GRAPHENE: A Precise Biomedical Literature Retrieval Engine with Graph Augmented Deep Learning and External Knowledge Empowerment

Sendong Zhao, Chang Su, Andrea Sboner, Fei Wang
Weill Cornell Medical College, Cornell University
New York, USA
{sez4001,csu4001,ans2077,few2001}@med.cornell.edu

ABSTRACT

Effective biomedical literature retrieval (BLR) plays a central role in precision medicine informatics. In this paper, we propose GRAPHENE, which is a deep learning based framework for precise BLR. GRAPHENE consists of three main different modules 1) graph-augmented document representation learning; 2) query expansion and representation learning and 3) learning to rank biomedical articles. The graph-augmented document representation learning module constructs a document-concept graph containing biomedical concept nodes and document nodes so that global biomedical related concept from external knowledge source can be captured, which is further connected to a BiLSTM so both local and global topics can be explored. Query expansion and representation learning module expands the query with abbreviations and different names, and then builds a CNN-based model to convolve the expanded query and obtain a vector representation for each query. Learning to rank minimizes a ranking loss between biomedical articles with the query to learn the retrieval function. Experimental results on applying our system to TREC Precision Medicine track data are provided to demonstrate its effectiveness.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; **Document representation**; **Learning to rank**; • **Computing methodologies** → *Neural networks*; • **Applied computing** → *Life and medical sciences.*

KEYWORDS

Biomedical Literature Retrieval, Graph Augmented Document Representation Learning, Deep Neural Networks, Learning to Rank

ACM Reference Format:
Sendong Zhao, Chang Su, Andrea Sboner, Fei Wang. 2019. GRAPHENE: A Precise Biomedical Literature Retrieval Engine with Graph Augmented Deep Learning and External Knowledge Empowerment. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM '19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3357384.3358038

CIKM '19, November 3–7, 2019, Beijing, China

ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3358038

1 INTRODUCTION

Precision medicine [3], with the goal of providing the right treatment to the right patient at the right time, holds great premise on treating complicated diseases such as cancer. Biomedical literature database stores the state-of-the-art information about the various aspects related to precision medicine (e.g., patients, diseases, genetics, treatments, etc.) and serves as a vital knowledge source. For example, with the prevalence of Next Generation Sequencing (NGS) technology, vast amounts of genetic variants can be identified for a specific patient on a specific tissue (e.g., tumor). The domain experts, such as clinicians, pathologists, oncologists, need to search the relevant literature on PubMed to interpret these genetic variants. Therefore, an efficient and accurate biomedical literature retrieval (BLR) system is crucial.

There are many challenges to building such an effective BLR system. For example, the vocabulary of professional terms in biomedical articles is typically large, and there are many semantic variations for the same biomedical concept (e.g., a specific genetic variant), the relations between these concepts are complicated (which could be explicit/implicit, direct/indirect, and known/unknown, etc.).

The goal of this paper is to develop a high-performance BLR system in view of the above challenges. Specifically, our task is to take patient information with the genetic variant as query and retrieve relevant biomedical articles from literature databases such as MEDLINE[1]. Here the patient information can be either in a structured form (disease name, variant, demographic, ...) or unstructured text. The output should be a list of biomedical articles which are ranked according to the relevance with the query.

There are mainly two types of retrieval models can be used for our scenario. The first type is the traditional document retrieval model which represents the query and documents both as one-hot representations and matches query and documents with similarity measures such as cosine similarity. In particular, one-hot representations can be generated through the bag-of-words model or TF-IDF model. In the bag-of-words model, a text (such as a query or a biomedical article) is represented as the bag (multiset) of its words, disregarding grammar and even word order but keeping multiplicity. TF-IDF model is a better approach compared to the bag-of-words model. It assigns a weighting to each word in the

[1]MEDLINE (https://www.nlm.nih.gov/bsd/medline.html) is a bibliographic database of life sciences and biomedical information. It includes bibliographic information for articles from academic journals covering medicine, nursing, pharmacy, dentistry, veterinary medicine, and health care.

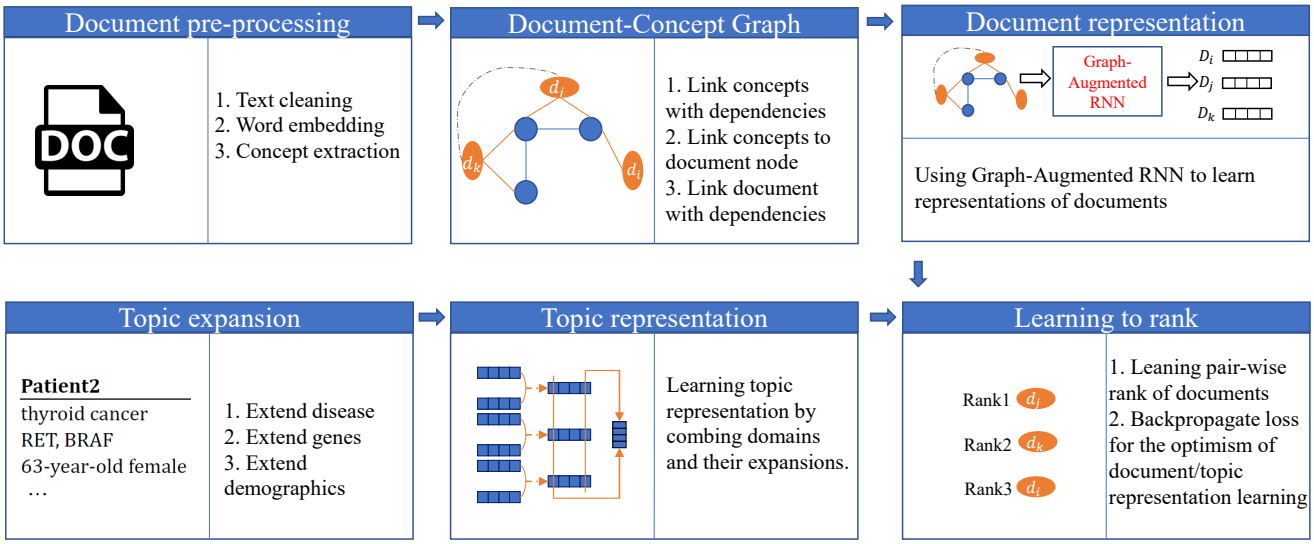

Figure 1: The framework of graph augmented deep learning with knowledge empowerment engine for for biomedical literature retrieval.

document and put that weighting in the vector, which is allowed to normalize the count/frequency of the word. Yet the drawback of the traditional retrieval model is apparent. Consider one takes "lung cancer" as the query, then the model can only retrieve the articles with exactly the term "lung cancer" in their contents. A notable exception is PubMed which expands the query by mapping it to related MeSH terms [21]. Although this increases recall, it often decreases precision which is crucial for precision medicine [13]. The second type is deep learning models, such as DeepMatch [22] and Delta [25], which have demonstrated superior performance in recent years. The advantage of these models comes from their flexible representation learning process for the query and the documents. In particular, the learned representations can encode the underlying semantics for queries and documents. This greatly enhanced the capturing of the semantically similar words but with different expressions such as "cancer" and "tumor", "diabetes" and "hyperglycemia". Therefore, take "diabetes" as query these models might find articles with "hyperglycemia". However, it is still difficult for these models to identify indirectly/implicitly related terms such as "diabetes" and "metformin", in which case we would need external knowledge base [36].

Our developed system is called GRAPHENE, which stands for **GR**aph **A**ugmented dee**P** learning wit**H** knowl**E**dge empowerme**N**t **E**ngine. Our model learns literature representations with local text and external structured knowledge and matches queries with the pair-wise learning to rank mechanism. The overall architecture of GRAPHENE is shown in Figure 1, which is composed of 3 main modules. The first is a document representation learning module. We propose to construct a document-concept graph, upon which a graph-augmented document representation is learned to encode both the underlying semantics of the literature and the information from non-local relevant medical concepts. The second is patient information query representation learning module, where we propose to expand query and learn representations through the convolution upon the expanded query. The third is a learning to rank module, which minimizes the pairwise ranking loss between the patient information and biomedical article to learn a partial order of relevance of biomedical articles. The organic integration of these components makes GRAPHENE possible to retrieve biomedical articles effectively and precisely.

Experimental results on TREC Precision Medicine data [27] demonstrate that our model can effectively retrieve most relevant biomedical articles with patient information. The results also show that our model outperforms those deep neural retrieval models which represent documents with textual information only, suggesting the necessity of encoding graph-based external knowledge into document representation.

The paper is organized as follows. Section 2 includes related studies in biomedical literature retrieval and document retrieval methodology. Section 3 describes the details of graph-augmented document representation learning algorithm. Section 4 describes the details of CNN-based query representation learning algorithm and pair-wise learning-to-rank algorithm. Experimental results are presented in Section 5, and the paper is concluded in Section 6.

2 RELATED WORK

2.1 Biomedical Literature Retrieval

There has already been a lot of research on biomedical literature retrieval [6–8, 25, 30, 39]. For example, Zheng and Wan [39] proposed to use the paragraph vector technique to learn the latent semantic representation of texts and treat the latent semantic representations and the original bag-of-words representations as two different modalities. They then proposed to use the multi-modality learning algorithm to retrieve biomedical literature for clinical decision support. Soldaini et al. [30] applied query reformulation techniques to

address the need of literature search based on case reports. Best Match [6] is the first relevance search algorithm for PubMed that leverages the intelligence of users and machine-learning technology as an alternative to the traditional sorting techniques. Delta [25] is a deep learning based model that applies convolution operation upon an updated document matrix in which each word is replaced with the most similar word in the query. However, this model takes the key topic word and other words in query with equal importance, making the retrieval out of focus.

2.2 Document Retrieval Models

The traditional document retrieval models represent both the queries and documents as one-hot representations and match query and documents with similarity measures like cosine similarity. As we stated in the introduction, the problem of this method is that the semantically similar query-document pairs without exact expression match cannot be identified.

In recent years, deep learning based retrieval models, such as DeepMatch [22], PACRR [17], Delta [25], Conv-KNRM [4], MASH RNN [19], have dominated the document retrieval research. In particular, these models have been focusing on 1) flexible representation learning for query and documents and 2) measuring the similarity between query and documents at different levels. Correspondingly, there are two main categories of deep neural information retrieval (IR) models. One is the representation-focused model, which tries to learn good representations for both query and documents with deep neural networks, and then conducts matching between the learned representations. Examples include DSSM [16], C-DSSM [9], ARC-I [15], Delta [25], MASH RNN [19]. The other is the interaction-focused model, which first builds local interactions (i.e., local matching signals) between the query and documents, and then uses deep neural networks to learn the overall matching score. Examples include DeepMatch [22], ARC-II [15], DRMM [11], ESR [35], PACRR [17], Conv-KNRM [4] and SMASH RNN [19].

These deep neural IR models can effectively exploit the underlying semantics for queries and documents, which can be good at matching similar but differently expressed words like "cancer" and "tumor", "diabetes" and "hyperglycemia". Therefore, take "diabetes" as query the model might find articles with "hyperglycemia". But these model cannot find articles indirectly/implicitly related to "diabetes" like the article with "metformin" because it requires external knowledge (metformin, against, diabetes). Although prior research generally confirms that external knowledge has great value for document retrieval [10, 29, 32], little research has been conducted to show the value of the external knowledge on deep neural IR models. There are studies trying to perform query expansion or leverage pre-trained embeddings of name entities using external knowledge bases. For example, Xiong and Callan [33] presented a simple and effective method to improve query expansion with Freebase. They proposed a supervised model to combine the information derived from Freebase descriptions and categories so that ‘effective terms can be selected for query expansion. In another paper, Xiong et al. [35] proposed a ranking technique for matching query and documents, where a knowledge graph is leveraged to pre-train the embeddings of the named entities in both query and documents.

Table 1: Symbols and descriptions.

Symbol	Description
q_k	k-th query with patient information or MeSH terms
d_i	document nodes, each of them consists of a sequence of k words $(x_1^{(i)}, x_2^{(i)}, ..., x_k^{(i)})$
$d_{p_k}^{(j)}$	the document which is ranked as j-th in the ranking list for query p_k
c_j	biomedical concept nodes
$\mathbf{h}_k^{(i)}$	the k-th hidden state of RNN in modelling document d_i
$\mathbf{g}_{d_i}^{(l)}$	the hidden state of document node v_{d_i} in the l-th layer of the graph neural network.
$\mathbf{g}_{c_i}^{(l)}$	the hidden state of biomedical node v_{c_i} in the l-th layer of the graph neural network.

3 REPRESENTATION LEARNING FOR DOCUMENT-CONCEPT GRAPH

In this section, we present a graph-augmented representation learning approach for documents and biomedical concepts through the Document-Concept Graph (DCG). There are two types of nodes in a DCG: the biomedical concepts and documents. There are also two types of edges in the graph. One links concepts and documents according to their co-occurrence relationships. The other links pairwise concepts if any semantic relationship can be identified between them in a specific knowledge base. With the DCG representation, the global topics of the documents are effectively encoded through the document-concept relationships. Moreover, such topics can further be enhanced with external knowledge sources through the transitions on the graph. In this paper, we denote scalars by lowercase letters, such as x; vectors by boldface lowercase letters, such as $\mathbf{x}$; and matrices by boldface upper case letters, such as $\mathbf{X}$. Table 1 lists the symbols and their descriptions that are used throughout this paper.

More formally, a DCG G consists of two types of nodes, the document nodes $\{d_i\}$ and the biomedical concept nodes $\{c_j\}$. A document d_i is composed of a sequence of k words $(x_1^{(i)}, x_2^{(i)}, ..., x_k^{(i)})$ and contains n medical concepts $(c_1^{(i)}, c_2^{(i)}, ..., c_n^{(i)})$. A specific biomedical concept c_j can be composed of a single word or multiple words, which corresponds to a medical named entity or a key topic in medical domain. We link a concept node c_j to a document node d_i if this document contains the concept c_j. We link two concept nodes c_u and c_v if there exists any relations between them in external knowledge bases. Given the document-concept graph G, our objective is to learn the representation for each document node.

3.1 Document-Concept Graph Construction

Figure 2 illustrates an example of DCG. The construction of a DCG consists of four steps: 1) tokenization; 2) word embedding; 3) medical concept extraction; 4) edge construction.

Tokenization. Given a biomedical article, tokenization is necessary to convert the texts into space-separated sequences of words (words may include punctuation or be followed by punctuation).

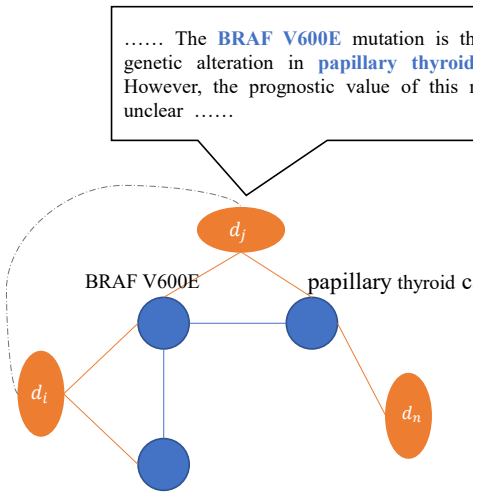

Figure 2: An example of the document-concept graph.

In our work, the Stanford CoreNLP [23] is leveraged to conduct tokenization on documents.

Word Embedding. After tokenization, unsupervised word representation learning models like Glove [26] and Word2Vec [24] are implemented on all documents to get semantic representation for each word which would be useful in later parts.

Medical Concept Extraction. Since each document might contain multiple medical concepts corresponding to different types of biomedical named entities, such as genes, diseases, chemicals, mutations, etc., which can be detected through medical named recognition and normalization methods and tools [31, 37].

Edge Construction. Given all detected medical concepts in documents, we can construct the edges in a DCG. It is straightforward to link the document and concept nodes by just checking whether the concept is included in the document as the same way in [38]. For the edges linking concept nodes, we extract them from external knowledge bases as in [40].

3.2 Document Representation Learning

Given the DCG G, our next step is to learn effective representations for documents, which is a so crucial factor for the performance of our system. In particular, we propose to learn such representation through a graph (DCG) augmented recurrent neural network (RNN) model, which is shown in Figure 3. Our model has three components:

- *Encoder*, which generates context-aware hidden representations for each biomedical document with a recurrent neural network;
- *Graph Module*, which is essentially the DCG capturing the relationships among the documents and biomedical concepts as we introduced in the last subsection.
- *Decoder*, which exploits the contextual information generated by the encoder and the graph to predict key topics of the documents. Our goal is to predict the title for the abstract of each biomedical article.

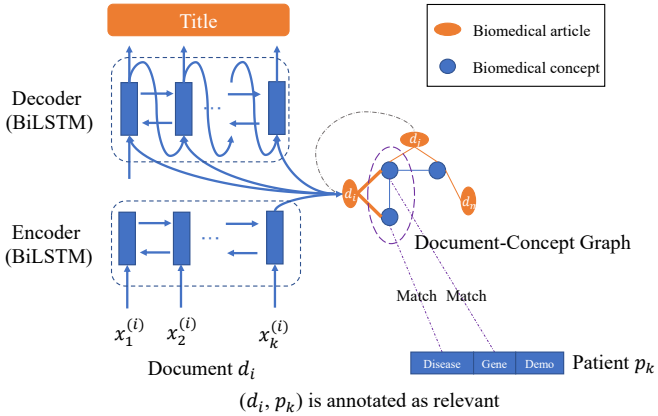

Figure 3: Graph-augmented RNN for document representation learning.

3.2.1 Encoder. We first use an encoder to generate document representations. Given a document d_i with word sequence $(x_1^{(i)}, x_2^{(i)}, ..., x_k^{(i)})$ of length k, each word $x_k^{(i)}$ is represented by a vector $\mathbf{x}_\mathbf{k}^{(i)}$, which is obtained from some pre-trained word embedding. We encode the document with a recurrent neural network defined as

$$\mathbf{h}_{1:k}^{(i)} = \text{RNN}(\mathbf{x}_{1:k}^{(i)}; \mathbf{0}; \Theta_{\text{enc}}) \quad (1)$$

where $\mathbf{x}_{1:k}^{(i)}$ represents the input word sequence $(x_1^{(i)}, x_2^{(i)}, ..., x_k^{(i)})$, $\mathbf{h}_{1:k}^{(i)}$ denotes the hidden states $[\mathbf{h}_1^{(i)}, \mathbf{h}_2^{(i)}, ..., \mathbf{h}_k^{(i)}]$. $\mathbf{0}$ denotes an all-zero vector. Θ_{enc} denotes parameters of the encoder. We implement the RNN as a bi-directional LSTM [14] to encode each document.

We construct the textual representation for document d_i by averaging the hidden states of its words, i.e. $\text{Enc}(d_i) = \frac{1}{k}(\sum_{t=1}^{k} \mathbf{h}_t^{(i)})$. These learned representations are then fed into the DCG as the initial representation of the document nodes. The final document representation will be the combination of the textual representation and the structural representation induced from the DCG. In the following, we introduce how such structural representations are learned.

3.2.2 The Graph Module. The graph module is designed to get the structural representations for document nodes via message passing in the DCG G. Specifically, we adopt a restricted graph convolutional network (GCN) to achieve this goal.

Given the DCG $G = (V, E)$, where for a document node v_{d_i} (representing document d_i), it has the encoding $\text{Enc}(d_i)$ capturing its textual information. For a concept node v_{c_j} (representing biomedical concept c_j), it has a pre-trained word embedding representation. The graph module enhances such representation with the graph structure in DCG. More concretely, it operates on local neighborhoods in the graph to integrate medical concepts information to document nodes. This means that we want the information flow from concept nodes to document nodes and the information flow between concept nodes, but we do not want the concept node to be influenced by the information from document nodes since we assume the semantic of a medical concept is much more concrete and stable than the document.

Precisely, at each layer of the GCN, each document node information from its neighboring concept nodes, i.e.,

$$\mathbf{g}_{d_i}^{(l+1)} = \delta\left(\sum_{m\in\mathcal{M}_i} \alpha f_m(\mathbf{g}_{d_i}^{(l)}, \mathbf{g}_{c_j}^{(l)})\right)$$

where $\mathbf{g}_{d_i}^{(l)} \in \mathbb{R}^{d^{(l)}}$ is the hidden state of document node v_{d_i} in l-th layer of the neural network, with $d^{(l)}$ being the dimenality of this layer's representation. Incoming messages from are accumulated and passed through an element-wise activa function $\delta(.)$, such as ReLU. $\mathcal{M}_i$ denotes the set of incoming sages for document node v_{d_i}. $f_m(.,.)$ is typically chosen to (message-specific) neural network-like function or simply a li transformation $f_m(\mathbf{g}_{d_i}^{(l)}, \mathbf{g}_{c_j}^{(l)}) = \mathbf{W}_d \mathbf{g}_{c_j}$ with a weight matrix α is a reward factor. The neighboring biomedical concepts article may differ in their importance. Therefore, it is reasonable to assign different weights to those medical concepts in sending message to their central document node. For those neighboring concept nodes which do not match any related queries, we set $\alpha = 1$. For those neighboring concept nodes which match related queries, we set $\alpha > 1$.

At each layer of the GCN, each concept node get information from its neighboring concept nodes, i.e.,

$$\mathbf{g}_{c_i}^{(l+1)} = \delta\left(\sum_{n\in\mathcal{N}_i} f_n(\mathbf{g}_{c_i}^{(l)}, \mathbf{g}_{c_j}^{(l)})\right) \tag{3}$$

where $\mathbf{g}_{c_i}^{(l)} \in \mathbb{R}^{d^{(l)}}$ is the hidden state of concept node v_{c_i} in the l-th layer of the neural network, with $d^{(l)}$ being the dimensionality of this layer's representation. Incoming messages of the form f_n are accumulated and passed through an element-wise activation function $\delta(.)$, such as ReLU. $\mathcal{N}_i$ denotes the set of incoming messages for concept node v_{c_i}. $f_n(.,.)$ is typically chosen to be a (message-specific) neural network-like function or simply a linear transformation $f_n(\mathbf{g}_{c_i}^{(l)}, \mathbf{g}_{c_j}^{(l)}) = \mathbf{W}_c \mathbf{g}_{c_j}$ with a weight matrix $\mathbf{W}_c$.

In each layer of the GCN, information is only propagated through neighboring nodes that are directly connected. Thus we can stack more GCN layers to get a larger node receptive field, i.e. each node can get information from more distant neighbors through transitivity on the graph. After L layers, for each document node v_{d_i} we obtain its structural representation $\text{GCN}(d_i) = \mathbf{g}_{d_i}^{(L)}$. Such representation will be combined with its corresponding textual representation $\text{Enc}(d_i)$ to form the final representation of each document, i.e.

$$\mathbf{v}_{d_i} = f_l(\text{Enc}(d_i) \circ \text{GCN}(d_i)) \tag{4}$$

where f_l is a linear transformation and $\circ$ is the concatenation operator. In this way the representation captured both the local textual information (induced from the encoder) and the global topic information (learned from the GCN) of each document.

3.2.3 Decoder. To make sure document representations can encode relevant local and non-local concepts, we design a decoder to see if key relevant information can be reproduced. The decoder takes as input the representation of each document $\mathbf{v}_{d_i}$ and generates a title for the document. In particular, given the document representation

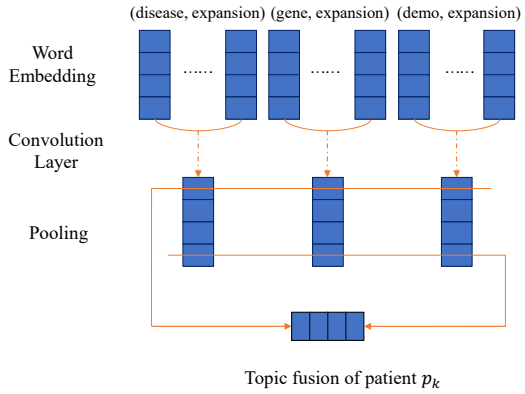

Figure 4: Query representation learning model based on CNN.

$\mathbf{v}_{d_i}$, we apply a recurrent neural network to generate a word or a medical concept once a time from a controlled vocabulary including words and medical entities. The decoder is defined as

$$w_{1:L}^{(i)} = \text{RNN}(\mathbf{h}_{1:L}^{(i)}, \mathbf{v}_{d_i}, \Theta_{\text{dec}}) \tag{5}$$

$w_{1:L}^{(i)}$ represents the generated word/entity sequence, L is the set max length of generated sequence. The process would stop early when 'EOS' (end-of-sequence) is generated, $\mathbf{h}_{1:L}^{(i)} = [\mathbf{h}_1^{(i)}, \mathbf{h}_2^{(i)}, ..., \mathbf{h}_L^{(i)}]$ denotes the hidden states. $\mathbf{v}_{d_i}$ serves as the initialization for each state, Θ_{dec} denotes parameters of the decoder. We implement the RNN as a bi-directional LSTM to generate word/entity sequence.

In this way, we can construct the loss to optimize over the model parameters using the sequence cross-entropy loss shown as follows

$$\mathcal{L}_{\text{graph}} = -\sum_{i=1}^{I}\sum_{l=1}^{L} p(w_l|d_i, w_{<l}; \Theta) \tag{6}$$

where I is the number of documents, L is the max length of generated sequence of word/entity, Θ represents all parameters involved in encoder, graph module, and decoder.

4 LEARNING TO RANK DOCUMENTS

In this section, we propose a learning-to-rank model to identify relevant biomedical articles for a given query (e.g., patient with genetic information in terms of treatment, prevention, and prognosis of the disease) with the document representation vector learned. First, we discuss how to effectively represent a query.

4.1 Representation learning for query

Each query contains information including the patient's disease name, the relevant genetic variants (which genes), basic demographic information (age, sex), etc. It is very common that the same disease and variant have different expressions. For example 'leucocythemia' and 'leukemia' are the same disease but expressed in different ways. Disease names are frequently expressed using abbreviations in literature. Different institutions may name the same genetic variants in different ways. All these aspects make it challenging to match the query to relevant biomedical articles.

Therefore, it is necessary to expand queries to incorporate these different expressions.

To get the flexible representation of the query, we propose to utilize a variety of knowledge bases in Table 2 to carefully expand gene, variant, and disease terms.

Table 2: Knowledge bases used for query expansion

Expansion	Knowledge bases
gene name expansion	NCBI GeneDB, HGNC, COSMIC, Entrez Gene Library, NCBI Homo Sapiens, PMDG
disease expansion	NCI thesaurus, MeSH hierarchy, SNOMED/Lexigram, SNOMED CT
variant expansion	COSMIC

Since the query is given as structured patient information including disease, variant, and demographic, it is crucial to combine them together and represent as a vector. We exploit convolutional neural network (CNN) to convolve structured items and obtain the vector representation of the query. Note that this part shares the same pre-trained word embedding vocabulary as is used in the document representation learning part. The CNN model is shown in Figure 4.

For the query of MeSH terms, we expand each term via knowledge bases in Table 2 and apply CNN to convolve each Mesh term and its expansion so as to share the same CNN model with clinical topics.

4.2 Learning to rank biomedical articles

For each (p_k, D) pair, where D is a ranking list $(d^1, d^2, d^3, ...)$ of biomedical articles with the superscript indicating the ranking order of the document and p_k is the k-th query, we define a relevant scoring function $f(p_k, d^i)$ as

$$f(p_k, d_i) = \left\| \mathbf{W} \times \mathbf{v}_{p_k} - \mathbf{v}_{d_i} \right\|_1 \tag{7}$$

where we use an ℓ_1 norm in the latent space, but other metrics could be used as well, $\mathbf{v}_{d_i}$ is the vector representation obtained from document representation learning and $\mathbf{v}_{p_k}$ is the vector representation obtained from patient representation learning, $\mathbf{W}$ is linear transformation matrix. The scoring function evaluates the relevance between p_k and d^i, smaller value indicates higher relevance between p_k and d^i . Therefore, we can generate the following pairs of partial order from the sample (p_k, D):

$$\begin{aligned} &f(p_k, d^1) < f(p_k, d^2); \\ &f(p_k, d^1) < f(p_k, d^3); \\ &f(p_k, d^2) < f(p_k, d^3); \\ &..... \end{aligned} \tag{8}$$

With these pairs, we can define the loss of pairwise learning to rank as follows

$$\mathcal{L}_{\text{rank}} = \sum_{k=1}^{K} \sum_{i=1}^{I} \sum_{j>i}^{J} max(0, f(p_k, d^i_{p_k}) - f(p_k, d^j_{p_k})) \tag{9}$$

This is a margin loss to make sure the score of $d^i_{p_k}$ is smaller than $d^j_{p_k}$ when $j > i$, where $d^i_{p_k}$ denotes the document is ranked as i-th for the query p_k. Our object is to minimize total loss on all training data. For a new query, the biomedical articles would be ranked according to the value of scoring function. The ranking model is trained to give more relevant documents a smaller score by tuning its parameters to minimize the pairwise maximum margin loss.

Therefore, the total loss of the entire ranking framework GRAPHENE is defined as

$$\mathcal{L} = \mathcal{L}_{\text{graph}} + \mathcal{L}_{\text{rank}} \tag{10}$$

The Adagrad stochastic gradient descent method is used to train the model with mini-batch mode.

5 EXPERIMENTS

This section describes the details of our empirical study on evaluating our proposed framework, including the ranking benchmark dataset, the experimental settings, and results.

5.1 Data set

We use the dataset from TREC Precision Medicine track (http://www.trec-cds.org/, 2017 and 2018) clinical scenarios and medical articles for empirical evaluation. We also exploited the MeSH[2] terms of each biomedical article in the first dataset as query and take the corresponding article to be the unique relevant document to be retrieved. The TREC dataset comprises 80 clinical scenarios (called topics) with structured patient scenarios including information related to disease, genetic variants, demographics, and other relevant factors. The dataset was curated by precision oncologists from MD Anderson. Therefore, it included relevant information about cancer patients such that participant systems can retrieve pertinent biomedical articles. The biomedical article corpus is composed of approximately 26.8 million MEDLINE abstracts with titles and associated MeSH terms. It is supplemented with two additional sets of abstracts: (i) 37,007 abstracts from recent proceedings of the American Society of Clinical Oncology (ASCO), and (ii) 33,018 abstracts from recent proceedings of the American Association for Cancer Research (AACR). These additional datasets were added to increase the set of potentially relevant treatment information. One reason for this is because the fact that the latest research is often presented at conferences such as ASCO and AACR prior to submission to journals (thus these proceedings may represent a more up-to-date snapshot of scientific knowledge than MEDLINE).

For each clinical topic, there is a list of relevant biomedical articles with human-labeled relevance scores. Therefore, it is very convenient to generate a ranking list of articles for each topic. We take 50 clinical topics of TREC PM track 2018 as the training set and the left 30 clinical topics of TREC PM track 2017 as the testing set. Due to the limited number of clinical topics for training a deep neural IR model, we use the MeSH terms of each biomedical article as the corresponding query to pre-train deep neural models. In the pre-training stage, for each unique ⟨MeSH term, article⟩ pair, we randomly sample a different article to generate a controlled pair ⟨MeSH terms, unrelated article⟩.

[2]MeSH (Medical Subject Headings, https://www.nlm.nih.gov/mesh/meshhome.html) is the National Library of Medicine's controlled vocabulary thesaurus. Each biomedical article is associated with a set of MeSH terms describing the content of the citation.

For each ⟨MeSH term, article⟩ pair, we can take the MeSH terms as query and search the corresponding article. Since the Mesh terms and article are one-one co-related. We can take the MeSH terms as another query set to conduct biomedical literature retrieval task. We split all pairs of ⟨MeSH, article⟩ and corresponding ⟨MeSH term, unrelated article⟩ into a training/validation/test set randomly, with the ratio of 8:1:1. The first part is for the training model, the second for hyper-parameter tuning, and the third for evaluation. Since there is only one matched article for a particular MeSH term query in most cases. Therefore, it is more reasonable to use Prec.1 and MRR instead of Prec.10 and NDCG.20 (which are explained in detail below) as the metrics to evaluate the performance on the query set of MeSH terms.

5.2 Evaluation Metrics and Settings

We introduce a set of metrics for evaluating the retrieval performance of the algorithms. All metrics have values in the range of [0, 1], with higher values for better rankings. When generating the ranked list, for documents with the same relevance scores, they will be ranked according to their IDs.

5.2.1 NDCG. Discounted Cumulative Gain (DCG) [18] is a relevance and rank correlation metric that penalizes placement of relevant documents at lower ranks. The traditional formula of DCG accumulated at a particular rank position n is defined as:

$$DCG(n) = \sum_{i=1}^{n} \frac{rel_i}{\log_2(i+1)} \tag{11}$$

Where rel_i is the graded relevance of the result at position i. Normalized Discounted Cumulative Gain (NDCG) then measures the relative DCG of a ranking compared to the best possible ranking for that data: $NDCG(n) = DCG(n)/IDCG(n)$, where $IDCG(n)$ is the $DCG(n)$ for the ideal ranking. When there are multiple queries, NDCG refers to the mean value across queries. We use the scaled relevance levels, and quote "NDCG.20" metrics for $n = 20$.

5.2.2 Precision at Rank, MRR and MAP. Average Precision measures, for a single query, is the weighted sum of the precision observed in a ranked list up to each specific rank weighted by the actual relevance score of the corresponding document, averaged over the number of relevant documents for that query. It is thus a ranking measure that factors out the size of the ranked list and the number of relevant documents, without any rank-based penalization or discounting. The Mean Average Precision (MAP) is the mean of the Average Precision across queries in our test dataset. The Precision at rank n metrics ("Prec.n") is the retrieval precision at rank n. The mean reciprocal rank (MRR) is the multiplicative inverse of the rank of the first correct answer.

5.2.3 Implementation Details. In the experiments, we evaluate the performance of GRAPHENE in two query sets for retrieving biomedical articles, including (1) clinical topics, and (2) MeSH terms. Our RNN network is a 3-layer BiLSTM with pre-trained word embeddings. Hyperparameters of our 2-layer GCN are set by the same values reported in Kipf and Welling [20]. We follow the training procedure outlined in Section 4.2 with the word embeddings setup in Section 5.4. The best setting of the reward factor α for those matched concept nodes is 1.6. We use a dropout rate of 0.5 and train the model with Adagrad SGD with the initial learning rate of 0.001 and momentum of 0.9 for 50 epochs. For baseline deep neural models, we use the same setting reported in their corresponding papers.

5.3 Baselines

We compared the performance of GRAPHENE with the following IR models as baselines.

BM25 [28] is a bag-of-words retrieval model that ranks a set of documents based on the appearance of the query terms in each document, regardless of the intra-relationship between the query terms within a document (e.g., their relative proximity).

DLH13 (DFR) [1] is a probabilistic retrieval model based on vector matching. However, they replace the term frequencies in the bag-of-words model with the probabilities inferred from a fitted Poisson model.

DeepMatch [22] is a deep learning architecture aiming at capturing the complicated matching relations between two objects from heterogeneous domains more effectively. DeepMatch is an interaction-focused model. It directly modeled the object-object interactions with a CNN-based architecture. In particular, it convolves on object-object interaction matrix and predicts if two objects are related.

Delta [25] constructs a "modified" document matrix first by replacing the words in the documents by the closest words in the query. Convolutions will be performed on this matrix to obtain a final relevance score.

5.4 Pre-trained word embeddings

We initialized the word embedding matrix with three types of pre-trained word embeddings respectively. The first is Word2Vec 100 dimensional embeddings trained on all 27.5 million MEDLINE biomedical articles in our data. The second is GloVe 100 dimensional embeddings trained on the same entire 27.5 million MEDLINE biomedical articles. The third is the randomly initialized 100 dimensional embeddings which are uniformly sampled from range $[-\sqrt{\frac{3}{dim}}, +\sqrt{\frac{3}{dim}}]$, where dim is the dimension of embeddings [12].

5.5 Results

This section presents the performance results of different retrieval models over the two benchmark query sets. The overall summary of the results is displayed in Table 3.

From the table we can observe that

- The deep neural IR models, Delta and DeepMatch, perform significantly better than the traditional retrieval models. This verifies the advantage of these deep neural models for retrieving relevant documents compared to retrieval models based on simple matching with one-hot representations.
- The TREC PM track models are the top two results reported at TREC precision track 2017. Since these two models utilize many external knowledge bases, rules, and ensemble strategies, these top two TREC PM track models produce very good results.
- Our GRAPHENE system achieves the best performance.

Table 3: Comparison of different retrieval models over the query sets MeSH terms and clinical topics. The value with * is infNDCG value.

Model Type	Model Name	MeSH terms			Clinical topics		
		Prec.1	MRR	MAP	NDCG.20	MAP	Prec.10
Traditional Retrieval Models	BM25	0.0927	0.2364	0.2359	0.2832	0.0908	0.2967
	DLH13	0.1374	0.2873	0.2856	0.4726	0.1280	0.4800
Deep Neural IR Models	DeepMatch	0.2111	0.3604	0.3589	0.6081	0.4375	0.6132
	Delta	0.3136	0.3897	0.3842	0.6796	0.4856	0.6831
TREC PM track	UD_GU_BioTM	-	-	-	0.4135*	-	0.6400
	UTD HLTRI	-	-	-	0.4593*	-	0.6172
Our Approach	GRAPHENE(-graph)	0.2745	0.3712	0.3674	0.6138	0.4693	0.6342
	GRAPHENE(-reward)	0.3354	0.4269	0.4243	0.6736	0.5732	0.6837
	GRAPHENE	0.3356	0.4296	0.4276	0.6907	0.5967	0.7024

In order to test the necessity of the different components in GRAPHENE, we also implemented two variants. GRAPHENE(-graph) excludes the graph module in document representation learning part. GRAPHENE(-reward) ignores the reward part when the passing message from concept nodes to document nodes by assigning equal weights on all messages. From the results we can see that both variants perform worse than the original GRAPHENE, which indicates that all components in GRAPHENE are important and necessary.

Moreover, we can also observe that GRAPHENE performs the best on the query set of MeSH terms, which is a one-one matching problem, which means that given a set of MeSH terms only one biomedical article should be matched in most cases. This also demonstrates the potential of our model in precise searching problems such as question answering, email search, citation recommendation, etc.

Table 4: Performance with different choices of pre-trained word embeddings on deep neural IR models

Word Embedding	Model	MAP	
		MeSH terms	Clinical topics
Random	DeepMatch	0.1826	0.2723
	Delta	0.3276	0.4054
	GRAPHENE	0.3629	0.4362
Word2Vec	DeepMatch	0.33326	0.4155
	Delta	0.3521	0.4645
	GRAPHENE	0.4131	0.5903
Glove	DeepMatch	0.3589	0.4375
	Delta	0.3842	0.4856
	GRAPHENE	0.4276	0.5967

We have also tested the effects of initialization with different strategies for pre-training the word embeddings described in Section 5.4. The results are shown in Table 4. From the table we can observe that

- Models using pre-trained word embeddings achieve a significant improvement as opposed to the ones using random embeddings.
- Models using GloVe embeddings outperforms using Word2Vec consistently for different neural models in different data sets.
- DeepMatch and Delta rely more heavily on pre-trained word embeddings compared to our proposed GRAPHENE.

The reason why pre-trained word embeddings affect less on our GRAPHENE is probably due to the encoder-decoder part of GRAPHENE to generate title can provide training of more customized word embeddings on large-scale abstract-to-title samples.

Moreover, we also investigated the importance of pre-training via the query set of MeSH terms for biomedical literature retrieval. We have performed experiments without pre-training on 'MeSH terms' and directly trained the model on 50 clinical topics. According to the results in Table 5, deep neural IR models without pre-training using MeSH terms query set drop sharply on the performance of biomedical literature retrieval. Compared to DeepMatch, the performance of Delta and our proposed GRAPHENE have a greater dependency on the pre-training with large-scale data.

Table 5: Performance without pre-train via MeSH terms on deep neural IR models

Model	NDCG.20	MAP	Prec.10
DeepMatch	0.1767	0.1265	0.1923
Delta	0.1134	0.0843	0.1243
GRAPHENE	0.1221	0.0752	0.1385

Figure 5 shows the overall scores of our model for biomedical literature retrieval across all 30 clinical topics as compared to Delta and DeepMatch. From the figures we can see that GRAPHENE performs consistently better than the baseline models for nearly all the topics across all evaluation measures on the query set of clinical topics. Delta outperforms GRAPHENE in 4 clinical topics and DeepMatch outperforms Delta in 3 clinical topics. Here, it is worth to mention that the first and second sub-figures in Figure 5 show the evaluation on top n documents and the third sub-figure in Figure 5 shows the evaluation on all related documents, and the evaluation on top n results is much more stable than the evaluation on all related documents.

The last question we investigated is whether the size of pre-training query set of MeSH terms would affect the final performance. As shown in Figure 6, we check the performance of deep neural

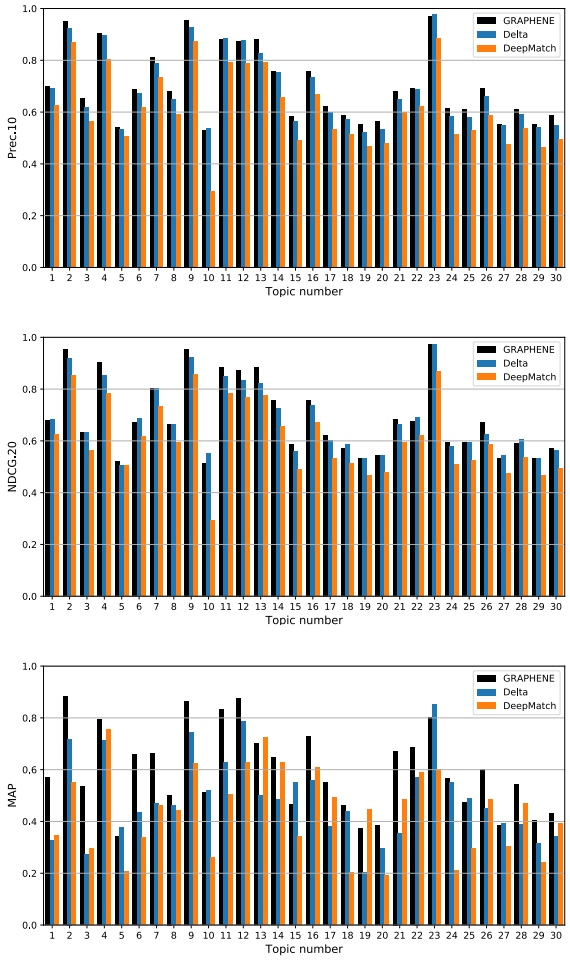

Figure 5: Prec.10 scores, NDCG.20 scores and MAP scores for each clinical topic.

IR models on biomedical literature retrieval of clinical topics by incrementally augmenting pre-training data. The results indicate clearly that the size of pre-training data plays a key role in all three deep neural IR models, and it is more critical to GRAPHENE and Delta, both of which are representation-focused models rely more heavily on large scale of pre-training data. This is consistent with the results of previous work [34].

Case Study: Last but not the least, we performed a case study to better understand the power of GRAPHENE. Table 6 shows an example of retrieved relevant documents that are placed at rank 1 by GRAPHENE, DeepMatch and Delta, with respect to a specific query. We highlight the biomedical concepts which are contained in query in blue bold text and the relevant medical concepts with entity in query blue italic text. From the results we can see that in the document retrieved by GRAPHENE, there are matched disease name "pancreatic cancer", matched gene variant "CDK6" and the relevant medical concept "arsenic trioxide" (treatment of the pancreatic cancer), while only the disease name "pancreatic cancer"

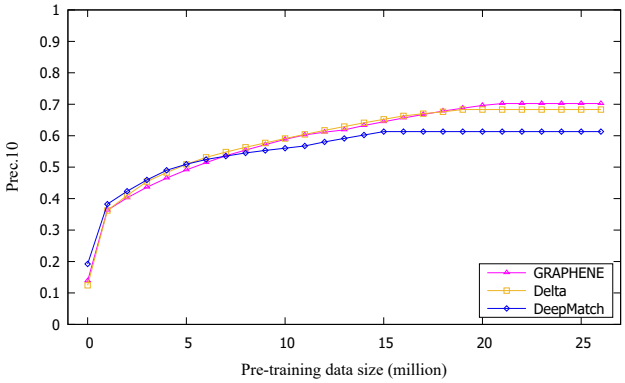

Figure 6: Comparison of deep neural IR models by incrementally augmenting pre-training query set of MeSH terms.

appears in the most relevant document identified by Delta and DeepMatch. This demonstrates the strong potential of GRAPHENE on identification of semantic similar items despite the different exact term expressions.

6 CONCLUSION

In this paper, we proposed a deep neural biomedical literature retrieval framework GRAPHENE which consists of graph-augmented document representation learning, query expansion, and representation learning, and learning to rank biomedical articles. The graph-augmented document representation learning is applied upon a document-concept graph which contains biomedical concept nodes and document nodes so that global biomedical concept co-occurrence can be explicitly modeled and graph convolution can be easily adapted. Query representation learning exploits a CNN-based model to convolve structured items of patients and output the vector representation for each query. A learning-to-rank algorithm is proposed to use partial order between biomedical articles with the given patient and learn the rank of relevant articles. Experimental results on TREC Precision Medicine track data provided compelling evidence to support that our model can effectively retrieve most relevant biomedical articles for a given query. The results also demonstrated that our GRAPHENE improves previous deep neural retrieval models which represent the document with textual information only, suggesting the necessity of encoding graph-based external knowledge bases into document representation.

Since the great success of pre-trained BERT model in modeling long text for many NLP tasks recently [2, 5], we want to integrate it into our present GRAPHENE to replace the RNN-based encoder-decoder module in our future work. Additionally, we also want to leverage advantages from both representation-focused models and interaction-focused models for our future work as they have respective advantages.

7 ACKNOWLEDGMENTS

This work is supported by NSF 1716432 and 1750326.

Table 6: Examples of matched documents ranked as top 1 by different deep neural IR models. Bold blue text is the medical concepts which appear in the query, italic blue text is the related medical concept.

Query	Model	Document
Pancreatic cancer; CDK6 Amplification; 48-year-old male	GRAPHENE	We have previously shown that *arsenic trioxide* blocks proliferation and induces apoptosis in human **pancreatic cancer** cells at low, ... expression of CDK2, CDK4, **CDK6**, and cyclin E were not affected ..... In summary, *arsenic trioxide* induced apoptosis in **pancreatic cancer** cells through activating the caspase cascade via the mitochondrial pathway ..... This old drug may be valuable for treatment of pancreatic cancer.
	Delta	..... An association has been reported between p16 mutations and **pancreatic cancer** ..... The second most frequent cancer was **pancreatic cancer** ..... of **pancreatic cancer** was ..... The estimated cumulative risk of developing **pancreatic cancer** in putative mutation ..... no cases of **pancreatic cancer** occurred. p16 mutation carriers have a considerable risk of developing **pancreatic cancer** .....
	DeepMatch	..... An association has been reported between p16 mutations and **pancreatic cancer** ..... The second most frequent cancer was **pancreatic cancer** ..... of **pancreatic cancer** was ..... The estimated cumulative risk of developing **pancreatic cancer** in putative mutation ..... no cases of **pancreatic cancer** occurred. p16 mutation carriers have a considerable risk of developing **pancreatic cancer** .....

REFERENCES

[1] Giambattista Amati. 2006. Frequentist and bayesian approach to information retrieval. In *Proceedings of the ECIR*. 13–24.

[2] Iz Beltagy, Arman Cohan, and Kyle Lo. 2019. SciBERT: Pretrained Contextualized Embeddings for Scientific Text. *arXiv preprint arXiv:1903.10676* (2019).

[3] Francis S Collins and Harold Varmus. 2015. A new initiative on precision medicine. *New England journal of medicine* 372, 9 (2015), 793–795.

[4] Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. Convolutional Neural Networks for Soft-Matching N-Grams in Ad-hoc Search. In *Proceedings of the WSDM*. 126–134.

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

[6] Nicolas Fiorini, Kathi Canese, Grisha Starchenko, Evgeny Kireev, Won Kim, Vadim Miller, Maxim Osipov, Michael Kholodov, Rafis Ismagilov, Sunil Mohan, et al. 2018. Best Match: new relevance search for PubMed. *PLoS biology* 16, 8 (2018), e2005343.

[7] Nicolas Fiorini, Robert Leaman, David J Lipman, and Zhiyong Lu. 2018. How user intelligence is improving PubMed. *Nature biotechnology* 36, 10 (2018), 937.

[8] Nicolas Fiorini, David J Lipman, and Zhiyong Lu. 2017. Cutting edge: towards PubMed 2.0. *Elife* 6 (2017), e28801.

[9] Jianfeng Gao, Patrick Pantel, Michael Gamon, Xiaodong He, and Li Deng. 2014. Modeling interestingness with deep neural networks. In *Proceedings of the EMNLP*. 2–13.

[10] Travis R Goodwin and Sanda M Harabagiu. 2016. Medical question answering for clinical decision support. In *CIKM*. 297–306.

[11] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the CIKM*. 55–64.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *ICCV*. IEEE Computer Society, Washington, DC, USA, 1026–1034.

[13] William Hersh, Susan Price, and Larry Donohoe. 2000. Assessing thesaurus-based query expansion using the UMLS Metathesaurus.. In *AMIA*. American Medical Informatics Association, 344.

[14] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[15] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *NIPS*. 2042–2050.

[16] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the CIKM*. 2333–2338.

[17] Kai Hui, Andrew Yates, Klaus Berberich, and Gerard de Melo. 2017. PACRR: A Position-Aware Neural IR Model for Relevance Matching. In *Proceedings of the EMNLP*. Association for Computational Linguistics, Copenhagen, Denmark, 1049–1058.

[18] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems* 20, 4 (2002), 422–446.

[19] Jyun-Yu Jiang, Mingyang Zhang, Cheng Li, Mike Bendersky, Nadav Golbandi, and Marc Najork. 2019. Semantic Text Matching for Long-Form Documents. (2019).

[20] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. In *ICLR*.

[21] Zhiyong Lu, Won Kim, and W John Wilbur. 2009. Evaluation of query expansion using MeSH in PubMed. *Information retrieval* 12, 1 (2009), 69–80.

[22] Zhengdong Lu and Hang Li. 2013. A deep architecture for matching short texts. In *NIPS*. 1367–1375.

[23] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *ACL System Demonstrations*. 55–60.

[24] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*. 3111–3119.

[25] Sunil Mohan, Nicolas Fiorini, Sun Kim, and Zhiyong Lu. 2018. A Fast Deep Learning Model for Textual Relevance in Biomedical Information Retrieval. In *Proceedings of the WWW*. 77–86.

[26] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the EMNLP*. 1532–1543.

[27] Kirk Roberts, Dina Demner-Fushman, Ellen M Voorhees, William R Hersh, Steven Bedrick, Alexander J Lazar, and Shubham Pant. 2017. Overview of the TREC 2017 precision medicine track. *NIST Special Publication* (2017), 500–324.

[28] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at TREC-3. *Nist Special Publication Sp* 109 (1995), 109.

[29] Harrisen Scells, Guido Zuccon, Bevan Koopman, Anthony Deacon, Leif Azzopardi, and Shlomo Geva. 2017. Integrating the framing of clinical questions via PICO into the retrieval of medical literature for systematic reviews. In *Proceedings of the CIKM*. 2291–2294.

[30] Luca Soldaini, Andrew Yates, and Nazli Goharian. 2017. Denoising Clinical Notes for Medical Literature Retrieval with Convolutional Neural Model. In *CIKM*. 2307–2310.

[31] Chih-Hsuan Wei, Hung-Yu Kao, and Zhiyong Lu. 2013. PubTator: a web-based text mining tool for assisting biocuration. *Nucleic acids research* 41, W1 (2013), 518–522.

[32] Chenyan Xiong and Jamie Callan. 2015. Esdrank: Connecting query and documents through external semi-structured data. In *CIKM*. ACM, 951–960.

[33] Chenyan Xiong and Jamie Callan. 2015. Query Expansion with Freebase. In *Proceedings of the 2015 International Conference on The Theory of Information Retrieval*. 111–120.

[34] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of the SIGIR*. ACM, 55–64.

[35] Chenyan Xiong, Russell Power, and Jamie Callan. 2017. Explicit semantic ranking for academic search via knowledge graph embedding. In *WWW*. International World Wide Web Conferences Steering Committee, 1271–1279.

[36] Sendong Zhao, Meng Jiang, Ming Liu, Bing Qin, and Ting Liu. 2018. CausalTriad: Toward Pseudo Causal Relation Discovery and Hypotheses Generation from Medical Text Data. In *ACM BCB*. ACM, 184–193.

[37] Sendong Zhao, Ting Liu, Sicheng Zhao, and Fei Wang. 2019. A Neural Multi-Task Learning Framework to Jointly Model Medical Named Entity Recognition and Normalization. In *AAAI*, Vol. 33. 817–824.

[38] Sendong Zhao, Quan Wang, Sean Massung, Bing Qin, Ting Liu, Bin Wang, and ChengXiang Zhai. 2017. Constructing and embedding abstract event causality networks from text snippets. In *WSDM*. ACM, 335–344.

[39] Ziwei Zheng and Xiaojun Wan. 2016. Graph-Based Multi-Modality Learning for Clinical Decision Support. In *Proceedings of the CIKM*. 1945–1948.

[40] Yongjun Zhu, Olivier Elemento, Jyotishman Pathak, and Fei Wang. 2018. Drug knowledge bases and their applications in biomedical informatics research. *Briefings in bioinformatics* (2018).