BERT-ER: Query-specific BERT Entity Representations for Entity Ranking

Shubham Chatterjee
shubham.chatterjee@unh.edu
University of New Hampshire
Durham, New Hampshire, USA

Laura Dietz
dietz@cs.unh.edu
University of New Hampshire
Durham, New Hampshire, USA

ABSTRACT

Entity-oriented search systems often learn vector representations of entities via the introductory paragraph from the Wikipedia page of the entity. As such representations are the same for every query, our hypothesis is that the representations are not ideal for IR tasks. In this work, we present BERT Entity Representations (BERT-ER) which are *query-specific* vector representations of entities obtained from text that describes how an entity is relevant for a query. Using BERT-ER in a downstream entity ranking system, we achieve a performance improvement of 13–42% (Mean Average Precision) over a system that uses the BERT embedding of the introductory paragraph from Wikipedia on two large-scale test collections. Our approach also outperforms entity ranking systems using entity embeddings from Wikipedia2Vec, ERNIE, and E-BERT. We show that our entity ranking system using BERT-ER can increase precision at the top of the ranking by promoting relevant entities to the top. With this work, we release our BERT models and query-specific entity embeddings fine-tuned for the entity ranking task.[1]

CCS CONCEPTS

• **Information systems → Retrieval models and ranking**; *Learning to rank*; *Similarity measures.*

KEYWORDS

Query-specific Entity Representations, Entity Ranking, BERT

ACM Reference Format:
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 (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3477495.3531944

1 INTRODUCTION

An important aspect of entity-oriented research pertains to the representation of entities. Commonly, the vector representation (embedding) of the introductory paragraph (lead text) from an entity's Wikipedia page is used as the entity's representation [35, 36, 58]. An issue with using the lead text is that it is a *static* description of the entity: Often, the lead text contains only generic information about the entity that is the same for every query and may not even be relevant for the query. For example, the entity "Food and Drug Administration" (FDA) is relevant to the topical keyword query "Genetically Modified Organism" as an organization that approved and released a kind of genetically engineered insulin; however, the lead text from the Wikipedia page of the FDA does not contain this information (Figure 1). In fact, the lead text has been found to be useful as an entity's description in less than 50% cases for the ClueWeb12 collection [16]. As the lead text is a static textual description of the entity, the corresponding entity embedding is *static* in nature, i.e., the embedding is the same for every query.

> The United States Food and Drug Administration (FDA or USFDA) is a federal agency of the Department of Health and Human Services. The FDA is responsible for protecting and promoting public health through the control and supervision of food safety, tobacco products, dietary supplements [...] The FDA's primary focus is enforcement of the Federal Food, Drug, and Cosmetic Act [...]

Figure 1: Lead text from the Wikipedia page of the entity "Food and Drug Administration". This text is a generic description of the entity and does not highlight the connection between the entity and the query "Genetically Modified Organism". Hence, the corresponding entity embedding is static. Our intuition is that such static embeddings may not be ideal for IR tasks.

Similarly, while entity embeddings obtained using graph embedding methods [5, 33, 51, 56] encode the general semantics and knowledge of entities available in a Knowledge Graph, the embeddings are static. Recently, models such as ERNIE [62] and E-BERT [44] have been proposed in an effort to inject information from Knowledge Graphs into BERT [13]. However, these models too use a static textual description of the entity, either from Freebase or Wikipedia, resulting in static embeddings.

Static entity embeddings obtained using Wikipedia or Knowledge Graphs are easy to pre-compute and store. They have also been shown to be useful for downstream (query-independent) knowledge-driven NLP tasks such as entity linking [19, 44, 60], entity typing [43, 62] and relation classification [44, 62]. However, our intuition is that such embeddings may not be ideal for IR tasks.

Often, a query and document are matched in the entity-space [35, 36, 40, 58] through the similarity between the embedding of the

[1] Data and code available at: https://github.com/shubham526/SIGIR2022-BERT-ER

SIGIR '22, July 11–15, 2022, Madrid, Spain

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3531944

entities mentioned in the query and the document. Static entity embeddings without any knowledge of the query may not be able to identify when two entities are similar/related in the context of the query. For example, the Wikipedia page of the entity "Food and Drug Administration" does not mention the entity "Robert Swanson", yet these two entities are similar/related in the context of the query "Genetically Modified Organism" because Robert Swanson was the founder of the company that produced the first genetically engineered insulin approved for use by the Food and Drug Administration. Our hypothesis is that an entity embedding that incorporates *query-specific* knowledge about the entity would be more beneficial in a downstream IR task. In this work, we use a *query-specific* textual description of an entity to encode the query-relevant information about entities using BERT.

Task. Given a query and an entity, produce a query-specific dense vector representation (embedding) of the entity.

We evaluate the impact of our query-specific BERT Entity Representations (BERT-ER) on a downstream entity ranking task: Given a keyword query, return a ranked list of entities ordered by relevance.

The prevalent approach for representing entities through the BERT embedding of the lead text is appealing because it is simple to implement and use; unfortunately, it leads to poor results, as we demonstrate in our experimental evaluation. We provide an equally easy-to-implement approach for obtaining query-specific entity embeddings using BERT that performs much better than the prevalent approach. This improvement is achieved by incorporating query-relevant information about the entity in its representation. To this end, we explore the utility of three types of **query-specific textual descriptions** (Figure 2) of entities for learning query-specific entity embeddings using BERT:

- **Aspect (top-level section from Wikipedia).** We identify the relevant top-level sections from an entity's Wikipedia page, and use the text of the highest ranked section as the entity's query-specific description. Prior work [40, 46] refers to the top-level sections as an entity's *aspects*. In this work, we too refer to the top-level sections from Wikipedia as an entity's aspects. We discuss this in more detail in Section 3.2.
- **PRF-passage.** This is the simplest and most straightforward query-specific textual description of an entity. The approach is based on Pseudo-Relevance Feedback [31] and entity linking. We use the text of the highest ranked pseudo-relevant candidate passage that mentions an entity as the entity's query-specific textual description. We discuss this in more detail in Section 3.3.
- **Entity-support passage.** An entity's support passage [4, 8, 25] is a PRF-passage that mentions the entity and explains to a human, why an entity is relevant to a query. We use the text of the highest ranked support passage as an entity's query-specific description. We discuss this in more detail in Section 3.4.

Contributions. The novel contribution of this work is new knowledge about query-specific entity embeddings that will not only benefit the IR community but also other related research areas. In the experimental evaluation, we demonstrate the benefits of using our query-specific BERT entity embeddings using several large entity ranking benchmarks consisting of a diverse set of queries (question answering, keyword queries, list search queries, etc.).

- We obtain query-specific BERT Entity Representations (BERT-ER) by incorporating the query-relevant knowledge about an entity into its representation. This query-relevant knowledge is obtained using pseudo-relevant candidate passages, support passages, and relevant aspects (top-level sections from Wikipedia).
- Using BERT-ER in our entity ranking system, we outperform the entity ranking system that uses the BERT embedding of the lead text of entities by 13–42% on two large-scale entity ranking test collections. We also outperform systems using entity embeddings from Wikipedia2Vec [59], ERNIE [62], E-BERT [44].
- We provide a detailed empirical evaluation demonstrating that compared to the prevalent entity embedding methods, our query-specific BERT entity embeddings yield better performance for IR tasks such as entity ranking.

2 RELATED WORK

2.1 Knowledge-Enhanced BERT

Recently, much effort has been spent on injecting knowledge into BERT [13]. Zhang et al. [62] propose ERNIE, a neural language model that uses additional knowledge encoder layers to integrate the knowledge from entities into the textual information from the underlaying layers. Peters et al. [43] propose KnowBert, a knowledge-enhanced BERT model that explicitly models entity spans in the input text and uses an entity linker trained jointly with the model to retrieve relevant entity embeddings. Wang et al. [55] propose KEPLER, a model based on RoBERTa [34] that maps texts and entities onto the same semantic space using the same language model and jointly optimizes the Knowledge Embedding and the Masked Language Modeling objectives. While ERNIE and KnowBert are based on adapting BERT to entity embeddings and involve additional pre-training, E-BERT proposed by Poerner et al. [44] adapts entity embeddings to BERT without any pre-training. E-BERT aligns Wikipedia2Vec [59] entity vectors with BERT's wordpiece vectors. E-BERT has been shown to outperform BERT, ERNIE, and KnowBert on question-answering, relation classification, and entity linking.

2.2 Entity Embeddings

Bordes et al. [5] propose TransE which learns embeddings for both entities and relations based on the idea that the relationship r between two entities h and t corresponds to a translation between the embedding of these entities. However, TransE has problems dealing with reflexive, one-to-many, many-to-one, or many-to-many relations between entities. Wang et al. [56] propose TransH to overcome this issue by representing each relation r with two vectors: the norm vector $\mathbf{w}_r$, and the translation vector $\mathbf{d}_r$. Both TransE and TransH assume that entities and relations are embedded in the same space. Lin et al. [33] propose TransR to address this issue by modelling entities and relations in distinct entity space and multiple relation spaces. TransR projects $\mathbf{h}$ and $\mathbf{t}$ to the aspects that a relation r focuses on using relation-specific mapping matrix $\mathbf{M}_r$. However, this means that for relation r, all entities share the same $\mathbf{M}_r$ irrespective of their types or attributes. Ji et al. [24] propose TransD to address this issue by using a unique mapping matrix for every entity-relation pair.

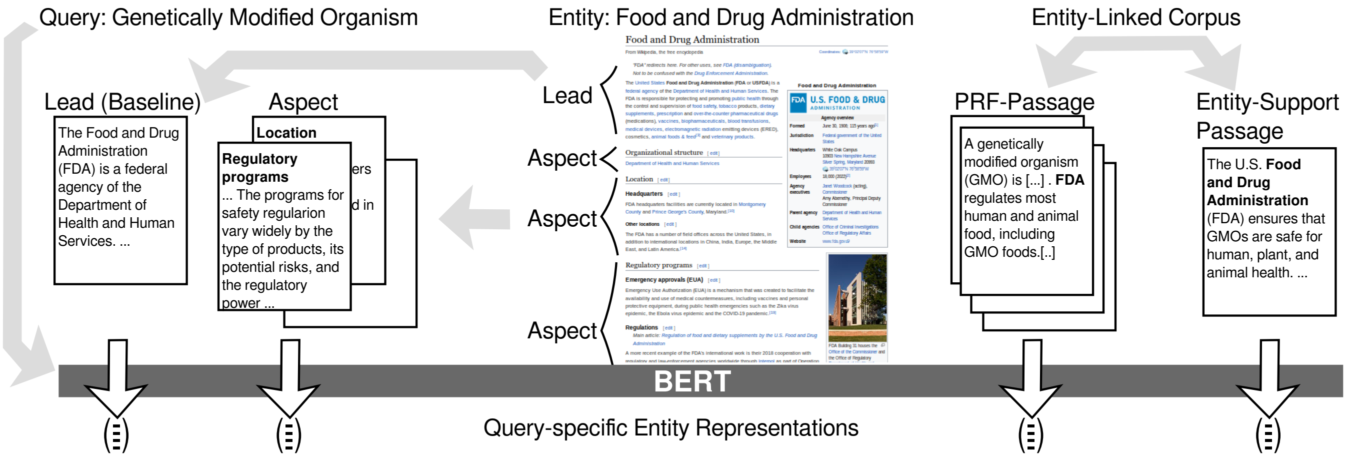

Query: Genetically Modified Organism
Entity: Food and Drug Administration
Entity-Linked Corpus
Lead (Baseline)
Aspect
The Food and Drug Administration (FDA) is a federal agency of the Department of Health and Human Services. ...
Location
Regulatory programs
... The programs for safety regularion vary widely by the type of products, its potential risks, and the regulatory power ...
Lead
Aspect
Aspect
Aspect
Food and Drug Administration
Organizational structure
Location
Regulatory programs
PRF-Passage
Entity-Support Passage
A genetically modified organism (GMO) is [...] . FDA regulates most human and animal food, including GMO foods.[..]
The U.S. Food and Drug Administration (FDA) ensures that GMOs are safe for human, plant, and animal health. ...
BERT
Query-specific Entity Representations

Figure 2: Example of query-specific entity representations for the query "Genetically Modified Organism" and entity "Food and Drug Administration". The lead text is a generic description of the entity and non-relevant for the query. The PRF-passage provides a query-specific description of the entity by first defining what a GMO is, then stating that the FDA regulates GMO food. As a result, the PRF-passage is a better textual description of the entity than the lead text. An issue with the PRF-passage is that the entity is not central to the discussion in the passage and the connection between FDA and GMO foods is made as a passing reference, i.e., passage is relevant to query but not to the entity. The support passage is a PRF-passage that is relevant to both the query and entity. The connection between entity and the query is central to the discussion in the support passage, and the support passage clarifies how the FDA regulates GMOs, including that the FDA allowed the use of the first genetically modified insulin (not shown in the figure). The aspect further clarifies the meaning of the entity in the context of the query – FDA is mentioned in the context of "Regulatory Programs" in the support passage; the text from the aspect elaborates on this deeper query-relevant meaning of the entity.

Xie et al. [57] propose a novel representation learning method for knowledge graphs taking advantages of entity descriptions present in knowledge bases. Yamada et al. [61] present *TextEnt*, a neural network model that learns distributed representations of entities and documents directly from a knowledge base using the introductory paragraph of an entity's Wikipedia article as descriptions.

Recently, Yamada et al. [59] proposed Wikipedia2Vec for learning embeddings of words and entities from Wikipedia based on the skip-gram model [38, 39]. Gerritse et al. [20] propose GEEER, an entity ranking system that re-ranks entities using Wikipedia2Vec. They show that entity embeddings from Wikipedia2Vec are useful for entity ranking. First, they compute the embedding-based score for an entity E as a weighted sum:

$$\text{Score}_{\text{emb}}(E, Q) = \sum_{e \in Q} C(e) \cdot \cos(\vec{E}, \vec{e}) \tag{1}$$

where $C(e)$ is the confidence score of each entity e in the query Q as returned by the entity linker TagMe [17]. The final score of an entity is derived from an interpolation of the embedding-based and retrieval scores:

$$\text{Score}_{\text{final}}(E, Q) = \lambda \cdot \text{Score}_{\text{emb}}(E, Q) + (1 - \lambda) \cdot \text{Score}_{\text{ret}}(E, Q)$$

where $\text{Score}_{\text{ret}}(E, Q)$ is the score of entity E obtained from a retrieval model, and $\lambda \in [0, 1]$.

We include the entity ranking method GEEER of Gerritse et al. [20] as a baseline. Further, we replace the Wikipedia2Vec embeddings in GEEER with embeddings from ERNIE and E-BERT, and include the resulting systems as baselines.

2.3 Entity Ranking

Ranking via unstructured retrieval models. Entity retrieval is often addressed in two steps. First, an *entity description* is constructed either from document corpora, Web documents such as the Wikipedia page of an entity, or knowledge bases such as DBpedia [32]. Second, these entity descriptions are ranked using a retrieval model such as BM25 [49], or language models. Alternatively, term dependencies may be incorporated using a Markov Random Field (MRF) [37]. For example, the Sequential Dependence Model (SDM) [37] based on the MRF assigns different weights to matching unigrams and bigarams of different types.

Ranking via fielded retrieval models. Fielded retrieval models are used to rank entities represented as a *fielded document* using, for example, the Wikipedia page of the entity. Zhiltsov et al. [63] propose the Fielded Sequential Dependence Model (FSDM) that estimates the feature functions for unigams and bigrams across multiple fields using field-specific background models. Nikolaev et al. [41] propose improvements over the FSDM by dynamically estimating the probability of unigrams and bigrams being mapped onto a field. Hasibi et al. [22] leverage entity links in queries and propose a parameter-free estimation of the field weights in FSDM.

Ranking via Learning-To-Rank. The current state-of-the-art entity ranking models are based on Learning-To-Rank (LTR). LTR approaches represent each query-entity pair as a feature vector, and learn the optimal way to combine these vectors through discriminative training. Schuhmacher et al. [52] propose several query-entity features, for example, whether the candidate entity is contained in the query, whether entities in queries and documents are connected in a Knowledge Graph, etc. Graus et al. [21] learn an optimal entity representation for entity retrieval by representing an entity as a fielded document. ENT-Rank [14] is a LTR model that combines information about an entity, the entity's neighbors, and context using a hypergraph. Recently, Chatterjee et al. [9] have shown that entity aspects are useful for entity retrieval. Their approach (referred to as LTR-ASP in this work) is based on LTR with a rich set of features derived from entity aspects. ENT-Rank and LTR-ASP represent the current state-of-the-art on our datasets. Hence, we include them as baselines.

Ranking via entity types. Type-aware entity retrieval models estimate the type-based similarity between an entity and the set of target types provided with the query. For example, Kaptein and Kamps [26] represent the types by concatenating the descriptions of entities that belong to that type, and then estimate the similarity by scoring the query against this representation. Balog et al. [1] represent the query and entity types using probability distributions, and then measure the similarity between the two distributions.

Ranking via entity relationships. Tonon et al. [53] address the ad-hoc entity retrieval task by identifying (relevant) entities from a Knowledge Graph that are related to entities present in a candidate set. Ciglan et al. [10] address the list search task of the Semantic Search Challenge [3] by identifying sets of semantically related entities from the underlying Knowledge Graph and scoring entities based on the relevance score of the sets it belongs to. Bron et al. [6] address the related entity finding task of the TREC Entity track [2] by modelling the relevance of an entity using a generative probabilistic model.

3 ENTITY REPRESENTATIONS

Given a query and an entity, we want to produce a query-specific dense vector representation (embedding) of the entity. In this work, we use *query-specific* entity descriptions, i.e., text that clarifies why an entity is relevant to a query, to create entity representations. Our assumption is that such a query-specific description provides a suitable and easy-to-implement method of providing the model with query-relevant information about an entity to learn the entity's embedding. We obtain query-specific BERT Entity Representations (BERT-ER) by fine-tuning BERT for the entity ranking task. We explore a range of different approaches (Sections 3.2 through 3.4) for obtaining query-specific entity descriptions and compare them empirically in Section 6.

3.1 Fine-tuning BERT

BERT-based neural re-ranking models such as MonoBERT and DuoBERT [42] have shown to be useful for the passage ranking task. Hence, we fine-tune a BERT model for entity ranking in two ways:

- Point-wise (MonoBERT-style) using the *cross-entropy loss.*
- Pair-wise (DuoBERT-style) using *margin ranking loss.*

The input to BERT is generated as follows:

$$\text{Input} = [CLS]\, t_1^q, t_2^q, ..., t_n^q\, [SEP]\, t_1^d, t_2^d, ..., t_m^d\, [SEP]$$

where t_i^q is a query token, t_j^d is a description token, and *[CLS]* and *[SEP]* are special tokens used by BERT. We use the L-dimensional embedding of the *[CLS]* token obtained from the last hidden layer of BERT as the query-specific embedding $\mathbf{e_Q}$ of an entity e.

Below, we discuss the different query-specific entity descriptions used to derive query-specific entity embeddings in this work.

3.2 Aspects: Top-Level Wikipedia Sections

We identify the top-level section from the Wikipedia page of an entity that is most relevant for the query, and use the text of the section to embed the entity. Following previous work [18, 40, 46, 48], we refer to the top-level sections from Wikipedia as *aspects*, and use a catalog of aspects provided by Ramsdell et al. [46].[2] This aspect catalog contains the top-level sections from the entire English Wikipedia together with section heading, text of the section, and the entities mentioned in the section.

To identify the most relevant top-level section (aspect) from an entity's Wikipedia page, we create a search index of aspects containing the full-text of all aspects from the catalog. We retrieve a candidate set of aspects (sections) $\mathcal{A}$ from this aspect index with the query using BM25.

An issue with directly using aspects from $\mathcal{A}$ is that many entities corresponding to the aspects in $\mathcal{A}$ may not even be relevant to the query. To remedy this, we leverage prior work on entity aspect linking. Entity Aspect Linking [40, 46] refines an entity link to an entity *aspect* link by clarifying the meaning of an entity from the context in which the entity has been mentioned, for example, the entity "Food and Drug Administration" in the context of its history or regulations.

We follow a useful assumption often encountered in entity-oriented research [11, 14, 47] to further improve the quality of the candidate set of aspects $\mathcal{A}$: The entities mentioned in passages from a candidate set of passages for the query are relevant for the query. We transfer this idea to entity aspects. First, we retrieve a candidate set of passages $\mathcal{D}$ for the query using BM25, then we retain only aspects $a \in \mathcal{A}$ that are linked to atleast one passage $p \in \mathcal{D}$ to obtain a filtered set of candidate aspects $\mathcal{A}'$. We use the text of the top-ranked aspect $a_e \in \mathcal{A}'$ of an entity e as the entity's description.

The downside of the the above approach is that often, Wikipedia articles are outdated or have some (negative) information removed. As a result, they do not contain all the query-relevant information.

3.3 Pseudo-Relevant Candidate Passages

To alleviate the above problem, we explore an alternative source of query-specific entity descriptions. We use ideas from Pseudo-Relevance Feedback [31] to obtain an entity's query-specific description: We use the candidate set of passages $\mathcal{D}$ for the query (obtained in Section 3.2) directly and use the text of the highest ranked passage $p_e \in \mathcal{D}$ that mentions the entity e (identified, for example, via entity links) as the entity's query-specific description.

This approach is easy to implement and based on a widely used Pseudo-Relevance Feedback technique. The downside is that although the candidate passage is relevant to the query, the entity may not be salient, i.e., *central* to the discussion in the passage, and

[2] https://www.cs.unh.edu/~dietz/eal-dataset-2020/

the connection between the query and entity may be made as a passing reference. In other words, the passage may be relevant to the query but not to the entity.

3.4 Entity Support Passage

To overcome the limitation from using PRF-passages as entity descriptions, we explore prior work on entity support passage retrieval [4, 8, 25] that identifies a passage that is relevant to both the query and the entity and elaborates on the connection between the query and entity. We extend the ideas from previous work on entity support passage retrieval to retrieve support passages for each entity (referred to as "target entity") in a candidate entity ranking. We use the entity ranking obtained using the combination of Pseudo-Relevance Feedback and Entity Context Model described in Section 4.3.

We want to focus on the relevant connections between the query and target entity while learning the query-specific representation of the target entity. The idea is that an entity (e.g., "Genetically Modified Crops") that is relevant to the query (e.g., Genetically Modified Organism) and mentioned frequently whenever the target entity (e.g., "Food and Drug Adminstration") is mentioned in some text, is a relevant connection between the query and target entity. Hence, we consider the other query-relevant entities that are mentioned frequently in the vicinity of the target entity as surrogates for such "relevant connections" between the query and target entity, and score a candidate support passage based on the number of such relevant connections (entities) the passage contains.

To implement this idea, we derive a filtered set of candidate passages $\mathcal{D}_e$ for a target entity e by retaining passages $p \in \mathcal{D}$ (obtained in Section 3.2) that mention the entity e. Then, we identify the k most frequently mentioned entities $e_x \in \mathcal{D}_e$. We re-rank passages $p \in \mathcal{D}_e$ for the entity e by the number of frequent entities e_x in the passage:

$$\text{Score}_e(p) = \sum_{e_x \in p} \text{Freq}(e_x \in \mathcal{D}_e)$$

where $\text{Freq}(e_x \in \mathcal{D}_e)$ is the number of times e_x appears in $\mathcal{D}_e$. We obtain the final score of a passage $p \in \mathcal{D}_e$ by interpolating the passage's score for the entity with the passage's score for the query (obtained from $\mathcal{D}$):

$$\text{Score}(p \mid e, Q) = \lambda \cdot \text{Score}_e(p) + (1 - \lambda) \cdot \text{Score}_Q(p) \quad \lambda \in [0, 1]$$

where λ is learnt using a machine learning method.

Additionally, we re-rank passages $p \in \mathcal{D}_e$ based on the salience[3] of the target entity e in the passage. Finally, we use the various support passage rankings obtained above as features to train a supervised Learning-To-Rank model, and produce one combined support passage ranking for each query and target entity. We use the text of the highest ranked support passage for each target entity as the target entity's query-specific description.

4 ENTITY RANKING

Given a keyword query Q, the entity ranking task is to return a ranked list of entities $\mathcal{E}$ from a Knowledge Graph ordered by relevance of each entity $e \in \mathcal{E}$ for the query Q. As discussed in Section 3.1, we use point-wise/pair-wise ranking methods to fine-tune BERT for the entity ranking task using our query-specific entity descriptions and obtain the vector representation $\mathbf{e}_Q$ of an entity $e \in \mathcal{E}$.

4.1 BERT-based Entity Ranking

To rank entities $e \in \mathcal{E}$ using BERT, we learn the scoring function $\text{Score}(e|Q)$ using the L-dimensional embedding $\mathbf{e}_Q$ of entity e as follows:

$$\text{Score}(e|Q) = W \cdot \mathbf{e}_Q{}^T + \mathbf{b}$$

where W is an $L \times L$ weight matrix, and $\mathbf{b}$ is an L-dimensional bias vector. We can implement this by passing the embedding $\mathbf{e}_Q$ through a fully-connected layer trained jointly with the model.

4.2 Combinations using Learning-To-Rank

As discussed in Section 2, the current state-of-the-art entity ranking models use list-wise Learning-To-Rank. Hence, using list-wise Learning-To-Rank, we combine the entity rankings obtained from BERT using the query-specific entity representations with other entity relevance features used in previous work [9, 11, 14] (discussed below) to obtain the final entity ranking for a query.

4.3 Other Entity Features

We include several entity features found to be useful by previous work in entity ranking. For example, considering each Wikipedia page to be an entity [27], we obtain an entity ranking by retrieving Wikipedia pages from a search index containing the full-text of Wikipedia pages. We also retrieve entities directly from an index containing the name and lead text of entities in a Knowledge Base.

Additionally, we also use an idea based on Pseudo-Relevance Feedback [31] and the Entity Context Model [11] that has been found to be a strong entity relevance indicator by prior work [9, 11, 14, 47] to obtain an entity ranking from a search index of paragraphs as follows: We represent a pseudo-relevant feedback set of paragraphs retrieved using the query as a bag-of-entities. To rank the entities in the bag, we weigh the frequency distribution of the entities by the retrieval score of the paragraphs.

5 EXPERIMENTAL METHODOLOGY

5.1 Datasets

We use two large-scale entity ranking test collections: TREC Complex Answer Retrieval and DBpedia-Entity v2.

TREC Complex Answer Retrieval (CAR). Given a topical keyword query such as "Genetically Modified Orgnism", the entity retrieval task of the TREC Complex Answer Retrieval (CAR) [15][4] track is to return a ranked list of entities based on whether the entity must, should, or could be mentioned in an article on this topic. In this work, we focus on the *page-level* queries, i.e., the title of the Wikipedia page as the query. The CAR dataset contains both manual and automatic entity ground truth, as well as an entity linked corpus consisting of paragraphs from the entire English Wikipedia. The automatic ground truth is constructed synthetically: all entities on the Wikipedia page corresponding to the query are relevant. The

[3] We use the salience detection system from Ponza et al. [45].

[4] http://trec-car.cs.unh.edu

manual ground truth was constructed after a manual assessment conducted by NIST using pool-based evaluation.

We use two subsets from the TREC CAR v2.1 data release.

- **BenchmarkY1-Train**. This subset is based on a Wikipedia dump from 2016. The ground truth is automatic with 117 page-level queries, and 3,031 positive entity assessments.
- **BenchmarkY2-Test**. This subset is based partly on a Wikipedia dump from 2018 and partly on the Textbook Question Answering [28] dataset (questions from middle school science curriculum). The ground truth is manual, with 27 page-level queries, and 3173 positive entity assessments.

The CAR dataset also contains a large collection of Wikipedia pages from 2016 in an easily parsable format (*unprocessedAllButBenchmark*). Query pages are excluded. We use this collection as our Knowledge Base to create the page/entity index representations used in Section 4.3.

DBpedia-Entity v2. The DBpedia-Entity v2 [23][5] is a collection of queries collected from several established entity retrieval benchmarking campaigns. It uses the DBpedia knowledge base (October 2015). The dataset contains the following categories of queries:

- **SemSearch ES** consisting of named entity queries (e.g., *brooklyn bridge*).
- **INEX-LD** consisting of IR-style keyword queries (e.g., *electronic music genres*).
- **List Search** consisting of queries which seek a list of entities (e.g., *Professional sports teams in Philadelphia*).
- **QALD-2** consisting of natural language queries (e.g., *Who owns Aldi?*).

Since we use the paragraphs, entity links, sections, etc. from Wikipedia, we use the version[6] of DBpedia-Entity v2 projected onto the Wikipedia dump from TREC CAR v2.1. Since our methods are not included in the assessment pool, we remove the unjudged entities retrieved by our methods to enable a fair comparison.

5.2 Evaluation Metrics

We use Mean Average Precision (MAP), Precision at R (P@R), and Normalized Discounted Cumulative Gain at 100 (NDCG@100) as our evaluation metrics. We conduct significance testing using paired-t-tests.

5.3 Details of Learning-To-Rank

We perform list-wise Learning-To-Rank (LTR) using Coordinate Ascent optimized for MAP. We use the toolkit Ranklips[7] for this purpose. We use 5-fold cross-validation for fine-tuning the BERT model as well as training the LTR model on both TREC CAR and DBpedia-Entity v2. The different subsets of queries available in the DBpedia-Entity v2 collection and CAR BenchmarkY2-Test were merged for training. Further details for using Ranklips can be found in the online appendix for this paper.

5.4 Feature Generating Retrieval Models

We produce entity rankings using the following retrieval models: (1) BM25, and (2) Query Likelihood with Dirichlet Smoothing ($\mu = 1500$), both with and without RM3-style query expansion. We use the default implementation of BM25 in Lucene. We also use the implementation of Query Likelihood in Lucene.

5.5 Details of BERT Fine-Tuning

Our model is implemented in PyTorch using HuggingFace. We use the `bert-base-uncased` version of BERT. For fine-tuning our model, we use the PyTorch implementation of the Cross-Entropy Loss[8] and Margin Ranking Loss.[9] The model is fine-tuned using the Adam [29] optimizer with a learning rate of $2e - 5$ and batch size of 8. We also use a linear learning rate schedule with 1000 warm-up steps.

5.6 Baselines

We include the following entity ranking systems as baselines:

(1) **BERT-LeadText++.** We fine-tune BERT for entity ranking using the *lead text* from an entity's Wikipedia page. The resulting entity rankings are used as features within a Learning-To-Rank system with other entity features (Section 4).
(2) **GEEER [20].** The entity retrieval system from Gerritse et al. [20] (described in Section 2) using Wikipedia2Vec [59] to re-rank entities.
(3) **GEEER-ERNIE.** Same as GEEER but using ERNIE [62] instead of Wikipedia2Vec.
(4) **GEEER-E-BERT.** Same as GEEER but using E-BERT [44] instead of Wikipedia2Vec.
(5) **GEEER-BERT.** Same as GEEER but using BERT [44] instead of Wikipedia2Vec. We use the *name* of the Wikipedia page of the entity to embed the entity using BERT.
(6) **ENT-Rank [14]** A Learning-To-Rank model that uses entity, neighbors, and text features.
(7) **LTR-ASP [9].** A Learning-To-Rank model that uses features based on entity aspects and entity support passages.
(8) **BM2F-CA.** Best-performing system on the DBpedia-Entity v2 dataset provided by the creators.

5.7 Research Questions

We address the following research questions in this work:

RQ1 Is it sufficient to use the lead text of an entity's Wikipedia page as the entity's description? Are query-specific entity descriptions better?

RQ2 To what extent do query-specific entity descriptions help improve entity ranking performance? What is the reason for this performance improvement?

RQ3 How do embeddings obtained using BERT-ER compare to those obtained using Wikipedia2vec for entity ranking? Which of these is better?

[5] https://github.com/iai-group/DBpedia-Entity
[6] https://github.com/TREMA-UNH/DBpediaV2-entity-CAR
[7] https://www.cs.unh.edu/~dietz/rank-lips/
[8] https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html
[9] https://pytorch.org/docs/stable/generated/torch.nn.MarginRankingLoss.html

Table 1: Results on BenchmarkY1-Train page-level using automatic ground truth. Trained using 5-fold cross-validation. $^\blacktriangle$ denotes significant improvement and $^\blacktriangledown$ denotes significant deterioration compared to BERT-LeadText++ (denoted $\star$) using a paired-t-test at $p < 0.05$.

	MAP	P@R	NDCG@100
BERT-LeadText++*	0.38*	0.41*	0.49*
GEEER [20]	0.15$^\blacktriangledown$	0.21$^\blacktriangledown$	0.30$^\blacktriangledown$
GEEER-E-BERT	0.13$^\blacktriangledown$	0.18$^\blacktriangledown$	0.26$^\blacktriangledown$
GEEER-ERNIE	0.14$^\blacktriangledown$	0.19$^\blacktriangledown$	0.26$^\blacktriangledown$
GEEER-BERT	0.14$^\blacktriangledown$	0.21$^\blacktriangledown$	0.28$^\blacktriangledown$
ENT-Rank [14]	0.32$^\blacktriangledown$	0.36$^\blacktriangledown$	0.46$^\blacktriangledown$
LTR-ASP [9]	0.49$^\blacktriangle$	0.50$^\blacktriangle$	0.63$^\blacktriangle$
BERT-ER++	**0.54**$^\blacktriangle$	**0.54**$^\blacktriangle$	**0.66**$^\blacktriangle$

6 RESULTS AND DISCUSSIONS

The overall results on CAR BenchmarkY1-Train are shown in Table 1, on CAR BenchmarkY2-Test in Table 2, and on DBpedia-Entity v2 in Table 3 (only best baselines shown due to lack of space). Below, we discuss the results with reference to the research questions outlined in Section 5.7. We use the query "Genetically Modified Organism" (GMO) as an illustrative query throughout our discussions below. In Tables 1 to 3, we refer to our entity ranking system as *BERT-ER++*. BERT-ER++ is the Learning-To-Rank combination of entity features described in Section 2.3 and entity rankings obtained by fine-tuning BERT using query-specific entity descriptions. In Table 4, BERT-ER is the Learning-To-Rank combination of all entity rankings obtained from BERT using query-specific entity descriptions (excluding the entity features described in Section 2.3).

6.1 Overall Results

From Tables 1 to 3, we observe that our entity ranking system BERT-ER++ outperforms all baselines in terms of all evaluation measures on both datasets. For example, on CAR BenchmarkY1-Train in Table 1, in comparison to BERT-LeadText++, we obtain an improvement of 42% in terms of MAP ($MAP = 0.38$ to $MAP = 0.54$). Similar results are observed in Tables 2 and 3. BERT-ER (Table 4) and BERT-ER++ especially improve on the recall-oriented measures MAP and NDCG@100. This shows that query-specific entity descriptions are more informative and useful than the lead text of an entity's Wikipedia article that has often been used in prior work. We discuss this further in Section 6.2.

BERT-ER and BERT-ER++ also obtain statistically significant improvements over the entity re-ranking systems using recent and state-of-the-art entity embedding methods: Wikipedia2Vec [59], ERNIE [62], and E-BERT [44]. For example, on CAR BenchmarkY1-Train in Table 1, GEEER [20] using Wikipedia2Vec obtain $MAP = 0.15$, GEEER-E-BERT obtains $MAP = 0.13$, and GEEER-ERNIE obtaines $MAP = 0.14$; our system BERT-ER++ obtains $MAP = 0.54$. Similar results are observed in Tables 2 and 3. This shows that our query-specific entity representations are able to capture the relevance of an entity for a query in a better way. We discuss this further with respect to Wikipedia2Vec in Section 6.3.

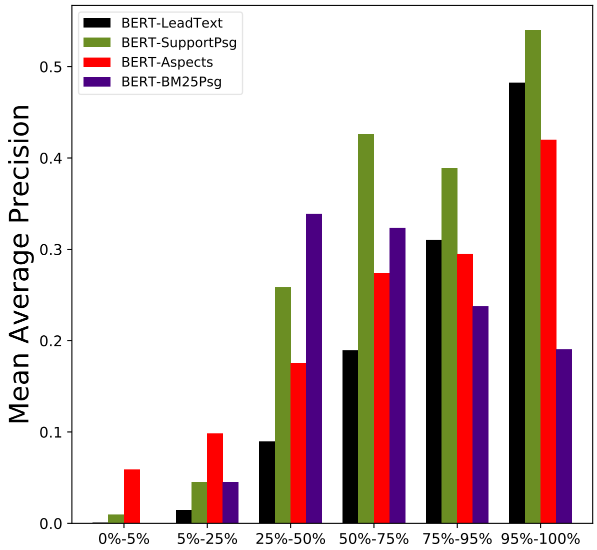


Figure 3: Difficulty test for MAP on CAR BenchmarkY1-Train, comparing entity rankings obtained by fine-tuning BERT using different query-specific entity descriptions. Baseline: BERT-LeadText. 5% most difficult queries for BERT-LeadText to the left and the 5% easiest ones to the right. Performance reported as macro-averages across queries. For the difficult queries (0-50%), relevant entities are found using BM25 passages, entity support passages, entity aspects. Hence, our entity ranking system outperforms several baselines.

6.2 Importance of Query-Specific Descriptions

To investigate why BERT-ER++ performs so well, we remove the other entity features from BERT-ER++ and analyze the results obtained by only BERT-ER. The results are shown in Table 4. This table shows the results of fine-tuning BERT for entity ranking using the individual query-specific entity descriptions obtained in Section 3 as well as a Learning-To-Rank combination of these (denoted as BERT-ER in the table). We use Equation 1 to rank entities using Wikipedia2Vec, E-BERT and ERNIE. We show results for only CAR BenchmarkY1-Train and DBpedia-Entity v2 (All).

Ablation study. From Table 4, we observe that BERT-ER outperforms BERT-LeadText on both datasets. On CAR BenchmarkY1-Train, BERT-ER achieves $MAP = 0.34$ whereas BERT-LeadText achieves $MAP = 0.16$. On DBpedia-Entity v2, BERT-ER achieves $MAP = 0.22$ whereas BERT-LeadText achieves $MAP = 0.07$. We also observe that BERT-SupportPsg, BERT-Aspects, and BERT-ER outperform Wikipedia2Vec, E-BERT and ERNIE on both datasets. This is because Wikipedia2Vec, E-BERT and ERNIE produce *query-independent* entity embeddings (embeddings have no knowledge of the query) using a query-independent textual description of an entity (often, the Freebase description or lead text). Hence, their performance on an IR task (here, entity ranking) is not good. BERT-SupportPsg and BERT-Aspects use *query-specific* entity embeddings obtained using query-specific entity descriptions. As a result,

Table 2: Results on BenchmarkY2-Test (separated by its subsets on Wikipedia and TQA) page-level using the manual ground truth. Trained using 5-fold cross-validation. $^{\blacktriangle}$ denotes significant improvement and $^{\blacktriangledown}$ denotes significant deterioration compared to $\star$. ENT-Rank results on BenchmarkY2-Test page-level unavailable.

	All			Textbook Question Answering			Wikipedia		
	MAP	P@R	NDCG@100	MAP	P@R	NDCG@100	MAP	P@R	NDCG@100
BERT-LeadText++*	0.25*	0.29*	0.44*	0.25*	0.28*	0.46*	0.24*	0.28*	0.40*
GEEER [20]	0.06$^{\blacktriangledown}$	0.11$^{\blacktriangledown}$	0.18$^{\blacktriangledown}$	0.06$^{\blacktriangledown}$	0.10$^{\blacktriangledown}$	0.19$^{\blacktriangledown}$	0.07$^{\blacktriangledown}$	0.12$^{\blacktriangledown}$	0.17$^{\blacktriangledown}$
GEEER-E-BERT	0.04$^{\blacktriangledown}$	0.08$^{\blacktriangledown}$	0.13$^{\blacktriangledown}$	0.03$^{\blacktriangledown}$	0.07$^{\blacktriangledown}$	0.12$^{\blacktriangledown}$	0.07$^{\blacktriangledown}$	0.12$^{\blacktriangledown}$	0.18$^{\blacktriangledown}$
GEEER-ERNIE	0.04$^{\blacktriangledown}$	0.08$^{\blacktriangledown}$	0.14$^{\blacktriangledown}$	0.03$^{\blacktriangledown}$	0.05$^{\blacktriangledown}$	0.10$^{\blacktriangledown}$	0.05$^{\blacktriangledown}$	0.09$^{\blacktriangledown}$	0.13$^{\blacktriangledown}$
GEEER-BERT	0.02$^{\blacktriangledown}$	0.07$^{\blacktriangledown}$	0.09$^{\blacktriangledown}$	0.01$^{\blacktriangledown}$	0.04$^{\blacktriangledown}$	0.06$^{\blacktriangledown}$	0.05$^{\blacktriangledown}$	0.12$^{\blacktriangledown}$	0.14$^{\blacktriangledown}$
LTR-ASP [9]	0.24	0.31$^{\blacktriangle}$	0.46$^{\blacktriangle}$	0.29	0.34$^{\blacktriangle}$	0.52$^{\blacktriangle}$	0.29	0.32$^{\blacktriangle}$	0.47$^{\blacktriangle}$
BERT-ER++	**0.33**$^{\blacktriangle}$	**0.36**$^{\blacktriangle}$	**0.54**$^{\blacktriangle}$	**0.33**$^{\blacktriangle}$	**0.37**$^{\blacktriangle}$	**0.55**$^{\blacktriangle}$	**0.34**$^{\blacktriangle}$	**0.36**$^{\blacktriangle}$	**0.50**$^{\blacktriangle}$

Table 3: Results on DBpedia-Entity v2 (separated by different subsets). Trained using 5-fold cross-validation. $^{\blacktriangle}$ denotes significant improvement and $^{\blacktriangledown}$ denotes significant deterioration compared to $\star$. Only best baselines shown.

	All			SemSearch_ES			ListSearch			INEX_LD			QALD2		
	MAP	P@R	NDCG@100	MAP	P@R	NDCG@100	MAP	P@R	NDCG@100	MAP	P@R	NDCG@100	MAP	P@R	NDCG@100
BERT-LeadText++*	0.45*	0.41*	0.68*	0.60*	0.54*	0.77*	0.43*	0.40*	0.69*	0.43*	0.40*	0.69*	0.34*	0.32*	0.60*
BM25F-CA [23]	0.45	0.43$^{\blacktriangle}$	0.68	0.61	0.55	0.78	0.44	0.43$^{\blacktriangle}$	0.68	0.42$^{\blacktriangledown}$	0.41$^{\blacktriangle}$	0.67$^{\blacktriangledown}$	0.37$^{\blacktriangle}$	0.36$^{\blacktriangle}$	0.46$^{\blacktriangledown}$
ENT-Rank [14]	0.48$^{\blacktriangle}$	0.44$^{\blacktriangle}$	0.71$^{\blacktriangle}$	0.59	0.50$^{\blacktriangledown}$	0.78	0.49$^{\blacktriangle}$	0.47$^{\blacktriangle}$	0.74$^{\blacktriangle}$	0.43	0.42$^{\blacktriangle}$	0.70$^{\blacktriangle}$	0.40$^{\blacktriangle}$	0.37	0.64$^{\blacktriangle}$
GEEER [20]	0.37$^{\blacktriangledown}$	0.38$^{\blacktriangledown}$	0.57$^{\blacktriangledown}$	0.56$^{\blacktriangledown}$	0.53$^{\blacktriangledown}$	0.72$^{\blacktriangledown}$	0.34$^{\blacktriangledown}$	0.38$^{\blacktriangledown}$	0.54$^{\blacktriangledown}$	0.34$^{\blacktriangledown}$	0.35$^{\blacktriangledown}$	0.55$^{\blacktriangledown}$	0.27$^{\blacktriangledown}$	0.29$^{\blacktriangledown}$	0.48$^{\blacktriangledown}$
LTR-ASP [9]	0.43$^{\blacktriangledown}$	0.39$^{\blacktriangledown}$	0.68	0.55$^{\blacktriangledown}$	0.47$^{\blacktriangledown}$	0.74$^{\blacktriangledown}$	0.43	0.41$^{\blacktriangle}$	0.69	0.41$^{\blacktriangledown}$	0.38$^{\blacktriangledown}$	0.67$^{\blacktriangledown}$	0.36$^{\blacktriangle}$	0.32	0.62$^{\blacktriangle}$
BERT-ER++	**0.50**$^{\blacktriangle}$	**0.46**$^{\blacktriangle}$	**0.72**$^{\blacktriangle}$	**0.63**$^{\blacktriangle}$	**0.57**$^{\blacktriangle}$	**0.81**$^{\blacktriangle}$	**0.51**$^{\blacktriangle}$	**0.47**$^{\blacktriangle}$	**0.74**$^{\blacktriangle}$	**0.47**$^{\blacktriangle}$	**0.44**$^{\blacktriangle}$	**0.71**$^{\blacktriangle}$	**0.41**$^{\blacktriangle}$	**0.38**$^{\blacktriangle}$	**0.65**$^{\blacktriangle}$

Table 4: Ablation study. Results on CAR BenchmarkY1-Train (Automatic) and DBpedia-Entity v2 (All) for entity ranking using different types of embeddings. $^{\blacktriangle}$ denotes significant improvement and $^{\blacktriangledown}$ significant deterioration compared to $\star$.

	CAR Y1-Train (Automatic)			DBpedia-Entity v2 (All)		
	MAP	P@R	NDCG@100	MAP	P@R	NDCG@100
BERT-LeadText*	0.16*	0.20*	0.25*	0.07*	0.08*	0.12*
Wikipedia2Vec [59]	0.10$^{\blacktriangledown}$	0.16$^{\blacktriangledown}$	0.23$^{\blacktriangledown}$	0.05$^{\blacktriangledown}$	0.07$^{\blacktriangledown}$	0.10$^{\blacktriangledown}$
E-BERT [44]	0.11$^{\blacktriangledown}$	0.13$^{\blacktriangledown}$	0.19$^{\blacktriangledown}$	0.09$^{\blacktriangle}$	0.15$^{\blacktriangle}$	0.22$^{\blacktriangle}$
ERNIE [62]	0.05$^{\blacktriangledown}$	0.10$^{\blacktriangledown}$	0.14$^{\blacktriangledown}$	0.09$^{\blacktriangle}$	0.12$^{\blacktriangle}$	0.16$^{\blacktriangle}$
BERT-BM25Psg	0.06$^{\blacktriangledown}$	0.07$^{\blacktriangledown}$	0.11$^{\blacktriangledown}$	0.08$^{\blacktriangle}$	0.10$^{\blacktriangle}$	0.14$^{\blacktriangle}$
BERT-SupportPsg	0.29$^{\blacktriangle}$	0.32$^{\blacktriangle}$	0.44$^{\blacktriangle}$	0.14$^{\blacktriangle}$	0.16$^{\blacktriangle}$	0.24$^{\blacktriangle}$
BERT-Aspects	0.22$^{\blacktriangle}$	0.28$^{\blacktriangle}$	0.37$^{\blacktriangle}$	0.18$^{\blacktriangle}$	0.21$^{\blacktriangle}$	0.30$^{\blacktriangle}$
BERT-ER	**0.34**$^{\blacktriangle}$	**0.36**$^{\blacktriangle}$	**0.48**$^{\blacktriangle}$	**0.22**$^{\blacktriangle}$	**0.23**$^{\blacktriangle}$	**0.35**$^{\blacktriangle}$

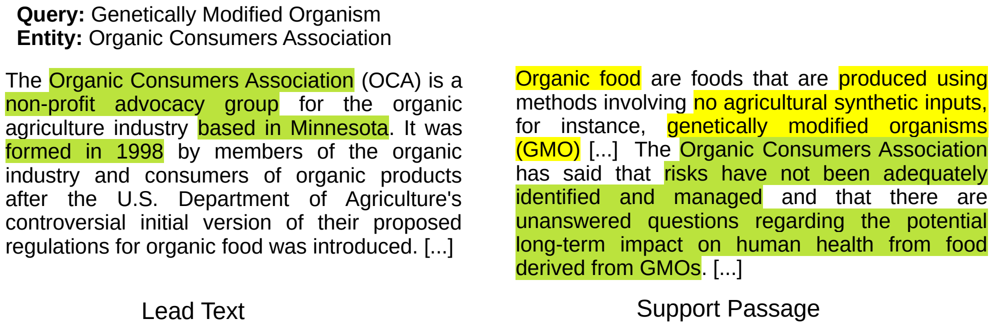
BERT-ER (combining BERT-SupportPsg, BERT-Aspects, and BERT-BM25Psg) is able to differentiate between relevant and non-relevant entities better and outperforms all other methods.

Lead text versus query-specific descriptions. To investigate the source of performance improvements due to query-specific entity descriptions, we analyze the results at the query-level by dividing the queries into different levels of difficulty according to the performance (MAP) of BERT-LeadText. We put the 5% most difficult queries for BERT-LeadText to the left and the 5% easiest ones to the right. Below, we discuss the results only with respect to CAR BenchmarkY1-Train but similar results are obtained on the other benchmarks.

From Figure 3, we observe that BERT-SupportPsg, BERT-Aspects, and BERT-BM25Psg perform well on the "difficult" queries (e.g., bins 0-50%) on which BERT-LeadText performs poorly. BERT-SupportPsg is always better than BERT-LeadText, even for queries where BERT-LeadText performs the best (bin 95–100%). BERT-Aspects are better than BERT-LeadText on 75% of the queries. We also notice that BERT-BM25Psg is complementary to BERT-LeadText: When the performance of BERT-LeadText is low, BERT-BM25Psg performs well, for example, in bin 25–50%, and vice-versa.

We find that BERT-SupportPsg improves performance (*helps*) on 92 queries, BERT-Aspects helps 95 queries, and BERT-BM25Psg helps 18 queries. On inspecting the top-100 entities for some queries that are helped, we find that compared to BERT-LeadText, BERT-SupportPsg, BERT-Aspects, and BERT-BM25Psg place relevant entities higher in the ranking. For example, BERT-LeadText places the relevant entity "Organic Consumers Association" at rank 57 whereas BERT-SupportPsg places it at rank 13 (see Figure 4). By promoting relevant entities higher up in the ranking, query-specific entity descriptions help to improve the precision at the top of the ranking. Moreover, we are able to improve performance on the "difficult" queries for BERT-LeadText using query-specific entity descriptions.

BM25 passage as description. From Table 4, we observe that on CAR BenchmarkY1-Train, BERT-BM25Psg obtains $MAP = 0.06$ whereas BERT-SupportPsg obtains $MAP = 0.29$. Using the difficulty test above, we find that BERT-SupportPsg obtains $MAP = 0.30$ on the lower 0–50% (difficult) queries where BERT-BM25Psg obtains $MAP = 0.15$. This shows that using an entity's support passage as



Query: Genetically Modified Organism
Entity: Organic Consumers Association

The Organic Consumers Association (OCA) is a non-profit advocacy group for the organic agriculture industry based in Minnesota. It was formed in 1998 by members of the organic industry and consumers of organic products after the U.S. Department of Agriculture's controversial initial version of their proposed regulations for organic food was introduced. [...]

Lead Text

Organic food are foods that are produced using methods involving no agricultural synthetic inputs, for instance, genetically modified organisms (GMO) [...] The Organic Consumers Association has said that risks have not been adequately identified and managed and that there are unanswered questions regarding the potential long-term impact on human health from food derived from GMOs. [...]

Support Passage

Figure 4: Example query and entity with description. Left: Lead text. The passage contains only generic knowledge (highlighted in green) about the entity and does not elaborate upon the connection between the query and entity. Right: Support passage. The passage contains information about both, the query (highlighted in yellow) and the entity (green), and elaborates that the entity "Organic Consumers Association" is relevant to GMOs because it regulates GMO food. This query-relevant knowledge helps BERT-SupportPsg learn that the entity is relevant for the query and promotes it up the ranking, from rank-57 placed by BERT-LeadText to rank-13.

its description is better than using a query-relevant BM25-passage that mentions the entity. As discussed in Section 3.4, this is because sometimes, the entity may not be salient to the discussion in the BM25-passage, and the connection between the query and entity may be made as a passing reference, i.e., although the passage is relevant to the query, it is non-relevant for the entity (see Figure 2 for example). A support passage addresses this issue because the support passage retrieval method only considers passages which are relevant to the query and mention the entity in a salient manner.

Take-away. Regarding **RQ1**, it is not always sufficient to use the lead text of an entity as the entity's description; query-specific entity descriptions are better. Regarding **RQ2**, although BERT-LeadText++ often performs well, our system BERT-ER++ using query-specific entity descriptions improves entity ranking performance by 13–42% over BERT-LeadText++. On its own (without other entity features from related work), BERT-ER outperforms not only BERT-LeadText but also entity rankings obtained using Wikipedia2Vec, ERNIE, and E-BERT. This performance boost is due to our system's ability to promote relevant entities to the top of the ranking while demoting non-relevant entities to the bottom. This is possible because query-specific descriptions help our model to learn query-relevant information and minimize the non-relevant information.

6.3 Analysis of Query-Specific BERT-ER

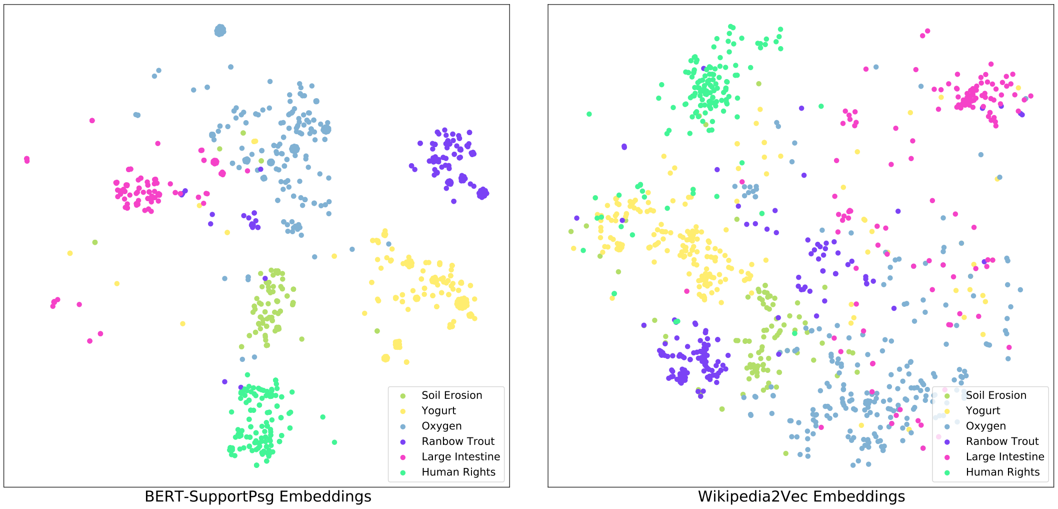
From Tables 1 to 4, BERT-ER and BERT-ER++ obtain statistically significant improvements over the entity re-ranking systems using entity embeddings from Wikipedia2Vec, ERNIE, and E-BERT. To analyze the source of this performance improvement, we compare the performance of our query-specific BERT-ER to Wikipedia2Vec (as our work heavily relies on Wikipedia and uses Wikipedia as a Knowledge Base). We use the Wikipedia2Vec entity embeddings with the graph component made available by Gerritse et al. [20].[10]

Overall results. From Table 4, we observe that BERT-ER outperforms Wikipedia2Vec on both CAR BenchmarkY1-Train and DBpedia-Entity v2. When performing the difficulty test described in Section 6.2, we find that BERT-ER obtains $MAP = 0.30$ for queries in the lower 0–50% range where Wikipedia2Vec obtains $MAP = 0.05$. Moreover, considering the query-specific descriptions individually, we observe that BERT-SupportPsg and BERT-Aspects consistently outperform Wikipedia2Vec on both datasets. This suggests that compared to Wikipedia2Vec, the entity embeddings obtained from BERT using query-specific entity descriptions capture the similarity/relevance of the entity for the query in a better way.

To verify this, we inspect the entity rankings for the query GMO obtained using Wikipedia2Vec and BERT-SupportPsg in Table 4. We find that BERT-SupportPsg places the relevant entity "Robert Swanson" at the top of the ranking (rank 3) compared to Wikipedia2Vec that places the entity at the bottom (rank 8). Moreover, BERT-SupportPsg demotes the non-relevant entity "Developmental Biology" that is placed higher by Wikipedia2Vec (rank 3) to the bottom of the ranking (rank 10). Our intuition is that this is because entity embeddings obtained using BERT-SupportPsg are query-specific and encode the query-relevant knowledge about the entity that is helpful for determining the entity's relevance for the query. The success of our approach demonstrates that the lead text often does not relate to the query, and compensating with query-specific descriptions leads to better performance.

Context-dependent entity relatedness. As discussed in Section 1, queries and documents are often matched in the entity-space through the cosine similarity of the embeddings of entities mentioned in the query and document. Hence, it is important that the entity embeddings are able to capture context-dependent similarity between entities. For example, the entities "Food and Drug Administration" and "Robert Swanson" are related in the context of GMOs since Robert Swanson was the founder of the company that produced the first genetically engineered insulin approved for use by the Food and Drug Administration. We find that compared to Wikipedia2Vec, our query-specific BERT entity embeddings capture this context-dependent similarity between two entities in a better way. For example, compared to Wikipedia2Vec, BERT-SupportPsg assigns a higher similarity to the entities above.

[10] Available from: https://github.com/informagi/GEEER



BERT-SupportPsg Embeddings

Wikipedia2Vec Embeddings

Figure 5: Visualizing clusters of relevant entities using t-SNE. We observe that relevant entities are better clustered using BERT-SupportPsg (left) than using Wikipedia2Vec (right).

Table 5: Results on BenchmarkY1-Train for clustering relevant entities. Evaluation measures: David-Bouldin score (lower better), Silhouette score (higher better), and Calinski-Harabasz score (higher better).

	David-Bouldin	Silhouette	Calinski-Harabasz
BERT-SupportPsg	3.87	-0.03	22.75
Wikipedia2Vec	5.29	-0.12	20.30

Clustering entities using embeddings. As an additional evaluation, we assess whether the embeddings satisfy the cluster hypothesis [30]: documents (entities) relevant to a query cluster together. We consider the embeddings of relevant entities as points in a vector space to be clustered and evaluate the quality of the resulting clusters. We use the following three metrics for evaluation: David-Bouldin score [12] (lower scores better), Silhouette score [50] (higher scores better), and Calinski-Harabasz score [7] (higher scores better). From Table 5, we observe that clusters formed using embeddings from BERT-SupportPsg are better than clusters formed using Wikipedia2Vec. We also present a t-SNE [54] visualization of the resulting clusters for some example queries in CAR BenchmarkY1-Train. As we observe from Figure 5, the relevant entities for a query (e.g., "Yogurt", and "Oxygen") are close together, and the clusters are better separated using BERT-SupportPsg than using Wikipedia2Vec.

Take-away. Regarding **RQ3**, our query-specific BERT-ER outperforms Wikipedia2Vec on all datasets. BERT-ER finds relevant entities for the (difficult) queries for which Wikipedia2Vec fails because compared to Wikipedia2vec, BERT-ER captures the context-dependent similarity between query-entity pairs in a better way. BERT-ER can promote relevant entities to the top of the ranking while demoting the non-relevant entities to the bottom.

7 CONCLUSION

We present BERT-ER, query-specific BERT Entity Representations learnt by fine-tuning BERT for the entity ranking task. In contrast to the prevalent approach of using the static lead text from an entity's Wikipedia page as the entity's description, we study the utility of three types of *query-specific* entity descriptions: pseudo-relevant candidate passage, entity support passage and entity aspect.

Using BERT-ER for entity ranking, we obtain a performance improvement of 13–42% (MAP) over a system using the lead text as the entity's description, across a diverse range of queries from two large-scale entity ranking test collections. We also outperform entity ranking systems using Wikipedia2Vec, E-BERT, and ERNIE. We show that query-specific descriptions help an entity ranking system by promoting relevant entities to the top of the ranking, thereby increasing the precision at the top of the ranking. We also demonstrate that compared to Wikipedia2Vec, BERT-ER representations can identify when entities are related in the context of the query in a better way. We also show, both qualitatively and quantitatively, that compared to Wikipedia2Vec, our query-specific BERT-ER produce better clusters of relevant entities.

In the long-term, we believe that our easy-to-implement approach to query-specific entity representations will lead to significant improvements in diverse IR and text analysis tasks, including question answering, and summarization. By demonstrating the importance of query-specific entity descriptions, we hope to promote more research in this area.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 1846017. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Krisztian Balog, Marc Bron, and Maarten De Rijke. 2011. Query Modeling for Entity Search Based on Terms, Categories, and Examples. *ACM Transactions on Information Systems* 29, 4, Article 22 (Dec. 2011), 31 pages. https://doi.org/10.1145/2037661.2037667

[2] Krisztian Balog, Pavel Serdyukov, and Arjen P. de Vries. 2010. *Overview of the TREC 2010 Entity Track.* Technical Report. Norwegian University of Science and Technology.

[3] Roi Blanco, Harry Halpin, Daniel M Herzig, Peter Mika, Jeffrey Pound, Henry S Thompson, and T Tran Duc. 2011. Entity Search Evaluation Over Structured Web Data. In *Proceedings of the 1st International Workshop on Entity-oriented Search* (Beijing, China) *(SIGIR '11, Vol. 14)*. Association for Computing Machinery, New York, NY, USA, 2181–2187.

[4] Roi Blanco and Hugo Zaragoza. 2010. Finding Support Sentences for Entities. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Geneva, Switzerland) *(SIGIR '10)*. Association for Computing Machinery, New York, NY, USA, 339–346. https://doi.org/10.1145/1835449.1835507

[5] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-Relational Data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2* (Lake Tahoe, Nevada) *(NIPS'13)*. Curran Associates Inc., Red Hook, NY, USA, 2787–2795.

[6] Marc Bron, Krisztian Balog, and Maarten de Rijke. 2013. Example Based Entity Search in the Web of Data. In *Advances in Information Retrieval, Proceedings of the 35th European Conference on IR Research (ECIR 2013)* (Moscow, Russia) *(Lecture Notes in Computer Science)*. Springer, Berlin, Heidelberg, 392–403. https://doi.org/10.1007/978-3-642-36973-5_33

[7] T. Caliński and J. Harabasz. 1974. A Dendrite Method for Cluster Analysis. *Communications in Statistics* 3, 1 (1974), 1–27. https://doi.org/10.1080/03610927408827101

[8] Shubham Chatterjee and Laura Dietz. 2019. Why Does This Entity Matter? Support Passage Retrieval for Entity Retrieval. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval* (Santa Clara, CA, USA) *(ICTIR '19)*. Association for Computing Machinery, New York, NY, USA, 221–224. https://doi.org/10.1145/3341981.3344243

[9] Shubham Chatterjee and Laura Dietz. 2021. Entity Retrieval Using Fine-Grained Entity Aspects. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, Canada) *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 1662–1666. https://doi.org/10.1145/3404835.3463035

[10] Marek Ciglan, Kjetil Nørvåg, and Ladislav Hluchý. 2012. The SemSets Model for Ad-Hoc Semantic List Search. In *Proceedings of the 21st International Conference on World Wide Web* (Lyon, France) *(WWW '12)*. Association for Computing Machinery, New York, NY, USA, 131–140. https://doi.org/10.1145/2187836.2187855

[11] Jeffrey Dalton, Laura Dietz, and James Allan. 2014. Entity Query Feature Expansion Using Knowledge Base Links. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Gold Coast, Queensland, Australia) *(SIGIR '14)*. Association for Computing Machinery, New York, NY, USA, 365–374. https://doi.org/10.1145/2600428.2609628

[12] David L. Davies and Donald W. Bouldin. 1979. A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-1, 2 (1979), 224–227. https://doi.org/10.1109/TPAMI.1979.4766909

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[14] Laura Dietz. 2019. ENT Rank: Retrieving Entities for Topical Information Needs through Entity-Neighbor-Text Relations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Paris, France) *(SIGIR'19)*. Association for Computing Machinery, New York, NY, USA, 215–224. https://doi.org/10.1145/3331184.3331257

[15] Laura Dietz and John Foley. 2019. TREC CAR Y3: Complex Answer Retrieval Overview.. In *Proceedings of Text REtrieval Conference (TREC)*.

[16] Laura Dietz, Michael Schuhmacher, and Simone Paolo Ponzetto. 2014. Queripidia: Query-specific Wikipedia Construction. *Proceedings of the Automatic Knowledge Base Construction (AKBC) Workshop* (2014).

[17] Paolo Ferragina and Ugo Scaiella. 2010. TAGME: On-the-Fly Annotation of Short Text Fragments (by Wikipedia Entities). In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management* (Toronto, ON, Canada) *(CIKM '10)*. Association for Computing Machinery, New York, NY, USA, 1625–1628. https://doi.org/10.1145/1871437.1871689

[18] Besnik Fetahu, Katja Markert, and Avishek Anand. 2015. Automated News Suggestions for Populating Wikipedia Entity Pages. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management* (Melbourne, Australia) *(CIKM '15)*. Association for Computing Machinery, New York, NY, USA, 323–332. https://doi.org/10.1145/2806416.2806531

[19] Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep Joint Entity Disambiguation with Local Neural Attention. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, 2619–2629. https://doi.org/10.18653/v1/D17-1277

[20] Emma J Gerritse, Faegheh Hasibi, and Arjen P de Vries. 2020. Graph-Embedding Empowered Entity Retrieval. In *Advances in Information Retrieval, Proceedings of the 42nd European Conference on Information Retrieval (ECIR 2020)* (Lisbon, Portugal) *(Lecture Notes in Computer Science)*. Springer, Cham, 97–110. https://doi.org/10.1007/978-3-030-45439-5_7

[21] David Graus, Manos Tsagkias, Wouter Weerkamp, Edgar Meij, and Maarten de Rijke. 2016. Dynamic Collective Entity Representations for Entity Ranking. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining* (San Francisco, California, USA) *(WSDM '16)*. Association for Computing Machinery, New York, NY, USA, 595–604. https://doi.org/10.1145/2835776.2835819

[22] Faegheh Hasibi, Krisztian Balog, and Svein Erik Bratsberg. 2016. Exploiting Entity Linking in Queries for Entity Retrieval. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval* (Newark, Delaware, USA) *(ICTIR '16)*. Association for Computing Machinery, New York, NY, USA, 209–218. https://doi.org/10.1145/2970398.2970406

[23] 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* (Shinjuku, Tokyo, Japan) *(SIGIR '17)*. Association for Computing Machinery, New York, NY, USA, 1265–1268. https://doi.org/10.1145/3077136.3080751

[24] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge Graph Embedding via Dynamic Mapping Matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Beijing, China, 687–696. https://doi.org/10.3115/v1/P15-1067

[25] Amina Kadry and Laura Dietz. 2017. Open Relation Extraction for Support Passage Retrieval: Merit and Open Issues. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Shinjuku, Tokyo, Japan) *(SIGIR '17)*. Association for Computing Machinery, New York, NY, USA, 1149–1152. https://doi.org/10.1145/3077136.3080744

[26] Rianne Kaptein and Jaap Kamps. 2013. Exploiting the Category Structure of Wikipedia for Entity Ranking. *Artificial Intelligence* 194 (Jan. 2013), 111–129. https://doi.org/10.1016/j.artint.2012.06.003

[27] Rianne Kaptein, Pavel Serdyukov, Arjen De Vries, and Jaap Kamps. 2010. Entity Ranking Using Wikipedia as a Pivot. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management* (Toronto, ON, Canada) *(CIKM '10)*. Association for Computing Machinery, New York, NY, USA, 69–78. https://doi.org/10.1145/1871437.1871451

[28] Aniruddha Kembhavi, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Are You Smarter Than a Sixth Grader? Textbook Question Answering for Multimodal Machine Comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4999–5007.

[29] Diederik P Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980* (2014).

[30] Oren Kurland. 2014. The Cluster Hypothesis in Information Retrieval. In *Advances in Information Retrieval, Proceedings of the 36th European Conference on IR Research (ECIR 2014)* (Amsterdam, The Netherlands) *(Lecture Notes in Computer Science)*. Springer, 823–826. https://doi.org/10.1007/978-3-319-06028-6_105

[31] Victor Lavrenko and W. Bruce Croft. 2001. Relevance-Based Language Models. *SIGIR Forum* 51, 2 (Aug. 2001), 260–267. https://doi.org/10.1145/3130348.3130376

[32] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2012. DBpedia–A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web Journal* 6, 2 (2012), 167–195. https://doi.org/10.3233/SW-140134

[33] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence* (Austin, Texas) *(AAAI'15)*. AAAI Press, 2181–2187.

[34] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR* abs/1907.11692 (2019). arXiv:1907.11692 http://arxiv.org/abs/1907.11692

[35] Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2018. Entity-Duet Neural Ranking: Understanding the Role of Knowledge Graph Semantics in Neural Information Retrieval. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, 2395–2405. https://doi.org/10.18653/v1/P18-1223

[36] Jarana Manotumruksa, Jeff Dalton, Edgar Meij, and Emine Yilmaz. 2020. CrossBERT: A Triplet Neural Architecture for Ranking Entity Properties. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, China) *(SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 2049–2052. https://doi.org/10.1145/3397271.3401265

[37] Donald Metzler and W. Bruce Croft. 2005. A Markov Random Field Model for Term Dependencies. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Salvador, Brazil) *(SIGIR '05)*. Association for Computing Machinery, New York, NY, USA, 472–479. https://doi.org/10.1145/1076034.1076115

[38] Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of the 2013 International Conference on Learning Representations*.

[39] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S., and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems*.

[40] Federico Nanni, Simone Paolo Ponzetto, and Laura Dietz. 2018. Entity-Aspect Linking: Providing Fine-Grained Semantics of Entities in Context. In *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries* (Fort Worth, Texas, USA) *(JCDL '18)*. Association for Computing Machinery, New York, NY, USA, 49–58. https://doi.org/10.1145/3197026.3197047

[41] Fedor Nikolaev, Alexander Kotov, and Nikita Zhiltsov. 2016. Parameterized Fielded Term Dependence Models for Ad-Hoc Entity Retrieval from Knowledge Graph. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Pisa, Italy) *(SIGIR '16)*. Association for Computing Machinery, New York, NY, USA, 435–444. https://doi.org/10.1145/2911451.2911545

[42] Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-Stage Document Ranking with BERT. *CoRR* abs/1910.14424 (2019). arXiv:1910.14424 http://arxiv.org/abs/1910.14424

[43] Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge Enhanced Contextual Word Representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 43–54. https://doi.org/10.18653/v1/D19-1005

[44] 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, Online, 803–818. https://doi.org/10.18653/v1/2020.findings-emnlp.71

[45] Marco Ponza, Paolo Ferragina, and Francesco Piccinno. 2018. SWAT: A System for Detecting Salient Wikipedia Entities in Texts. *Computational Intelligence* (04 2018). https://doi.org/10.1111/coin.12216

[46] Jordan Ramsdell and Laura Dietz. 2020. A Large Test Collection for Entity Aspect Linking. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management* (Virtual Event, Ireland) *(CIKM '20)*. Association for Computing Machinery, New York, NY, USA, 3109–3116. https://doi.org/10.1145/3340531.3412875

[47] Hadas Raviv, David Carmel, and Oren Kurland. 2012. A Ranking Framework for Entity Oriented Search Using Markov Random Fields. In *Proceedings of the 1st Joint International Workshop on Entity-Oriented and Semantic Search* (Portland, Oregon, USA) *(JIWES '12)*. Association for Computing Machinery, New York, NY, USA, Article 1, 6 pages. https://doi.org/10.1145/2379307.2379308

[48] Ridho Reinanda, Edgar Meij, and Maarten de Rijke. 2016. Document Filtering for Long-Tail Entities. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* (Indianapolis, Indiana, USA) *(CIKM '16)*. Association for Computing Machinery, New York, NY, USA, 771–780. https://doi.org/10.1145/2983323.2983728

[49] Stephen Robertson and Hugo Zaragoza. 2009. *The probabilistic relevance framework: BM25 and beyond.* Now Publishers Inc.

[50] Peter J. Rousseeuw. 1987. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. *J. Comput. Appl. Math.* 20 (1987), 53–65. https://doi.org/10.1016/0377-0427(87)90125-7

[51] Andrew Runge and Eduard Hovy. 2020. Exploring Neural Entity Representations for Semantic Information. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*. Association for Computational Linguistics, Online, 204–216. https://doi.org/10.18653/v1/2020.blackboxnlp-1.20

[52] Michael Schuhmacher, Laura Dietz, and Simone Paolo Ponzetto. 2015. Ranking Entities for Web Queries Through Text and Knowledge. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management* (Melbourne, Australia) *(CIKM '15)*. Association for Computing Machinery, New York, NY, USA, 1461–1470. https://doi.org/10.1145/2806416.2806480

[53] Alberto Tonon, Gianluca Demartini, and Philippe Cudré-Mauroux. 2012. Combining Inverted Indices and Structured Search for Ad-Hoc Object Retrieval. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Portland, Oregon, USA) *(SIGIR '12)*. Association for Computing Machinery, New York, NY, USA, 125–134. https://doi.org/10.1145/2348283.2348304

[54] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing Data Using t-SNE. *Journal of Machine Learning Research* 9, 11 (2008).

[55] Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021. KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation. *Transactions of the Association for Computational Linguistics* 9 (2021), 176–194. https://doi.org/10.1162/tacl_a_00360

[56] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence* (Québec City, Québec, Canada) *(AAAI'14)*. AAAI Press, 1112–1119.

[57] Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. 2016. Representation Learning of Knowledge Graphs with Entity Descriptions. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence* (Phoenix, Arizona) *(AAAI'16)*. AAAI Press, 2659–2665.

[58] Chenyan Xiong, Russell Power, and Jamie Callan. 2017. Explicit Semantic Ranking for Academic Search via Knowledge Graph Embedding. In *Proceedings of the 26th International Conference on World Wide Web* (Perth, Australia) *(WWW '17)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1271–1279. https://doi.org/10.1145/3038912.3052558

[59] Ikuya Yamada, Akari Asai, Jin Sakuma, Hiroyuki Shindo, Hideaki Takeda, Yoshiyasu Takefuji, and Yuji Matsumoto. 2020. Wikipedia2Vec: An Efficient Toolkit for Learning and Visualizing the Embeddings of Words and Entities from Wikipedia. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, 23–30.

[60] Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2016. Joint Learning of the Embedding of Words and Entities for Named Entity Disambiguation. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Berlin, Germany, 250–259. https://doi.org/10.18653/v1/K16-1025

[61] Ikuya Yamada, Hiroyuki Shindo, and Yoshiyasu Takefuji. 2018. Representation Learning of Entities and Documents from Knowledge Base Descriptions. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, Santa Fe, New Mexico, USA, 190–201. https://aclanthology.org/C18-1016

[62] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced Language Representation with Informative Entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 1441–1451. https://doi.org/10.18653/v1/P19-1139

[63] Nikita Zhiltsov, Alexander Kotov, and Fedor Nikolaev. 2015. Fielded Sequential Dependence Model for Ad-Hoc Entity Retrieval in the Web of Data. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Santiago, Chile) *(SIGIR '15)*. Association for Computing Machinery, New York, NY, USA, 253–262. https://doi.org/10.1145/2766462.2767756