EDM-BERT: Dynamic Wikipedia Representations in Knowledge Graph Enhanced BERT Entity Ranking

Susanne van de Logt
Radboud University
Nijmegen, The Netherlands
susanne.vandelogt@ru.nl

Luke van Leijenhorst
Radboud University
Nijmegen, The Netherlands
luke.vanleijenhorst@ru.nl

Tom Rust
Radboud University
Nijmegen, The Netherlands
tom.rust@ru.nl

ABSTRACT

Entity linking is used for natural language processing (NLP) tasks to match entities to a query. Modern techniques use neural information retrieval (IR) models to improve classical entity linking. In this paper, we extend a neural entity linking model, EM-BERT, by adding a dynamic representation of entities instead of a static representation, the introductory paragraph of the Wikipedia page matching the entity. We show that using the most relevant paragraph based on BM25 as entity representation (dynamic representation) did not improve the re-ranking compared to the original model.

ACM Reference Format:
Susanne van de Logt, Luke van Leijenhorst, and Tom Rust. 2023. EDM-BERT: Dynamic Wikipedia Representations in Knowledge Graph Enhanced BERT Entity Ranking. In *Proceedings of some conference (Conference acronym 'Bert and Ernie).* ACM, New York, NY, USA, 4 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Pretrained BERT-based models can capture much world knowledge and answer many tasks in information retrieval. However, these models do not perform well on non-popular terms and on complex queries. To improve these pretrained models, new models have been developed which are enriched with entity information contained within knowledge graphs (KG). This entity information is often taken from a Wikipedia KG and can contain the introductory paragraph of the Wikipedia page or simply the entire page. Both are considered static representations. It can also be the case that the entity information contains the paragraph most relevant for the specific query, which is a dynamic representation. In this project, we investigate, if we can improve the EM-BERT [2] model by changing the static entity information representation into a dynamic representation. We suspect that this new model would improve entity-oriented tasks, since BERT-ER showed that a dynamic representation can improve the precision of tasks using keyword queries [1]. In this paper, we thus investigate the following research question: *To what extent does enhancing EM-BERT with dynamic entity representations improve entity retrieval?*

2 RELATED WORK

BERT with static representation. It has been shown that BERT based models perform better on entity retrieval tasks than traditional retrieval models [2]. However, these models do not always perform well on sparse facts and complex queries for which knowledge graph information is required. To overcome these disadvantages, multiple models have been build to add knowledge about entities to the neural model. One of these models is E-BERT [4], which adds knowledge using Wikipedia2vec vector representations of entities. It transforms these vectors to the wordpiece space of BERT to make sure that both can be combined in the BERT model. This model is not yet applied on a retrieval task, a model that uses E-BERT and is applied on a retrieval task is EM-BERT [2]. EM-BERT ranks a list of entities for a given query using E-BERT and monoBERT. Entities in the query and passage matching an entity are annotated and used as input for the E-BERT part of the model. The other non-entity words are used as input for the monoBERT model. The graph embedding used in the model is Wikipedia2vec and as entity descriptions, the introductory paragraph of the Wikipedia page of that entity is used. This is a static representation of the entities. The paper showed that mainly the evaluation metrics of queries which included entities improved, as well as queries related to less popular entities.

BERT with dynamic representation. According to Chatterjee et al. [1], the introductory paragraph is not always the ideal representation for information retrieval tasks, since it does not always contain relevant information for the query. That is why this paper uses three different query-specific or dynamic representations:

(1) The top-level relevant paragraph of a Wikipedia page, also referred to as aspect.
(2) The highest pseudo-relevant candidate passage or PRF-passage.
(3) The paragraph containing an explanation of why the entity is relevant for the query or entity-support passage.

We focus on the first, which uses the most relevant paragraph based on BM25 and the context of the entity using entity-aspect linking. They showed that by using aspects instead of the first paragraph, the performance improves on tasks in which, given a keyword query, a ranked list of entities is returned.

3 METHOD

To investigate our research question, we extend the EM-BERT model created by Gerritse et al[2] to create EDM-BERT, see Figure 1. In EM-BERT, to add knowledge graph information to improve the model, Wikipedia2vec embeddings are used as knowledge graph embeddings. The embeddings are learned using a neural network

Conference acronym 'Bert and Ernie, June 03–05, 2023, Woodstock, NY
© 2023 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/XXXXXXX.XXXXXXX

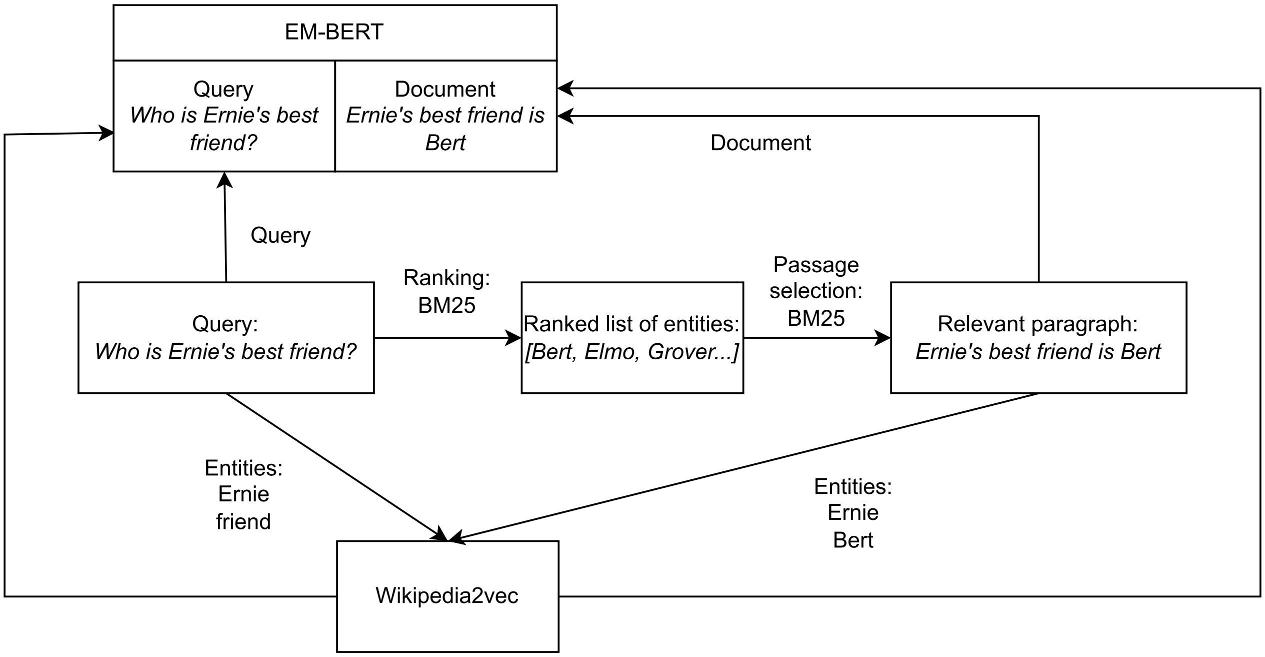

Figure 1: Representation of EDM-BERT

based on the following loss function:

$$L = L_w + L_e + L_a \tag{1}$$

In this formula L_w is based on the neighbouring words of an entity (word-based skip-gram), L_a is based on the neighbouring words of a hyperlink (anchor context) and L_e is based on the neighbouring entities of each entity in the Wikipedia knowledge graph (link graph). The embeddings have to be transformed to BERT vector space to be able to combine the entities with the monoBERT, which is done using the E-BERT model. Then both the entity vectors and wordpiece vectors are used as input to the monoBERT model which calculates a score for each query-document pair. To transform these entities, we annotated the entities with `ENTITY/`, while keeping a version which can be split into wordpieces to use for the BERT model, so for example the sample text

```
Aristotle was a Greek philosopher and
    polymath during the Classical period in
    Ancient Greece.
```

becomes

```
Aristotle ENTITY/Aristotle was a Greek
    ENTITY/Greek philosopher ENTITY/
    philosopher and polymath ENTITY/polymath
     during the Classical period ENTITY/
    Classical_period in Ancient Greece
    ENTITY/Ancient_Greece.
```

The other words that are not annotated are separated into word pieces and is used as input to 'regular' BERT. The entities are used as input for E-BERT and the vectors in the entity vector space are transformed to the BERT space. Then for the retrieval part, BM25 is used to rank the documents or in this case entities and monoBERT is used to rerank the top k documents. In the original EM-BERT implementation, the model only used the abstracts of the Wikipedia pages for the ranking of the entities. In our implementation, we experiment with two different approaches:

(1) Take the most relevant paragraph of a Wikipedia page for the given query instead of the abstract. We then use BM25 to figure out what paragraph of a Wikipedia page is the most relevant to the query by getting a BM25 score for each paragraph and selecting the maximum value.
(2) In our second approach we simply take the first 2000 characters of the Wikipedia page. We could not take the full page because training time would be too long, estimated time of 200 hours.

We then compare these approaches by looking at the Mean Reciprocal Rank (MRR), Normalized Discounted Cumulative Gain (NDCG), Recall and Precision at various ranks calculated on the DBpediaEntity v2 test collection [3].

Out of these metrics, we mainly focus on the NDCG[6]. This is the DCG normalized by the Ideal DCG, defined as:

$$\text{NDCG@}K(q) = \frac{\text{DCG@}K(q)}{\text{IDCG@}K(q)}. \tag{2}$$

With the DCG defined as:

$$\text{DCG@}K(q) = \sum_{i=1}^{\min(K,D_q)} \frac{r(q,i)}{\log_2(1+i)}. \tag{3}$$

Where $r(q,i)$ is the label for the document ranked at position i and D_q the number of ranked documents for query q.

4 EXPERIMENTAL SETUP

To answer our research question, we compare the EM-BERT model with static entity representation with the same model with dynamic entity representation and our additional static representation of taking the first 2000 characters. To do this, we use a large Wikipedia dump of 50Gb[1], released October 2015, out of which we retrieve the full Wikipedia articles. From these Wikipedia articles, we also retrieve the best paragraph for a given query by using BM25 to calculate a score for every paragraph and taking the maximum value. We use the MS MARCO passage data set for our first stage fine-tuning and then we use the DBpedia-Entity v2 collection for our second stage fine-tuning.

First, we parse the XML file of the Wikipedia dump to create an index we can use for the EM-BERT model[2]. To do this, we clean up the file and annotate the entities using regular expressions. We did not use the REL entity linker due to time constraints. Instead, we opted to solely take the Wikipedia references to other entities. After parsing the file, we create an index of Wikipedia titles and their entire page text. Then we separate the page text into paragraphs and rank these paragraphs using BM25 based on the current query. To use this new model with dynamic representation, we use the pygaggle file `evaluate_document_ranker.py` where we made some changes to how the index is loaded. In EDM-BERT, this is either by taking the first 2000 characters of the text (static but longer representation) or splitting the text into paragraphs and using BM25 to calculate the most relevant paragraph for the given query (dynamic representation).

As baseline, we use EM-BERT with the first paragraph of our Wikipedia dump as entity representation. This is the most similar to what is used in the original EM-BERT paper.

To test the performance, we use the Normalized Discounted Cumulative Gain (NDCG) at ranks 10 and 100. Along with the precision at rank 1, the recall at rank 3, 50 and a 1000 and finally the Mean Reciprocal Rank (MRR).

5 RESULTS AND DISCUSSION

In this section, we first evaluate our results of running the original EM-BERT model, the baseline, and then evaluate the differences with our model.

The resulting scores on the DBpedia-entity v2 collection can be found in Table 1 and Figure 2. This table shows that the NDCG@10 for running the original EM-BERT model is 0.531 and the NDCG@100 is 0.606. In the original paper [2] these scores were 0.461 and 0.495, respectively. We expected that these scores would be more similar, since we ran the same model on the same data set, short abstract of the Wikipedia dump of October 2015a. This difference may be caused by the fact that we may have run the model with different settings or files, but as far as we know, we ran the model identically to the paper. Another possibility is that our code may contain an error at some point. To improve this, the method could be extensively discussed with experts in the field of entity retrieval.

Table 1 also shows that our proposed model, EDM-BERT, performs worse on all metrics we tested. Looking at the precision and recall, this means that more irrelevant documents are ranked higher and less relevant documents are selected by EDM-BERT compared to EM-BERT. Also, the first relevant document is at a lower position for EDM-BERT on average and at higher positions we have less relevant documents. Only NDCG@100 is better compared to the results of the original paper, but it is difficult to compare these results since there was a large difference between the results of the original paper and our results for the static representation. There are a couple of reasons why we think that EM-BERT may have outperformed our model even though we expected the results the other way around. Firstly, we used identifiers, "[[]]", in the XML file representing links to other Wikipedia pages to determine which entities were present in the text. However, not all entities in the text may have been linked to other Wikipedia pages, since these have to be manually linked by Wikipedia editors. On the other hand, this method was also used for EM-BERT on only the first paragraph, which performs better than the original paper instead of worse. So it is unlikely that the worse performance is caused by the method of entity linking. Another option is that even though other paragraphs are more relevant for the query, they may not always contain the entity to answer the query, where the introductory paragraph may be more general, but does contain the right information.

To investigate further, why our model may have performed worse than the original EM-BERT model, we ran EM-BERT with the first paragraph of our annotated and cleaned XML file and with the first 2000 characters. The model using the first paragraph performs comparable to the original EM-BERT model. The NDCG metrics and Recall@50 are slightly better, while the Precision@1 and Recall@3 are slightly worse. This suggests that running our model with our index created from the full Wikipedia dump does work comparable to using the short abstracts directly, which was done in the paper. So the difference was not caused by a difference in the Wikipedia dump used. Running on the first 2000 characters resulted into worse metrics than the original EM-BERT and EM-BERT on the first paragraph. The idea of running this model was to see whether the full Wikipedia page performed better than a single paragraph. However, running the full page would take 200 hours on estimate, so we took the first 2000 characters. The disadvantage is that the paragraphs directly following the introductory paragraph may not be the most relevant paragraphs for the query, since the relevant information could be at the bottom of the page. This is why the results may be quite similar to the original EM-BERT and EM-BERT on first paragraph, but a bit worse since it contains a lot of redundant information as well.

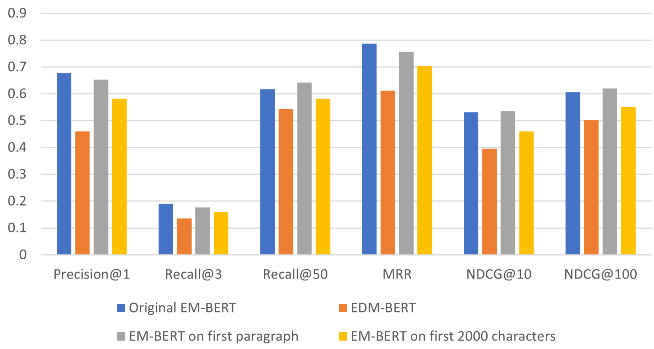

Figure 2: Evaluation metrics EM-BERT different EM-Bert Models

[1] Wikipedia dump http://downloads.dbpedia.org/wiki-archive/Downloads2015-10.html
[2] https://github.com/Lukevanl/EDM-BERT

Table 1: Comparison of EM-BERT and EDM-BERT

Model	Precision@1	Recall@3	Recall@50	MRR	NDCG@10	NDCG@100
Original EM-BERT (paper)	-	-	-	-	0.461	0.495
Original EM-BERT	0.677	0.191	0.617	0.787	0.531	0.606
EDM-BERT	0.460	0.136	0.543	0.612	0.396	0.502
EM-BERT on first paragraph	0.653	0.176	0.642	0.757	0.536	0.620
EM-BERT on first 2000 characters	0.582	0.160	0.581	0.704	0.460	0.551

6 FUTURE WORK

We investigated if our dynamic representation approach improves scoring of an entity retrieval task compared to a static representation. We found that our dynamic model did not perform better than the baseline. To further investigate why this is the case and how this new model could still be improved the following approaches can be taken. First of all, the model could be run multiple times and checked for consistency. In this research we only ran all models once, due to time constraints. Another important addition is significance testing to make sure that the current difference is indeed significant, which we expect given the large difference. The same model could be run using the entire Wikipedia page as a representation for an entity. This however makes EDM-BERT much slower, since it has to process a lot more data. To solve the problem of differing passage length for the entities, a cut-off value could be used to set a standard length for all passages. This length could be based on the average length of the introductory paragraph.

Future research extending this paper could include using other ranking methods. For now, we have used a simple BM25 model to find the paragraph that is the most relevant to our query, but we could also use other methods to rank these paragraphs. Other methods earlier used in entity search are GEER and BM25F-CA [2] [1]. One could also look at methods that rank paragraphs based on how similar the meaning of a paragraph is compared to a query instead of just the amount of terms that co-occur. This can be done by using techniques from the field of Semantic Textual Similarity (STS). Furthermore, we link entities using their Wikipedia references based on the annotation in the XML file. Alternatively, we could use a different entity linker, such as the Radboud Entity Linker [5]. This would improve the comparison with the original paper, since EM-BERT originally used REL. Another improvement would be to focus more on the context of an entity by adding entity-aspect linking or using the entity-support passage, which was done in BERT-ER [1]. Furthermore, our dynamic representation of a document uses the paragraph most relevant to a query. However, it could be that the paragraph is relevant, but does not necessarily contain the answer to the query. We could extend our dynamic representation to still include part of the abstract to make sure we have some general information about the entity present. We could also return all paragraphs belonging to the entity to EM-BERT, instead of returning a single paragraph per entity. With this, we re-rank the paragraphs and would then get a ranked list of paragraphs, from which many may belong to the same entity. Next, we can merge these multiple paragraphs for a single entity back to a representation for a single entity. With this method and a suitable representation, we may find the entities that are well represented in most of the paragraphs.

7 CONCLUSION

All in all, we think that using dynamic representations may still have potential in the field of entity retrieval. However, expert knowledge and several additions, improvements and debugging should be added to our current implementation to get a better view of the potential of this method.

REFERENCES

[1] Shubham Chatterjee and Laura Dietz. 2022. BERT-ER: Query-Specific BERT Entity Representations for Entity Ranking. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Madrid, Spain) *(SIGIR '22)*. Association for Computing Machinery, New York, NY, USA, 1466–1477. https://doi.org/10.1145/3477495.3531944

[2] Emma J. Gerritse, Faegheh Hasibi, and Arjen P. de Vries. 2022. Entity-aware Transformers for Entity Search. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. https://doi.org/10.1145/3477495.3531971

[3] Faegheh Hasibi, Fedor Nikolaev, Chenyan Xiong, Krisztian Balog, Svein Erik Bratsberg, Alexander Kotov, and Jamie Callan. 2017. DBpedia-Entity V2: A Test Collection for Entity Search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '17)*. ACM, 1265–1268. https://doi.org/10.1145/3077136.3080751

[4] Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2020. E-BERT: Efficient-Yet-Effective Entity Embeddings for BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, 803–818. https://doi.org/10.18653/v1/2020.findings-emnlp.71

[5] Johannes M. van Hulst, Faegheh Hasibi, Koen Dercksen, Krisztian Balog, and Arjen P. de Vries. 2020. REL: An Entity Linker Standing on the Shoulders of Giants. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*. ACM.

[6] Yining Wang, Liwei Wang, Yuanzhi Li, Di He, and Tie-Yan Liu. 2013. A Theoretical Analysis of NDCG Type Ranking Measures. In *Proceedings of the 26th Annual Conference on Learning Theory (Proceedings of Machine Learning Research, Vol. 30)*, Shai Shalev-Shwartz and Ingo Steinwart (Eds.). PMLR, Princeton, NJ, USA, 25–54. https://proceedings.mlr.press/v30/Wang13.html