Neural Entity Context Models

Pooja H. Oza
pho1003@wildcats.unh.edu
University of New Hampshire
Durham, NH, USA

Shubham Chatterjee
shubham.chatterjee@ed.ac.uk
University of Edinburgh
Edinburgh, Scotland

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

ABSTRACT

A prevalent approach of entity-oriented systems involves retrieving relevant entities by harnessing knowledge graph embeddings. These embeddings encode entity information in the context of the knowledge graph and are static in nature. Our goal is to generate entity embeddings that capture what renders them relevant for the query. This differs from entity embeddings constructed with static resource, for example, E-BERT. Previously, Dalton et al. [3] demonstrated the benefits obtained with the Entity Context Model, a pseudo-relevance feedback approach based on entity links in relevant contexts. In this work, we reinvent the Entity Context Model (ECM) for neural graph networks and incorporate pre-trained embeddings. We introduce three entity ranking models based on fundamental principles of ECM: (1) Graph Attention Networks, (2) Simple Graph Relevance Networks, and (3) Graph Relevance Networks. Graph Attention Networks and Graph Relevance Networks are the graph neural variants of ECM, that employ attention mechanism and relevance information of the relevant context respectively to ascertain entity relevance. Our experiments demonstrate that our neural variants of the ECM model significantly outperform the state-of-the-art BERT-ER [2] by more than 14% and exceeds the performance of systems that use knowledge graph embeddings by over 101%. Notably, our findings reveal that leveraging the relevance of the relevant context is more effective at identifying relevant entities than the attention mechanism. To evaluate the efficacy of the models, we conduct experiments on two standard benchmark datasets, DBpediaV2 and TREC Complex Answer Retrieval. To aid reproducibility, our code and data are available.[1]

ACM Reference Format:
Pooja H. Oza, Shubham Chatterjee, and Laura Dietz. 2023. Neural Entity Context Models. In *Woodstock '23: ACM Symposium on Neural Gaze Detection, June 03–05, 2023, Woodstock, NY*. ACM, New York, NY, USA, 10 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Entity Ranking is a task that involves retrieving a ranked list of entities from a knowledge repository such as Wikipedia, for a given query. It is an active research area and plays a pivotal role in various NLP tasks, including entity linking, question answering, and more.

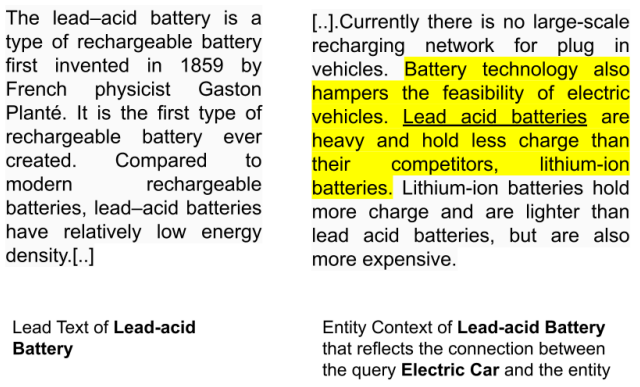

Figure 1: Example of query *Electric car* and the relevant entity *Lead-acid battery*. On the left side, the lead text of *Lead-acid battery* is given (https://en.wikipedia.org/wiki/Lead-acid_battery). The lead text contains the static and generic description of the entity and does not contain any relation between the query and the entity. On the right side, the pseudo-relevance feedback document, which serves as entity context, elaborates on the connection between the relevant entity *Lead-acid battery* and the query *Electric car*.

A predominant approach for entity ranking is using entity embeddings obtained using graph embedding methods. These knowledge graph entity embeddings are static embeddings that encode the semantics and knowledge of entities in the context of a Knowledge Graph. Recently, knowledge-enhanced pre-trained language models such as ERNIE [36] and E-BERT [22] have been proposed. These models integrate entity information from the Knowledge Graph into BERT embeddings. However, such embeddings also inject static entity information, either through textual description or knowledge graph entity embeddings, which leads to static embeddings. These entity embeddings are used to determine the relevance of entities by capturing the similarity between the entities and the entities mentioned in the queries. The static nature of such entity embeddings does not capture the information of what makes an entity relevant for a query. Our work aims to address this limitation by creating embeddings that specifically capture the information that makes an entity relevant to a given query.

In 2014, Dalton et al. [3] demonstrated that entity features can be leveraged to obtain relevant entities which can be further used to drive the text ranking task. In particular, they found that the entity ranking feature, named Entity Context Model (ECM), based on retrieved text passages that contain query terms and entity links is the most effective entity ranking feature. The overarching idea of ECM, which we depict in Figure 2, is that the entity context in

[1] https://github.com/TREMA-UNH/neural-entity-context-models.git

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

the relevant documents can help us to identify relevant entities. Entity links serve to disambiguate mentions of entities in the text by connecting them to their corresponding entities in a knowledge base. Thereby, entity links introduce connections between relevant text and relevant entities. Consequently, it can be stated that entities are represented through their entity contexts.

In our work, we focus on enhancing the entity ranking task by adapting the fundamental idea of ECM. The ECM model demonstrates that the entity contexts help to capture intricate relationships between the queries and the entities, aiding in the identification of relevant entities. For instance, for the query *"Electric Car"*, one of the relevant entities is "`Lead-acid battery`". This relevant entity is not explicitly mentioned in the Wikipedia page of *"Electric Car"* to indicate a direct connection. The textual description of the relevant entity "`Lead-acid battery`" also does not contain any clear signal that indicates a connection between the query *"Electric Car"* and the relevant entity, as shown in Figure 1. However, the entity context obtained via pseudo-relevance feedback documents contains information that shows the connection between the query and the relevant entity through the entity link.

To achieve our goal, we generate entity embeddings that capture the contextual information contributing to the identification of an entity's relevance to a specific query. In essence, we aim to specialize the entity embeddings for the task of entity ranking, thereby making the retrieval process more adaptive to the query. To this end, we introduce three entity ranking models which are variants of the ECM approach: (1) **Graph Attention Networks (GAT)**, (2) **Simple Graph Relevance Network (Sim-GRN)** and (3) **Graph Relevance Networks (GRNs)**. **GAT**, is a neural variant of ECM that incorporates neural components through graph neural networks and pre-trained embeddings. This variant uses the attention mechanism to model the entity representations. While **Sim-GRN** and **GRN** models are also built on the foundational principles of ECM, we further utilize the relevance information of the entity contexts as a signal to enhance entity representations by capturing relevance to the query. **Sim-GRN** is a simple entity ranking feature that utilizes the relevance information to determine the relevance of entities without incorporating any neural components. **GRN** is a neural variant that also capitalizes on the relevance information of entity contexts while incorporating the neural components through graph neural networks and pre-trained embeddings. In contrast to **GAT**, which relies on attention mechanism, *GRN* utilizes relevance information of entity contexts to model the entity representations. In summary, we introduce three variants of the ECM approach as below:

- **GAT:** A graph neural network variant that utilizes attention mechanism
- **Sim-GRN:** A simple traditional entity ranking feature which uses relevance information
- **GRN:** A graph neural variant of ECM based on relevance information

Entity Ranking Task: Given a user's information need q, we return a list of entities E ranked by their relevance to the query. We assume access to a text corpus consisting of text passages with entity links. These entity links can be obtained via entity linkers such as WAT [21] or REL [27] or by hyperlinks to Wikipedia Pages. We also assume that for each linked entity, we have a text description available which is a lead text of the entity's Wikipedia page.

We evaluate the performance on two standard entity ranking benchmarks, DBpediaV2 and TREC Complex Answer Retrieval.

Contributions: The novel contribution of this work is entity retrieval models that are based on the foundational principle of ECM such that they generate entity embeddings that capture the relevancy of entities for the queries for the entity ranking task.

- We reinvent the ECM model using pre-trained embeddings and graph neural networks and show that the underlying assumptions of ECM hold even in the neural version.
- We introduce the Graph Relevance Networks model which incorporates relative relevance information of the entity contexts to determine entities relevance.
- Our graph neural models outperform the SOTA BERT-ER [2] baseline by 17-80% and also outperform entity ranking systems that use knowledge graph embeddings.

2 RELATED WORK

2.1 Knowledge-enhanced BERT Models

Knowledge-enhanced BERT models infuse knowledge into the BERT model through knowledge graph embeddings such as TransE [1] and Wiki2Vec [34]. ERNIE [36] integrates entity information in the BERT model by utilizing TransE entity embeddings during the pretraining phase. It aligns the TransE entity embeddings with the BERT word embedding corresponding to the initial wordpiece token of each entity mention to generate encoded embeddings in a common embedding space. E-BERT [22] adapts Wiki2Vec entity embeddings to BERT without additional pretraining. Utilizing the shared embedding space of Wiki2Vec, E-BERT learns a weight matrix by linearly transforming Wiki2Vec word embeddings into BERT-like embeddings. Using the learned weight matrix, it constructs a function to align the entity embeddings of Wiki2Vec with the BERT word embeddings. KEPLER [30] utilizes entity descriptions corresponding to the entities in relation triples and jointly optimizes Knowledge Graph and Language Model representations. KELM [16] injects knowledge in the BERT model via multi-relational subgraphs from the Knowledge Graph and text.

2.2 Knowledge Graph Embeddings

Knowledge Graph embeddings serve as vector-based representations for entities in a Knowledge Graph. These embeddings encapsulate both semantic and structural characteristics of the entities they represent. Bordes et al. [1] introduced a model called TransE, which employs a translational approach to learn the embeddings of both entities and their associated relations. In TransE, the model operates under the assumption that a relation $\boldsymbol{r}$ acts as a translational link between two entities $\boldsymbol{h}$ and $\boldsymbol{t}$. Both entities and relations are mapped to a common vector space in TransE. However, a limitation of TransE is that it is geared toward 1-to-1 relations and struggles with 1-to-N, N-to-1, and N-to-N relationships. To address this, the TransH [31] model was developed, which assigns two vectors to each relation $\boldsymbol{r}$. Another model, TransR [14], takes it a step further by assigning a unique vector space for each relation $\boldsymbol{r}$, into which the entities $\boldsymbol{h}$ and $\boldsymbol{t}$ are then projected in the context of that relation.

Yamada et al. [35] have introduced Wiki2Vec, a model that learns embeddings for both entities and words by leveraging text and structural data from Wikipedia. Gerritse et al. [6] uses Wiki2Vec knowledge graph embeddings to determine the similarity score between the initial entity candidate set and the entities linked in the queries as described in Equation 1.

$$F(E,Q) = \sum_{e \in E(Q)} s(e) \cdot \cos(\overrightarrow{E}, \overrightarrow{e}) \quad (1)$$

The initial entity candidate set is then re-ranked using interpolation, through the Learning-to-Rank approach, with the similarity-scored ranking.

$$score_{total}(E,Q) = (1-\lambda) \cdot score_{other}(E,Q) + \lambda \cdot F(E,Q) \quad \lambda \in [0,1]$$

We use it as a reference baseline. Additionally, we also provide other baselines where we replace Wiki2Vec entity embeddings with ERNIE and E-BERT entity embeddings in GEEER [7] system.

2.3 Entity Ranking

Ranking through Fielded Retrieval Models. Models based on the Markov Random Field [17] represent a joint distribution over the terms from an entity's description and the information from semi-structured data about the entity. For example, the Sequential Dependence Model [17] and its variants [19, 37] estimate the weights for unigrams and bigrams by representing entities using multiple fields. Hasibi et al. [10] estimate the field weights using entity annotations in the queries whereas Raviv et al. [23] model the different representations of an entity (description, type, and name) jointly with the query terms.

Ranking through Probabilistic Models. Liu and Fang [15] propose Latent Entity Space (LES) based on a generative probabilistic framework that constructs a high-dimensional latent entity space. In contrast, Xiong and Callan [32] proposes EsdRank, which is a discriminative framework that marginalizes over a joint distribution of entities and documents. Raviv et al.[24] suggest entity-based language models, while Xiong et al. [33] use a duet model of entities and words.

Ranking through Pseudo-Relevance Feedback Documents. Schuhmacher et al. [25] employs pseudo-relevance feedback method for entity ranking by utilizing entity links found in web documents. The entities receive higher rankings when they are mentioned in the higher ranking feedback documents. ENT-Rank [4] combines information about an entity, the entity's neighbors, and context using Learning-To-Rank on a hypergraph of entities. Various features from feedback runs such as entity mentions, entity co-occurrences, etc. and features from entities are combined to determine the relevance of entities.

Ranking through Language Models. Recent work has also focused on using Transformers [28] for entity ranking. EM-BERT [8] incorporates the Wiki2Vec graph embeddings into BERT and performs a two-stage fine-tuning on passage and entity ranking tasks. BERT-ER [2] leverages BERT to generate query-specific entity representations using query-specific entity descriptions and evaluate them on the entity ranking task. BERT-ER employs various entity descriptions constructed using aspects, pseudo-relevance candidate passages, and entity support passages. They combine various features such as frequency of the entity, entity salience, etc. to select the entity support passage that is relevant to both the query and entity. Our work differs from BERT-ER as we learn to model the entity representations through entity contexts of every entity in our ranking model.

3 BACKGROUND

The underlying assumption of the Entity Context Model (ECM) is that the more relevant the entities, the more frequently they are mentioned near query terms in relevant documents. The original ECM as proposed by Dalton et al. [3], first retrieves documents using a traditional text retrieval model, such as query likelihood, the Sequential Dependence Model [18], or BM25. The top k documents are then selected as a feedback set, and entity links within these documents are identified. For each entity link, an entity context consisting of 50 tokens (or alternatively 8 tokens) to the left and right is extracted. For any entity e that has an entity link in the feedback set, all corresponding link contexts are collected and concatenated to form a pseudo-document representation for that entity. Using the same traditional text retrieval model, all entities are then ranked based on their pseudo-documents.

4 APPROACH

In this section, we introduce three variants based on the foundational principles of the ECM approach.

4.1 Graph Attention Networks

Graph perspective: The original ECM model can be conceptualized as a message-passing on a graph, by denoting each passage d and entity e in the feedback set as a node in the graph. Every mention of an entity e in a passage d is represented as an edge (d, e). Therefore, each entity e has a neighborhood of passages d where the edges are directed from d to e. Edge weights and initial node weights are set to 1.0. The process of concatenating the link contexts to form a pseudo-document representation for entity e can be considered as aggregating the neighborhood of node e.

$$\vec{e} = (\sum_{d \in P_e} 1.0 \cdot \vec{d}) \quad (2)$$

Graph Attention Networks (GAT):. We revitalize the ECM model in the neural version through graph neural networks in our GAT model. Neural Message-Passing Graph Networks [9, 38] serve as a generalization of traditional random walk models and are applicable to deep learning. Instead of heuristically defining scalar updates, they learn vector-valued message representations that propagate along edges to optimize a training objective.

In graph neural networks, a recurrence of updates to node[2] representations e is based on messages sent from adjacent nodes d. These messages can be composed of a weight $\vec{a}_{d \to e}$ and a node representation $\vec{d}$. The messages are then aggregated and used to

[2]For consistency with ECM, we refer to source nodes as d and target nodes as e. We note that these ideas can be generalized to any graph.

update the node representation e.

$$\vec{e} := \texttt{upd}(\texttt{proj}\,(\vec{e}) + \sum_{d:d\sim e} \underbrace{\vec{a}_{d\to e} \cdot \texttt{proj}\left(\vec{d}\right)}_{\text{message from } d \text{ to } e})$$

In this work, $\texttt{proj}\,(\ldots)$ refers to linear projections with bias term in latent space.

A popular example of such networks are GATs proposed by Veličković et al. [29], where the message weight $\vec{a}$ is modeled with multi-head scaled dot product attention. After projecting each node representation into multiple query, key, and value vectors, the attention is computed as $\vec{a}_{d\to e} = \frac{1}{\sqrt{d_k}}\texttt{proj}\left(\vec{d}\right) \cdot \texttt{proj}\,(\vec{e})$. The attention scores are then used to compute a weighted sum of the value vectors $\texttt{proj}\left(\vec{d}\right)$, which is used to update the representation $\vec{e}$ of the receiving node.

The underlying assumption of this paradigm is that the node representations of entities $\vec{e}$ and passages $\vec{d}$ are sufficiently expressive to model the relevance of edges. However, there is a downside to this approach whenever the graph contains many non-relevant nodes. To avoid the non-relevant information overwhelming the node representation of an entity, the attention mechanism would need to learn when to reduce the edge weight to zero, thereby removing edges that would lead to wrong ranking decisions. In the case of GATs the notion of relevance would need to be recovered from the representations of passages $\vec{d}$ and entities $\vec{e}$, which can be challenging when the representations are not sufficiently expressive and $\vec{d}$ not have knowledge of the query.

These representations are commonly derived from textual content, such as BERT for passages or entities. This is addressed by query-specific node representations, such as a MonoBERT-style cross-encoder [20] or a ColBERT-style late-interaction models [12] or DPR-style bi-encoders [11] of query and passage text.

4.2 Simple Graph Relevance Networks

We contemplate a simpler version of ECM, **Simple Graph Relevance Networks (Sim-GRN)**, as depicted in Figure 2. We use relevance information like the reciprocal rank of the entity link contexts (in this case passages) to model the importance of each passage in the graph.

We first entity link all the documents in the collection and split each document into passages of approximately 50 terms. We then retrieve the top k relevant documents using a traditional retrieval model, which serves as a feedback set P. For each entity present in the feedback set P, the passage text serves as the entity link context. To obtain the relevance of each entity e, we combine the reciprocal ranks of the link context (in this case passages) and rank the entities based on the aggregated reciprocal ranks. In this simpler version, we use the aggregated relative relevance of the link contexts, thus the entities are represented by the relevance information of the link contexts.

To aggregate the relevance information of passages, we follow the weighting scheme of RM3 query expansion model [13] using entity links: For every passage document d in the feedback set P, we get the relevance distribution as $p(d|q)$.

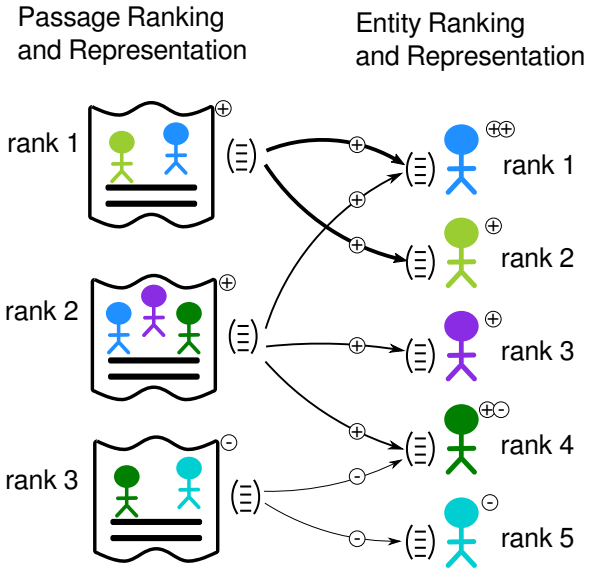

Figure 2: Spreading of relevant (+) and less relevant (-) information through the GRN. In Sim-GRN and Special-GRN only the relevance rank information is transmitted (no content representation). In GRN reciprocal rank information is used in lieu of graph attention to aggregate content representations. In GAT no relevance information is used, instead attention is derived from the content representations.

Different aggregation methods for relevance information of passages can be used, summing reciprocal ranks or rank scores, or using the geometric mean. In this work, we follow the weighting of an RM3 query expansion model [13] just using linked entities instead of words: Derive a categorical relevance distribution over top retrieved documents models $p(d|q)$. Derive per document distributions over expansion terms (here: entity links) as denoted $p(e|d)$. Then the expansion distribution is derived via marginalization as $p(e|q) = \sum_d p(e|d)p(d|q)$. Similar ideas also gave rise to work on Latent Entity Space [15] and EsdRank [32].

In our experiments below, the score of Sim-GRN model is as follows (d denotes passages)

$$\text{score}(e|q) = \sum_{d\in P_e} \underbrace{\frac{\left(\frac{1}{rank(d)}\right)}{\sum d'\left(\frac{1}{rank(d')}\right)}}_{p(d|q)} \underbrace{\frac{\text{count } e \in d}{\text{total entity count in } d}}_{p(e|d)} \tag{3}$$

4.3 Graph Relevance Networks

Graph perspective: Similar to the original ECM model, the score aggregation of the Sim-GRN model can be viewed as a message-passing on a graph as depicted in Figure 2. Edge weights are defined based on the reciprocal rank of the passages as given in equation 4. On this graph, a message-passing model is applied, such as Random Walks with Restarts [26]. After the first iteration, all entity nodes will obtain a new node weight by summing over all adjacent passages ($d \sim e$):

$$\text{score}(e|q) = \sum_{d \in P_e} \underbrace{\frac{1}{rank(d)}}_{p(d|q)} \cdot \underbrace{\text{count } e \in d}_{p(e|d)} \tag{4}$$

$$\text{nodeweight}(e) := \sum_{d:d \sim e} \text{weight}_{d \to e} \cdot \text{nodeweight}(d) \tag{5}$$

$$= \sum_{d \in P_e} \underbrace{p(d|q)p(e|d)}_{\text{edge weight}} \cdot \underbrace{1}_{\text{passage representation}} \tag{6}$$

Random walks can also retain node weights from previous iterations, relating to residual connections in neural networks.

Graph Relevance Networks (GRN):. We hypothesize that the relevance information of the entity contexts can provide the signal to distinguish the relevant nodes from the non-relevant ones to enable more expressive representations of entities. While GAT employs an attention mechanism, we use relevance information from a retrieval engine, to model the edge relevance in the graph network, as depicted in Figure 2. We replace the attention mechanism with relevance information associated with edges $\vec{r}$. This relevance information is provided by a search engine and does not need to be re-discovered from the nodes' vector representations.

Various metrics from a a retrieval engine can be used for the relevance-based edge weight $\vec{r}_{d \to e}$: rank scores or reciprocal ranks across multiple retrieval models and index representations as well as similarity information. We use a simple relevance representation that aligns with Sim-GRN: the passage reciprocal rank: $\vec{r}_{d \to e} = p(d|q)p(e|d)$.

We incorporate the textual content representations of entities, queries and passages from pre-trained resources from BERT. The concrete setup we study in this work is defined as follows:

(1) Raw embedding: We obtain raw embedding content representations for query, entity, and passage from pre-trained embeddings.
(2) Query-specific representations: Using a hadamard product, we obtain query-specific representations entities as $\vec{e} = \texttt{proj}(\vec{q}_{\text{raw}}) \odot \texttt{proj}(\vec{e}_{\text{raw}})$. We reduce the parameter space by first linearly down-projecting to dimensionality d, have hadamard product between the down projected query and entity embeddings. Passage representations $\vec{d}$ are projected down and query-specific passage embeddings are obtained in the similar fashion as entity.
(3) GRN: The entity's node representation is updated via $\vec{e} := \texttt{upd}(\texttt{proj}(\vec{e}) + \sum_{d:d \sim e} \vec{r}_{d \to e} \cdot \texttt{proj}\left(\vec{d}\right))$
(4) Ranking: The rank score is obtained via linear projection to a scalar $\text{score}(e|q) = \texttt{proj}(\vec{e})$
(5) Loss function: The model is trained end-to-end (fixing only pre-trained embeddings), optimizing a pair-wise rank loss defined on a subsample of posive/negative entity pairs.

Special-GRN. A special case of the GRN as defined in Equation 6 when nodes are represented by 1-dimensional vectors of 1 or 0. The result differs slightly as the proj (..) operator introduces a constant scale factor and bias term, which could be set during training. We refer to this special case model as Special-GRN.

5 EXPERIMENTAL EVALUATION

We use two standard datasets to evaluate our models: DBpediaV2 and TREC Complex Answer Retrieval (CAR).

5.1 Datasets and Corpus

Datasets: DBpediaV2 consists of four different types of queries: (1) INEX-LD contains IR-styled keywords. e.g., "electronic music genre"; (2) SemSearchES contains short one entity search type of queries, e.g., "brooklyn bridge" (3); QALD2 consists of natural questions which are answerable by entities, e.g., "who is the mayor of Berlin?"; (4) ListSearch which consists of queries searching for a list of entities, e.g., "Professional sports team in Philadelphia". TREC CAR dataset consists of topical queries such as "air pollution".

Corpus: As background corpus for both experiments, we use 20 million deduplicated passages from Wikipedia as provided in the TREC CAR passage corpus. These contain entity links that were manually inserted by the page author.

5.2 Feedback Set

For every query, we retrieve the top 1000 passages as feedback set P using a traditional ECM passage ranking model. We use query likelihood to retrieve initial candidate passage set and entity link them. For each entity link in the passage, we consider the entire passage as the entity context, as opposed to considering only 50 tokens. We form pseudo-document representation for each entity by concatenating all the corresponding entity contexts. Using query likelihood, the entities are ranked based on their pseudo-documents. Subsequently, the score for each passage is determined as the summation of scores of its linked entities. Finally, passages are ranked based on these scores.

5.3 Entity Candidate Sets

For the methods Special-GRN, Tuned BERT, GAT, and GRN, we build an entity candidate set E where we consider all the entity links present in the feedback set P. For every query, we further subsample the entity candidate set E by randomly selecting 100 relevant i.e., positive and 100 non-relevant i.e., negative samples. Thus, for every query the final candidate entity set E consists of in total 200 entities, with 100 positive and 100 negative samples. Also, for each entity in the entity candidate set E we use the entity's Wikipedia page lead text as its representation. The text of the passages P serves as the entity link contexts.

5.4 Training

We train the models on a pair-wise ranking loss, where the goal is to rank the relevant entities higher than the non-relevant entities of queries by learning better representations of the entities. In ranking problems, it is important to select informative negative samples to learn high-quality ranking models. To select the negative samples for training the models, we use two approaches (1) Candidate Negatives and (2) In-batch Negatives, which we describe below:

Candidate Negatives. We define the sub-sampled 100 negative samples from the candidate entity set E for every query as Candidate Negatives. This setting is more suitable for our approach as in the

ECM model, the goal is to identify the relevant entities through the entity link context.

In our work, we also utilize ***in-batch negatives*** along with the candidate negatives to train the models. We provide the results of the effectiveness of in-batch negatives on the overall performance in Section 7.4.

Training Sets. We use 5-fold cross-validation to train DBpedia V2 (DBpediaV2) experiments. TREC CAR dataset consists of a very large training dataset which is divided in several subsets such as Benchmark Y1, Benchmark Y2, Fold0-5. In our work, we train the models on total of 317 queries from two subsets, 117 from Benchmark Y1 Train and 200 queries from Fold0 subset.

Model Selection. We train for 50 epochs with a batch size of 1000 for all the experiments. We consider the model with the highest MAP for evaluation set as the best model. We optimize with `pytorch`'s Adam using a learning rate of 2e-5. We use a 1-head attention for the GAT model. The model training takes maximum 3 hours to train on 1 NVIDIA A40 GPU.

5.5 Evaluation

Evaluation Sets and Metrics. For TREC CAR dataset, we use Benchmark Y2 Test consisting of 65 topical queries as evaluation set. For DBpediaV2 experiments, we use the evaluation set provided by the dataset.

We use `trec_eval` metrics (mean) average precision (MAP), Precision at number of relevant entities (P@R), and normalized cumulative gain with cutoff rank 100 (NDCG). On TREC Complex Answer Retrieval Y2 Test (CAR) we use evaluate complete rankings, but on DBpediaV2, due to many missing judgments, we evaluate entities that are explicitly judged as positive or negative.

Significance Testing. We test for the significance of improvements with a one-sided test at $p < 0.05$. We denote significant improvement over BERT-ER [2] with $\triangle$ and improvement over Tuned BERT with †.

5.6 Input Embeddings

In our experiments, we use the pre-trained 768-dimensional BERT embeddings [3] of the `[CLS]` token for entities, queries, and passages.

6 EXPERIMENTS

6.1 Baselines

- **Trad-BM25:** Ranking Wikipedia page content with BM25.
- **Sim-GRN:** Simple Graph Relevance Network on 1000 expansion passages.
- **GEEER [7]:** As an external reference system we use GEEER [7]. GEEER uses Wiki2Vec embeddings and an entity ranking (but does not use passage information). We will use the top 1000 entities of the Sim-GRN run. The queries are annotated by TagMe entity linker and Wiki2Vec embeddings are used for both, queries and entities representations.
- **GEEER-ERNIE:** Same as GEEER but instead of Wiki2Vec embeddings, we use ERNIE entity embeddings.

[3] DistillBERT: `distilbert-base-uncased`

Table 1: Results on benchmarkY2-test TREC-CAR using automatic ground truth. $\triangle$ denotes significant improvement over BERT-ER [2] and † indicates significant improvement over Tuned BERT (only tested for MAP) using a paired t-test at $p < 0.05$. Here, NDCG denotes NDCG@100.

Models	MAP	P@R	NDCG
Trad-BM25	0.012	0.037	0.073
Sim-GRN	0.146	0.221	0.353
GEEER [7]	0.144	0.215	0.349
GEEER-ERNIE	0.145	0.218	0.353
GEEER-EBERT	0.144	0.218	0.349
BERT-ER [2]	0.263	0.319	0.482
Special-GRN	0.316$\triangle$	0.366	0.543
Tuned BERT	0.469$\triangle$	0.568	0.679
GAT	0.471$\triangle$	**0.580**	0.677
GRN	**0.494**$\triangle$†	0.565	**0.695**

- **GEEER-EBERT:** Same as GEEER but instead of Wiki2Vec embeddings, we use E-BERT entity embeddings.
- **Tuned BERT:** Ranking entities with fine-tuned query-specific entity embeddings, ignoring the graph in step 3 of the model.
- **BERT-ER [2]:** The SOTA entity ranking system that generates query-specific entity representations to rank the entities.

6.2 Our Variants

- **GAT:** Graph Attention Networks by replacing the message with normalized dot product attention $\vec{r}_{d\to e} = \frac{1}{\sqrt{d_k}}\vec{e}\vec{d}$ of query-specific representations in step 3.
- **GRN:** Graph Relevance Network using the passage reciprocal rank as message $\vec{r}_{d\to e}$. Query-specific representations $\vec{e}$ and $\vec{d}$ are derived from entities, passages, and queries as described above.
- **Special-GRN:** Reducing the GRN model to the special case of Sim-GRN as described in Section 4.3. Implemented by skipping steps 1 and 2, by setting $\vec{d} = 1$ and $\vec{e} = 0$ and using the same relevance weight $\vec{r}$ as in GRN.

7 RESULTS

Through our experiments, we address the following research questions:

- **RQ1:** Do the neural variants of ECM help to improve the entity ranking task?
- **RQ2:** To what extent is graph structure helpful to identify relevant entities?
- **RQ3:** Is relevance information a more effective indicator than attention mechanism in the graph network for a ranking task?

7.1 RQ1: Overall Performance

From Tables 1 and 2, we observe that both the entity ranking models, GAT and GRN, based on the underlying concept of ECM significantly outperform the state-of-the-art entity ranking system BERT-ER [2] between 14-88% in MAP for both, TREC CAR and DBpediaV2, datasets.

Table 2: Results on DBpediaV2. △ denotes significant improvement over BERT-ER [2] and † indicates significant improvement over Tuned BERT(only tested for MAP) using a paired t-test at $p < 0.05$. Here, NDCG denotes the NDCG@100 evaluation metric. We highlight the best-performing models in bold.

Models	ALL			QALD_2			ListSearch			INEX-LD			SemSearch		
	MAP	NDCG	P@R	MAP	NDCG	P@R	MAP	NDCG	P@R	MAP	NDCG	P@R	MAP	NDCG	P@R
Trad-BM25	0.363	0.562	0.374	0.257	0.466	0.279	0.300	0.501	0.348	0.306	0.532	0.330	0.608	0.769	0.558
Sim-GRN	0.298	0.498	0.339	0.269	0.476	0.286	0.354	0.557	0.424	0.236	0.439	0.299	0.330	0.516	0.355
GEEER [7]	0.318	0.512	0.365	0.279	0.485	0.302	0.365	0.563	0.444	0.250	0.447	0.323	0.377	0.552	0.399
GEEER-ERNIE	0.301	0.501	0.345	0.271	0.478	0.288	0.356	0.557	0.424	0.236	0.439	0.305	0.340	0.524	0.369
GEEER-EBERT	0.319	0.515	0.363	0.276	0.486	0.298	0.363	0.563	0.435	0.254	0.452	0.320	0.386	0.560	0.409
BERT-ER [2]	0.500	**0.723**	0.461	0.410	0.658	0.381	0.507	0.739	0.472	0.469	**0.714**	0.437	0.629	**0.810**	0.570
Special-GRN	0.604△†	0.713	0.608	0.585△	0.711	0.582	0.669△†	**0.760**	**0.681**	**0.523**△†	0.653	0.528	0.633△	0.723	0.634
Tuned BERT	0.595△	0.708	**0.612**	0.580△	0.706	**0.589**	0.658△	0.755	**0.681**	0.506△	0.640	**0.532**	0.629	0.720	**0.639**
GAT	0.587△†	0.703	0.606	0.566△†	0.698	0.573	0.649△†	0.750	0.679	0.496△	0.634	0.531	0.631△	0.721	0.636
GRN	**0.607**△†	0.714	0.610	**0.592**△†	**0.714**	**0.589**	**0.671**△†	**0.760**	0.680	0.520△†	0.651	0.527	**0.634**△	0.722	0.637

Both entity ranking models achieve significant improvement of more than 200% in terms of MAP (*MAP=0.144 to MAP=0.471 and MAP=0.494*), over entity ranking method based on static knowledge graph embeddings such as GEEER [7] for TREC CAR. We observe a similar performance increase for DBpediaV2 in Table 2. These results show that the graph neural models grounded in the notion of ECM improve the entity ranking task and thus support our ***RQ1***. We provide further evidence through exploration of the performance at the query level.

Query-level Analysis. We further investigate the performance of the neural models, GAT and GRN with the state-of-the-art entity ranking system BERT-ER at the query level. We divide the queries into percentiles based on their difficulty for a baseline. Queries on which the baseline obtains a lower MAP performance are considered to be more difficult queries.

From Figure 3, we observe that both, GAT and GRN, perform better for all the queries than the BERT-ER method. This indicates that the graph neural models improve the ranking quality consistently across all queries. We observe the same phenomenon for the DBpediaV2 dataset (results omitted). This shows that generating entity embeddings specifically for entity ranking task that captures the entity contextual information indicating what makes an entity relevant to a particular query is important to determine the relevance of entities.

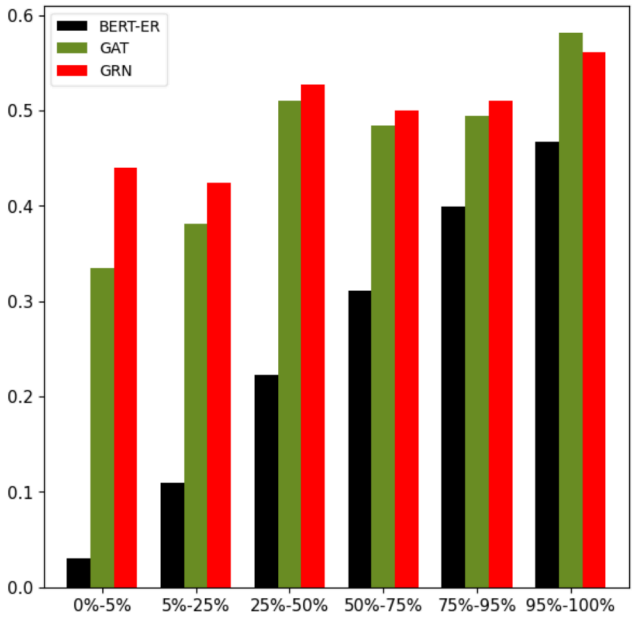

Figure 3: Dividing CAR queries into different percentiles of difficulty for BERT-ER method, as measured in per-query MAP performance. The y-axis displays the MAP performance of each method on queries within. The x-axis displays different percentiles: Most difficult 5% queries for the BERT-ER are on the left side and the easiest 5% queries are on the right side.

7.2 RQ2: Importance of the Graph Structure

We observe from Tables 1 and 2, that for both the datasets, graph-based models such as GRN, Special-GRN, and GAT obtain consistently good performance compared to the Tuned BERT and BERT-ER method. Special-GRN includes only the relevance information through reciprocal rank without drawing on embeddings of the entity contexts. In contrast, the graph models of GAT and GRN include the entity contextual information along with the entity representation of itself via self-loop and GRN also incorporates the reciprocal rank information.

GAT and GRN: From Tables 1 and 2, we find that the graph based models GAT and GRN improves the performance between 0.4-5% compared to Tuned BERT system for CAR dataset. These models also achieve significant MAP improvement of 79% and 87% respectively over the BERT-ER system. We observe a similar pattern for DBpediaV2 where both GAT and GRN outperform Tuned BERT and BERT-ER. This indicates the significance of the entity contextual information for the generation of entity embeddings to determine the relevance of entities. It also demonstrates the importance of the graph structure induced by the entity links. This finding is in line with the conclusions of several pre-neural studies [3, 5, 15], but we show that it still holds in the era of Neural Networks.

Special-GRN. The Special-GRN model is similar in approach to the traditional Sim-GRN method, with the graph induced from

entity links. We observe that Special-GRN improves the MAP performance significantly for the DBpediaV2 dataset, however, for TREC CAR the performance of Special-GRN is significantly lower compared to Tuned BERT.

We suspect that the reason for the higher performance of the Special-GRN model for DBpediaV2 dataset is the characteristics of the dataset itself that do not need any additional entity information to identify the relevant entities. For example, one of the subsets of DBpediaV2, the *SemSearch* subset includes queries such as "brooklyn bridge" that needs only lexical overlap between the query terms and entity mentions to identify more than 80% of the relevant entities. This indicates that additional contextual information of any entity is not required necessarily by many queries to identify relevant entities. Hence, while entity link information is useful (as the dataset is constructed from Wikipedia pages), entity contextual information might not be overly helpful for many queries of the DBpediaV2 dataset. Lack of entity contextual information can thus be a potential reason for Special-GRN to perform on par with the other graph models.

We note that the graph structures, in particular, help to elevate the performance of recall-oriented metrics for both datasets.

7.3 RQ3: Attention vs Relevance

We hypothesized that the relevance information can serve as an effective signal to distinguish relevant information from non-relevant. Indeed we find that for both benchmarks, the best-performing method uses the relevance information provided by the search engine. From the Tables 1 and 2, it is evident that incorporating relevance information significantly enhances the accuracy of the entity ranking task.

Compared to GRN, GAT needs to rediscover the semantic information about relevance from the entity context representations. This is an indirect way and only possible due to our query-specific passage and entity representations. Moreover, it is unnecessary, as the candidate set automatically comes with the relevance information that merely needs to be used.

We further investigate how relevance information affects the ranking of the entities. For example, one of the queries "*protecting the water supply*" place relevant entities such as "Flocculation" and "Cleveland" in higher ranking positions of 1 and 14 out of the top 14 relevant entities, whereas these entities are placed at 19 and 29 in GAT. We further expand on the understanding of the higher rankings in GRN below.

Example. The GRN model includes the relevance information of the entities through the reciprocal rank of the contexts. As an example, we study the entity rankings of the query "*glaciers*" for GRN and GAT to understand the effect of the inclusion of the relevance information on the rankings.

As mentioned previously, GRN uses reciprocal rank information along with the entity context and entity representation as opposed to only entity context and entity representation information in GAT. Thus, entities that are present in more relevant documents to the query are considered to be more relevant in GRN. Hence, relevant entities should be placed higher in the rankings. We observe that for the query "*glaciers*", in GRN all the top 12 ranking entities are relevant while GAT contain only 6 relevant entities in the top 12 at various ranks.

We further examine one of the relevant entities "*Crevasse*" which ranks at a position 9 by GRN, whereas GAT place them at the rank 21 after various non-relevant entities. To understand the reason behind the higher ranking, we determine the sum of the reciprocal ranks for the connected passages to the entity "*Crevasse*". We find that "*Crevasse*" is connected to 31 relevant passages and the sum of the reciprocal ranks is 0.182. This relevance information provides a signal to the graph model GRN to provide more weight to the entity "*Crevasse*" compared to other entities with lower reciprocal rank sum. We understand that the reciprocal of the rank usually results in a smaller number. We observe the same pattern for other relevant entities that are placed at higher rankings in GRN.

7.4 Effect of In-batch Negatives

As mentioned earlier in Section 5.4, we train models on training sets with different negative samples to understand the influence of negative sampling on the results of the ranking models. Despite incorporating both in-batch and candidate negatives, we observe no performance gains across benchmarks. We suspect the potential reason for no improvement in the performance is that the query-specific entity representations generate representations that are too diverse between the queries and hence exploring these representations as negatives is not helpful to elevate the performance.

8 CONCLUSION

In this paper, we introduce three entity ranking models that are built on the foundational principles of the ECM approach. In contrast to the predominant approaches that exploit static entity embeddings, our neural models focus on entity contexts through pseudo-relevance feedback documents.

GAT, a graph neural variant of ECM, extends the paradigm by including the text representations of entity contexts and entities. A simpler traditional Sim-GRN, which utilizes the relevance information from the retrieval engine. We also introduce GRN, a graph neural variant, which models the entity representations by using the relevance information along with entity contexts and entities.

On two benchmarks consisting of a diverse range of queries, we demonstrate the benefits of our entity ranking models. We find that the relevance-based GRN model works best for both CAR and DBpediaV2. On both datasets our approach significantly outperforms a range of strong baseline systems, such as BERT-ER [2], GEEER [7], and Tuned BERT. By developing graph neural ranking models based on a theoretical underpinning, we achieved significant performance improvements. We demonstrate that incorporating relevance information from retrieval engines, a less expensive alternative, into graph neural networks is more effective for the entity ranking task than the attention mechanism.

Acknowledgements

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] 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.

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

[3] 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

[4] 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

[5] Faezeh Ensan and Ebrahim Bagheri. 2017. Document retrieval model through semantic linking. In *Proceedings of the tenth ACM international conference on web search and data mining*. 181–190.

[6] Emma J. Gerritse, Faegheh Hasibi, and Arjen P. de Vries. 2020. Graph-Embedding Empowered Entity Retrieval. In *Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part I* (Lisbon, Portugal). Springer-Verlag, Berlin, Heidelberg, 97–110. https://doi.org/10.1007/978-3-030-45439-5_7

[7] 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

[8] Emma J. Gerritse, Faegheh Hasibi, and Arjen P. de Vries. 2022. Entity-Aware Transformers for Entity Search. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Madrid, Spain) *(SIGIR '22)*. Association for Computing Machinery, New York, NY, USA, 1455–1465. https://doi.org/10.1145/3477495.3531971

[9] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In *International conference on machine learning*. 1263–1272.

[10] 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

[11] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 6769–6781. https://doi.org/10.18653/v1/2020.emnlp-main.550

[12] Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 39–48.

[13] Victor Lavrenko and W Bruce Croft. 2017. Relevance-based language models. In *ACM SIGIR Forum*, Vol. 51. ACM New York, NY, USA, 260–267.

[14] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *AAAI*.

[15] Xitong Liu and Hui Fang. 2015. Latent entity space: a novel retrieval approach for entity-bearing queries. *Information Retrieval Journal* 18 (2015), 473–503.

[16] Yinquan Lu, Haonan Lu, Guirong Fu, and Qun Liu. 2021. KELM: knowledge enhanced pre-trained language representations with message passing on hierarchical relational graphs. *arXiv preprint arXiv:2109.04223* (2021).

[17] 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

[18] 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*. 472–479.

[19] 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

[20] Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with BERT. *arXiv preprint arXiv:1910.14424* (2019).

[21] Francesco Piccinno and Paolo Ferragina. 2014. From TagME to WAT: A New Entity Annotator. In *Proceedings of the First International Workshop on Entity Recognition & Disambiguation* (Gold Coast, Queensland, Australia) *(ERD '14)*. Association for Computing Machinery, New York, NY, USA, 55–62. https://doi.org/10.1145/2633211.2634350

[22] 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

[23] 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

[24] Hadas Raviv, Oren Kurland, and David Carmel. 2016. Document retrieval using entity-based language models. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 65–74.

[25] 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

[26] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. 2006. Fast random walk with restart and its applications. In *Sixth international conference on data mining (ICDM'06)*. IEEE, 613–622.

[27] Johannes M Van Hulst, Faegheh Hasibi, Koen Dercksen, Krisztian Balog, and Arjen P de Vries. 2020. Rel: An entity linker standing on the shoulders of giants. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2197–2200.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[29] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph Attention Networks. In *International Conference on Learning Representations*.

[30] 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.

[31] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. *Proceedings of the AAAI Conference on Artificial Intelligence* 28, 1 (Jun. 2014). https://doi.org/10.1609/aaai.v28i1.8870

[32] Chenyan Xiong and Jamie Callan. 2015. EsdRank: Connecting Query and Documents through External Semi-Structured Data. 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, 951–960. https://doi.org/10.1145/2806416.2806456

[33] Chenyan Xiong, Jamie Callan, and Tie-Yan Liu. 2016. Bag-of-Entities Representation for Ranking. 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, 181–184. https://doi.org/10.1145/2970398.2970423

[34] 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.

[35] 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, Online, 23–30. https://doi.org/10.18653/v1/2020.emnlp-demos.4

[36] 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

[37] 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

[38] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks: A review of methods and applications. *AI open* 1 (2020), 57–81.