated as a binary classification task. Using BERT as a cross-attention encoder, a representation of each mention-entity pair is classified as a true link or not. The final entity is chosen based on the classification probabilities, the entity with the highest probability is chosen as final linked entity. This architecture is pictorially represented in figure 7

Interestingly, Wu et al. (2019) propose something very similar. One of the differences in the approach is that for the retrieval step, Wu et al. (2019) use approximate nearest neighbors search while Agarwal and Bikel (2020) use cosine similarity. Apart from that, most of the architecture seems very similar as both use a dual-encoder for first stage retrieval and a cross-encoder for re-ranking.

5 Dual Encoders for Question Answering

One of the most efficient uses of the dual encoder architecture is for question answering tasks. Various models of dual encoders have different strengths and weaknesses when it come to this task as is demonstrated by Dong et al. (2022).

Five variants of dual encoders are tested which are, Siamese Dual-Encoder(SDE), Asymmetric Dual-Encoder (ADE), ADE with shared token embedder (ADE-STE), ADE with frozen token embedder (ADE-FTE) and ADE with shared projection layer (ADE-SPL). The first two variants have been described in section 2. The remaining three are variants of ADE such that certain parts of the networks are shared. Figure 8 encapsulates the differences in the architectures of these five variants. Parameter sharing in different parts of the dual encoder architecture produce different variants. The orange and green components in Figure 8 are distinctly parameterised for question and answer encoder network respectively. The blue component is shared. Grey components are frozen.

The results of the experiments are interesting. SDE tend to perform better in comparison to the

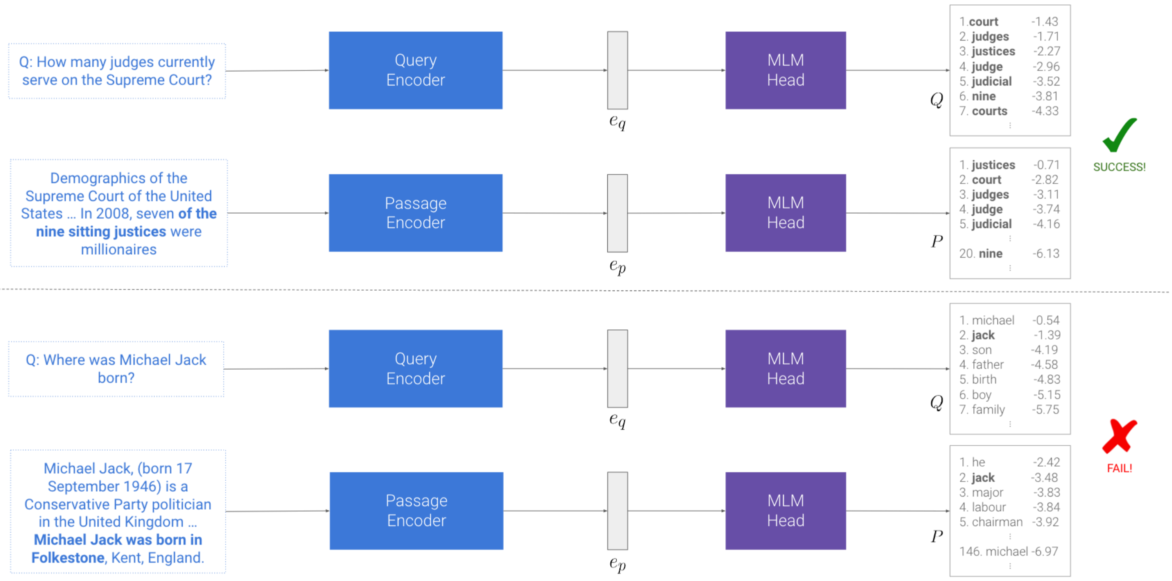


Figure 9: Ram et al. (2022)

ADE. The authors of Dong et al. (2022) speculate that the reason for this is the fact that different encoders used for question and answers map them to different parameter spaces that are not aligned.

Since the SDE share parameters, the embeddings for the question and answers are forced to be in the same vector space. This assumption is confirmed by conducting an analysis on the embeddings where the question and answer embeddings are generated, following which t-SNE, Van der Maaten and Hinton (2008) is used to project and cluster the embeddings to 2-dimensional space.

To improve the performance of the ADE, the other variants of dual encoders are constructed. Freezing and sharing token embedders bring minimal improvements for ADE's which suggest that token embedders might not be the best way to fix the gap between SDE and ADE. Another method used is by sharing projection layers which according to experiments performed by Dong et al. (2022) do improve the performance of asymmetric dual encoders.

6 How do Dual Encoders represent text?

Dual encoders perform surprisingly well for many tasks and surprisingly poor for others. The reasons for its success or failure are not very clear. To shed light on this, Ram et al. (2022) attempt to interpret the representations produced by the dual encoder. This can be done by projecting the representations to the vocabulary space by passing them to a Masked Language Model(MLM) head. These projections are found to be highly interpretable by humans.

The Query Encoder Enc_Q encodes the query, q to obtain its representation, e_q. Similarly, the Passage Encoder Enc_p, encodes the passage p to obtain its representation e_p. The MLM head is then applied to obtain the vocabulary projection,

$$Q = MLM - HEAD\ (\ e_q\ )$$
$$P = MLM - HEAD\ (\ e_p\ )$$

The MLM-HEAD can be defined as a function that takes $h \in R^d$ as input and returns a probability distribution P over the vocabulary V such that,

$$MLM - HEAD(h)[i] = \frac{exp(\mathbf{v_i}^\top g(h))}{\sum_{j \in \mathcal{V}} exp(\mathbf{v_i}^\top g(h))} \tag{4}$$

$g : R^d \rightarrow R^d$ is a function that adds non-linearity and $v_i \in R^d$ is the static embedding of the *ith* item in V.

Figure 9 demonstrates the method. Surprisingly, the query projections seem to contain words that imply that the model had implicitly performed *query expansion*, Rocchio Jr (1971) by "expanding" the terms the query includes which would help in finding a relevant passage. Similarly, the passage projections seem to almost anticipate the queries that could be asked for the passage. This is showcased in the first panel of figure 9. For the successful case, Q even includes the answer(nine) to the query(How many judges currently serve on the Supreme Court?).

This exercise also sheds light on why/how these models can fail. Dense retrievers tend to ignore some of the words in the passage; for example, in the failure case in figure 9, the word *Michael* is ranked low even when it seems to be an important token. The authors refer to this as *token amnesia*. In an attempt to overcome this, the authors suggest complimenting the dense representations with lexical information. The paper demonstrates that this method improves the performance of dual encoders on various retrieval tasks. Hence, the method of vocabulary projection can be used to 'detect' problems in how a dual encoder encodes the text, which can be used to improve the performance of the encoder on various tasks.

7 Conclusion

This report is an attempt to shed light on the different ways dual encoders can be used in the field of information retrieval. Various variants of the dual encoder architectures and their positives as well as negatives were presented. A short description of random projection is presented and the fact that projecting a dense matrix to a lower dimension does not depend on the original dimension is highlighted. Dual encoders are compared against sparse retrievers and it is mentioned that dual encoders are not as efficient in encoding longer documents compared to sparse models. Dual encoders can be used for entity linking tasks, various papers that use the architecture are summarised and presented. Similarly, the use of dual encoders for question answering tasks is also presented. A method of 'fixing' the drawback of the dual encoder architecture is highlighted which prompts further work using this architecture.

References

Oshin Agarwal and Daniel M Bikel. 2020. Entity linking via dual and cross-attention encoders. *arXiv preprint arXiv:2004.03555*.

Pierre Baldi and Yves Chauvin. 1993. Neural networks for fingerprint recognition. *Neural Computation*, 5(3):402–418.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6.

Davide Chicco. 2020. Siamese neural networks: An overview. In *Methods in Molecular Biology*, pages 73–94. Springer US.

S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. IEEE.

Laura Dietz and Nick Craswell. Trec complex answer retrieval overview.

Zhe Dong, Jianmo Ni, Daniel M. Bikel, Enrique Alfonseca, Yuan Wang, Chen Qu, and Imed Zitouni. 2022. Exploring dual encoder architectures for question answering.

Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning dense representations for entity retrieval.

Daniel Gillick, Alessandro Presta, and Gaurav Singh Tomar. 2018. End-to-end retrieval in continuous space.

William B Johnson. 1984. Extensions of lipschitz mappings into a hilbert space. *Contemp. Math.*, 26:189–206.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering.

Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. 2020. Learning dense representations of phrases at scale.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345.

Aditya Menon, Sadeep Jayasumana, Ankit Singh Rawat, Seungyeon Kim, Sashank Reddi, and Sanjiv Kumar. 2022. In defense of dual-encoders for neural ranking. In *International Conference on Machine Learning*, pages 15376–15400. PMLR.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. 2021a. Large dual encoders are generalizable retrievers.

Jianmo Ni, Gustavo Hernández Ábrego, Noah Constant, Ji Ma, Keith B. Hall, Daniel Cer, and Yinfei Yang. 2021b. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models.

Michael J. Pyrcz. 2019. Machine learning: Random projection.

Ori Ram, Liat Bezalel, Adi Zicher, Yonatan Belinkov, Jonathan Berant, and Amir Globerson. 2022. What are you token about? dense retrieval as distributions over the vocabulary. *arXiv preprint arXiv:2212.10380*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Joseph John Rocchio Jr. 1971. Relevance feedback in information retrieval. *The SMART retrieval system: experiments in automatic document processing*.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Santosh Vempala. 2004. The random projection method, volume 65 of dimacs series in discrete mathematics and theoretical computer science. *American Mathematical Society*.

Yanmeng Wang, Jun Bai, Ye Wang, Jianfei Zhang, Wenge Rong, Zongcheng Ji, Shaojun Wang, and Jing Xiao. 2022. Enhancing dual-encoders with question and answer cross-embeddings for answer retrieval.

Wikipedia. 2023a. Normal distribution — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Normal%20distribution&oldid=1141349215. [Online; accessed 04-March-2023].

Wikipedia. 2023b. Rademacher distribution — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Rademacher%20distribution&oldid=1068885318. [Online; accessed 04-March-2023].

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2019. Scalable zero-shot entity linking with dense entity retrieval.