Textual Evidence Mining via Spherical Heterogeneous Information Network Embedding

Xuan Wang[1*], Yu Zhang[1*], Aabhas Chauhan[1], Qi Li[2], Jiawei Han[1]
[1]Department of Computer Science, University of Illinois at Urbana-Champaign, IL, USA
[2]Department of Computer Science, Iowa State University, IA, USA
[1]{xwang174, yuz9, aabhasc2, hanj}@illinois.edu, [2]qli@iastate.edu

2020 IEEE International Conference on Big Data (Big Data) | 978-1-7281-6251-5/20/$31.00 ©2020 IEEE | DOI: 10.1109/BigData50022.2020.9377958

***Abstract*—Scientific literature, as one of the major knowledge resources, provides abundant textual evidence that has great potential to support high-quality scientific hypothesis validation. In this paper, we study the problem of *textual evidence mining* in scientific literature: given a scientific hypothesis as a query triplet, find the textual evidence sentences in scientific literature that support the input query. A critical challenge for textual evidence mining in scientific literature is to retrieve high-quality textual evidence without human supervision. Because it is non-trivial to obtain a large set of human-annotated articles containing evidence sentences in scientific literature. To tackle this challenge, we propose EVIDENCEMINER, a high-quality textual evidence retrieval method for scientific literature without human-annotated training examples. To achieve high-quality textual evidence retrieval, we leverage heterogeneous information from both existing knowledge bases and massive unstructured text. We propose to construct a large heterogeneous information network (HIN) to build connections between the user-input queries and the candidate evidence sentences. Based on the constructed HIN, we propose a novel HIN embedding method that directly embeds the nodes onto a spherical space to improve the retrieval performance. Quantitative experiments on a huge biomedical literature corpus (over 4 million sentences) demonstrate that EVIDENCEMINER significantly outperforms baseline methods for unsupervised textual evidence retrieval. Case studies also demonstrate that our HIN construction and embedding greatly benefit many downstream applications such as textual evidence interpretation and synonym meta-pattern discovery.**

***Index Terms*—textual evidence mining; heterogeneous information network; spherical graph embedding**

I. INTRODUCTION

With the vast amount of data and advanced computational technologies, scientists nowadays can generate massive untested *scientific hypotheses* within a short period of time. The generated scientific hypotheses are usually relational triplets fitting the schema of existing knowledge bases. For example, in Table I, *(resveratrol, inhibit, pancreatic cancer)* is a scientific hypothesis indicating that "*resveratrol*" can be a potential drug treatment for "*pancreatic cancer*". The generated scientific hypotheses need to be validated before turned into real knowledge. However, manual validation could be laborious and error-prone. To automatically validate the massive scientific hypotheses, scientists often seek evidence from a variety of knowledge resources, such as existing knowledge repositories and massive unstructured text. Existing human-curated knowledge repositories usually have limited coverage of scientific evidence, whereas massive unstructured text provides abundant *textual evidence* that has great potential to support high-quality scientific hypothesis validation. The textual evidence in scientific literature is defined as a claim sentence that supports the scientific hypothesis as a query triplet. For example Table I, S1 and S2 are textual evidence sentences supporting the query triplet *(resveratrol, inhibit, pancreatic cancer)*. Sentences that either miss the query entities (e.g., S3), express a different relation from the query relation (e.g., S4) or are non-claim sentences (e.g., S5) are not considered as textual evidence sentences. In this paper, we study the problem of *textual evidence mining* in scientific literature: given a scientific hypothesis as a query triplet, find the textual evidence sentences from scientific literature that support the input query.

TABLE I
EXAMPLES OF TEXTUAL EVIDENCE SENTENCES GIVEN USER-INPUT QUERIES AS RELATIONAL TRIPLETS.

ID	Queries	Textual Evidence Sentences
S1	*(resveratrol, inhibit, pancreatic cancer)*	**Resveratrol** *inhibits* the growth and development of **pancreatic cancer**. ✓
S2	*(resveratrol, inhibit, pancreatic cancer)*	**Resveratrol** *induces apoptosis in* **pancreatic cancer** cells. ✓
S3	*(genistein, affect cotreatment, brca1)*	Both SAHA and olaparib downregulated the expression of **BRCA1**. ✗
S4	*(cldn6, increase expression, sb431542)*	**SB431542** led to a *decrease* of the binding activity for **CLDN6** promoter. ✗
S5	*(quercetin, increase activity, cftr)*	We sought to determine the effect of **quercetin** on *increasing* **CFTR**. ✗

The last several years have witnessed increasing interests and efforts in textual evidence mining (see Section II). However, the existing methods cannot be directly applied to our task of textual evidence mining in scientific literature. Traditional textual evidence mining or claim mining tasks aim to verify the claims as natural language statements whereas our task aims to verify the scientific hypotheses as relational triplets. Some existing methods for textual evidence mining are supervised learning methods [21, 22, 17] assuming a set of human-annotated articles containing evidence sentences be given as the training examples. It is non-trivial to obtain a large set of human-annotated training examples, especially in scientific literature where expert annotations are expensive. Several recent studies [1, 18, 38] have been proposed to use limited or no human-annotated training data for textual evidence mining.

*The first two authors contributed equally to this work and should be considered as joint first authors.

However, these methods are developed for specific domains (e.g., the online debate corpus) using heuristic rules that cannot be directly applied to scientific literature. Another relevant task is fact checking [31, 52, 43, 28, 15, 40, 10] that aims to verify the correctness of the relational triplets in the knowledge graphs with evidence from various resources such as text. Our goal is more focused on retrieving the correct textual evidence rather than validating the trustworthiness of the query triplets. In summary, a critical challenge for textual evidence mining in scientific literature is to retrieve high-quality textual evidence without human supervision.

To tackle this challenge, we propose EVIDENCEMINER, a high-quality textual evidence retrieval method for scientific literature without human-annotated training examples (Figure 1). To achieve high-quality textual evidence retrieval, we leverage heterogeneous information resources including both existing knowledge bases (KBs) and massive unstructured text. We propose to construct a large heterogeneous information network (HIN) to build connections between the user-input queries and the candidate evidence sentences using these heterogeneous information resources. The nodes in the HIN include the relational triplets in existing KBs, the candidate evidence sentences in massive unstructured text and the words, entities and informative textual patterns automatically extracted from both existing KBs and massive unstructured text.

Based on the constructed HIN, we propose a novel HIN embedding method to represent each node in a low-dimensional vector space that effectively preserves the semantic similarity between the nodes. Since the semantic similarity is obtained as the cosine similarity in the embedding space, we propose to directly embed the nodes onto a spherical space as a natural adaptation to improve the similarity-based retrieval performance. The HIN embedding is pre-computed offline to support fast online retrieval. For online retrieval, the user-input query is processed and connected with the nodes in the HIN. The embedding of the query is inferred by the embeddings of its one-hop neighbors in the HIN. The candidate evidence sentences are ranked based on the cosine similarity between their embeddings and the query embedding.

EVIDENCEMINER shows a great promise to extract high-quality textual evidence in scientific literature to support a large-scale scientific hypothesis validation. Our contributions are summarized as follows:

- We propose EVIDENCEMINER, a high-quality textual evidence retrieval method for scientific literature without human-annotated training examples.
- To achieve a high-quality textual evidence retrieval, we propose to construct a large HIN to leverage information from both existing KBs and massive unstructured text and to build connections between the input queries and the candidate evidence sentences.
- To further improve the performance, we propose a novel HIN embedding method that directly embeds the nodes onto a spherical space for the cosine similarity-based retrieval.

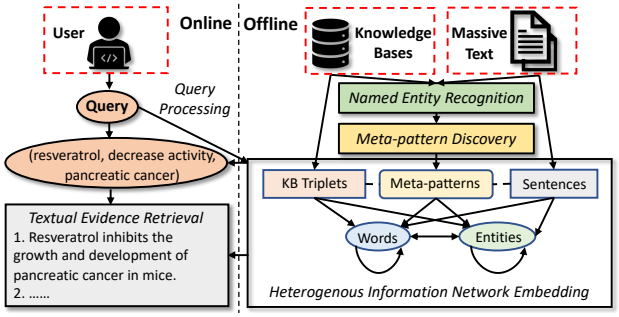


Fig. 1. The overall framework of EVIDENCEMINER.

- Quantitative experiments on a huge biomedical literature corpus (over 4 million sentences) demonstrate that EVIDENCEMINER significantly outperforms baseline methods on unsupervised textual evidence retrieval.
- Case studies also demonstrate that our HIN construction and embedding greatly benefit many downstream applications such as textual evidence interpretation and synonym meta-pattern discovery.

II. RELATED WORK

Textual Evidence Mining. The last several years have witnessed increasing interests and efforts in textual evidence mining [21, 22, 17, 1, 18, 38, 2, 44]. Traditional evidence mining or claim mining tasks aim to verify the claims as natural language statements whereas our task aims to verify the scientific hypotheses as relational triplets. Some tasks output the evidence sentences labeled as positive, negative or neutral in regard to the input claim. However, the evidence sentences in scientific literature are usually objective statements that do not have a strong sentiment in the opinion they hold. Therefore, the expected output of our task is a ranked list of sentences indicating how strongly they are coherent with the input query. Other tasks have a similar output with our task, such as context-dependent claim mining. Some previous studies of context-dependent claim mining assume a small set of human-annotated articles containing the claims be given as the training examples, such as [17]. It is non-trivial to retrieve the set of human-annotated articles, especially in scientific literature where expert annotations are expensive.

Several recent studies have been proposed to use limited or no human-annotated training data for claim mining [1, 18, 38]. Al-Khatib *et al.* [1] propose a distantly-supervised claim mining method for the online debate corpus. They introduce an argumentativeness mapping function, which coarsely maps sentences in each section of the online debates into claims for distant supervision. Levy *et al.* [18] recently propose an unsupervised claim mining method. They define a heuristic rule for candidate claim generation in the online debate corpus, which assumes a claim sentence must have a preceding word "that" in it. These distantly-supervised or unsupervised claim mining methods are developed for specific domains (e.g., the online debate corpus) using heuristic rules that cannot be directly used for claim mining in scientific literature.

Another relevant task is fact checking [31, 52, 43, 28, 15, 40, 10] that aims to verify the correctness of the relational triplets in the knowledge graphs with evidence from various resources such as text. For example, Defacto [15] proposes to transform triplets in a knowledge graph to natural language sentences, and uses a web search engine to find web pages containing those sentences. However, the transformation from relational triplets to natural language sentences may introduce errors. It is more natural to match a knowledge triplet directly against massive unstructured text. Our goal is more focused on retrieving the correct textual evidence rather than validating the trustworthiness of the query triplets.

Heterogeneous Information Network Embedding. Due to the hypothesis that real-world objects and their interactions are often multi-typed, heterogeneous information networks (HIN) [39] have been widely used as a more generic superclass of traditional networks. HIN embedding techniques can be divided into three major categories: random walk-based, proximity-based, and message-passing approaches. Among random walk-based methods, ESim [35] learns node embeddings using meta-path guided sequence sampling and noise contrastive estimation; Metapath2Vec [8] samples node sequences through heterogeneous random walks following specific meta-paths; HIN2Vec [11] exploits different types of links among nodes; JUST [13] proposes Jump/Stay random walks that do not rely on pre-defined meta-paths; HeteSpaceyWalk [12] introduces a scalable embedding framework based on heterogeneous personalized spacey random walk. Among proximity-based methods, PTE [41] decomposes the whole network into bipartite graphs and embeds nodes based on first and second order proximities; AspEm [36] mines different aspects from the HIN and preserves the proximity between nodes in the same aspect instance; HEER [37] extends PTE by considering typed-closeness. Among message passing-based methods, R-GCN [34] extends graph convolutional networks by introducing an edge type-aware aggregation function; HetGNN [51] assumes that each node is associated with different features (e.g., attributes, text and image) and encodes these features before aggregating them; GATNE [6] considers both transductive and inducting settings in node embedding. One can refer to two recent surveys [50, 9] for a more detailed introduction to these approaches. We would like to emphasize that most of these approaches embed nodes into a Euclidean space, while our proposed embedding method consider a spherical space to better model the cosine similarity between nodes.

III. Problem Formulation

The input to EvidenceMiner is a query triplet q, an existing knowledge base Ψ with entity type schema $\mathcal{T}_\Psi$ and a corpus of document collections $\mathcal{D}$. We assume a universe of entities $\mathcal{E} = \{e\}$ where the type of each entity mention $\mathcal{T}(e) \in \mathcal{T}_\Psi$. The query q can be treated as a relational triplet $\langle h, r, t\rangle$, where $h \in \mathcal{E}$ is the head entity, r is the relation and $t \in \mathcal{E}$ is the tail entity. The relation $r = \langle w_1, w_2, ...w_{|r|}\rangle$ is a sequence of words w_i. The knowledge base $\Psi = \{\langle h_\Psi, r_\Psi, t_\Psi\rangle\}$ can be treated as a set of relational triples, where $h_\Psi \in \mathcal{E}$, $r_\Psi = \langle w_1, w_2, ...w_{|r_\Psi|}\rangle$ and $t_\Psi \in \mathcal{E}$. The document $\mathcal{D} = \{s_1, s_2, ...s_{|\mathcal{D}|}\}$ is a set of sentences that are used as candidate evidence sentences in our task. Each sentence $s = \langle w_1, w_2, ...e_1, e_2, ...\rangle$ is a sequence containing both words w_i and entity mentions e_i where the type of each entity mention $\mathcal{T}(e_i) \in \mathcal{T}_\Psi$. We define the meta-patterns $p = \langle w_1, w_2, ...t_1, t_2, ...\rangle$ as sequences containing both words and entity type tokens $t_i \in \mathcal{T}_\Psi$. The meta-patterns can be extracted from both the triplets in the KB Ψ and the sentences in the corpus $\mathcal{D}$ after replacing the entity mentions with their corresponding entity types. We consider the problem of ranking candidate evidence sentences $s \in \mathcal{D}$ given a user-input query q.

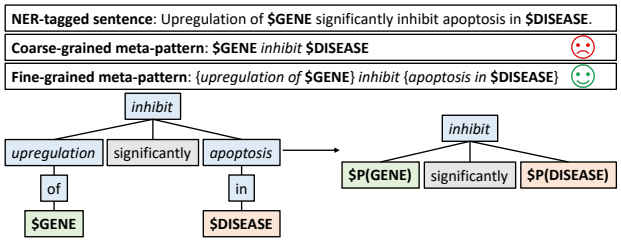


Fig. 2. Example of fine-grained meta-pattern extraction.

Problem 1 (Textual Evidence Retrieval): Given a query $q = \langle h, r, t\rangle$, where $h \in \mathcal{E}$ is the head entity, $r = \langle w_1, w_2, ...w_{|r|}\rangle$ is the relation of a word sequence and $t \in \mathcal{E}$ is the tail entity, rank the candidate evidence sentences $s \in \mathcal{D}$ by their semantic similarities with q.

We would like EvidenceMiner to develop an appropriate representation that carries semantics for the input query and the candidate evidence sentences. Note that we restrict the input queries by one criterion: there exists at least one entity or word in the query that has appeared in the input knowledge base or the input corpus. We think this is a reasonable restriction because if the input query does not satisfy the above criterion, it is unlikely that any textual evidence sentence can be retrieved from our input corpus to support this query. With that in mind, we construct a large HIN and propose a spherical HIN embedding technique for high-quality textual evidence retrieval.

IV. Data Preparation

The overall framework of EvidenceMiner is shown in Figure 1. Given a user-input query, EvidenceMiner automatically generates a ranked list of evidence sentences indicating how strongly they support the input query. EvidenceMiner has three major components: data preparation, HIN embedding and unsuperivsed textual evidence retrieval. The first two steps (data preparation and HIN embedding) are pre-computed offline based on the input KB and the input corpus. The textual evidence retrieval step is performed online given a user-input query. In this section, we briefly discuss the data preparation component. Taken a KB and a corpus as the input, we conduct two steps of data preparation: named entity recognition and meta-pattern discovery.

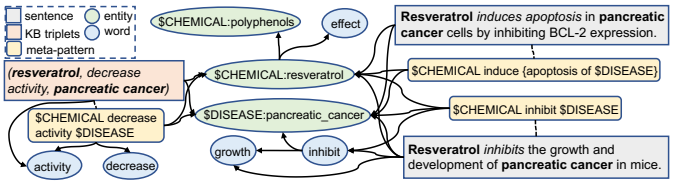


Fig. 3. Heterogeneous information network for explainable textual evidence mining.

Named Entity Recognition (NER). The entities are important connections between the input queries and the evidence sentences. For the entities in the input KB, we can directly get the entity type information from the KB schema. For the entities in the input corpus, we use PubTator [48], a state-of-the-art tool for biomedical named entity recognition [47, 46]. We use three major biomedical entity types in our experiments, i.e., genes, chemicals and diseases.

Meta-pattern Discovery. The meta-patterns (i.e., the textual pattern containing entity types) are not only important connections between the input queries and the evidence sentences but also potential interpretations of the retrieved evidence sentences. Meta-patterns containing at least two entity types (e.g., "$CHEMICAL inhibit $DISEASE") are defined as relational meta-patterns that are used for our HIN construction. For the meta-patterns in the input KB, we can directly get them from the triplets by replacing the entities with their corresponding entity types. For the meta-patterns in the input corpus, we use CPIE [45] and WW-PIE [20], two state-of-the-art methods for meta-pattern extraction in the biomedical literature. Moreover, we consider the fine-grained meta-patterns that can be derived from the nested meta-patterns extracted by WW-PIE [20]. The reason is that the coarse-grained meta-patterns do not preserve the local properties for the entities, which can lead to a semantic drift between the extracted meta-patterns and the original sentences. For example, in Figure 2, a coarse-grained meta-pattern extraction will extract "$Gene inhibit $DISEASE" from the NER-tagged sentence. However, the sentence in Figure 2 actually indicates "$Gene inhibit the death of the cells in $DISEASE" that has the opposite meaning to "$Gene inhibit $DISEASE". The fine-grained meta-pattern preserves both the top-level relationship ("inhibit") and the bottom-level entity properties ("apoptosis in $DISEASE"). Fine-grained meta-pattern extraction is a careful way to balance the trade-off between the coverage and the accuracy of meta-pattern matching to the input corpus.

V. HIN Construction and Embedding

Based on the extracted information above, we construct a large heterogeneous information network (HIN) from the input KB and the input corpus. A spherical HIN embedding technique is then proposed to represent each node in a low-dimensional vector space that best preserves the semantic similarity between nodes. In this section, we discuss the HIN construction and embedding steps in detail.

A. HIN Construction

An HIN is defined as a graph $G = (\mathcal{V}, \mathcal{E})$ with a node type mapping $\phi : \mathcal{V} \rightarrow \mathcal{T}_{\mathcal{V}}$ and an edge type mapping $\psi : \mathcal{E} \rightarrow \mathcal{T}_{\mathcal{E}}$. Either the number of node types $|\mathcal{T}_{\mathcal{V}}|$ or the number of relation types $|\mathcal{T}_{\mathcal{E}}|$ is larger than 1. As shown in Figure 3, we include five types of nodes in our HIN: the candidate evidence sentences in the input corpus (s), the relational triplets in the input KB (q), and the meta-patterns (p) and entities/words (w)[1] extracted from both the input KB and the input corpus. To capture the interactions between these nodes, we consider the following four types of edges in our HIN.

Sentence – Entity / Word. The sentence–entity/word edges describe the fact that two entities/words tend to have similar semantics when they co-occur in a sentence. The edge weight between an entity/word w and a sentence s is the term frequency of w in s (denoted as $TF(w, s)$). For example, in Figure 3, a sentence such as "resveratrol inhibits the growth and development of pancreatic cancer" is linked with the words "inhibit" and "growth" and the entity "$DISEASE:pancreatic_cancer" that appear in this sentence.

Triplet – Entity / Word. We add an edge between between a triplet q and an entity/word w if w appears in q. For instance, in Figure 3, the KB triplet "*(resveratrol, decrease activity, pancreatic cancer)*" is linked with both its relation words (e.g., "activity") and entities (e.g., "$DISEASE:pancreatic_cancer").

Meta-Pattern – Entity / Word. The meta-pattern–entity/word edges encode pattern-level entity/word co-occurrences. To be specific, some sentences can be matched by fine-grained meta-patterns (e.g., "$CHEMICAL inhibit $DISEASE"), and the edge weight between a meta-pattern p and an entity/word w is the sum of the term frequency of w in each sentence matched with p (i.e., $\sum_{s \text{ matched with } p} TF(w, s)$).

Entity / Word – Context. According to [25, 41], each word in a sentence should be similar to not only the entire sentence semantics, but also its local context. To be specific, given a sequence of words and entities $w_1 w_2 ... w_n$, the local context of w_i is defined as $\mathcal{C}(w_i, h) = \{w_j : i - h \leq j \leq i + h, i \neq j\}$, where h is the context window size. For example, in Figure 3, "CHEMICAL:resveratrol" has the context "$CHEMICAL:polyphenols" and "effect". The edge weight between an entity/word w_i and its context w_j is the number of times that w_j appears in $\mathcal{C}(w_i, h)$.

B. Spherical HIN Embedding

Given the constructed HIN, we jointly embed different types of nodes into the same latent space to characterize their relationships. Being aware of edge heterogeneity, we consider the four edge types one by one.

Sentence – Entity / Word. To preserve the proximity between a sentence s and an entity/word $w \in s$ in the joint embedding space, inspired by the softmax function proposed in word

[1] We use "w" to denote both entities and words for ease of notation.

embedding [25] and network embedding [42], we define the following conditional probability:

$$p(w|s) = \frac{\exp(\boldsymbol{e}_w^T \boldsymbol{e}_s)}{\sum_{w' \in \mathcal{W}} \exp(\boldsymbol{e}_{w'}^T \boldsymbol{e}_s)}, \tag{1}$$

where $\boldsymbol{e}_w$ and $\boldsymbol{e}_s$ are the embedding vectors of w and s, respectively; $\mathcal{W}$ is the set of entities/words appearing in the whole corpus.

Given a positive sentence–entity/word pair (s, w_p) (i.e., w_p appears in s), our goal is to maximize the log-likelihood $\log p(w_p|s)$ during embedding learning. Inspired by studies on knowledge graph embedding [5], we adopt the following margin-based ranking loss:

$$\max\Big(0, \gamma + \log p(w_n|s) - \log p(w_p|s)\Big). \tag{2}$$

Here, w_n is a negative entity/word regarding s; $\gamma > 0$ is a hyperparameter indicating the expected margin between the positive pair (s, w_p) and the negative pair (s, w_n). Based on the definition of $p(w|s)$ in Eq. (1), we have

$$\begin{aligned} &\log p(w_n|s) - \log p(w_p|s) \\ =\ &\log\left(\frac{\exp(\boldsymbol{e}_{w_n}^T \boldsymbol{e}_s)}{\sum_{w' \in \mathcal{W}} \exp(\boldsymbol{e}_{w'}^T \boldsymbol{e}_s)}\right) - \log\left(\frac{\exp(\boldsymbol{e}_{w_p}^T \boldsymbol{e}_s)}{\sum_{w' \in \mathcal{W}} \exp(\boldsymbol{e}_{w'}^T \boldsymbol{e}_s)}\right) \\ =\ &\log\big(\exp(\boldsymbol{e}_{w_n}^T \boldsymbol{e}_s)\big) - \log\big(\exp(\boldsymbol{e}_{w_p}^T \boldsymbol{e}_s)\big) \\ =\ &\boldsymbol{e}_{w_n}^T \boldsymbol{e}_s - \boldsymbol{e}_{w_p}^T \boldsymbol{e}_s. \end{aligned} \tag{3}$$

Therefore, the objective function of the sentence–entity/word edges can be defined as follows.

$$\begin{aligned} \mathcal{J}_{SW} &= \sum_s \sum_{w_p} \sum_{w_n} \max\Big(0, \gamma + \log p(w_n|s) - \log p(w_p|s)\Big) \\ &= \sum_s \sum_{w_p} \sum_{w_n} \max\Big(0, \gamma + \boldsymbol{e}_{w_n}^T \boldsymbol{e}_s - \boldsymbol{e}_{w_p}^T \boldsymbol{e}_s\Big). \end{aligned} \tag{4}$$

Triplet – Entity / Word. To encourage the closeness between the triplet q and its linked entity/word w in the embedding space, we can define the conditional probability $p(w|q)$ in a form similar to that in Eq. (1).

$$p(w|q) = \frac{\exp(\boldsymbol{e}_w^T \boldsymbol{e}_q)}{\sum_{w' \in \mathcal{W}} \exp(\boldsymbol{e}_{w'}^T \boldsymbol{e}_q)}, \tag{5}$$

where $\boldsymbol{e}_q$ is the triplet embedding. Then, following the derivation above, the objective function of the triplet–entity/word edges is

$$\mathcal{J}_{TW} = \sum_q \sum_{w_p} \sum_{w_n} \max\Big(0, \gamma + \boldsymbol{e}_{w_n}^T \boldsymbol{e}_q - \boldsymbol{e}_{w_p}^T \boldsymbol{e}_q\Big). \tag{6}$$

Meta-Pattern – Entity / Word. The meta-pattern–entity/word part is similar to the previous two edge types. We first define $p(w|p)$ using the softmax function, and then adopt the margin-based ranking loss:

$$\mathcal{J}_{PW} = \sum_p \sum_{w_p} \sum_{w_n} \max\Big(0, \gamma + \boldsymbol{e}_{w_n}^T \boldsymbol{e}_p - \boldsymbol{e}_{w_p}^T \boldsymbol{e}_p\Big). \tag{7}$$

Here, $\boldsymbol{e}_p$ denotes the meta-pattern embedding.

Entity / Word – Context. To model the relationship between each word and its context, Mikolov et al. [25] propose the Skip-Gram model where $\mathcal{C}(w_i, h)$ is predicted given the center word w_i. Following [25], we define the following conditional probability:

$$p(\mathcal{C}(w_i, h)|w_i) = \prod_{w_j \in \mathcal{C}(w_i, h)} \frac{\exp(\boldsymbol{c}_{w_j}^T \boldsymbol{e}_{w_i})}{\sum_{w' \in \mathcal{W}} \exp(\boldsymbol{c}_{w'}^T \boldsymbol{e}_{w_i})}. \tag{8}$$

Note that each word w has two embedding vectors in Eq. (8): $\boldsymbol{e}_w$ when w is viewed as a center word and $\boldsymbol{c}_w$ when w is a context word [25].

To maximize the log-likelihood $\log p(\mathcal{C}(w_i, h)|w_i)$, we adopt the margin-based ranking loss again. For each positive pair (w, w_p) where $w_p \in \mathcal{C}(w, h)$, we generate a negative pair (w, w_n), and the loss function can be defined as

$$\mathcal{J}_{WW} = \sum_w \sum_{w_p} \sum_{w_n} \max\Big(0, \gamma + \boldsymbol{c}_{w_n}^T \boldsymbol{e}_w - \boldsymbol{c}_{w_p}^T \boldsymbol{e}_w\Big). \tag{9}$$

Given the objective of each edge type, our embedding learning process can be formulated as a joint optimization problem as follows.

$$\begin{aligned} &\min_{\{\boldsymbol{e}_w\}, \{\boldsymbol{e}_s\}, \{\boldsymbol{e}_q\}, \{\boldsymbol{e}_p\}, \{\boldsymbol{c}_w\}} \mathcal{J} = \mathcal{J}_{SW} + \mathcal{J}_{TW} + \mathcal{J}_{PW} + \mathcal{J}_{WW}, \\ &\text{s.t.} \quad ||\boldsymbol{e}_w||_2 = ||\boldsymbol{e}_s||_2 = ||\boldsymbol{e}_q||_2 = ||\boldsymbol{e}_p||_2 = ||\boldsymbol{c}_w||_2 = 1. \end{aligned} \tag{10}$$

Note that we impose an L2-norm constraint that all embeddings should be unit vectors. In other words, our embedding space is a high-dimensional sphere. Therefore, the embedding learning step here is different from many existing HIN embedding approaches in the Euclidean space [35, 8, 11, 50]. We propose this spherical HIN embedding objective for two reasons: (1) Many downstream tasks in both text mining [16, 49] and network mining [35, 8] can be formulated as a similarity search problem in the embedding space. In these cases, the embedding vectors will be normalized onto a sphere to calculate their cosine similarity. In our EVIDENCEMINER framework, we will face the same problem of cosine similarity calculation in text evidence retrieval (see Section VI-B). Therefore, we propose to directly embed the nodes onto a spherical space. In fact, this strategy has proved to be effective in text embedding [23]. (2) Without these constraints, the gap between positive and negative pairs (e.g., $\boldsymbol{e}_{w_n}^T \boldsymbol{e}_s - \boldsymbol{e}_{w_p}^T \boldsymbol{e}_s$) can approach $-\infty$ when $||\boldsymbol{e}_s||_2$ becomes arbitrarily large, which makes the optimization problem trivial.

Optimization. Directly computing the embedding objective (or its gradient) requires summing over all entities/words, sentences, triplets and meta-patterns, which is computationally expensive for large-scale datasets. Therefore, we adopt the sampling technique introduced in [42, 41]. At each iteration, we alternatively sample from the four types of positive pairs (i.e., (s, w_p), (q, w_p), (p, w_p) and (w_p, w)) according to the edge weights. For each positive pair (e.g., (s, w_p)), we sample a negative pair (e.g., (s, w_n)) from the noise distribution [25, 42]. Then we can calculate the Euclidean gradient $\nabla^E \mathcal{J}$

of the embedding vectors based on the positive and negative pairs. Given that all embedding vectors reside on a sphere, we apply the Riemannian stochastic gradient method [4] for optimization. Specifically, we calculate the Riemannian gradient ∇^R on a sphere based on the Euclidean gradient ∇^E according to the following equation [23]:

$$\nabla^R \mathcal{J}(\boldsymbol{e}) = (\boldsymbol{I} - \boldsymbol{e}\boldsymbol{e}^T)\nabla^E \mathcal{J}(\boldsymbol{e}). \tag{11}$$

Using the Riemannian gradient, we can update the embedding vectors in the spherical space. For more details, one can refer to the work on spherical text embedding [23, 24].

VI. Unsuperivsed Textual Evidence Retrieval

The above two steps (data preparation and HIN embedding) are pre-computed offline to support fast online retrieval. Given a user-input query, we automatically retrieve the textual evidence online by generating a ranked list of sentences indicating how strongly they support the input query. In this section, we discuss the query processing and textual evidence retrieval in detail.

A. Query Processing

Given a user-input query, the query is first processed and connected with the nodes in the constructed HIN. The embedding of the query is inferred by one-hop neighbors of the query in the HIN.

Query-HIN Connections. The input queries should be in a format of relational triplets that later can be added into the existing knowledge bases. Given an input query, we first try to match the query entities and the query relation words directly to the constructed HIN. The matched query entities and words will build edges between the input query and the candidate evidence sentence in the constructed HIN. As we discussed in Section III, we assume there exists at least one entity or word in the query that can be matched to the constructed HIN. Otherwise, it is unlikely that any textual evidence sentences can be extracted from our input corpus to support this query.

Query Embedding. After the query is connected with the nodes in the constructed HIN, we infer the query embedding based on its one-hop neighbors in the HIN. To be specific, we calculate the query embedding $\boldsymbol{e}_q$ based on the node embeddings $\boldsymbol{e}_w$, where w are the one-hop neighbors of the query q in the HIN (denoted as $N(q)$).

$$\boldsymbol{e}_q = \frac{1}{|N(q)|} \sum_{w \in N(q)} \boldsymbol{e}_w. \tag{12}$$

The query embedding is a maximum likelihood estimator that maximize the probability of this query generating all its one-hop neighbors in the HIN.

B. Unsuperivsed Textual Evidence Retrieval

After we get the query embedding, we generate a ranked list of evidence sentences based on the semantic similarity between the input query and the candidate evidence sentences in the input corpus. We use the cosine similarity to measure the semantic similarity between the query and the sentences in the HIN, which is also our motivation to directly embed the nodes onto a spherical space. In particular, we rank the sentences s for the input query q with a similarity score $S(q, s)$:

$$S(q, s) = \frac{\boldsymbol{e}_q^T \boldsymbol{e}_s}{\|\boldsymbol{e}_q\| \|\boldsymbol{e}_s\|}. \tag{13}$$

For example, in Figure 3, the sentence "resveratrol induces apoptosis in pancreatic cancer cells by inhibiting BCL-2 expression" could be a retrieved textual evidence for a query such as "*(resveratrol, decrease activity, pancreatic cancer)*".

VII. Experiments

In this section, we demonstrate the effectiveness of EvidenceMiner in unsupervised textual evidence retrieval. We compare the textual evidence retrieval performance of EvidenceMiner with three types of baseline methods. We also conduct case studies to demonstrate that our HIN embedding greatly benefits many downstream applications such as textual evidence interpretation and synonym meta-pattern discovery.

A. Experimental Setup

Datasets. To test the effectiveness of EvidenceMiner, we collect a huge biomedical literature corpus (a subset of PubMed[2]) that contains the most recent 20 years' publications for Molecular Biology[3]. This corpus contains 512,316 abstracts with 4,657,908 sentences and is used as our input corpus. We choose a widely used biomedical knowledge base (Comparative Toxicogenomics Database[4]) as our input knowledge base. CTD contains 29,754,943 relational triplets for three major biomedical entity types (i.e., genes, chemicals and diseases). Based on the input corpus and knowledge base, we randomly selected 122 triplets from CTD and retrieved all the candidate evidence sentences by a simple keyword matching. Then we ask domain experts (graduate students in biomedicine) to manually label all the candidate evidence sentences as correct or not for each input query. Among all the 122 query triplets, 91 of them have at least one retrieved evidence sentence labeled as correct. So we use the 91 queries to compare the performance of different baseline methods in the following experiments. Interestingly, we observe that more than 80% of the queries have only one or two textual evidence sentences labeled as correct in our input corpus. It indicates the sparsity of textual evidence in our task, which requires the textual evidence retrieval methods to be highly accurate in order to achieve an effective retrieval performance. A detailed discussion of the queries used in our experiments can be found in Section VIII.

Evaluation Metrics. We apply standard IR metrics to compare the textual evidence retrieval performance as follows:

- **Average Precision@k**: average of precision at k for each $\langle q, s \rangle$.

[2]https://www.ncbi.nlm.nih.gov/pubmed/

[3]https://www.ncbi.nlm.nih.gov/nlmcatalog/?term=Molecular+Biology%5Bst%5D

[4]http://ctdbase.org

TABLE II
COMPARISON OF THE UNSUPERIVSED TEXTUAL EVIDENCE RETRIEVAL PERFORMANCE WITH BASELINE METHODS.

Method	Precision		nDCG		Recall	
	@1	@2	@1	@2	@1	@2
BM25	0.544	0.406	0.544	0.622	0.452	0.624
LDA	0.022	0.022	0.022	0.034	0.022	0.034
Sent2Vec	0.067	0.044	0.067	0.072	0.049	0.071
Sentence-BERT	0.189	0.144	0.189	0.211	0.131	0.202
TransE	0.111	0.006	0.111	0.111	0.111	0.111
DeepWalk	0.589	**0.422**	0.589	0.638	0.458	0.617
LINE (1st order)	0.289	0.194	0.289	0.301	0.232	0.284
LINE (2nd order)	0.556	0.394	0.556	0.607	0.454	0.595
ESim	0.356	0.239	0.356	0.349	0.246	0.315
Metapath2Vec	0.300	0.228	0.300	0.339	0.228	0.338
EVIDENCEMINER	**0.622**	0.417	**0.622**	**0.653**	**0.502**	**0.630**

- **Average nDCG@k**: average of normalized discounted cumulative gain at k for each $\langle q, s\rangle$.
- **Average Recall@k**: average of recall at k for each $\langle q, s\rangle$.

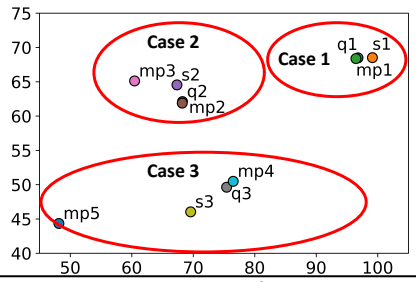
Baseline Methods. We compare EVIDENCEMINER with three types of baseline methods for unsuperivsed textual evidence retrieval as follows:

- **Traditional IR methods**: We use *BM25* [33] and *Latent Dirichlet Allocation (LDA)* [3] as the representative traditional IR methods. For the LDA baseline, we consider the cosine similarity between LDA-assigned weights for both documents and queries to rank the relevant documents.
- **Text embedding-based methods**: We use *Sent2Vec* [27] and *Sentence-BERT* [32] as the representative text embedding-based methods. Sent2Vec is an extension of the CBOW word embedding model [25] which learns distributed representations of sentences. Sentence-BERT uses a pre-trained sentence transformer model to encode the corpus sentences and queries using BERT [7].
- **Graph embedding-based methods**: We use both homogeneous graph embedding methods (*DeepWalk* [30] and *LINE* [42]) and heterogeneous graph embedding methods (*TransE* [5], *ESim* [35] and *Metapath2Vec* [8]) as representative graph embedding-based methods. EVIDENCEMINER is also a heterogeneous graph embedding-based method. Some related work of HIN embedding is introduced in Section II.

B. Performance of Unsuperivsed Textual Evidence Retrieval

To demonstrate the effectiveness of EVIDENCEMINER in unsupervised textual evidence retrieval, we compare EVIDENCEMINER with the baseline methods using the standard IR evaluation metrics (Precision@1,2, nDCG@1,2 and Recall@1,2). The results are shown in Table II.

In general, the traditional keyword-based retrieval methods and the graph embedding-based methods both perform better than the text-embedding based methods. One reason that the text-embedding based methods do not perform well could be that these methods capture the overall semantics of the query as a whole instead of paying special attention to the concrete query entities. As a result, their top-retrieved sentences may have a close semantic meaning with the query relation but miss some query entities. Comparing the traditional keyword-based retrieval methods and the graph embedding-based methods, the graph embedding methods show better performance



ID	HIN node
q1	(cftr, therapeuti, amikacin)
q2	(trpm8, increase activity, icilin)
q3	(nahs, decrease expression, angii)
s1	Amikacin may be a superior choice for suppression therapy in patients that carry a premature stop mutation in CFTR.
s2	Icilin is recognized as a chemical agonist of nociceptors and can activate TRPM8 channels.
s3	The pretreatment of HUVECs mimicked the inhibitory effect of NaHS on the expression of p65 induced by AngII.
mp1	$GENE therapeuti $CHEMICAL
mp2	$GENE increase {activity $CHEMICAL}
mp3	$CHEMICAL increase {$GENE activity}
mp4	$CHEMICAL decrease {expression $GENE}
mp5	$CHEMICAL decrease {elevated $GENE levels}

Fig. 4. Embeddings of queries and retrieved evidence sentences with interpretable meta-patterns generated by tSNE.

(e.g., DeepWalk, LINE (2nd order) and EVIDENCEMINER outperform BM25). Among all the baseline graph embedding methods, DeepWalk and LINE (2nd order) show the best performance. The reason that TransE does not perform well could be that our graph is not a knowledge graph and do not have different types on the edges. The reason that the baseline HIN embedding methods (e.g., Metapath2Vec) do not perform well could be that their embeddings are obtained based on human-defined meta-paths in the HIN. The human-defined meta-paths may not be inclusive enough on our large HIN. Also, we use equal weights for all the meta-paths and do not consider their different effectiveness. Among all the methods, EVIDENCEMINER is the most competitive for unsupervised textual evidence retrieval. The reason that EVIDENCEMINER achieves a good performance could be that EVIDENCEMINER effectively leverages the heterogeneous information from both the input KBs and input corpus. Moreover, the proposed spherical HIN embedding could be a more natural fit for the cosine similarity-based retrieval tasks. If we change the spherical embedding in EVIDENCEMINER to a Euclidean embedding, the performance is close to LINE (2nd order).

C. HIN Embedding Applications

To further demonstrate the effectiveness of EVIDENCEMINER HIN embedding, we conduct case studies to show that our HIN embedding greatly benefits many downstream applications. In this section, we discuss two interesting applications: textual evidence interpretation and synonym meta-pattern discovery.

Textual Evidence Interpretation. Since all the heterogeneous information is embedded into the same space, we can interpret the top-retrieved evidence sentences by their most closely

TABLE III
COMPARISON OF THE SYNONYM META-PATTERN DISCOVERY PERFORMANCE.

Method	Precision			Recall		
	@5	@10	@20	@5	@10	@20
PATTY+	0.560	0.660	0.330	0.160	0.368	0.368
TruePIE	0.680	0.600	0.430	0.198	0.357	0.504
EVIDENCEMINER	**0.919**	**0.840**	**0.580**	**0.264**	**0.473**	**0.644**

related meta-patterns. We also use the cosine similarity to measure the node similarity between the sentences and meta-patterns in the HIN.

We show some case studies in Figure 4 of the generated explanations with meta-patterns in our HIN. For example, in case 1, the top-retrieved sentence "Amikacin may be a superior choice for suppression therapy in patients that carry a premature stop mutation in CFTR" has a most closely related meta-pattern "$GENE therapeuti $CHEMICAL", which very well explains why it is retrieved as a top evidence for the query "*(cftr, therapeuti, amikacin)*". Similarly, in case 2, the top-retrieved sentence "Icilinis recognized as a chemical agonist of nociceptors and can activate TRPM8 channels" has a most closely related meta-pattern "$GENE increase activity $CHEMICAL", which very well explains why it is retrieved as a top evidence for the query "*(trpm8, increase activity, icilin)*". Also, in case 3, the top-retrieved sentence "The pretreatment of HUVECs mimicked the inhibitory effect of NaHS on the expression of p65 induced by AngII" has the most closely related meta-patterns "$CHEMICAL decrease expression $GENE" and "$CHEMICAL decrease elevated $GENE levels", which very well explains why it is retrieved as a top evidence for the query "*(nahs, decrease expression, angii)*". The meta-patterns in the HIN not only help build connections between the input queries and candidate evidence sentences but also give additional interpretations to our retrieval results. The interpretation of the retrieved textual evidence has not been addressed by existing methods. It is key to the evidence usability in the scientific domain.

Synonym Meta-pattern Discovery. Another interesting application is synonym meta-pattern discovery. Meta-pattern discovery [14] is an important task for pattern-based open information extraction. It aims to automatically extract massive structured relationships from text without any pre-defined relation types. One major challenge for the existing meta-pattern discovery methods is to effectively group synonym meta-patterns together to reduce the pattern redundancy for downstream applications such as knowledge base completion. We compare EVIDENCEMINER with two baseline methods and found our HIN embedding is much more effective for synonym meta-pattern grouping. Here we only consider unsupervised synonym meta-pattern grouping methods based on the entities extracted by the meta-patterns and the relation words in the meta-patterns.

- **PATTY+** [26]: *PATTY* is a feature-based method that discovery synonym meta-patterns by their common entity extractions. Since the common entity extractions are usually sparse in scientific literature, PATTY almost cannot find any synonym patterns in our corpus. We enhance it as PATTY+ that represents each meta-pattern with both its entity extractions and relation words.
- **TruePIE** [19]: *TruePIE* is a text embedding-based method that represents each pattern as the concatenation of two text embedding vectors: the average word embedding of the relation words and the average transE embedding between the head and the tail entities. It measures the meta-pattern similarity by the cosine similarity between two meta-pattern embedding vectors.
- **EVIDENCEMINER**: EVIDENCEMINER is our HIN embedding-based method. For a fair comparison, the HIN embedding used here is obtained from a sub-graph in our original HIN that only includes the nodes of the meta-patterns, the entities extracted by the meta-patterns and the relation words in the meta-patterns.

We evaluate the synonym meta-pattern grouping results by manually labeling the top-20 meta-patterns in text retrieved by each baseline method given 10 seed meta-patterns in the knowledge base. We compare the performance using standard IR evaluation metrics (Precision@5,10,20 and Recall@5,10,20) in Table III and Table IV. From Table III, we observe that EVIDENCEMINER performs substantially better than PATTY+ and TruePIE for synonym meta-pattern discovery. Since the features are usually very sparse in scientific literature, PATTY+ often has a low recall for meta-pattern grouping. TruePIE relies heavily on the relation word embeddings, which ignores the local context of the extracted entities and leads to a semantic drift for the synonym meta-pattern grouping. For example, in Table IV, the top results of TruePIE for the seed meta-pattern "$CHEMICAL increase abundance $GENE" is "{limited availability of $CHEMICAL} increase $GENE" that has exactly the opposite meaning due to the local context "limited availability". Although TruePIE also includes the transE embedding between the head and the tail entities, this embedding can be noisy because there is no clear direction between the head and tail entities of the scientific triplets. EVIDENCEMINER not only has better performance for meta-pattern grouping but also produce more diverse grouping results. For example, in Table IV, the top results of EVIDENCEMINER for the seed meta-pattern "$CHEMICAL decrease activity $DISEASE" include "$CHEMICAL decrease $DISEASE cell", "$CHEMICAL decrease $DISEASE cell adhesion and migration" and "$CHEMICAL decrease $DISEASE growth".

VIII. DISCUSSION

As we mentioned in Section VII-A, more than 80% of the queries have only one or two textual evidence sentences labeled as correct in our input corpus. According to the domain experts, one reason could be that we are dealing with highly specific query triplets (e.g., *(nahs, decrease expression, angii)*) in scientific research instead of general query triplets (e.g., *(smoking, cause, cancer)*) in the news or social media.

TABLE IV
TOP-5 SYNONYM META-PATTERNS IN TEXT GIVEN A SEED META-PATTERN IN THE KNOWLEDGE BASE.

$CHEMICAL increase abundance $GENE		
PATTY+	TruePIE	EVIDENCEMINER
{treatment with $CHEMICAL} increase $GENE ✓	{limited availability of $CHEMICAL} increase $GENE ✗	$CHEMICAL increase {the $GENE gene expression} ✓
{transactivation of $GENE} increase $CHEMICAL ✗	$CHEMICAL increase {the level of $GENE} ✓	{the drug, CHEMICAL,} increase {the expression of $GENE} ✓
{ICV injection of $GENE} increase $CHEMICAL ✗	$CHEMICAL increase {$GENE levels} ✓	$GENE increase {the chemotherapeutic drug $CHEMICAL} ✗
$CHEMICAL increase {$GENE activity} ✓	$GENE increase {the intracellular levels of $CHEMICAL} ✗	$CHEMICAL increase {$GENE stability} ✓
{expression of $GENE} increase $CHEMICAL ✗	$CHEMICAL increase {the level stability of $GENE} ✓	$CHEMICAL increase {$GENE expression} ✓
$CHEMICAL decrease response to substance $GENE		
PATTY+	TruePIE	EVIDENCEMINER
$CHEMICAL decrease {$GENE secretion} ✓	{$CHEMICAL from folium isatidis} decrease {$GENE expression} ✓	$CHEMICAL response bind $GENE ✗
$CHEMICAL decrease {$GENE production} ✓	{stimulation with $CHEMICAL alone} increase {basal $GENE secretion} ✗	{$CHEMICAL treatment of cells} decrease {$GENE expression} ✓
$CHEMICAL response bind $GENE ✗	{chronic inhibition of $GENE} increase {$CHEMICAL synthesis} ✗	$CHEMICAL treatment decrease {$GENE expression} ✓
$CHEMICAL decrease {$GENE expression} ✓	{$CHEMICAL compound} decrease {$GENE protein levels} ✓	$CHEMICAL decrease {$GENE expression} ✓
{a decrease in $GENE} mediate $CHEMICAL ✗	$CHEMICAL decrease {basal $GENE transport activity} ✓	$CHEMICAL decrease {$GENE production} ✓
$CHEMICAL decrease activity $DISEASE		
PATTY+	TruePIE	EVIDENCEMINER
{$DISEASE activity} of $CHEMICAL ✗	{$CHEMICAL treatment} result in {the $DISEASE activity} ✗	$CHEMICAL decrease {activity $DISEASE} ✓
$CHEMICAL decrease {$DISEASE growth} ✓	{the addition of $CHEMICAL} decrease $DISEASE ✓	{treatment with $CHEMICAL} decrease {$DISEASE cells} ✓
{$CHEMICAL treatment} decrease $DISEASE ✓	$CHEMICAL decrease {$DISEASE growth}✓	$CHEMICAL decrease {$DISEASE cell} ✓
$CHEMICAL decrease {$DISEASE growth in MKR} ✓	$CHEMICAL decrease {$DISEASE growth in MKR} ✓	$CHEMICAL decrease {$DISEASE cell adhesion and migration} ✓
$CHEMICAL decrease {$DISEASE cell} ✓	{$CHEMICAL treatment} decrease $DISEASE ✓	$CHEMICAL decrease {$DISEASE growth} ✓

Considering that we are randomly sampling from about 30 million queries to find evidence from about 4 million sentences, this sparsity of the ground-truth evidence sentences can be expected. Another reason could be that we are only dealing with explicit evidence within one sentence without considering implicit evidence that could span multiple sentences. Among all the textual evidence in the biomedical literature, more than half of them span multiple sentences [29]. This is one limitation of our current work that could not deal with cross-sentence evidence retrieval. It could be an interesting problem that can be explored in the future.

As we mentioned in Section I, it is non-trivial to retrieve a large set of human-annotated articles for textual evidence retrieval, especially in scientific literature where expert annotations are expensive. Therefore, we are conducting textual evidence retrieval under the unsupervised setting and we do not compare the performance with the supervised methods (e.g., supervised text-embedding or supervised graph embedding). However, it could be an interesting direction to explore if some weak or distant supervision can be further incorporated to improve the textual evidence retrieval performance.

IX. CONCLUSION

We studied the problem of textual evidence mining in scientific literature: given a scientific hypothesis as a query triplet, find the textual evidence sentences in scientific literature that support the input query. We propose EVIDENCEMINER, a novel method for unsuperivsed textual evidence retrieval in scientific literature without human-annotated training examples. To achieve high-quality textual evidence retrieval, We propose to construct a large HIN that leverages heterogeneous information from both existing knowledge bases and massive unstructured text to build connections between the user-input queries and the candidate evidence sentences. Based on the constructed HIN, we propose a novel HIN embedding method that directly embeds the nodes onto a spherical space to improve the unsupervised retrieval performance. Quantitative experiments on a huge biomedical literature corpus demonstrate that EVIDENCEMINER significantly outperforms baseline methods for unsupervised textual evidence retrieval. Case studies also demonstrate that our HIN embedding greatly benefits many downstream applications such as textual evidence interpretation and synonym meta-pattern discovery. Further improvements can be explored such as improving the method for finding textual evidence across multiple sentences.

ACKNOWLEDGMENT

Research was sponsored in part by US DARPA KAIROS Program No. FA8750-19-2-1004 and SocialSim Program No. W911NF-17-C-0099, National Science Foundation IIS-19-56151, IIS-17-41317, IIS 17-04532, and IIS 16-18481, and DTRA HDTRA11810026. Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and should not be interpreted as necessarily representing the views, either expressed or implied, of DARPA or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright annotation hereon.

REFERENCES

[1] K. Al-Khatib, H. Wachsmuth, M. Hagen, J. Köhler, and B. Stein. Cross-domain mining of argumentative text through distant supervision. In *NAACL'16*, pages 1395–1404, 2016.

[2] A. Allot, Q. Chen, S. Kim, R. Vera Alvarez, D. C. Comeau, W. J. Wilbur, and Z. Lu. Litsense: making sense of biomedical literature at sentence level. *Nucleic acids research*, 2019.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3:993–1022, 2003.

[4] S. Bonnabel. Stochastic gradient descent on riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229, 2013.

[5] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. In *NIPS'13*, pages 2787–2795, 2013.

[6] Y. Cen, X. Zou, J. Zhang, H. Yang, J. Zhou, and J. Tang. Representation learning for attributed multiplex heterogeneous network. In *KDD'19*, pages 1358–1368, 2019.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL'19*, pages 4171–4186, 2019.

[8] Y. Dong, N. V. Chawla, and A. Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *KDD'17*, pages 135–144, 2017.

[9] Y. Dong, Z. Hu, K. Wang, Y. Sun, and J. Tang. Heterogeneous network representation learning. In *IJCAI'20*, pages 4861–4867, 2020.

[10] J. Du, J. Z. Pan, S. Wang, K. Qi, Y. Shen, and Y. Deng. Validation of growing knowledge graphs by abductive text evidences. In *AAAI'19*, pages 2784–2791, 2019.

[11] T.-y. Fu, W.-C. Lee, and Z. Lei. Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In *CIKM'17*, pages 1797–1806, 2017.

[12] Y. He, Y. Song, J. Li, C. Ji, J. Peng, and H. Peng. Hetespaceywalk: a heterogeneous spacey random walk for heterogeneous information network embedding. In *CIKM'19*, pages 639–648, 2019.

[13] R. Hussein, D. Yang, and P. Cudré-Mauroux. Are meta-paths necessary? revisiting heterogeneous graph embeddings. In *CIKM'18*, pages 437–446, 2018.

[14] M. Jiang, J. Shang, T. Cassidy, X. Ren, L. M. Kaplan, T. P. Hanratty, and J. Han. Metapad: Meta pattern discovery from massive text corpora. In *KDD'17*, pages 877–886, 2017.

[15] J. Lehmann, D. Gerber, M. Morsey, and A.-C. N. Ngomo. Defacto-deep fact validation. In *ISWC'12*, pages 312–327, 2012.

[16] O. Levy, Y. Goldberg, and I. Dagan. Improving distributional similarity with lessons learned from word embeddings. *TACL*, 3:211–225, 2015.

[17] R. Levy, Y. Bilu, D. Hershcovich, E. Aharoni, and N. Slonim. Context dependent claim detection. In *COLING'14*, pages 1489–1500, 2014.

[18] R. Levy, S. Gretz, B. Sznajder, S. Hummel, R. Aharonov, and N. Slonim. Unsupervised corpus–wide claim detection. In *Proc. Work. Arg. Min.*, pages 79–84, 2017.

[19] Q. Li, M. Jiang, X. Zhang, M. Qu, T. P. Hanratty, J. Gao, and J. Han. Truepie: Discovering reliable patterns in pattern-based information extraction. In *KDD'18*, pages 1675–1684, 2018.

[20] Q. Li, X. Wang, Y. Zhang, Q. Li, F. Ling, C. H. Wu, and J. Han. Pattern discovery for wide-window open information extraction in biomedical literature. In *BIBM'18*, pages 420–427. IEEE, 2018.

[21] M. Lippi and P. Torroni. Argument mining: A machine learning perspective. In *Int. Work. Theor. App. Form. Arg.*, pages 163–176, 2015.

[22] M. Lippi and P. Torroni. Argumentation mining: State of the art and emerging trends. *TOIT*, 16(2):10, 2016.

[23] Y. Meng, J. Huang, G. Wang, C. Zhang, H. Zhuang, L. Kaplan, and J. Han. Spherical text embedding. In *NeurIPS'19*, pages 8208–8217, 2019.

[24] Y. Meng, Y. Zhang, J. Huang, Y. Zhang, C. Zhang, and J. Han. Hierarchical topic mining via joint spherical tree and text embedding. In *KDD'20*, pages 1908–1917, 2020.

[25] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS'13*, pages 3111–3119, 2013.

[26] N. Nakashole, G. Weikum, and F. Suchanek. Patty: a taxonomy of relational patterns with semantic types. In *EMNLP-CoNLL'12*, pages 1135–1145, 2012.

[27] M. Pagliardini, P. Gupta, and M. Jaggi. Unsupervised learning of sentence embeddings using compositional n-gram features. In *NAACL-HLT'18*, pages 528–540, 2018.

[28] J. Pasternack and D. Roth. Knowing what to believe (when you already know something). In *COLING'10*, pages 877–885, 2010.

[29] N. Peng, H. Poon, C. Quirk, K. Toutanova, and W.-t. Yih. Cross-sentence n-ary relation extraction with graph lstms. *TACL*, 5:101–115, 2017.

[30] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *KDD'14*, pages 701–710, 2014.

[31] H. Rashkin, E. Choi, J. Y. Jang, S. Volkova, and Y. Choi. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *EMNLP'17*, pages 2931–2937, 2017.

[32] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP'19*, pages 3973–3983, 2019.

[33] S. Robertson, H. Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *FnT Inf. Ret.*, 3(4):333–389, 2009.

[34] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling. Modeling relational data with graph convolutional networks. In *ESWC'18*, pages 593–607, 2018.

[35] J. Shang, M. Qu, J. Liu, L. M. Kaplan, J. Han, and J. Peng. Meta-path guided embedding for similarity search in large-scale heterogeneous information networks. *arXiv preprint arXiv:1610.09769*, 2016.

[36] Y. Shi, H. Gui, Q. Zhu, L. Kaplan, and J. Han. Aspem: Embedding learning by aspects in heterogeneous information networks. In *SDM'18*, pages 144–152, 2018.

[37] Y. Shi, Q. Zhu, F. Guo, C. Zhang, and J. Han. Easing embedding learning by comprehensive transcription of heterogeneous information networks. In *KDD'18*, pages 2190–2199, 2018.

[38] E. Shnarch, C. Alzate, L. Dankin, M. Gleize, Y. Hou, L. Choshen, R. Aharonov, and N. Slonim. Will it blend? blending weak and strong labeled data in a neural network for argumentation mining. In *NAACL'18*, pages 599–605, 2018.

[39] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *PVLDB*, 4(11):992–1003, 2011.

[40] Z. H. Syed, M. Röder, and A.-C. Ngonga Ngomo. Factcheck: Validating rdf triples using textual evidence. In *CIKM'18*, pages 1599–1602, 2018.

[41] J. Tang, M. Qu, and Q. Mei. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *KDD'15*, pages 1165–1174, 2015.

[42] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. Line: Large-scale information network embedding. In *WWW'15*, pages 1067–1077, 2015.

[43] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*, 2018.

[44] X. Wang, Y. Guan, W. Liu, A. Chauhan, E. Jiang, Q. Li, D. Liem, D. Sigdel, J. Caufield, P. Ping, et al. Evidenceminer: Textual evidence discovery for life sciences. In *ACL'20 System Demonstrations*, pages 56–62, 2020.

[45] X. Wang, Y. Zhang, Q. Li, Y. Chen, and J. Han. Open information extraction with meta-pattern discovery in biomedical literature. In *ACM-BCB'18*, pages 291–300. ACM, 2018.

[46] X. Wang, Y. Zhang, Q. Li, X. Ren, J. Shang, and J. Han. Distantly supervised biomedical named entity recognition with dictionary expansion. In *BIBM'19*, pages 496–503, 2019.

[47] X. Wang, Y. Zhang, X. Ren, Y. Zhang, M. Zitnik, J. Shang, C. Langlotz, and J. Han. Cross-type biomedical named entity recognition with deep multi-task learning. *Bioinformatics*, 35(10):1745–1752, 2019.

[48] C.-H. Wei, H.-Y. Kao, and Z. Lu. Pubtator: a web-based text mining tool for assisting biocuration. *Nucleic Acids Res.*, 41(W1):W518–W522, 2013.

[49] C. Xing, D. Wang, C. Liu, and Y. Lin. Normalized word embedding and orthogonal transform for bilingual word translation. In *NAACL-HLT'15*, pages 1006–1011, 2015.

[50] C. Yang, Y. Xiao, Y. Zhang, Y. Sun, and J. Han. Heterogeneous network representation learning: Survey, benchmark, evaluation, and beyond. *arXiv preprint arXiv:2004.00216*, 2020.

[51] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla. Heterogeneous graph neural network. In *KDD'19*, pages 793–803, 2019.

[52] S. Zhao, B. Cheng, and H. Yang. An end-to-end multi-task learning model for fact checking. In *Proceedings of the First Workshop on Fact Extraction and VERification*, pages 138–144, 2018.