Dense Retrieval with Entity Views

Seminar "Modern Infomation Retrieval", Summer 2023

Johannes Gabriel Sindlinger

Heidelberg University
Prof. Dr. Michael Gertz / Ashish Chouhan
Institute of Computer Science
Database Systems Research Group
johannes.sindlinger@stud.uni-heidelberg.de

June 15, 2023

Relevance of Paper

Authors:

- Hai Dang Tran (Max Planck Institute for Informatics)
- Andrew Yates (University of Amsterdam)

Publication Date: 17 October 2022

Publication Conference: CIKM '22: Proceedings of the 31st ACM International Conference on Information & Knowledge Management

Citations: 0

Downloads on ACM[1]: 316

[1] last retrieved June 3, 2023

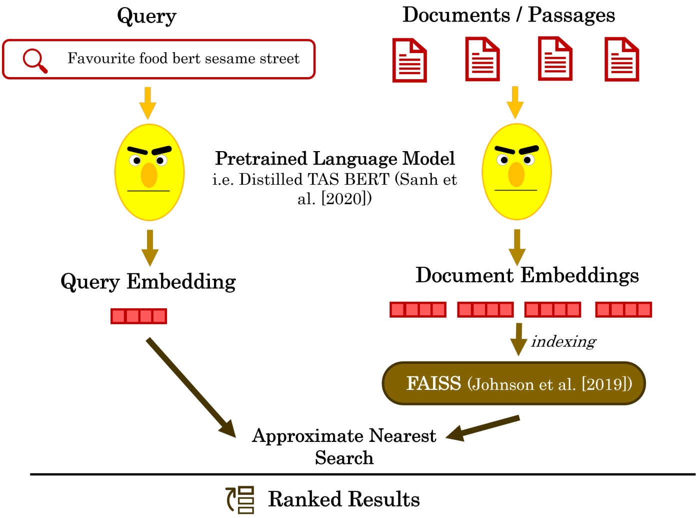
Outline

1. What's the issue? – Motivation
2. What has been already there? – Related Work
3. What‘s new? – Methodology
4. What's the outcome? – Results

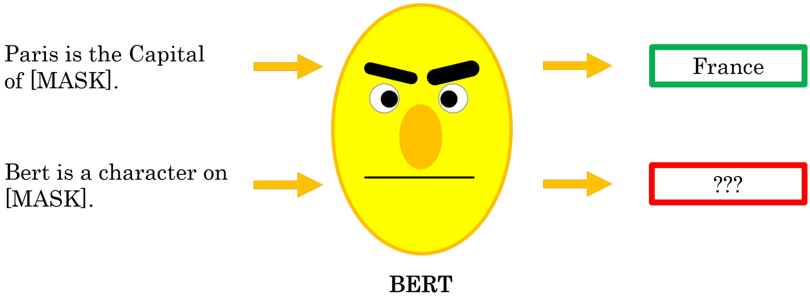
Outline

1. What's the issue? – Motivation
2. What has been already there? – Related Work
3. What's new? – Methodology
4. What's the outcome? – Results

Introduction – Bi-Encoder Model



Motivation



⇒ Language models do not fully capture information about real-world entities, especially for uncommon entities.

Example

Bert / BERT | Sesame Street | Boring Stories

Query q

Favourite book bert sesame street

Document d

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.

Outline

1. What's the issue? – Motivation
2. What has been already there? – Related Work
3. What‘s new? – Methodology
4. What's the outcome? – Results

Related Work

Pre neural IR models: Extending queries with entity descriptions or features (e.g. synonyms, relationships)

Within neural IR models:

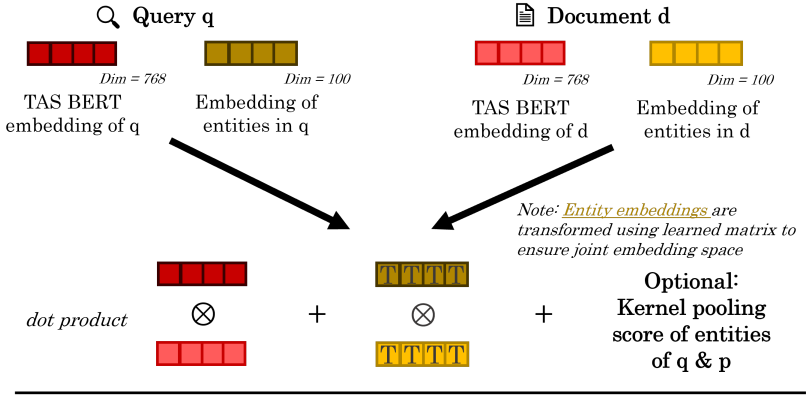
- Interaction-based ranking methods (e.g. KNRM[2])
- Including entities within learning procedure (e.g. ERNIE[3])

What's new? Including entity representations **independently** of pre-trained language model.

[2] Xiong et al. [2017], additionally applied to the approach of this paper
[3] Sun et al. [2020]

Outline

General Model



Final ranking result

Choose document with highest overall score

Extracting Entities

Query q

Favourite book bert sesame street

Entity Linker

i.e. Dexter (Ceccarelli et al. [2013])

Bert / BERT

Sesame Street

Knowledge base

i.e. Wikipedia2Vec (Yamada et al. [2018])

Embedding 1

Embedding 2

Dim = 100

Kernel Pooling Score – KNRM

Idea: Find documents where entities of documents and query have high similarity value using kernel-based neural ranking model.

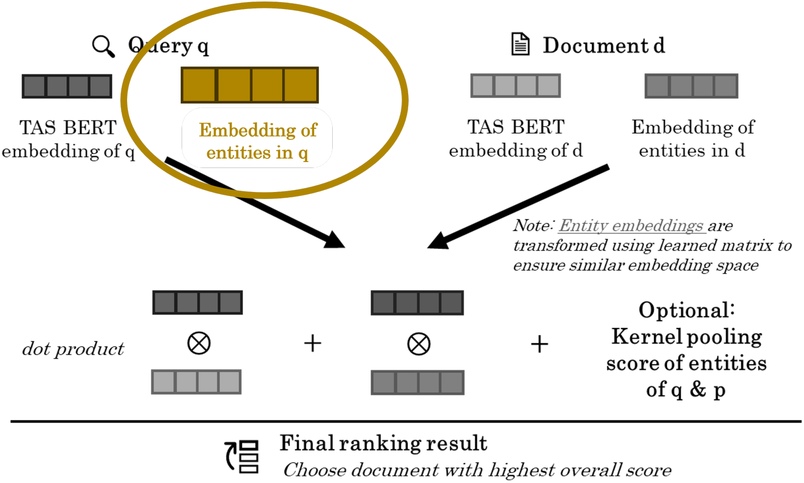
⇒ **Interaction-based** approach

1. Get entity interaction matrix between the set of entities within query $\mathbf{E}(q)$ and document $\mathbf{E}(d)$:

$$T_{i,j} := sim(\mathbf{E}_i(q), \mathbf{E}_j(d))$$

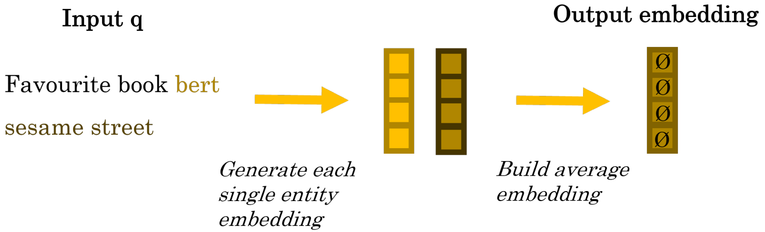
2. Build k kernels for each entity of q using radial basis function, which creates differentiable histograms around given μ and σ^2
3. Calculate a pooled / summarized representation of kernels and apply final (learned) ranking layer on that.

Generating Embeddings – Queries

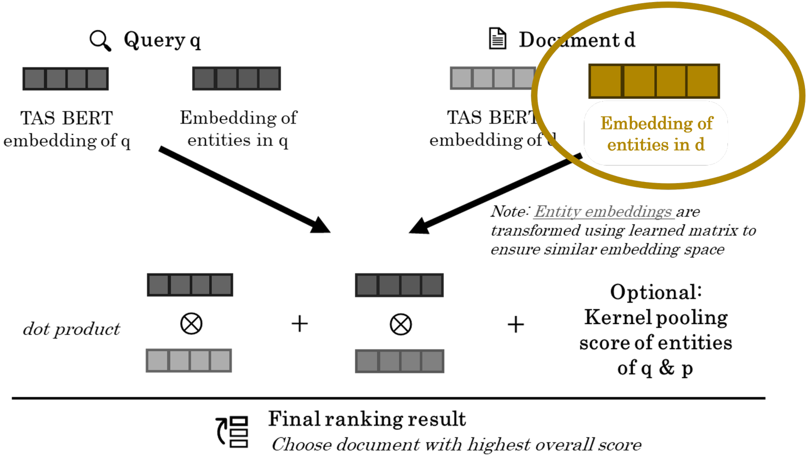


Generating Embeddings – Queries

Idea: Combine all Entities within Query (since query is unknown in advance, keep this method for all approaches)



Generating Embeddings – Documents



Generating Embeddings – Documents

- Single Entity Representation (EVA[4] Single)
- Query-Aware Single Entity Representation (EVA Single-QA)
- Multiple Entity View Representation (EVA Multi)

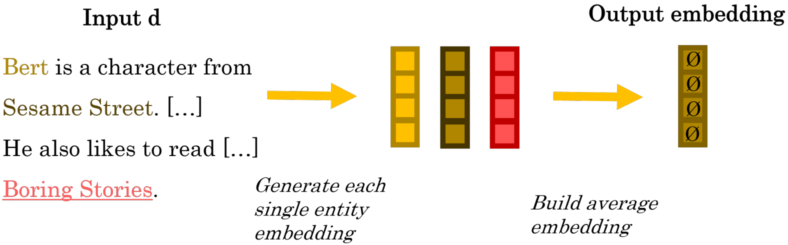
⇒ Optionally for all models: Adding Kernel pooling score (i.e. KNRM)

[4] EVA $\widehat{=}$ Entity Views in Dense Retrieval

EVA Single

Idea: Combine all Entities within Document / Passage

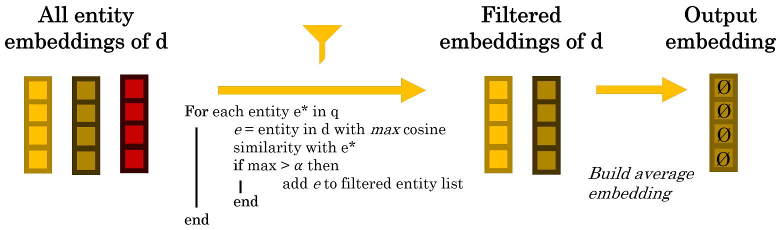
⇒ Problem: No focus on query information, possibly including irrelevant entities



EVA Single-QA

Assumption: Query is known before calculations

Idea: Select only entities in document with high similarity to query entities

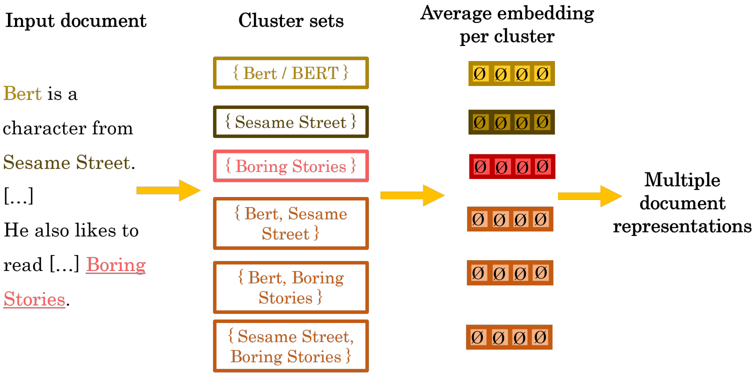


⇒ Problem: Large latency, EVA Multi solves this issue

EVA Multi / 1

Idea: Different queries require different views on entities

⇒ Build embeddings for clusters of entities within document.



EVA Multi / 2

Building clusters removes assumption knowing the query in advance.

Why?

Recall filtering algorithm for EVA Single-QA: For each entity in the query q at maximum one close entity of the document d is selected.

$\Rightarrow$ |Entities in q| $\geq$ |Filtered embeddings of d|

Analyzing data: $>$ 99% queries contain at maximum 2 entities.

$\Rightarrow$ Clusters of size 1 and 2 are enough, can be enumerated easily.

Outline

Experimental Setup

- Training data: MS MARCO[5] (300,000 training samples, 7,127 test samples)
- Evaluation data: MS MARCO Dev, TREC Deep Learning (DL) Track 2019[6], TREC DL 2020[7], TREC DL HARD[8]
- Evaluation metrics: nDCG@10, MRR@10, MAP@1000

[5] Nguyen et al. [2016]
[6] MacAvaney et al. [2019]
[7] MacAvaney et al. [2020]
[8] Yates et al. [2020]

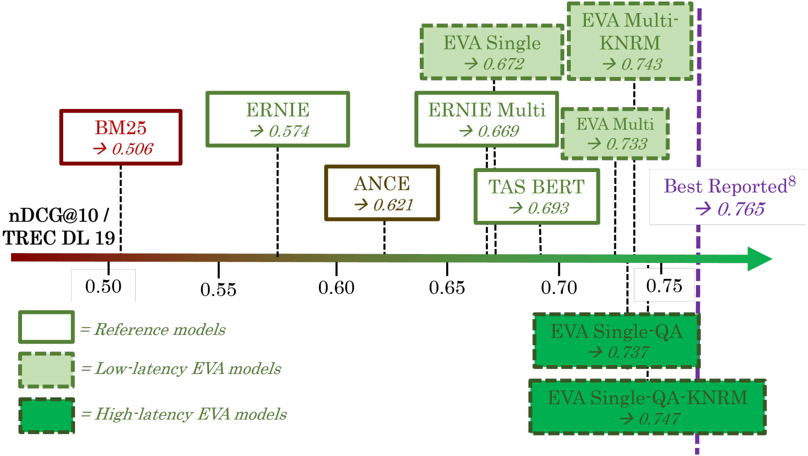
Results

Methods	Latency (ms)	TREC DL 19 nDCG	TREC DL 19 MRR	TREC DL 19 MAP	TREC DL 20 nDCG	TREC DL 20 MRR	TREC DL 20 MAP	DL HARD nDCG	DL HARD MRR	DL HARD MAP	MS MARCO Dev nDCG	MS MARCO Dev MRR	MS MARCO Dev MAP
Low latency (<100 ms)													
BM25	13	0.506	0.702	0.301	0.480	0.653	0.286	0.285	0.465	0.159	0.228	0.184	0.193
ANCE	25	0.621	0.763	0.361	0.605	0.786	0.373	0.335	0.446	0.193	0.368	0.311	0.317
ERNIE Tuned	29	0.574	0.728	0.326	0.573	0.760	0.348	0.287	0.388	0.163	0.320	0.267	0.274
ERNIE Multi	70	0.669†	0.822	0.422†	0.631†	0.891†	0.394†	0.329	0.452	0.198	0.344†	0.291†	0.296†
TAS BERT	28	0.693	0.835	0.442	0.673	0.812	0.451	0.360	0.472	0.224	0.395	0.334	0.340
EVA Single	40	0.672	0.853	0.429	0.642	0.813	0.428	0.363	0.481	0.224	0.374	0.316	0.322
EVA Multi	76	0.733	0.853	**0.483**	**0.694**	**0.855**†	**0.456**	0.397†	0.521†	0.240	**0.407**†	0.346†	0.350†
EVA Multi-KNRM	74	**0.743**†	**0.879**	0.482	0.680	0.827	0.440	**0.402**†	**0.532**†	**0.253**	0.406†	**0.347**†	**0.351**†
Higher latency (>100 ms)													
EVA Single-QA	2039	0.737	0.862	0.443	**0.701**	**0.856**	0.444	0.389	0.515	0.221	0.402	0.342	0.346
EVA Single-QA-KNRM	3839	**0.747**	**0.874**	**0.447**	0.685	0.838	0.439	0.397	0.534	0.232	0.405	0.347	0.351
BM25 + T5 (Zero-Shot)	5052	0.718	0.865	0.443	0.683	0.837	**0.462**	**0.408**	**0.585**	**0.238**	**0.443**	**0.380**	**0.383**
Best Reported	-	0.765	0.928	0.503	0.803	0.915	0.545	0.408	0.585	0.238	-	0.463	-

Results

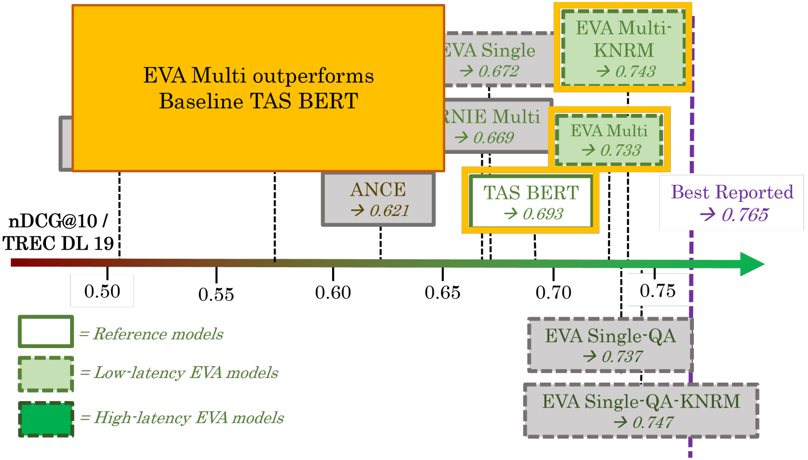
Methods	Latency (ms)	TREC DL 19 nDCG	TREC DL 19 MRR	TREC DL 19 MAP	TREC DL 20 nDCG	TREC DL 20 MRR	TREC DL 20 MAP	DL HARD nDCG	DL HARD MRR	DL HARD MAP	MS MARCO Dev nDCG	MS MARCO Dev MRR	MS MARCO Dev MAP
Low latency (<100 ms)													
BM25	13	0.506	0.702	0.301	0.480	0.653	0.286	0.285	0.465	0.159	0.228	0.184	0.193
ANCE	25	0.621	0.763	0.361	0.605	0.786	0.373	0.335	0.446	0.193	0.368	0.311	0.317
ERNIE Tuned	29	0.574	0.728	0.326	0.573	0.760	0.348	0.287	0.388	0.163	0.320	0.267	0.274
ERNIE Multi	70	0.669†	0.822	0.422†	0.631†	0.891†	0.394†	0.329	0.452	0.198	0.344†	0.291†	0.296†
TAS BERT	28	0.693	0.835	0.442	0.673	0.812	0.451	0.360	0.472	0.224	0.395	0.334	0.340
EVA Single	40	0.672	0.853	0.429	0.642	0.813	0.428	0.363	0.481	0.224	0.374	0.316	0.322
EVA Multi	76	0.733	0.853	**0.483**	**0.694**	**0.855†**	**0.456**	0.397†	0.521†	0.240	**0.407†**	0.346†	0.350†
EVA Multi-KNRM	74	**0.743†**	**0.879**	0.482	0.680	0.827	0.440	**0.402†**	**0.532†**	**0.253**	0.406†	**0.347†**	**0.351†**
Higher latency (>100 ms)													
EVA Single-QA	2039	0.737	0.862	0.443	**0.701**	**0.856**	0.444	0.389	0.515	0.221	0.402	0.342	0.346
EVA Single-QA-KNRM	3839	**0.747**	**0.874**	**0.447**	0.685	0.838	0.439	0.397	0.534	0.232	0.405	0.347	0.351
BM25 + T5 (Zero-Shot)	5052	0.718	0.865	0.443	0.683	0.837	**0.462**	**0.408**	**0.585**	**0.238**	**0.443**	**0.380**	**0.383**
Best Reported	-	0.765	0.928	0.503	0.803	0.915	0.545	0.408	0.585	0.238	-	0.463	-

Exemplary Results

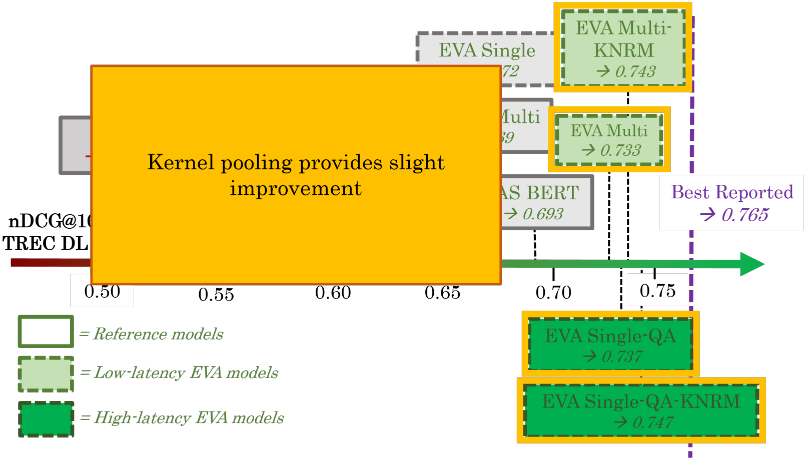


Test

Takeaways

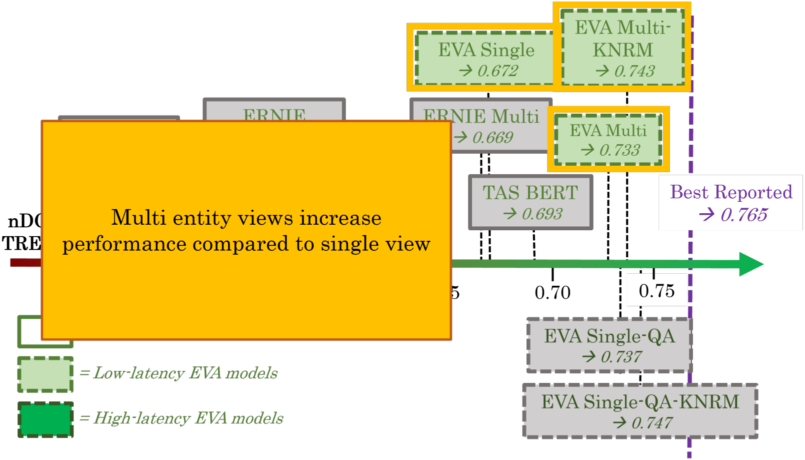

EVA Multi outperforms
Baseline TAS BERT
VA Single
→ 0.672
EVA Multi-
KNRM
→ 0.743
RNIE Multi
→ 0.669
EVA Multi
→ 0.733
ANCE
→ 0.621
TAS BERT
→ 0.693
Best Reported
→ 0.765
nDCG@10 /
TREC DL 19
0.50
0.55
0.60
0.65
0.70
0.75
= Reference models
= Low-latency EVA models
= High-latency EVA models
EVA Single-QA
→ 0.737
EVA Single-QA-KNRM
→ 0.747

Takeaways



Kernel pooling provides slight improvement

Takeaways

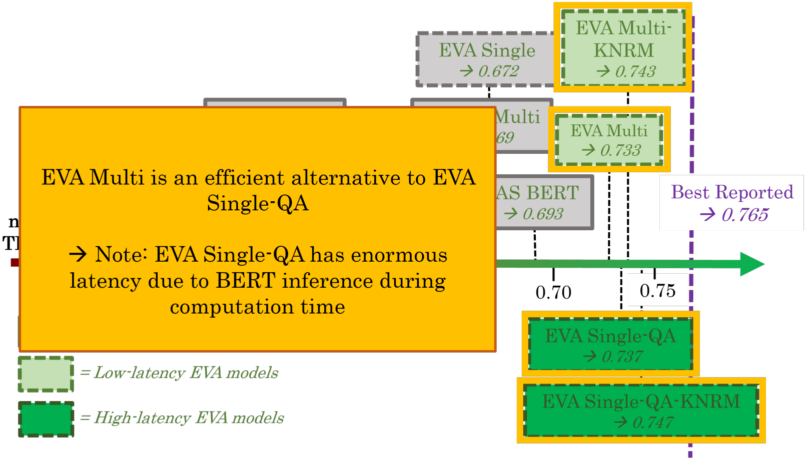


Multi entity views increase performance compared to single view

Takeaways

EVA Multi is an efficient alternative to EVA Single-QA

→ Note: EVA Single-QA has enormous latency due to BERT inference during computation time



Personal opinion

Likes

- Adding entities outperform basic Bi-Encoder approach significantly
- Multi-View approach seems reasonable and increase efficiency as well effectiveness
- Interpretable intuition

Dislikes

- Focus on entities is irrelevant for many queries, i.e. 43.5% of queries during training process are reported to have 0 entities.
- Only focusing on TAS BERT and ERNIE as Pre-trained language model.

Possible Improvements: Adding other attributes in addition to entities, e.g. metadata (geographical, time, etc.), keyword embeddings, ...

Questions

Thanks for your attention!

Questions?

References I

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.

Benjamin Heinzerling and Kentaro Inui. Language models as knowledge bases: On entity representations, storage capacity, and paraphrased queries. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2021.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

References II

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.

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.

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.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, A distilled version of BERT: Smaller, faster, cheaper and lighter. In *Advances in Neural Information Processing Systems*, 2020.

References III

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.

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.

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.

References IV

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.

Calculating KNRM I

1. Let $\mathbf{E}(q)$ be the set of entities within query q, $\mathbf{E}(d)$ the set of entities within document d. Then the entity interaction matrix is given as:

$$T_{i,j} := sim(\mathbf{E}_i(q), \mathbf{E}_j(d))$$

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

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

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

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

Calculating KNRM II

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

$$\phi(T) = \sum_{i=1}^{|\mathbf{E}(q)|} \log \overrightarrow{K(\mathbf{E}_i(q))}$$

5. Get final kernel pooling score by applying a learned ranking layer

$$\mathbf{S}_{\text{kp}} = \tanh(w^T \phi(T) + b)$$

Definitions: Pretrained Language Model Representation

Definition 1

Given a query or passage as text t the textual representation $\mathbf{R}_{text}(t)$ of t is formed by passing t to a pre-trained language model (PLM), i.e. distilled TAS (Sanh et al. [2020]). So it yields:

$$\mathbf{R}_{text}(t) = \mathrm{PLM}_{\mathrm{CLS}(t)}$$

Definition 2

Let $\mathbf{E}_{all}(q)$ be the set of all entities mentioned in query q. The query entity representation $\mathbf{R}_{all}(q)$ is then the average embedding of entities in $\mathbf{E}_{all}(q)$.

Definitions: Single Entity Representation

Definition 3

The query-independent passage entity representation $\mathbf{R}_{all}(p)$ is defined as the average embedding of entities in $\mathbf{E}_{all}(p)$.

Definition 4

The total representation of a passage or query in the setting of EVA-Single is defined as:

$$\mathbf{R}_{\text{single\_total}}(t) = \mathbf{R}_{\text{text}}(t) \oplus (W_{\text{entity}}^{T} \cdot \mathbf{R}_{\text{all}}(t))$$

The matrix W^{entity} is learned during training from MS MARCO dataset.

Definitions: Query-aware Entity Representation / 1

Definition 5

Let $\mathbf{R}_{focus}(p)$ be the set of passage entities which have maximum similarity with query entities. The query-aware passage entity representation $\mathbf{R}_{focus}(p)$ is the average embedding of entities in $\mathbf{E}_{focus}(p)$. See Algorithm 1 for details.

Definition 6

The transformed entity representation $\mathbf{R}_{\text{trans}}(t)$ of text t is defined as:

$$\mathbf{R}_{\text{trans}}(t)^{\intercal} = \begin{cases} \mathbf{R}_{\text{all}}(t)^{\intercal}\mathbf{W}_{\text{entity}} & \text{if } t \text{ is a query} \\ \mathbf{R}_{\text{focus}}(t)^{\intercal}\mathbf{W}_{\text{entity}} & \text{if } t \text{ is a passage} \end{cases}$$

Definitions: Query-aware Entity Representation / 2

Definition 7

The query-aware total representation $\mathbf{R}_{\text{total}}(t)$ of query or passage t is defined as:

$$\mathbf{R}_{\text{total}}(t) = \mathbf{R}_{\text{text}}(t) \oplus \mathbf{R}_{\text{trans}}(t)$$

where $\oplus$ is the concatenation operator.

Definition 8

Given a set of entities $\mathbf{X}$, the kernel pooling signal $\mathbf{S}_{\text{kp}}(\mathbf{X}, t)$ of $\mathbf{X}$ with the text t is defined as:

$$\mathbf{S}_{\text{kp}}(\mathbf{X}, t) = \begin{cases} 1, & \text{if } t \text{ is a query,} \\ \mathbf{S}_{\text{knrm}}(\mathbf{X}, t), & \text{if } t \text{ is a passage} \end{cases}$$

Definitions: Query-aware Entity Representation / 3

Definition 9

The query-aware total representation with kernel pooling, $\mathbf{R}_{\text{knrm}}(t)$, of text t is:

$$\mathbf{R}_{\text{knrm}}(t) = \mathbf{R}_{\text{total}}(t) \oplus \mathbf{S}_{\text{kp}}(\mathbf{E}_{\text{all}}(q), t)$$

Corollary 10

The final score of the query-aware passage entity representation is given as:

$$\begin{aligned}\mathbf{S}_{knrm}(q,p) &= \mathbf{R}_{knrm}(q) \otimes \mathbf{R}_{knrm}(p) \\ &= (\mathbf{R}_{text}(q) \otimes \mathbf{R}_{text}(p)) + (\mathbf{R}_{rans}(q) \otimes \mathbf{R}_{rans}(p)) \\ &\quad + \mathbf{S}_{kp}(\mathbf{E}_{all}(q), p)\end{aligned}$$

Algorithm: Query-aware passage entity representation

Algorithm 1 Query-aware passage entity representation

Input: Query q and passage p

Output: Query entity representation for q and query-aware passage entity representation for p

$E_{all}(q) \leftarrow$ set of entities in q
$R_{all}(q) \leftarrow$ average embedding of entities in $E_{all}(q)$
$E_{focus}(p) \leftarrow \{\}$
for e_q in $E_{all}(q)$ **do**
    $e_p \leftarrow$ entity in p having the maximum cosine similarity with e_q
    if cosine similarity$(e_p, e_q) > \alpha$ **then**
        $E_{focus}(p) \leftarrow E_{focus}(p) \cup \{e_p\}$
    end if
end for
$R_{focus}(p) \leftarrow$ average embedding of entities in $E_{focus}(p)$
return $R_{all}(q), R_{focus}(p)$

Definitions: Multiple Entity Representation / 1

Definition 11

Given passage p and an entity cluster C in p, let $\mathbf{R}_{cluster}(C)$ be the average embedding of entities in C. The transformed cluster representation $\mathbf{R}_{trans\_cluster}(C)$ of C is then:

$$\mathbf{R}_{trans\_cluster}(C)^T = \mathbf{R}_{cluster}(C)^T \cdot W_{entity}$$

Definition 12

Given passage p and an entity cluster C in p, the cluster total representation $\mathbf{R}_{\text{total\_cluster}}(C, p)$ of passage p with cluster C is given as:

$$\mathbf{R}_{\text{total\_cluster}}(C, p) = \mathbf{R}_{\text{text}}(p) \oplus \mathbf{R}_{\text{trans\_cluster}}(C)$$

Definitions: Multiple Entity Representation / 2

Definition 13

Given passage p and an entity cluster C in p, the cluster total representation with KNRM $\mathbf{R}_{\text{total\_cluster\_KNRM}}(C, p)$ of passage p and cluster C is defined as follows:

$$\mathbf{R}_{\text{total\_cluster\_KNRM}}(C, p) = \mathbf{R}_{\text{total\_cluster}}(C, p) \oplus \mathbf{S}_{\text{kernel\_pooling\_signal}}(C, p)$$

Algorithm: Multiple Cluster Total Representations

Algorithm 2 Multiple Cluster Total Representations of Passage

Input: Passage p, Maximum cluster size M
Output: Multiple cluster total representations of p

$E(p) \leftarrow$ set of all entities in p
$\mathit{clusters} \leftarrow \emptyset$
for every non-empty subset $C \subset E(p)$ with size $l \leq M$ **do**
    if $l = 1$ or (every pair of entities in C has Cosine similarity $> \beta$) **then**
        $\mathit{clusters} \leftarrow \mathit{clusters} \cup C$
    end if
end for
$\mathit{total\_reps} \leftarrow \emptyset$
for C in $\mathit{clusters}$ **do**
    $R_{C,p} \leftarrow$ cluster total representation of p with cluster C
    $\mathit{total\_reps} \leftarrow \mathit{total\_reps} \cup R_{C,p}$
end for
return $\mathit{total\_reps}$

Definitions: nDCG@10 I

$$\text{nDCG@10} = \frac{\text{DCG@10}}{\text{IDCG@10}}$$

$\Rightarrow$ In context of this paper rankings are based on a labeled four-point scale where 0 is non-relevant and 3 is perfectly relevant.

Definitions: nDCG@10 II

Derivation:

1. Discounted Cumulative Gain (DCG): The DCG at a particular position is calculated as the sum of the relevance scores of the ranked items up to that position, discounted by a logarithmic function.

$$\text{DCG@10} = \sum_{i=1}^{10} \frac{rel_i}{\log_2(i+1)}$$

2. Ideal Discounted Cumulative Gain (IDCG): The IDCG represents the maximum achievable DCG value at a given position.

$$\text{IDCG@10} = \sum_{i=1}^{10} \frac{rel_{(i)}}{\log_2(i+1)}$$

Definitions: MRR@10

$$\text{MRR@10} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\text{Rank}_i}$$

where N represents the total number of queries, and Rank_i represents the rank of the first relevant item (within the top 10) for the i-th query. If no relevant item was found for a query within the top 10, the respective value is set to be 0.

$\Rightarrow$ In context of this paper rankings are based on binarized judgments where four-point scale from nDCG@10 is used: Only labels of 2 and 3 are treated as relevant.

Definitions: MAP@1000

$$\text{MAP@1000} = \frac{1}{1000} \sum_{k=1}^{1000} \text{Precision@k} \times \text{Relevance@k}$$

where Precision@k represents the precision at position k and Relevance@k represents the binary relevance label (1 if relevant, 0 if non-relevant) at position k.

$\Rightarrow$ In context of this paper rankings are based on binarized judgments where four-point scale from nDCG@10 is used: Only labels of 2 and 3 are treated as relevant.

Training Data: Summary statistics of Queries

Entities	Training Queries		Testing Queries	
	Count	**Fraction**	**Count**	**Fraction**
0	130353	0.435	3442	0.483
1	149073	0.497	3232	0.454
2	19207	0.064	416	0.058
3+	1367	0.004	37	0.005
Total	300000		7127	
Average	0.640		0.587	

Table: Summary statistics of the queries.

Training Data: Summary Statistics of the Passage Collection

Entities	Training Passages		Testing Passages	
	Count	Fraction	Count	Fraction
0-2	201932	0.337	3309263	0.375
3-5	261200	0.435	3731425	0.422
6-7	82416	0.137	1103501	0.125
8+	54452	0.091	697634	0.078
Total	600000		8841823	
Average	3.87		3.63	

Table: Summary statistics of the passage collection.

Model Selection: Varying Aggregation Operators

Table: Varying Aggregation Operators

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

Hyperparameter Tuning: Varying Parameters M and β

- M = Upper Bound for Clusters when building multiple cluster representations
- β = Threshold of considering pairs of entities as similar / relevant.

Table: Varying Parameters M and β

Params M	Params β	Index	Dev nDCG	Dev MRR	Dev2E nDCG	Dev2E MRR
1	-	×3.6	0.406	0.347	0.236	0.203
2	0.9	×3.7	0.406	0.347	0.236	0.203
2	0.7	×5.0	0.405	0.347	0.234	0.204
2	0.5	×7.8	0.407	0.349	0.257	0.226
3	0.5	×13.5	0.407	0.349	0.256	0.226