Heidelberg University
Institute of Computer Science
Summer Term 2023
Seminar: Modern Information Retrieval
Instructors: Prof. Dr. Michael Gertz
Nicolas Reuter
Ashish Chouhan
Jayson Salazar
John Ziegler

Seminar Paper

Dense Retrieval with Entity Views

Name:	Johannes Gabriel Sindlinger
Student ID:	3729339
Study Programme:	Master, Computer and Data Science (3rd semester)
Email:	johannes.sindlinger@stud.uni-heidelberg.de
Date of Submission:	02/08/2023

I, **Johannes Gabriel Sindlinger**, hereby certify that I have written the paper entitled **Dense Retrieval with Entity Views** in the seminar **Modern Information Retrieval** in the **Summer Term 2023** with **Prof. Dr. Michael Gertz** and **Nicolas Reuter**, **Ashish Chouhan**, **Jayson Salazar** and **John Ziegler** independently and only with the aids indicated within the work. I have clearly marked citations as well as the use of external sources, texts and aids according to the rules of scientific practice.

I am aware that I am not allowed to claim other people's texts and text passages as my own and that a violation of this basic rule of scientific work is considered an attempt to deceive and cheat, which will result in appropriate consequences. These consist of the grading of the examination performance with „not sufficient“ (5,0) as well as further measures if necessary.

Furthermore, I confirm that this work has not been presented in the same or similar form in any other seminar.

Heidelberg, 02/08/2023 ________________

Contents

1 Introduction

In the field of information retrieval, finding relevant information efficiently and effectively from large collections has been a challenge for a long time. This task involves finding documents or passages that are most relevant to a particular query or topic. Conventional methods of information retrieval often rely on sparse representations such as BM-25 (Robertson and Zaragoza [13]), which can miss subtle semantic connections and fail to fully capture the context of the information being searched for.

Dense retrieval, on the other hand, has emerged as another approach that attempts to overcome the limitations of sparse representations by using dense vector embeddings of documents and queries. These embeddings are usually generated using large language models such as BERT (Devlin et. al [5]) and allow to encode richer semantic information. Consequently, they provide a more comprehensive understanding of the underlying content.

For dense retrieval, two fundamentally different approaches have evolved: bi-encoder and cross-encoder. The bi-encoder approach involves creating distinct vector representations for queries and documents, which are then compared to determine relevance scores, while the cross-encoder approach directly measures the similarity between the query and document by taking both as input, potentially leading to more accurate results but with higher computational cost.

Apart from that, the field of information retrieval seems to evolve from a more classical approach of only ranking best matching documents of a related query to a 'rich search' (Balog [2]) ranking, which is often referred to 'Entity-oriented search', as Balog [2] describes. In this context, 'rich search' refers to the fact that answers to queries more often contain specific information about entities, facts or other structured data. In consequence, information about entities are quite relevant to solve this more sophisticated task.

However, large language models such as BERT tend to suffer from a problem that arises particularly in the context of querying documents: entities such as persons, locations, or organizations are not represented entirely within the models, as Heinzerling and Inui [6] were able to illustrate. Thus, large language models are likely to represent socially relevant entities that occur in frequent documents, but at the same time entities that are rather rare not.

As an example, the two authors refer to the training process of BERT model, which uses a masked language model approach. During training phase, specific words or tokens are taken from existing documents and are randomly masked. The model is then charged to predict the masked tokens based on the context of the surrounding words. France is therefore correctly predicted with a high probability for the example 'Paris is the capital of [MASK]' (Heinzerling and Inui, [6]), while prediction might be ambiguous for the rather rare entity Sesame Street in the case of 'Bert is a character on [MASK]' (Heinzerling and Inui, [6]). In particular, entities that were not present during the training process and only emerged afterwards cannot be captured within language models at all.

The outlined relevance of entities in the retrieval of relevant information and the issue that these entities are not or only partially captured by large language models were the major reasons for Tran and Yates [17] to work on the paper that is presented in this seminar report. In the following, the work of Tran and Yates will be presented in detail and critically examined. The elaborations will be structured as follows: First, in section 2 the work is put into the context of current research in the field of information retrieval. Then, in section 3, the ideas and methods used by Tran and Yates are presented. This is followed by a presentation of the results in section 4, before concluding with a critical assessment and personal opinion in section 5.

2 Related Work

Entities have been a subject of interest in the field of information retrieval since the emergence of knowledge repositories like Wikipedia. Research in this context can be divided into the time before and after the advent of large language models such as BERT [5].

2.1 Entities in Sparse Retrieval

Prior to the advent of large language models such as BERT [5], research focused on sparse retrieval methods like BM25, where entities from knowledge bases were used to enhance query understanding. Efforts were made to expand queries using Wikipedia descriptions of entities [22] and extract additional features like synonyms and relationships from linked knowledge bases [4]. Balog [2] provided a comprehensive meta-analysis of related research.

2.2 Entities in Dense Retrieval

While some research has already addressed the consideration of entities in sparse retrieval, the amount of work in the field of dense retrieval in this context is rather limited.

Some work focuses on integration of entities within interaction-based methods, treating textual and entity information as common inputs for ranking models. An example of this approach is the work of Xiong et al. [18, 20]: They developed a method to build ranking features by incorporating an attention mechanism of word embeddings and entity embeddings and could significantly outperform baselines for word-based and entity-based learning to rank systems.

Interaction-based methods such as these can be considered as extensions of the cross-encoder approach of dense retrieval and therefore potentially face the same issue of high computational complexity as described in section 1. All documents and query pairs must be processed at runtime, which can lead to significant time and resource overhead. This leads to a slow overall retrieval process and can be inefficient, especially with large data sets.

In the context of the broad field of natural language processing tasks, entities have also been considered in several research papers and have been shown to have a significant impact on its tasks. The most prominent example is the ERNIE model (Sun et al. [15]), which incorporates knowledge from both pre-training tasks and external knowledge graphs, enabling it to achieve better contextual understanding and knowledge integration in natural language processing tasks.

ERNIE builds upon BERT's methods, particularly the masked language model, and tailors them to a more context-sensitive learning approach. ERNIE employs a masking strategy during its learning procedure, where the model is required to predict not only single words or tokens, but also several consecutive words. These consecutive words originate from different subtasks within the ERNIE model, including basic-level masking, phrase-level masking, and entity-level masking. The basic-level masking follows the standard token masking approach used in BERT, while the phrase-level masking groups small sets of tokens together to form conceptual units in the language to learn. Of most relevance to this seminar report is the entity-level masking stage, where the model tries to accurately predict entities that can span multiple tokens. This leads to the model becoming highly sensitive towards entities, as demonstrated by Sun et al. in their work on ERNIE [15]. ERNIE can be fine-tuned for information retrieval tasks, offering valuable insights and serving as a reference model for Tran and Yates' work [17].

The existing concepts of dense retrieval, including interaction-based methods and models like ERNIE, do not fully explore entities independently from the underlying large language model. This represents the main novelty of Tran and Yates' work [17], which will be elaborated in the following sections.

3 Methodology

As described above, the main contribution of Tran and Yates in the discussed work is to consider entities independently of the underlying large language model. In order to do so, they combine embeddings of an arbitrary large language model with embeddings of the entities from the respective documents and queries. For the embeddings of entities, they introduce some kind of multiple views on entities: Different sets of entities within queries and documents represent different perspectives on the queries and documents. Depending which view a query displays, different documents or parts of documents might be relevant.

In contrast to the previously described methods in subsection 2.2, the separate embeddings of queries and documents are not inserted as components into a learned framework, but are merely merged into a joint vector space. As a result, a final embedding in the joint vector space is derived, which is used to calculate similarities between different vectors. Therefore, Tran and Yates, stick to the bi-encoder model (see section 1), which allows indexing of embeddings for documents and enables fast ranking computations via ANN search.

3.1 General Model

The proposed method of Tran and Yates incorporates embeddings of documents and entities independently. To achieve a joined vector space, they merge embeddings of both for each query and document. This general approach is illustrated in Figure 1: Since Tran and Yates follow the bi-encoder approach (see section 1), embeddings for queries and documents are created independently of each other using a pre-trained large language model.

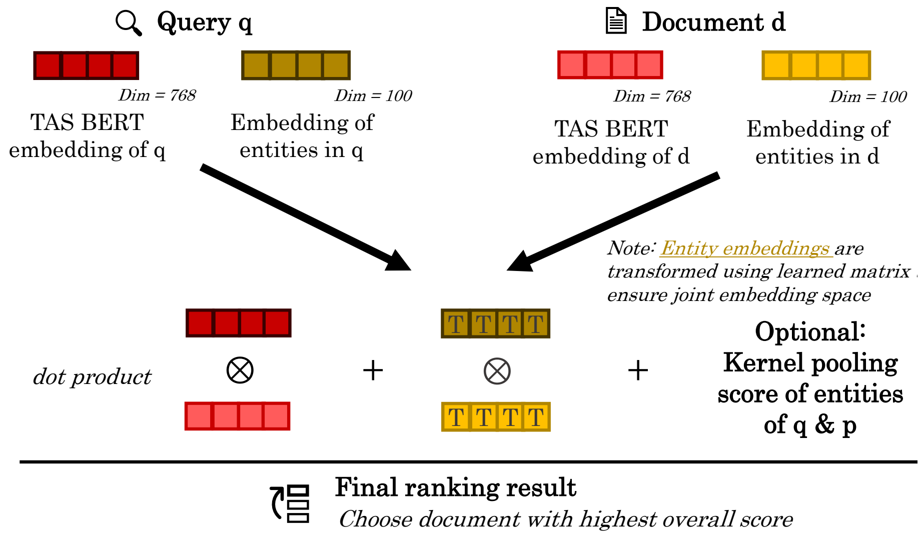

Query q
Document d
Dim = 768
Dim = 100
Dim = 768
Dim = 100
TAS BERT embedding of q
Embedding of entities in q
TAS BERT embedding of d
Embedding of entities in d
Note: Entity embeddings are transformed using learned matrix ensure joint embedding space
dot product
⊗
+
⊗
+
T T T T
T T T T
Optional: Kernel pooling score of entities of q & p
Final ranking result
Choose document with highest overall score

Figure 1: General model

In their work, Tran and Yates use a distilled version of BERT (Sanh et al. [14]), which was fine-tuned via TAS BERT approach by Hofstätter et al. [7] for information retrieval task. The distilled version of BERT is used by Tran and Yates with the objective of maintaining low computational costs while losing little effectiveness.

The term TAS BERT of the method of Hofstätter et al. refers to the term ’topic aware sampled BERT’, whereby sampling during training process of the information retrieval system is meant. In concrete, within TAS BERT approach the queries of the training dataset are clustered in advance based on different topics. During training process, query samples are extracted only from clusters of identical topics. This is intended to make the information retrieval model more sensitive to differing topics.

In a usual dense retrieval setting, the generated embeddings of tokens are then compared using similarity measurements. When a query is submitted to a system, the ranking of the best documents with respect to the query is carried out, based on the calculated similarity score between the duets of the given query and all documents.

For their model, Tran and Yates use the identical principle, but enhance it with embeddings of entities. In addition to the textual embeddings of a pre-trained language model, for each query and each document a single entity embedding is generated. Subsequently, the textual embedding is combined with the entity embedding in order to create an embedding that contains both the information about the semantic context and the information about the entities. The term combination here refers to concatenation of both vectors.

Experimental results of Tran and Yates showed that concatenation yields the best results for combining both embeddings of text and entities compared to others like max pooling and sum pooling. Table 1 displays the respective analysis of Tran and Yates based on their experimental setup, which is introduced in detail in section 4. Apart from the analytical point of view, the approach of concatenation offers the advantage that it can be understood quite intuitively.

Operators	**MS MARCO Dev**		
	nDCG	**MRR**	**MAP**
Sum	0.393	0.335	0.339
Max	0.388	0.330	0.334
Concat	0.396	0.341	0.343

Table 1: Varying Aggregation Operators of Embedding Concatenation

However, there is an issue to be solved in the context of concatenation: Since the vectors of the word embeddings and the entity embeddings are generated in a different vector space, this setup leads to biased similarity results. More precisely, the dimension of the word embeddings of the distilled TAS BERT model is 768 and of the word embeddings 100. Moreover, the magnitudes of the embeddings do not necessarily coincide. To overcome this problem, Tran and Yates introduce a transformation matrix $W \in \mathbb{R}^{100 \times 100}$ for the vectors of the embeddings. So let $\mathbf{E}(t)$ be the embedding of entities of text t, the transformed entity embedding is given as:

$$\mathbf{R}_{entity}(t) = W^T \cdot \mathbf{E}(t) \tag{1}$$

The values of the matrix are determined during the training process of the entire model and thus reflect a meaningful transformation of the embeddings into a joint vector space.

Given text t and its corresponding word embedding $\mathbf{R}_{text}(t)$ derived by the pre-trained language model, the final vector representation of t is then calculated as:

$$\mathbf{R}_{final}(t) = \mathbf{R}_{text}(t) \oplus \mathbf{R}_{entity}(t) \tag{2}$$

As a further optional addition to their model, Tran and Yates include an external scoring source that measures the relationship between query entities and documents. They use a kernel pooling score that is intended to measure the importance of entities of queries which also occur within documents. Accordingly, this approach requires knowledge of the query and document at runtime. Therefore, this approach belongs to methods interaction-based approaches of incorporating entities within retrieval tasks, as described in section 2.

This kernel pooling score, called KNRM-signal will be elaborated in detail in the following subsection. Together with the KNRM-signal S_{knrm} of a query q and a document d the final vector representation of the model of Tran and Yates resolves to

$$\mathbf{R}_{final\_knrm}(t) = \begin{cases} \mathbf{R}_{text}(t) \oplus \mathbf{R}_{entity}(t) + 1 & t \text{ is query} \\ \mathbf{R}_{text}(t) \oplus \mathbf{R}_{entity}(t) + S_{knrm}(t) & t \text{ is document} \end{cases} \tag{3}$$

Since the interaction of entities in queries with each other is always ideal, the maximum value $S_{knrm}(t)$ will be reached whenever t is a query. As shown in subsection 3.2, the supremum of $S_{knrm}(t)$ is 1, therefore the value for $S_{knrm}(t)$ is set to 1, if t is a query.

3.2 KNRM Signal

Tran and Yates introduce the KNRM signal, originally developed by Xiong et al. [19], as an additional scoring mechanism to extend their basic approach by a separate framework. Unlike its original use, Tran and Yates adapt the KNRM model to specifically capture the interaction between entities in queries and documents. The calculation of the KNRM signal follows these steps:

1. Let $X(q)$ be the set of all entity embeddings within query q, $X(d)$ any set of entity embeddings which occur in document d. For EVA Single and EVA Single-QA models (see subsubsection 3.3.3) $X(d)$ contains all entities with the given document, for EVA Multi (see subsubsection 3.3.3) $X(d)$ contains only entities of specific subsets of entities of d. The entity interaction matrix is defined as:

   $$T_{i,j} := sim(X_i(q), X_j(d)) \ ,$$

   where $X_i(q)$ and $X_j(d)$ are the i-th and j-th embedding of q and d. The similarity function is given as cosine similarity. Embeddings are generated via Wikipedia2Vec (Yamada et. al [23]), as described in subsubsection 3.3.1.

2. Build k kernels using radial basis function, which creates differentiable histograms around given μ and σ^2.

   $$K_l(X_i(q)) = \sum_{j=1}^{|X(p)|} \exp\left(-\frac{(T_{i,j} - \mu_i)^2}{2\sigma_i^2}\right)$$

3. Pool / Summarize the k results into a k-dimensional feature vector:

   $$\overrightarrow{K(X_i(q))} = [K_1(X_i(q)), \ldots, K_k(X_i(q))]$$

4. Build kernel-pooled representation $\phi(T)$ by calculating log-sum for each query entity:

   $$\phi(T) = \sum_{i=1}^{|X(q)|} \log \overrightarrow{K(X_i(q))}$$

5. Get final kernel pooling score by applying a learned ranking layer. Note that $\tanh(\cdot) \in (-1, 1)$ and therefore $\sup S_{knrm} = 1$:

$$S_{\text{knrm}} = \tanh(w^T \phi(T) + b)$$

Despite being an interaction-based model that requires scoring during runtime for all query-document pairs, the computational complexity of the KNRM approach remains limited. The computations involved are relatively straightforward, and the additional learned layer does not significantly increase the computational overhead. Furthermore, these computations can be performed in parallel with the other components of Tran and Yates' approach. Empirical results concerning the latency of the models, with and without the KNRM signal, validate this assertion (see section 4).

3.3 Generating Entity Embeddings

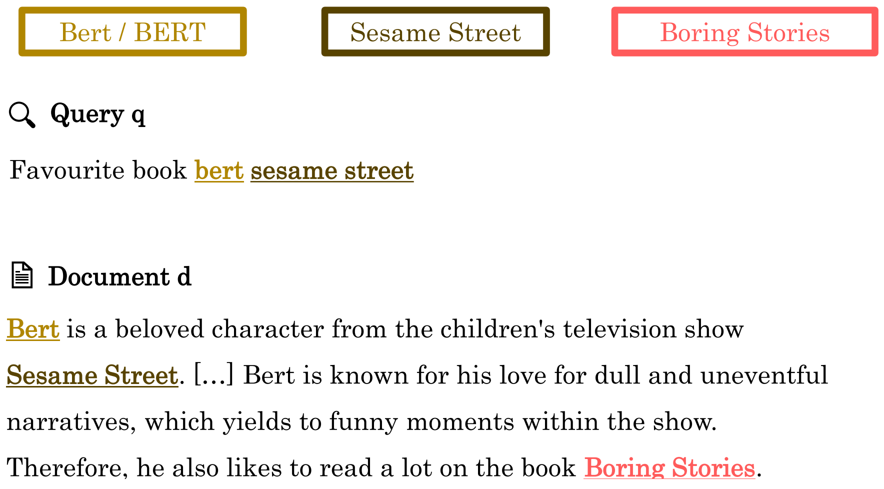
As described in subsection 3.1, Tran and Yates create embeddings for both word tokens and entities in queries and documents. While the word token embeddings are generated using TAS BERT through the dense retrieval approach, the primary focus of their contribution lies in the creation of entity embeddings. In particular, since queries and documents usually contain more than one entity, they need to be aggregated to a single embedding.

In order to do so, the example given in Figure 2 shall be introduced: Given the query 'Favourite book bert sesame street' and a corresponding document that mentions Bert as a character of Sesame Street who enjoys the book Boring Stories, there are three entities: Bert, Sesame Street, and Boring Stories.

3.3.1 Extracting Entities

To aggregate multiple entities in a document or query, Tran and Yates first extract these entities from the text using external frameworks Dexter (Ceccarelli et al. [3]) and Wikipedia2Vec (Yamada et. al [23]).

The Dexter framework is an entity linkage system that resolves entity mentions in text to corresponding entities in a knowledge base. In particular, Dexter employs a combination of methods, including named entity recogni-



Favourite book bert sesame street

Bert is a beloved character from the children's television show Sesame Street. [...] Bert is known for his love for dull and uneventful narratives, which yields to funny moments within the show. Therefore, he also likes to read a lot on the book Boring Stories.

Figure 2: Example query and example document

tion and pattern matching, to perform this task.

Wikipedia2Vec, on the other hand, leverages the structure of Wikipedia to generate dense vector representations (embeddings) for words, articles, and entities. It utilizes the Word2Vec algorithm (Mikolov et al. [10]) to capture semantic relationships from Wikipedia, producing high-dimensional embeddings.

The extraction procedure involves submitting a document or query to Dexter, which extracts entity mentions from the given text. These entity names are then passed on to the knowledge base, which transforms them into embeddings. Wikipedia2Vec provides vectors in dimension 100 as default, Tran and Yates keep this value in their model. Figure 3 visualizes this process.

3.3.2 Combining Entities for Queries

For now, multiple embeddings for each entity within a query or a document are generated by applying the procedure outlined in subsubsection 3.3.1. The aggregation of these generated embeddings differs based on whether a query or a document is considered, given the usual brevity of queries compared to documents.

For queries, Tran and Yates adopt a straightforward approach. They aggregate the embeddings of entities by averaging all entity embeddings across all

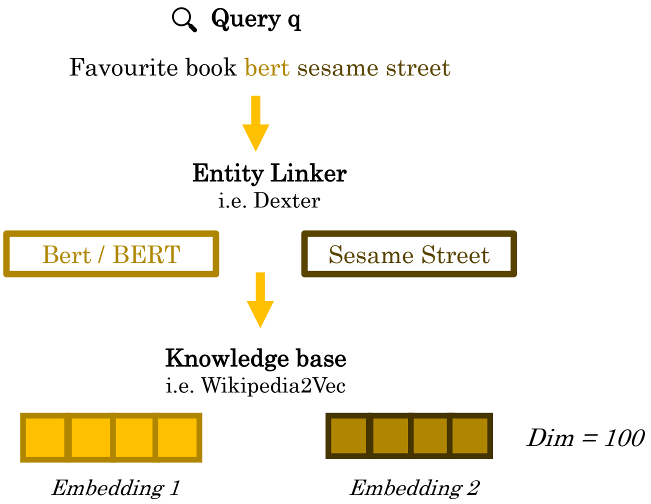


Figure 3: Process of entity extraction

dimensions. Figure 4 visualizes this procedure.

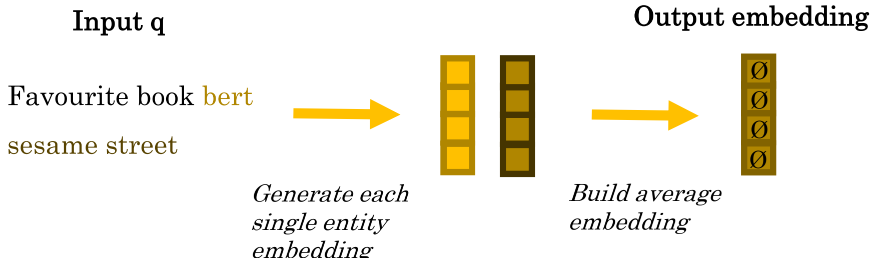


Figure 4: Process of generating a single entity embedding for a query

3.3.3 Combining Entities for Documents

Documents usually contain many different entities, as the example in Figure 2 indicates. Additionally, documents might also cover different aspects of a topic, so entities from very different areas might appear within them. This factor adds more complexity to the aggregation of embeddings for documents than for queries. To overcome this issue, Tran and Yates elaborated three different methods to aggregate embeddings of documents, which build upon each other:

- Single Entity Representation (EVA Single)
- Query-Aware Single Entity Representation (EVA Single-QA)
- Multiple Entity View Representation (EVA Multi)

The term EVA refers to Entity Views in Dense Retrieval. The concept of Entity Views, the name-giving term for the paper by Tran and Yates, is used in particular in the third method EVA Multi and will be introduced in the following.

EVA Single The first approach of aggregating the entities of a document is similar to the one used for queries. The idea of the Single Entity Representation approach is to extract all entities of a document and then create a single average output embedding. Analogous to Figure 4 for queries, this approach is visualized in Figure 5.

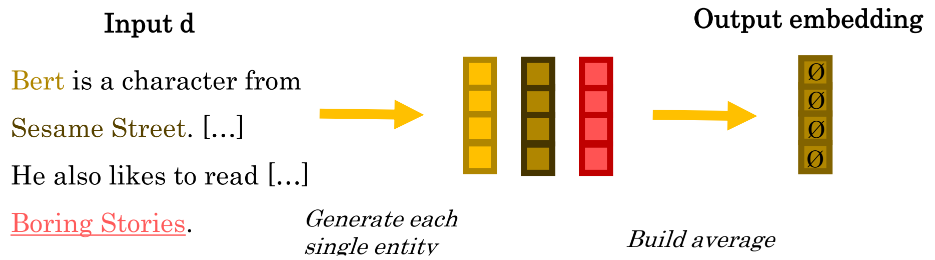


Figure 5: Process of generating an entity embedding for a document following the EVA Single approach

This approach does not take into account that documents can cover various topics. The query information is completely discarded and thus partially irrelevant entities are accounted for calculations of the output embedding. This leads to a bias within the ranking results, as it can be observed in section 4.

EVA Single-QA To address this problem, Tran and Yates introduce the Query-Aware Single Entity Representation, which creates embeddings focusing on the needs of the query. However, this requires the assumption that the query is known before calculations, which leads to increased computational complexity. This is due to the fact that computations for all duets of query and documents must now be performed during runtime, precomputations of embeddings for documents and indexing is no longer possible. This effect is reflected within the results (see section 4), which prove a high latency in the case of the EVA-Single QA approach.

The underlying idea of the EVA single QA model is to filter the entities of a document based on the information of a given query and select only entities with high similarity to a query entity. Figure 6 visualizes this process.

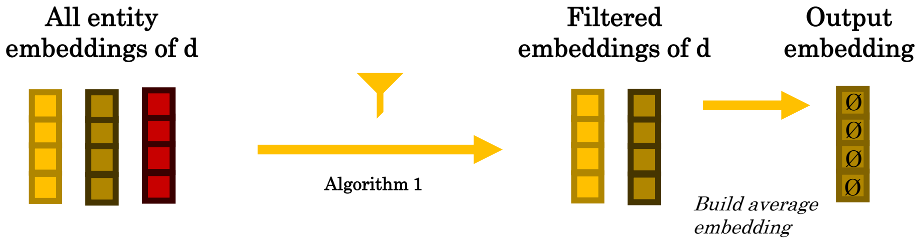


Figure 6: Process of generating an entity embedding for a document following the EVA Single-QA approach

Filtering is done using algorithm 1, which is applied to all duets of a query q and document d: For each entity within q the algorithm searches for the entity within d having the maximum cosine similarity. If this similarity extends some threshold, the respective entity of d will be added to the filtered list $X_{focus}(d)$. As a final step, the single output embedding for document d is calculated as the average of all entity embeddings within the filtered list $X_{focus}(d)$.

EVA Multi The EVA single-QA approach suffers from the issue that queries must be known before algorithm 1 can be applied and the query-aware document representation can be retrieved. With the third approach, called EVA-Multi, Tran and Yates have found a solution to this problem with only minor negative side effects.

They analyzed their training data and realized that the number of entities in queries do not exceed two in the vast majority of instances. Based on the experimental setup described in section 4, they found that 99.6 % of the 300,000 training instances and 99.5 % of test instances in the MS Marco Dev dataset contain two or fewer entities. Table 2 shows the respective analysis of

Algorithm 1 Query-aware document entity representation

```
Input: Query $q$ and document $d$, threshold $\alpha$
Output: Filtered entity embedding list $X_{focus}(d)$ of $d$
$X(q) \leftarrow$ set of embeddings of entities in $q$
$X_{focus}(d) \leftarrow \{\}$
for $e$ in $X(q)$ do
    $e^* \leftarrow$ entity embedding in $d$ having the maximum cosine similarity with $e$
    if cosine similarity$(e^*, e) > \alpha$ then
        $X_{focus}(d) \leftarrow X_{focus}(d) \cup \{e^*\}$
    end if
end for
return $X_{focus}(d)$
```

queries. Therefore, Tran and Yates focused their research on the assumption that it is sufficient to consider a maximum of two entities in queries.

Entities	**Training Queries**		**Testing Queries**	
	Count	**Fraction**	**Count**	**Fraction**
0	130,353	0.435	3,442	0.483
1	149,073	0.497	3,232	0.454
2	19,207	0.064	416	0.058
3+	1,367	0.004	37	0.005
Total	300,000		7,127	
Average	0.640		0.587	

Table 2: Summary statistics of the queries.

If one applies algorithm 1 under this assumption, one notices that at most two entities remain in the filtered embedding list X_{focus}. This is due to the fact that the algorithm iterates over the set of all entities in the given query once. Since X_{focus} therefore only contains no item, a single item or at maximum two items, the amount of all possible sets that are eligible for X_{focus} is limited by $|\{\}| + |X(d)| + \binom{|X(d)|}{2}$, where $|X(d)|$ corresponds to the number of entities in d.

Tran and Yates take advantage of this and introduce clusters of entities that can be seen as different views on a document. For the EVA Multi approach, all possible single itemsets and sets of pairs of the entities are generated. Sets of pairs are only considered, if cosine similarity between the two items within a pair extend a predefined threshold β to ensure only reasonable entity views are generated. Final output embeddings are again calculated by averaging the embeddings of all items within each set. When applying the optional KNRM signal (see subsection 3.2) to this approach, the entity interaction matrix T is build only upon the set of the entity embeddings of a single cluster and not on all entities within the respective document.

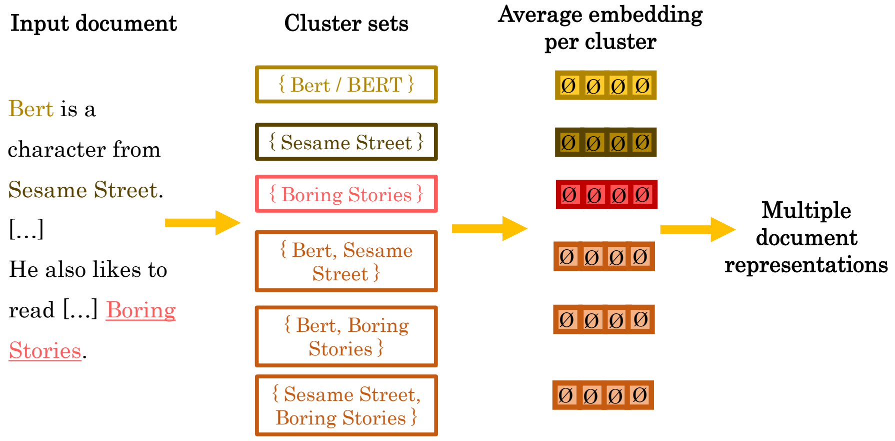

Input document
Cluster sets
Average embedding per cluster
Bert is a character from Sesame Street. [...] He also likes to read [...] Boring Stories.
{ Bert / BERT }
{ Sesame Street }
{ Boring Stories }
{ Bert, Sesame Street }
{ Bert, Boring Stories }
{ Sesame Street, Boring Stories }
Ø Ø Ø Ø
Multiple document representations

Figure 7: Process of generating entity embeddings for a document following the EVA Multi approach

All the calculations can be performed independently of the information about queries and thus allow indexing. In contrast to the two previously mentioned methods EVA Single and EVA Single-QA, several entity output embeddings are now generated per document, which have to be taken into account during retrieval ranking.

Figure 7 visualizes the process of generating the embeddings of multiple entity views / clusters. For the given example of three entities within an input document, six different views on entities and document entity representations are generated.

4 Results

In the following, the concrete implementation of the described models in section 3 and the resulting results will be presented. The focus of the elaborations are the major findings of the work of Tran and Yates, accordingly not all results are listed herein. Details can be studied in the paper by Tran and Yates [17].

4.1 Experimental Setup

Tran and Yates implemented their models in a TensorFlow (Abadi et al. [1]) setup: For the pretrained language model, they chose a distilled BERT (Sanh et al. [14]), fine-tuned using TAS BERT approach (Hofstätter et al. [7]), as described in subsection 3.1. Further, to determine the embeddings, Tran and Yates used Dexter (Ceccarelli et al. [3]) and Wikipedia2Vec (Yamada et. al [23]), as described in subsection 3.3.

Training was done using pairwise hinge loss on four Quadro RTX 8000 GPUs in parallel on 300,000 training samples of the MS MARCO dataset (Nguyen et al. [11]). For the two models that allow indexing, i.e. EVA Single and EVA Multi (see subsubsection 3.3.3), the end-trained model was used to index embeddings for documents.

Evaluation was performed on 7,127 test samples from the MS Marco dataset and the datasets TREC Deep Learning (DL) Track 2019 (MacAvaney et al. [8]), TREC DL 2020 (MacAvaney et al. [9]), TREC DL HARD (Yates et al. [24]). Evaluation metrics were chosen to be nDCG@10, MRR@10, MAP@1000.

Tran and Yates compared their own models EVA Single, EVA Single-QA and EVA Multi (see subsubsection 3.3.3), each with and without KNRM signal (see subsection 3.2), against different baselines:

- BM25: The most prominent example of exact matching paradigm, using sparse representations (Robertson et al. [13])
- TAS BERT: Fine-tuned BERT model using topic-aware sampling strategy as described in subsection 3.1.
- ANCE: A state-of-the-art bi-encoder model that employs a sophisticated negative sample mining strategy during training process (Xiong et al. [21]).
- BM25 + T5 (Zero-Shot): First stage ranking using BM25 plus a T5-cross-encoder (Nogueira et al. [12]) to rerank the top 1000 results of first stage retrieval.
- ERNIE: Fine-tuned model of ERNIE v2 (Sunh et. al. [16]), which is an optimized version of base ERNIE as described in section 2.

- ERNIE Multi: Similar to the EVA Multi approach, but using ERNIE as a pre-trained language model for word embeddings.
- Best Reported: The best results of the MS MARCO leader board or respectively the best results of the corresponding papers of the TREC DL datasets (MacAvaney et al. [8], MacAvaney et al. [9], Yates et al. [24]).

4.2 Efficiency

To evaluate their different models, Tran and Yates studied both efficiency and effectiveness. Full results for both can be found in the research paper of Tran and Yates [17]. Considering the aspect of latency, the results of all examined models are shown in Table 3. The table shows the average search time per query across all evaluation datasets for each model, ran on the same server.

Methods	**Latency (ms)**
Low latency (<100 ms)	
BM25	13
ANCE	25
ERNIE Tuned	29
ERNIE Multi	70
TAS BERT	28
EVA Single	40
EVA Multi	76
EVA Multi-KNRM	74
Higher latency (>100 ms)	
EVA Single-QA	2,039
EVA Single-QA-KNRM	3,839
BM25 + T5 (Zero-Shot)	5,052
Best Reported	-

Table 3: Analysis of effectiveness of EVA models and baselines

4.3 Effectiveness

Since the results of the effectiveness of the various models are consistent across all combinations of evaluation metric and choice of dataset, this report will only provide exemplary results based on a selected metric of a particular evaluation dataset. Thus, these are representatives of the results of all metrics and data sets. The main outcomes, which are described in the following subsection 4.4, can therefore be derived based on all results. The nDCG@10 metric and the TREC DL Track 2019 dataset (MacAvaney et al. [8]) will serve as the example. The results are presented in a visual layout in Figure 8.

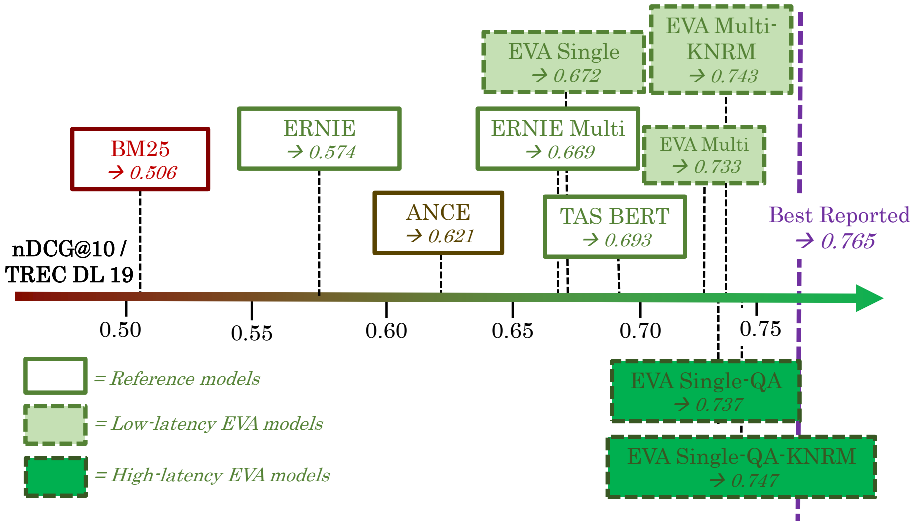


Figure 8: Exemplary results of EVA models and baselines for TREC DL 19 dataset and evaluation metric nDCG@10

4.4 Impact

Based on the results from subsection 4.3 and subsection 4.2, the main outcomes to be attributed to Tran and Yates' work are:

1. **Enriching pre-trained language models with entity embeddings improve effectiveness significantly**: The effectiveness results reveal that the models proposed by Tran and Yates, EVA Multi and EVA Single-QA, both with and without KNRM signal, significantly outperform the various baselines. In the given example in Figure 8, the value for nDCG@10 of the best performing baseline TAS BERT is 0.693. In contrast, the value for EVA Multi is 0.733, for EVA Single-QA 0.737, which thus corresponds to an increase in effectiveness of 5.8 % (EVA Multi) and 6.3 %. Further, the models are only slightly off the best reported results.

2. **Multiple entity views increase performance to single view**: Considering the performance of the first introduced approach EVA Single, which incorporates entities in documents without a specific focus on the query information, one observes poor results. The EVA Single model even performs worse than the baseline TAS BERT. This is due to the fact that entirely irrelevant entities within documents are taken into account during the retrieval process, thus biasing the results (i.e. see subsubsection 3.3.3). However, if only entities relevant to the respective query are taken into account, which is the case for EVA Single-QA and EVA Multi this leads to a significant increase in effectiveness compared to all baselines, as explained above.

3. **KNRM signal provides slight improvement of effectiveness**: Comparing the results of the EVA models with additional KNRM signal and without KNRM signal, one observes slightly better results of the EVA models with KNRM signal. This is certainly the case for the example in Figure 8, where the EVA Multi KNRM model with an nDCG@10 value of 0.748 is slightly higher than the value of the EVA Multi model of 0.733. The same applies in the setting of the EVA Single-QA model. However, the effect is small; for the TREC DL 2020, the EVA Multi approach even outperforms the same approach with additional KNRM signal. It can be concluded that the impact of introducing entity embeddings is stronger than that of introducing the KNRM signal.

4. **Removing known query assumptions has minor impact on effectiveness, but increases efficiency drastically**: Best results of effectiveness across all EVA models and baselines are achieved by EVA Multi and EVA Single-QA, as described above. However, the key difference between EVA Multi and EVA Single-QA is the assumption of knowing queries at runtime for EVA Single-QA and the need for large language model inference at runtime (see subsubsection 3.3.3). This leads to large differences in latency compared to the EVA multi-model.. As Table 3 shows, the latency for the EVA Multi models is 74 ms with KNRM signal and 76 ms without, while the latency for both of the EVA Single-QA models exceeds two seconds. In a real-world scenario, this would be excessive; users of an information retrieval system do not

usually are willing to wait such a long time for results. However, since the effectiveness results of EVA Multi and EVA Single-QA differ only marginally, the EVA Multi approach provides a good trade-off between efficiency and effectiveness.

5 Discussion

Tran und Yates provide einen Ansatz der

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] Krisztian Balog. *Entity-oriented search*. Springer Nature, 2018.

[3] Diego Ceccarelli, Claudio Lucchese, Salvatore Orlando, Raffaele Perego, and Salvatore Trani. Dexter: an open source framework for entity linking. In *Proceedings of the sixth international workshop on Exploiting semantic annotations in information retrieval*, pages 17–20, 2013.

[4] Jeffrey Dalton, Laura Dietz, and James Allan. Entity query feature expansion using knowledge base links. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 365–374, 2014.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[6] Benjamin Heinzerling and Kentaro Inui. Language models as knowledge bases: On entity representations, storage capacity, and paraphrased queries. *arXiv preprint arXiv:2008.09036*, 2020.

[7] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 113–122, 2021.

[8] Sean MacAvaney, Andrew Yates, Sergey Feldman, Wei Guo, Yixing Hua, Tom Kenter, Federico Nanni, Bhaskar Mitra, Navid Rekabsaz, and Hamed Zamani. Overview of the trec 2019 deep learning track. In *Proceedings of The 28th Text REtrieval Conference (TREC 2019)*, 2019.

[9] Sean MacAvaney, Andrew Yates, Sergey Feldman, Wei Guo, Yixing Hua, Tom Kenter, Bhaskar Mitra, Federico Nanni, Navid Rekabsaz,

and Hamed Zamani. Overview of the trec 2020 deep learning track. In *Proceedings of The 29th Text REtrieval Conference (TREC 2020)*, 2020.

[10] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26:3111–3119, 2013.

[11] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. Ms marco: A human generated machine reading comprehension dataset. *choice*, 2640:660, 2016.

[12] Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. Document ranking with a pretrained sequence-to-sequence model. *arXiv preprint arXiv:2003.06713*, 2020.

[13] Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, March 2009. ISSN 1554-0669.

[14] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[15] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*, 2019.

[16] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. Ernie 2.0: A continual pre-training framework for language understanding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8968–8975, 2020.

[17] Hai Dang Tran and Andrew Yates. Dense retrieval with entity views. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 1955–1964, 2022.

[18] Chenyan Xiong, Jamie Callan, and Tie-Yan Liu. Word-entity duet representations for document ranking. In *Proceedings of the 40th Interna-*

tional ACM SIGIR conference on research and development in information retrieval, pages 763–772, 2017.

[19] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of the 40th International ACM SIGIR conference on research and development in information retrieval*, pages 55–64, 2017.

[20] Chenyan Xiong, Russell Power, and Jamie Callan. Explicit semantic ranking for academic search via knowledge graph embedding. In *Proceedings of the 26th international conference on world wide web*, pages 1271–1279, 2017.

[21] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*, 2020.

[22] Yang Xu, Gareth JF Jones, and Bin Wang. Query dependent pseudo-relevance feedback based on wikipedia. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 59–66, 2009.

[23] Ikuya Yamada, Akari Asai, Jin Sakuma, Hiroyuki Shindo, Hideaki Takeda, Yoshiyasu Takefuji, and Yuji Matsumoto. Wikipedia2vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from wikipedia. *arXiv preprint arXiv:1812.06280*, 2018.

[24] Andrew Yates, Sean MacAvaney, Bhaskar Mitra, Navid Rekabsaz, Hamed Zamani, Chenyan Li, Xiang Xu, Zhuyun Dai, Saptarshi Pal, Hui Fang, et al. Overview of the trec 2020 deep learning for hard information retrieval (dlhard) track. In *Proceedings of The 29th Text REtrieval Conference (TREC 2020)*, 2020.