A Hierarchical Attention Retrieval Model for Healthcare Question Answering

Ming Zhu
Dept. of Computer Science
Virginia Tech, Arlington, VA
mingzhu@vt.edu

Aman Ahuja
Dept. of Computer Science
Virginia Tech, Arlington, VA
aahuja@vt.edu

Wei Wei
Google AI
Mountain View, CA
wewei@google.com

Chandan K. Reddy
Dept. of Computer Science
Virginia Tech, Arlington, VA
reddy@cs.vt.edu

ABSTRACT

The growth of the Web in recent years has resulted in the development of various online platforms that provide healthcare information services. These platforms contain an enormous amount of information, which could be beneficial for a large number of people. However, navigating through such knowledgebases to answer specific queries of healthcare consumers is a challenging task. A majority of such queries might be non-factoid in nature, and hence, traditional keyword-based retrieval models do not work well for such cases. Furthermore, in many scenarios, it might be desirable to get a short answer that sufficiently answers the query, instead of a long document with only a small amount of useful information. In this paper, we propose a neural network model for ranking documents for question answering in the healthcare domain. The proposed model uses a deep attention mechanism at word, sentence, and document levels, for efficient retrieval for both factoid and non-factoid queries, on documents of varied lengths. Specifically, the word-level cross-attention allows the model to identify words that might be most relevant for a query, and the hierarchical attention at sentence and document levels allows it to do effective retrieval on both long and short documents. We also construct a new large-scale healthcare question-answering dataset, which we use to evaluate our model. Experimental evaluation results against several state-of-the-art baselines show that our model outperforms the existing retrieval techniques.

CCS CONCEPTS

• **Information systems** → **Language models**; **Learning to rank**; **Question answering**.

KEYWORDS

Neural Networks, Information Retrieval, Consumer Healthcare, Question Answering

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313699

ACM Reference Format:
Ming Zhu, Aman Ahuja, Wei Wei, and Chandan K. Reddy. 2019. A Hierarchical Attention Retrieval Model for Healthcare Question Answering. In *Proceedings of the 2019 World Wide Web Conference (WWW'19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3308558.3313699

1 INTRODUCTION

With the growth of the Web in recent years, a vast amount of health-related information is now publicly available on the Internet. Many people use online health information platforms such as WebMD[1] and Patient[2] to search for information regarding the symptoms, diseases, or any other health-related information they are interested in. In addition to consumers, often doctors and healthcare professionals need to look into knowledgebases that contain detailed healthcare information about diseases, diagnosis, and procedures [8, 31]. Despite the abundance of available information, it might be difficult for healthcare consumers to navigate through these documents to get the required healthcare information. Hence, effective retrieval techniques are required to allow consumers to efficiently use such platforms. Since healthcare documents usually include several details about the disease such as it's symptoms, preventive measures, and common treatments, they are usually more elaborate, compared to other factual documents, which describe well-known facts (e.g., population of a town, capital of a city, or any other entity), and are very specific in nature. Hence, in such cases, it might be desirable to provide the consumers with a short piece of text that succinctly answers their queries. Furthermore, many questions that users have about health-related topics are very abstract and open-ended in nature, and hence traditional search methods do not work well in such cases.

Prompted by the success of deep neural networks in language modeling, researchers have proposed several techniques that apply neural networks for effective information retrieval [9, 21] and question answering [33, 38]. This has been facilitated primarily due to the development of large training datasets such as TREC [35] and SQuAD [25]. However, both these datasets are primarily composed of factoid queries / questions, and the answers are generally short in length. Hence, systems trained on such datasets cannot perform

[1] https://www.webmd.com/
[2] https://patient.info/

What would happen if I didn't take antithyroid medicines?
It is usually advisable to treat an overactive thyroid gland (hyperthyroidism). Untreated hyperthyroidism can cause significant problems with your heart and other organs. It may also increase your risk of complications should you become pregnant. However, in many cases there are other treatment options. That is, radioactive iodine or surgery may be suitable options. See the separate leaflet called Overactive Thyroid Gland (Hyperthyroidism) for details of these other treatment options .

Figure 1: An example of a healthcare question, and it's corresponding answer. The question and answer do not have any overlapping words. The highlighted text corresponds to the most relevant answer snippet from the document.

well in a setting where a large proportion of the queries are non-factoid and open-ended, and the documents are relatively longer in length. Figure 1 shows an example of a typical question that a consumer would have regarding antithyroid medicines, and it's corresponding answer paragraph, selected from the website Patient. This problem and the domain provides some unique challenges which require us to build a more comprehensive retrieval system.

- **Minimal overlap between question and answer words:** There is minimal or no word overlap between the question and answer text. As there are no matching terms, traditional keyword-based search mechanisms will not work for answering such questions.
- **Length of question and answer:** The question is longer than a typical search engine query. The answer is also typically longer than a sentence. Although, for illustration purposes, we show a short paragraph, in many cases, the answer, as well as the document containing it, might be even longer. Hence, neural semantic matching algorithms will not be effective in such cases, as they are ideally designed for short sentences. Therefore, an effective retrieval system would require a mechanism to deal with documents of varied lengths.
- **Non-factoid nature:** The question is very open-ended in nature, and does not ask for any specific factual details. As such, a majority of the machine comprehension models are trained on datasets like SQuAD, which are comprised of factoid QA pairs. Such systems do not work well in a setting where the desired answer is more elaborate.

To overcome these problems, we propose **HAR**, a **H**ierarchical **A**ttention **R**etrieval model for retrieving documents for healthcare related queries. The proposed model uses a cross-attention mechanism between the query and document words to discover the most important words that are required to sufficiently answer the query. It then uses a hierarchical inner attention, first over different words in a sentence, and then over different sentences in a document, to successively select the document features that might be most relevant for answering the query. Finally, it computes a similarity score of a document with the query, that could be used to rank different documents in the corpus, given a query. The use of hierarchical attention also enables it to find the most important sentences and words, that could be important to answer a query, without the need of using an explicit machine comprehension module. To evaluate the performance of our model, we construct a large scale healthcare question answering dataset, using knowledge articles collected from the popular health services website Patient. Although we use this model in the healthcare domain, where the questions are usually non-factoid in nature, and the documents are longer due to the presence of detailed description about different medical procedures, our model is more generic, and can be used in any domain where the questions are open-ended, and the documents are longer.

The rest of this paper is organized as follows: Section 2 gives an overview of the existing techniques related to our work. In Section 3, we describe our proposed neural retrieval model called HAR, and provide the details about its architecture and the training procedure, including the optimization for the HAR model. The details about the data collection and annotation have been described in Section 4. In Section 5, we give details about our experimental evaluation, and the metrics and baseline techniques used in the evaluation process. Finally, Section 6 concludes the paper, with possible directions for future research.

2 RELATED WORK

2.1 Document Ranking

Document retrieval and ranking is a classical problem in the information retrieval community, which has attracted significant interest from researchers for many years. Early methods in informational retrieval were largely based on keyword-based query-document matching [27, 29, 30]. With the advancement of machine learning algorithms, better retrieval mechanisms have been proposed. Logistic Inference [7] used logistic regression probabilities to determine the relevance between queries and documents. In [14], the authors used Support Vector Machine (SVM) based approach for retrieval, which allows the retrieval system to be trained using the search engine click logs. Other traditional techniques in information retrieval include boosting-based methods [6, 41]. TF-IDF based similarity [26] and Okapi BM25 [28] are the most popularly used term-based techniques for document search and ranking. However, such techniques usually do not perform well, when the documents are longer [18], or have minimal exact word overlap with the query.

2.2 Neural Information Retrieval

With the success of deep neural networks in learning feature representation of text data, several neural ranking architectures have been proposed for text document search. Deep Structured Semantic Model (DSSM) [13] uses a simple feed-forward network to learn the semantic representation of queries and documents. It then computes the similarity between their semantic representations using cosine similarity. Convolutional Deep Structured Semantic Model (CDSSM) [34] uses convolutional layers on word trigram features, while the model proposed in [22] uses the last state outputs of LSTM encoders as the query and document features. Both these models then use cosine similarity between query and document representations, to compute their relevance. In [12], the authors

propose convolutional neural network models for semantic matching of documents. The Architecture-I (ARC-I) model proposed in this work also uses a convolutional architecture to create document-level representation of query and document, and then uses a feed-forward network to compute their relevance. The InferSent Ranker [11] proposed recently also uses a feed forward network to compute the relevance between query and documents, by summing up their sentence embeddings. However, all these methods use the document-level semantic representation of queries and documents, which is basically a pooled representation of the words in the document. However, in majority of the cases in document retrieval, it is observed that the relevant text for a query is very short piece of text from the document. Hence, matching the pooled representation of the entire document with that of the query does not give very good results, as the representation also contains features from other irrelevant parts of the document.

To overcome the problems of document-level semantic-matching based IR models, several interaction-based IR models have been proposed recently. In [9], the authors propose Deep Relevance Matching Model (DRMM), that uses word count based interaction features between query and document words, while the Architecture-II (ARC-II) proposed in [12] uses convolution operation to compute the interaction features. These features are then fed to a deep feed-forward network for computing the relevance score. The models proposed in [4, 40] use kernel pooling on interaction features to compute similarity scores, while MatchPyramid [23] uses the dot product between query and document word vectors as their interaction features, followed by convolutional layers to compute the relevance score. Other methods that use word-level interaction features are attention-based Neural Matching Model (aNMM) [42], that uses attention over word embeddings, and [36], that uses cosine or bilinear operation over Bi-LSTM features, to compute the interaction features. The Duet model proposed in [21] combines both word-level interaction features, as well as document-level semantic features, in a deep CNN architecture, to compute the relevance. One common limitation of all these models is that they do not utilize the inherent paragraph and sentence level hierarchy in documents, and hence, they do not perform well in case of longer documents. By using a powerful cross attention mechanism between query and document words, our model can effectively determine the most relevant document words for a query. It then uses hierarchical inner attention over these features, which enables it to effectively deal with long documents. This is especially helpful in cases where the relevant information in the document is a very small piece of text.

2.3 Information Retrieval for Healthcare

Early works in the domain of information retrieval for medicine and healthcare used tradition search methods such as TF-IDF [19] and BM25 [17]. MedQA [45] uses hierarchical clustering along with TF-IDF for answering definitional questions asked by physicians. Th question-answering system proposed in [5] aimed at helping clinicians to search for treatments for any disease. However, their system is tailored to answer one specific type of questions, and cannot be used to answer open-ended questions. As discussed in [15], physicians also often need to use such systems, and they have limited time to browse through every returned document. Hence, medical retrieval needs to be accurate, and should precisely serve the requirements of the users. Retrieval in this domain is complex, attributed to the non-factoid nature of queries, and longer documents. Hence, traditional IR techniques or semantic matching algorithms do not work well on such datasets.

3 THE PROPOSED MODEL

In this section, we introduce our proposed Hierarchical Attention Retrieval (HAR) model, which uses deep attention mechanism for effective retrieval. The detailed architecture of our model is shown in Fig. 2. HAR is a novel neural network model that uses two powerful attention mechanisms to overcome the shortcomings of existing document retrieval models. Given a query q, the model computes a relevance score r_i with each candidate document d_i in the document knowledgebase $\mathcal{D}$. The different components of our model are described in detail below.

3.1 Word Embeddings

The input layer in our model is an embedding lookup function which converts the query and document words into fixed K-dimensional word vectors using a lookup matrix $\boldsymbol{E} \in \mathbb{R}^{V \times K}$ of V pre-trained word embeddings such as GloVe [24] or Word2Vec [20]. Let $\{w_t^q\}_{t=1}^m$ be the words in q. Let l be the number of sentences in document d, and $\{w_t^{id}\}_{t=1}^n$ be the words in sentence i in d. This layer converts each of the words in q and d into the word vectors $\{e_t^q\}_{t=1}^m$ and $\{e_t^{id}\}_{t=1}^n$, respectively. Here, m and n are the number of words in query and each of the document sentences, respectively.

3.2 Encoder

We use two bidirectional RNN (Bi-RNN) [32] encoders to encode the inter-document temporal dependencies within query and documents words, respectively. This layer consists of two RNN layers in different directions, whose output is concatenated to get the H-dimensional contextual representation of each word. We choose GRU [2] over vanilla-RNN or LSTM [10] because of its high performance and computational efficiency. Since we split the documents into short sentences, GRU performs equally well as LSTM, because the encoder does not need to deal with very long sequences.

3.2.1 Query Encoder. The query encoder contains a simple Bi-GRU layer, which takes the query word embeddings $\{e_t^q\}_{t=1}^m$ as the input, and outputs the contextual representation $U^q = \{u_t^q\}_{t=1}^m \in \mathbb{R}^{m \times H}$.

$$u_t^q = BiGRU_Q(u_{t-1}^q, e_t^q) \tag{1}$$

3.2.2 Document Encoder. Since documents are usually longer than queries, we encode each sentence in the document separately, using a sentence-level Bi-GRU encoder. Given a sentence i, this layer takes the sentence word embeddings $\{e_t^{id}\}_{t=1}^n$ as the input, and returns the contextual word embeddings $U^{id} = \{u_t^{id}\}_{t=1}^n \in \mathbb{R}^{n \times H}$. After encoding each of the l sentences in the document through this encoder, the new document representation is $\{U^{1d}, .., U^{ld}\} \in \mathbb{R}^{l \times n \times H}$.

$$u_t^{id} = BiGRU_D(u_{t-1}^{id}, e_t^{id}) \tag{2}$$

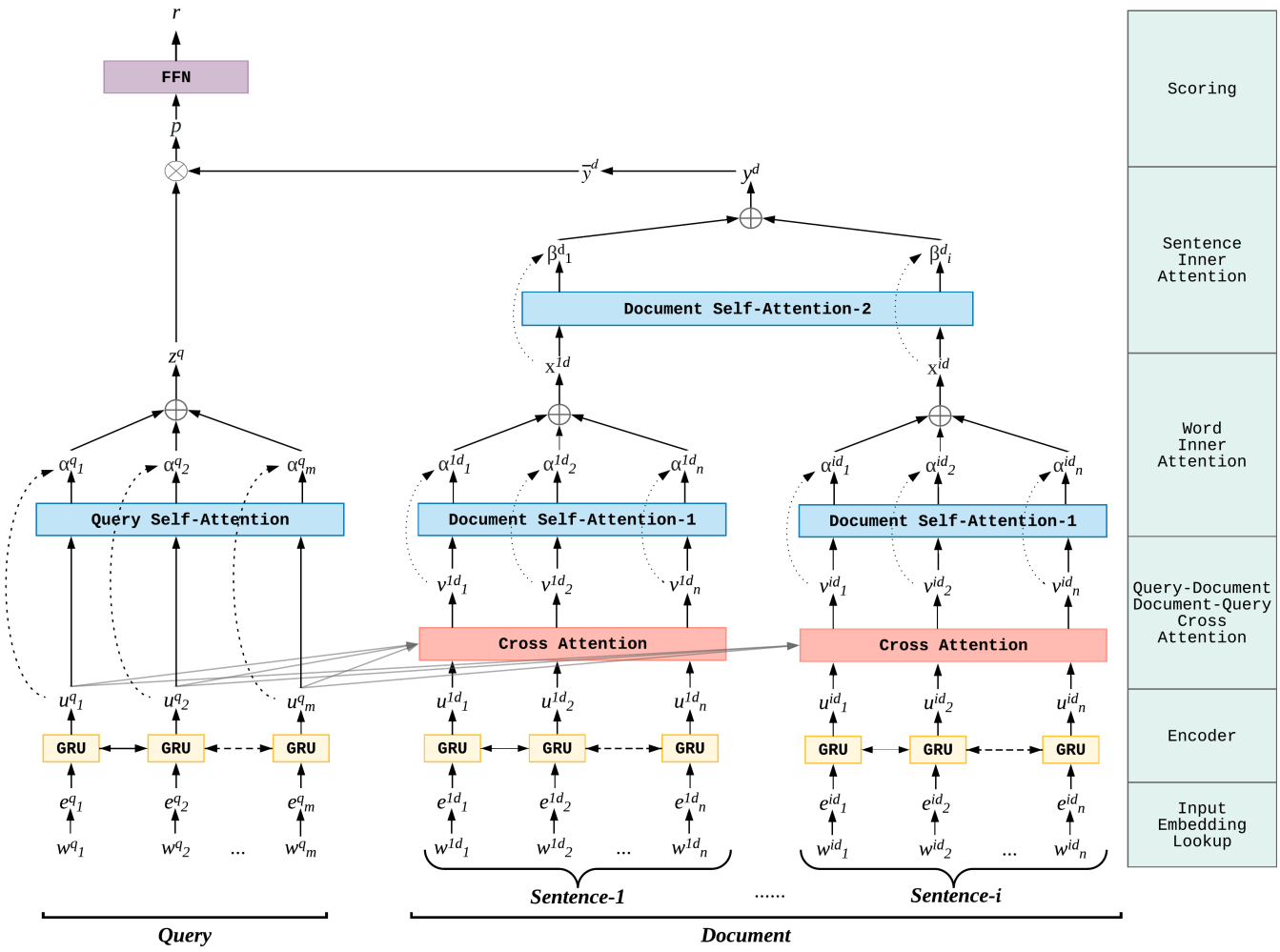

Figure 2: Architecture of the proposed HAR model.

3.3 Cross Attention between Query and Document

This layer is used to fuse the information from query words to the document words. It computes the relevance of each query word with respect to each word in the document. As we use a hierarchical modeling for documents, this layer can compute the attended embeddings of each word in sentence i in d with each word in the query q. We use the cross attention mechanism proposed in [39, 44], as this method has been proven to show superior performance in state-of-the-art reading comprehension systems. The attention layer computes the relevance between each pair of query and document words, using their contextual embeddings generated by the respective encoders. To calculate the relevance of query words with respect to document words, and vice-versa, we use bi-directional attention mechanism [33], which is composed of document-to-query attention $D2Q$ and query-to-document attention $Q2D$. This is done by first computing a similarity matrix $S \in \mathbb{R}^{n \times m}$, which is then normalized over each row and column, using a normalization operation such as softmax. This generates normalized similarity matrices $\overline{S}_{D2Q} \in \mathbb{R}^{n \times m}$ and $\overline{S}_{Q2D} \in \mathbb{R}^{n \times m}$, respectively. Finally, the attention matrices $A_{D2Q} \in \mathbb{R}^{n \times H}$ and $A_{Q2D} \in \mathbb{R}^{n \times H}$ can be computed as described below.

Let $s_{xy} \in \mathbb{R}$ be an element of the similarity matrix S from row x and column y. Given $U^{id} \in \{U^{1d}, .., U^{ld}\}$ and U^q as inputs, the final output $V^{id} = \{v_t^{id}\}_{t=1}^{n} \in \{V^{1d}, .., V^{ld}\}$ of the cross attention layer can be computed as follows:

$$s_{xy} = w_c^T \cdot [u_x^{id}; u_y^q; u_x^{id} \odot u_y^q] \tag{3}$$

$$\overline{S}_{D2Q} = \text{softmax}_{\text{row}}(S) \tag{4}$$

$$\overline{S}_{Q2D} = \text{softmax}_{\text{col}}(S) \tag{5}$$

$$A_{D2Q} = \overline{S}_{D2Q} \cdot U^q \tag{6}$$

$$A_{Q2D} = \overline{S}_{D2Q} \cdot \overline{S}_{Q2D}{}^T \cdot U^{id} \tag{7}$$

$$V^{id} = [U^{id}; A_{D2Q}; U^{id} \odot A_{D2Q}; U^{id} \odot A_{Q2D}] \in \mathbb{R}^{n \times 4H} \tag{8}$$

In the above equations, ; is the concatenation operation, $\odot$ is element-wise multiplication, $\cdot$ is matrix multiplication, and $w_c \in \mathbb{R}^{3H}$ is a trainable weight vector.

3.4 Query Inner Attention

To encode variable length queries into a fixed size embedding, we use the self attention mechanism proposed in [16]. The importance of different words varies from document to document, and is dependent on the context in which they are used. This layer allows the model to give higher priority to more important words while creating a pooled representation of the query. This ensures that the query representation contains features from more significant words. Let A be the dimension of the pooled representation. Given query features $U^q = \{u_t^q\}_{t=1}^m$ as the input, this layer generates a pooled representation of $z^q \in \mathbb{R}^H$ as follows:

$$c_t^q = w_q^T(\tanh(W_q u_t^q)) \tag{9}$$

$$\alpha_t^q = \frac{\exp(c_t^q)}{\sum_{j=1}^m \exp(c_j^q)} \tag{10}$$

$$z^q = \sum_{t=1}^m \alpha_t^q u_t^q \tag{11}$$

3.5 Document Hierarchical Inner Attention

Since documents are longer in length, it is not necessary that the entire document is relevant to a query. In fact, in most cases, it is observed that part of the document that is relevant to a query is just a few sentences. Even inside each sentence, different words might have varying relevance to the query. Furthermore, because of the varied lengths of documents, a mechanism is required to get a fixed-dimensional representation of the document. Hence, we use a two-level hierarchical inner attention (as proposed in [43]), to get the document embedding.

3.5.1 Level-1: Attention over words in a sentence. The first level in our hierarchical attention encodes each sentence independently from other sentences at word-level, resulting in a fixed-dimensional representation of each sentence. This layer computes the importance of each word within the sentence, and then creates a pooled representation of each sentence weighted by the attention weights. For each sentence i in the document d, this layer takes the output vectors $V^{id} = \{v_t^{id}\}_{t=1}^n \in \mathbb{R}^{n \times 4H}$ from the cross-attention layer as the input, and returns a sentence vector $x^{id} \in \mathbb{R}^{4H}$.

$$c_t^{id} = w_{d1}^T(\tanh(W_{d1} v_t^{id})) \tag{12}$$

$$\alpha_t^{id} = \frac{\exp(c_t^{id})}{\sum_{j=1}^n \exp(c_j^{id})} \tag{13}$$

$$x^{id} = \sum_{t=1}^n \alpha_t^{id} v_t^{id} \tag{14}$$

3.5.2 Level-2: Attention over sentences in a document. To ensure that the sentences more relevant to the query are given higher importance while computing the similarity score, we use a second inner attention, to compute the document representation. This layer takes the sentence embeddings $\{x_i^d\}_{i=1}^l$ as the input, and returns a document vector $y^d \in \mathbb{R}^{4H}$ as the output.

$$b_i^d = w_{d2}^T(\tanh(W_{d2} x^{id})) \tag{15}$$

$$\beta_i^d = \frac{\exp(b_i^d)}{\sum_{j=1}^l \exp(b_j^d)} \tag{16}$$

$$y^d = \sum_{j=1}^l \beta_j^d x^{jd} \tag{17}$$

3.6 Score Computation

The final layer in our model computes the score between the query representation z^q and document representation y^d. Since the dimension of y^d is 4 times the dimension of z^q, we first pass y^d through a feed-forward layer to compute $\overline{y}^d \in \mathbb{R}^H$. After this, we compute the similarity vector $p \in \mathbb{R}^H$ by performing element-wise multiplication of z^q and $\overline{y}^d$. Finally, we pass p through a feed-forward network to compute the final relevance score $r \in \mathbb{R}$.

$$\overline{y}^d = w_{d3}^T y^d + b_{d3} \tag{18}$$

$$p = z^q \odot \overline{y}^d \tag{19}$$

$$r = w_f^T p + b_f \tag{20}$$

3.7 Optimization

3.7.1 Negative sampling. Many retrieval datasets, such as the ones created using user click-logs, only have the query-document pairs, which serve as the positive data for the model. However, for the model to have sufficient discriminative power to give a score to every document proportional to their relevance with the query, the model also needs negative query-document pairs during the training process. Hence, we do negative sampling to generate negative data samples of query-document pairs for our model. For each query, the negative samples are composed of the following:

- *Irrelevant negative samples:* For the model to have a sufficient discriminative power that is needed to distinguish documents at a high level, we sample negative documents that have very low relevance to the query.
- *Partially relevant negative samples:* We define partially relevant negative documents as those that might have some relevance to the query, either due to some overlapping words, or because they are from the same topic, but do not contain the correct answer for the query. As suggested in [37], having such samples in the training dataset gives a higher discriminative power to the model,

as compared to a model trained with randomly sampled negative pairs.

3.7.2 Loss function. We use pairwise maximum margin loss [14] as the objective function to be minimized. Given a query q, positive document d^{pos}, and k negatively sampled documents $\{d_1^{neg}, .., d_k^{neg}\}$, the loss is given by:

$$\mathcal{L} = \sum_{i=1}^{k} max(0, M - score(q, d^{pos}) + score(q, d_i^{neg})) \quad (21)$$

Here, M is the margin, by which we want the score of positive query-document pair to exceed that of a negative query-document pair.

4 HEALTHQA DATASET

We will now introduce the dataset created in this work to train and evaluate the proposed HAR model. We call this dataset *HealthQA*. It consists of question and document pairs from the healthcare domain. The details of this dataset are described below:

4.1 Knowledge Articles

To create HealthQA dataset, we collected healthcare articles from the popular health-services website Patient. We scraped all the articles from the *Health Topics* section of Patient. The website contains articles from a diverse set of healthcare domains such as child health, mental health, sexual health, details about treatments and medications, and several other healthcare domains. The articles on this website are much more detailed, as compared to other healthcare knowledgebases like MedlinePlus[3]. In total, we collected 1,235 health articles, with each article having an average of 6 sections. As the sections themselves are very long in these articles, we use each section as one document.

Figure 3 shows the distribution of percentage of documents with respect to the length of the documents (number of words). We can see that the proportion of documents with less than 50 words is very less (5%). The dataset contains a large number of documents with 100-200 words, and a high proportion of documents containing more than 200 words.

4.2 Question-Answer Pair Generation

To create healthcare-related questions, we employed human workers from diverse age groups, and from different countries. For the dataset to have a diverse set of questions and answers that different people might have about healthcare, we hired six annotators, consisting of a combination of freelancers, graduate and undergraduate students. To ensure high quality of the dataset, and low error rates, we ensured that all the annotators had good English skills. For each document, workers were instructed to create 1 to 3 questions that can be asked using the information given in the document. They were encouraged to use simple language in the queries, so that the questions follow the style of those asked by a common person, without any domain expertise in healthcare. All the generated questions also underwent an additional round of cross validation by the authors, and any query-document pairs with errors or insufficient context were either corrected or discarded.

[3]https://medlineplus.gov/

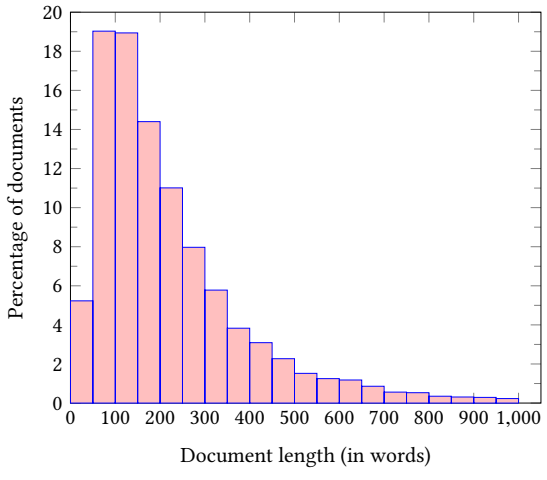

Figure 3: Percentage of documents vs. number of words.

It was found that many articles on Patient have several subtitles, roughly one subtitle per paragraph, and in most cases, these subtitles could be rephrased into valid questions. Some of the titles have incomplete context, which can be made into a valid question by rephrasing them. Hence, workers were also allowed to use the subtitles as questions, by rephrasing them into valid questions.

Figure 4 shows the percentage of different types of questions in our dataset. The questions with the type “How" and “Why" are mostly non-factoid in nature. Such questions are open-ended, and require detailed answers. Although the questions with the type “What", that are generally factoid, have a large proportion in our dataset, after manual analysis, we found that a large proportion of such questions are also non-factoid. Examples of such questions include *“What is the outlook of gaming disorder?"*.

Table 1: Statistics of the HealthQA dataset.

Number of articles	1,235
Number of documents (article sections)	7,355
Number of questions	7,517
Average length of questions (in words)	8.04
Average length of documents (in words)	233.4
Average number of sentences in documents	13.54
Average length of sentences (in words)	17.24

5 EXPERIMENTAL RESULTS

5.1 Evaluation Metrics

We compare the performance of HAR with various baseline techniques using the following evaluation metrics:

- **Recall@K:** Recall@K for a query is defined as the ratio of the number of relevant documents in top-K retrieved documents, with respect to the total number of relevant documents for that query. This is averaged over all the queries in the dataset. Since, in our case, each query has only one relevant document, Recall@K

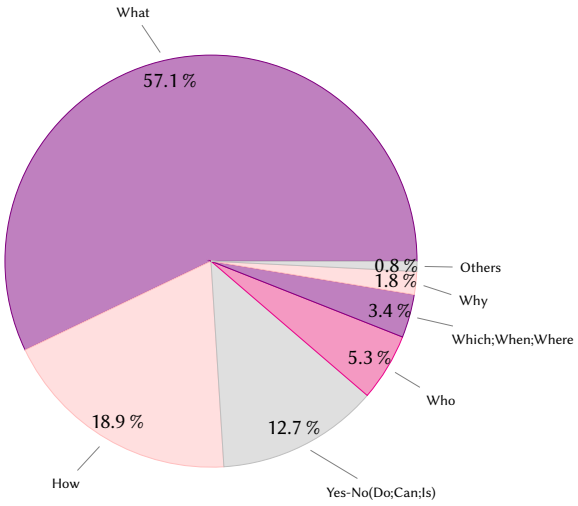

Figure 4: Percentage of questions by type.

denotes the percentage of queries whose correct document was present in the top-K retrieved documents.

$$Recall@K = \frac{1}{Q_{test}} \sum_{i=1}^{Q_{test}} \frac{\#\text{ relevant documents in top-K for query } i}{\#\text{ total relevant documents for query } i} \tag{22}$$

- **Mean Reciprocal Rank (MRR):** The reciprocal rank is defined as the inverse of the rank of the first correct document d^{pos} for a given query. MRR is the mean of the reciprocal rank for all the queries in the test set. Since we have only one correct document for every query, MRR is well suited for our evaluation. As compared to Recall@K, MRR also takes into account the ranking order in the evaluation.

$$MRR = \frac{1}{Q_{test}} \sum_{i=1}^{Q_{test}} \frac{1}{rank(d^{pos}) \text{ of query } i} \tag{23}$$

5.2 Baseline Methods

For performance comparison, we use the following state-of-the-art retrieval models as the baselines:

- **TF-IDF:** This is the standard baseline for retrieval tasks, that uses TF-IDF representation for both the query and document, and cosine similarity as the similarity function.
- **CDSSM [34]:** It is an extension of DSSM, that uses a CNN-based model on letter trigrams from query and document as input features, and then computes the cosine similarity to calculate the relevance between query and document representation.
- **ARC-II [12]:** This model first computes the interaction feature vector between query and document using CNN layers. It then computes the score for the query-document interaction vector using feed-forward network.
- **MV-LSTM [36]:** It is a neural semantic matching model that was proposed to find semantic similarity between a pair of sentences. The model uses the word embeddings obtained by passing the sentences through a Bi-LSTM, and then computes an interaction vector using cosine similarity, or a bilinear operation. It finally passes the interaction vector through a feed-forward network to compute the similarity score. In our implementation, we use cosine similarity to compute the interaction vector.
- **DRMM [9]:** It is a state-of-the-art neural ranking model that uses cosine similarity between query and document word vectors to compute their similarity, and then computes a histogram-like interaction vector by binning the cosine similarity scores into pre-defined intervals. It then passes these features through a feed-forward network to compute the scores.
- **KNRM [40]:** It is a recently proposed neural ranking model that first computes cosine similarity between each query word with each of the document words. It then performs kernel pooling, followed by a feed-forward network to compute the relevance score.
- **aNMM [42]:** This model first computes an interaction matrix by computing the cosine similarity between each of the query and document words. Similar to DRMM, this model also performs binning to compute a fixed-dimensional interaction vector. However, instead of using the counts of word-pairs that fall into a bin as its features, this model uses the total sum of the similarity between those word pairs as the bin features. It also uses an attention mechanism over the query word vectors, which is then combined with the interaction vector to compute the final relevance scores.
- **Duet [21]:** It is a recently proposed hybrid neural matching model that uses the word-level interaction, and document-level similarity, in a deep CNN architecture, to compute similarity between two documents.
- **MatchPyramid [23]:** This model computes pair-wise dot product between query and document word vectors to compute an interaction matrix. It then passes this matrix through CNN layers with dynamic pooling to compute the similarity score.

5.3 Implementation Details

We implemented our model in Keras [3], with TensorFlow [1] as the backend. The model was trained using Adadelta optimizer [46], with an initial learning rate of 2.0. We used one Bi-GRU layer (one forward and one backward), and each GRU layer had an output dimension of 150 units. The maximum number of words in query, and each document sentence was set to 15, and sentences greater than 15 words were split into multiple sentences. The maximum number of sentences in each document was set to 20. All the attention layers had a dimension of 300. We used a feed-forward network with 3 layers to compute the final score from the similarity vector. Additionally, we used dropout of 0.2 after each layer. For each query, we had one positive document, 3 partially relevant negative documents, and 6 irrelevant negative documents. For all the neural baselines, we used an open-source implementation MatchZoo[4]. The TF-IDF baseline was implemented using the `scikit-learn`[5] package.

We used GloVe [24] pre-trained word vectors with 300 dimensions. Since healthcare documents contain some medical words which cannot be found in GloVe, we used randomly initialized embeddings for such out-of-vocabulary words. We experimented with

[4]https://github.com/NTMC-Community/MatchZoo
[5]http://scikit-learn.org/stable/index.html

training GloVe word vectors on our document corpus, but it was found that the original pre-trained GloVe outperformed our medical word embeddings. We hypothesize that it was due to the fact that our corpus contained far lesser documents than those used in the original GloVe. Also, it is observed that people usually do not use very complex medical words in their queries, and hence, most of the queries were composed of words that were present in the pre-trained GloVe.

The dataset was split into three parts: train, validation, and test. The number of queries in each split were 5,274, 1,109, and 1,134 respectively.

5.4 Results

In this section, we present the results obtained in the experimental evaluation of the proposed HAR model.

5.4.1 Quantitative comparison against state-of-the-art models. Table 2 shows the performance of HAR against the baseline retrieval models on the HealthQA dataset. For computing Recall@K, we set K as 1, 3, and 5. As we can see, the results for HAR are consistently better than all other baselines, across all the metrics, which can be attributed to its strong performance.

- **Effect of long documents on model performance:** We can observe from Table 2 that TF-IDF has very low performance on our dataset. With the exception of ARC-II, the performance of TF-IDF is consistently lower than all the other models. This can be attributed to the non-factoid nature of our dataset, as well as the long length of documents in the corpus. Hence, a keyword-based method cannot perform well on such dataset, since its performance largely depends on the word overlap between the query and the document. In such cases, embedding-based methods yield better performance in general, as they can correlate queries and documents based on their semantic representation.
- **Document-representation based semantic similarity methods:** One key observation from our results is that neural models that match query with documents solely based on their document-level representation do not work well for retrieval problems. CDSSM computes the representation of query and document separately, and then computes the similarity between their vector representations. Although using semantic representations can help in dealing with the problems that arise where the queries do not have matching words with the documents, this concept only works in problems such as sentence or paraphrase matching, where the length of both the documents being matched is similar. In case of retrieval, queries are typically much shorter compared to the documents. Moreover, the actual part in the document that is relevant to the query is only a few words or sentences. Hence, the vector representation of the document contains features from other parts of the documents that are irrelevant to the query. This leads to the poor performance of such models for retrieval tasks.
- **Effect of word interactions:** With the exception of ARC-II, other baselines such as MV-LSTM, DRMM, KNRM, aNMM, Duet, and MatchPyramid use some form of embedding-based pair-wise keyword interaction in the feature representation of the query-document pair. This results in their better performance compared to other baselines. Using word-level interaction features based on their vector representation allows the models to deal with the problems faced by traditional word matching methods such as TF-IDF. However, by computing interaction in the early stages of the model, these models do not have any mechanism to incorporate the underlying structure of the query and document in the interaction feature generation or scoring process. This leads to their poor performance in a setting where the documents are longer in length.

By using a cross attention mechanism between the query and document, HAR is able to model the interaction features between the query and document words, while retaining the overall semantic meaning of the document sentences. The self attention mechanism then facilitates focussing on sentences and words in the document that are most relevant to the query. This mechanism helps HAR to achieve the highest performance compared to all other baseline methods. Most importantly, HAR achieves a considerably high MRR and Recall@1 against all other methods. This implies that, in most of the cases, our model is able to rank the correct document with the highest score, demonstrating a higher reliability of HAR.

In Figure 5, we show the performance of HAR and other baseline methods on each of the question types given in Figure 4. Although questions of the type *"what"* are relatively easier to answer (as they contain many factoid questions), our model outperforms other baselines on these questions. We can also see that HAR gives high performance on question types *"how"* and *"why"*, which are non-factoid in nature, and difficult for a retrieval system. The performance of HAR is consistently higher than the baselines across all other question categories as well.

5.4.2 Qualitative results. For qualitative evaluation of the performance of HAR, we show an example of a question, and the retrieved document, obtained by MatchPyramid and HAR, in Figure 6. We compare with MatchPyramid, since it is the strongest baseline among all the other methods. The question shown here is about *"the effect of wisdom tooth removal on brushing"*. Although the document returned by MatchPyramid is relevant to the topic, which is *wisdom teeth removal*, it is not the one that can correctly answer the query. MatchPyramid computes interaction features at an early stage in the model. It does not retain the original query and document, and computes scores solely based on the interaction features. Due to this, the main intent of the question can sometimes be lost, as shown in the example here. By using a powerful attention mechanism, HAR has the ability to discriminate between two similar documents, based on the intent of the query. Hence, HAR is able to retrieve the correct document for the question.

5.4.3 Using HAR for answer extraction. As mentioned earlier, an added advantage of using the hierarchical attention mechanism is that it allows the model to discover the most probable answer snippet from the long document. This can be done by comparing the attention weights of different sentences in level-1 of hierarchical inner attention over documents. Since sentences with high attention weights have more contribution towards generating the document representation, they are likely to be more relevant to the query, as compared to sentences with low attention weights.

In Figure 7, we illustrate how the attention weights can be used to extract the most probable answer from the document. We show

Table 2: Comparison of HAR model with other baseline models on HealthQA dataset.

Model name	MRR	Recall@1	Recall@3	Recall@5
TF-IDF	60.597	36.576	81.740	96.909
CDSSM	64.461	44.274	81.335	93.418
ARC-II	50.366	29.847	61.858	78.539
MV-LSTM	75.145	58.882	90.262	97.385
DRMM	74.082	57.078	90.081	99.008
KNRM	70.965	54.914	84.040	94.139
aNMM	74.717	58.251	90.352	98.287
Duet	69.659	53.291	84.310	93.868
MatchPyramid	81.816	69.432	93.688	98.918
HAR	**87.877**	**78.900**	**96.844**	**99.639**

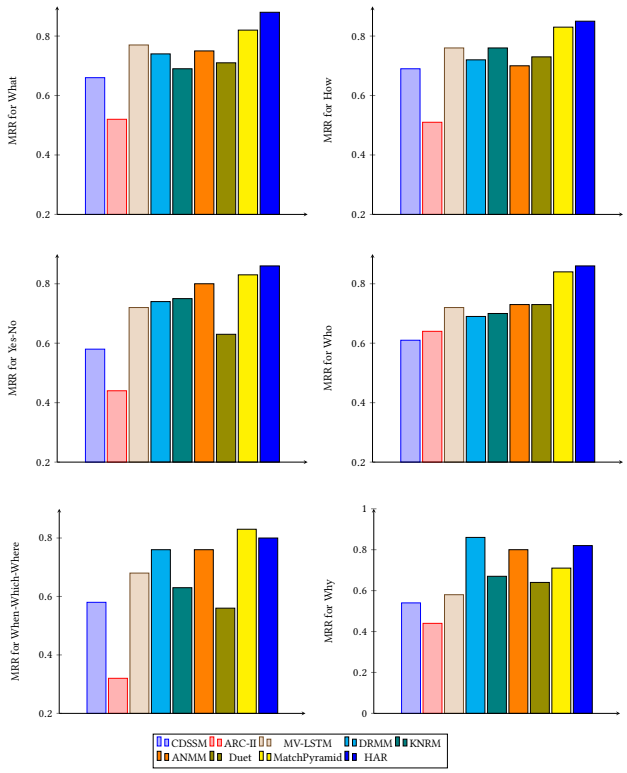

Figure 5: Performance of HAR and baseline methods on different types of questions.

one question, and it's corresponding highest-ranked document. The self-attention weights over the query words are highlighted in blue, while the self-attention weights at sentence level (level-1) of the document hierarchical attention are shown in red.

The question shown in Figure 7 is about *Alopecia Areata*, which is a condition that leads to hair loss in many people, mainly because the immune system of a person starts attacking the hair follicles. The question asks about the diagnosis procedure for this condition, demonstrated by the use of the word *test*. Since diagnosis is the main intent of this question, the feature representation uses highest

Will removal of wisdom teeth affect my brushing ?	
MatchPyramid answer (incorrect): Most people need two or three days off work to recover from the effects of the anaesthetic after they have had wisdom teeth removed. Those with a more physical job may need an extra day or two. If your job involves driving you should be fine to return after 24 - 48 hours after having had anaesthetic, but you need to use your own judgement and seek advice from your dentist if you are unsure.	**HAR answer (correct):** You may find it uncomfortable to brush shortly after having wisdom teeth removed, especially around any wounds. For this reason, your dentist might recommend that you use a mouthwash to help keep the area clean.

Figure 6: An example of a question and its answer retrieved by MatchPyramid (left) and HAR (right).

attention weight for the word *test*, followed by other words that are useful in the question. The document here has the highest attention weight for the first sentence, that contains the answer to this question. Other highlighted sentences in the document are those which can provide additional information supporting the answer for this question.

Do I need any tests, for Alopecia Areata ?

Usually not . the diagnosis is usually based on the typical appearance of the bald patches . If there is doubt about the cause of the hair loss , sometimes some blood tests or a skin scraping from a bald patch may be done to rule out other causes . A small sample of skin (skin biopsy) is sometimes taken to look at under the microscope . Occasionally you may need some tests to check for other autoimmune diseases . For example , if you have certain other symptoms , you might need to have blood tests to check your blood count and thyroid function . Usually blood tests come back entirely normal with alopecia areata .

Figure 7: An example of a question and its answer document with highlightings based on the attention weights. The attention weights for the question are obtained from the query self attention, and those for document are obtained from level-2 self attention.

5.5 Performance Analysis

5.5.1 Effect of attention mechanism. To quantitatively evaluate the effect of various components used in our HAR on the model performance, we compare the performance of HAR against its two variants. These are described below:

- **HAR without cross-attention:** To evaluate the effect of using cross-attention mechanism on model performance, we evaluate the performance of a variant of HAR that does not use cross-attention between query and document words. This model uses an inner attention over query words, and a hierarchical inner attention over document words and sentences. By removing the cross attention, the model is not able to use the interaction features between query and document word vectors in the scoring process. We refer to this model as *HAR-WCA*.
- **HAR without cross and hierarchical attention:** This is an even simpler version of HAR that neither uses cross-attention between query and document words, nor the hierarchical inner attention in the document. This model uses similar encoder as HAR for query and document, and then uses only one level of inner attention to get the query and document representations. It then computes the scores between these vectors similar to the scoring process used in HAR. It does not incorporate the underlying document structure in the scoring process due to the removal of hierarchical attention. We refer to this model as *HAR-simple*.

Model name	MRR	Recall@1	Recall@3	Recall@5
HAR-simple	82.139	69.883	94.770	99.008
HAR-WCA	83.667	72.047	95.942	99.369
HAR	**87.877**	**78.900**	**96.844**	**99.639**

Table 3: Performance comparison of HAR and its variants.

Table 3 shows the performance comparison of HAR with two of its variants. The performance of both *HAR-WCA* and *HAR-simple* is worse than the full model. We believe that since *HAR-simple* uses a single long encoder for documents, it is not able to embed the contextual dependencies in the encoded embeddings. Also, it does not use cross-attention, thereby ignoring the keyword-interaction features. The performance of *HAR-WCA* is slightly better than that of *HAR-simple*. Since each sentence sequence is much smaller than the full document, the encoded representation is able to embed the context of the sentence in each word.

5.5.2 Hyperparameter sensitivity and model convergence. As mentioned earlier, by splitting the documents into short sentences and using the hierarchical attention mechanism, HAR does not need to deal with long sequences. This allows the model to be achieve high performance using computationally efficient GRU, which can effectively model short sequences. We also find that by using GRU, the model is able to converge quickly, as compared to a model that uses LSTM. We also evaluated the performance of HAR by varying different parameters of our model, such as the number of GRU hidden units and the number of feed-forward layers. We find that these parameters only have a marginal impact (~1% MAP reduction) on the performance of our model. In Figure 8, we show the convergence of HAR and other baselines over training epochs. We show the MRR of different models on test queries, as the number of training epochs increase. We can see that HAR converges faster as compared to other baselines, demonstrating a better learning ability of our model.

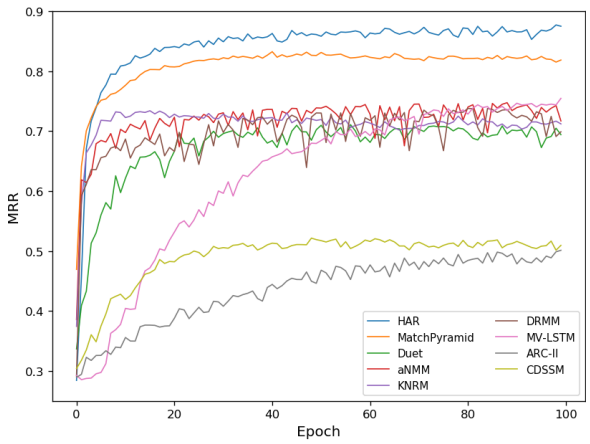

Figure 8: MRR convergence with training epochs.

6 CONCLUSION

In this paper, we proposed a novel deep neural network architecture to rank documents for healthcare related queries. The model uses a combination of powerful attention mechanisms to develop a robust retrieval system. This attention mechanism also enables the model to discover highly probable answer snippets from the documents, without the need for using a computationally expensive machine comprehension module. The model has been carefully designed by considering the special characteristics of question-answering in healthcare domain, such as the open-ended nature of queries, and longer document length.

To evaluate the proposed HAR model, we constructed a novel consumer-oriented healthcare question answering dataset, HealthQA. This dataset is comprised of questions that consumers typically ask about health-related topics. We evaluated our proposed model on this dataset, against several state-of-the-art baseline techniques. Our experimental results show that our model outperforms these techniques by a wide margin. We also show how our model can be used to extract the most probable answer snippets from the highly-ranked documents. We hope that our proposed model will be useful for both healthcare and information retrieval communities, to make healthcare information more accessible to the people.

ACKNOWLEDGEMENTS

This work was supported in part by the US National Science Foundation grants IIS-1619028, IIS-1707498, and IIS-1838730, and NVIDIA Corporation.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. TensorFlow: a system for large-scale machine learning. In *Proceedings of*

the 12th USENIX conference on Operating Systems Design and Implementation. USENIX Association, 265–283.

[2] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1724–1734.

[3] François Chollet et al. 2015. Keras.

[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 Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 126–134.

[5] Dina Demner-Fushman and Jimmy Lin. 2006. Answer extraction, semantic clustering, and extractive summarization for clinical question answering. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 841–848.

[6] Yoav Freund, Raj Iyer, Robert E Schapire, and Yoram Singer. 2003. An efficient boosting algorithm for combining preferences. *Journal of machine learning research* 4, Nov (2003), 933–969.

[7] Fredric C Gey. 1994. Inferring probability of relevance using the method of logistic regression. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*. Springer-Verlag New York, Inc., 222–231.

[8] Paul N Gorman, Joan Ash, and Leslie Wykoff. 1994. Can primary care physicians' questions be answered using the medical journal literature? *Bulletin of the Medical Library Association* 82, 2 (1994), 140.

[9] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 55–64.

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

[11] Phu Mon Htut, Samuel Bowman, and Kyunghyun Cho. 2018. Training a Ranking Function for Open-Domain Question Answering. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*. 120–127.

[12] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *Advances in neural information processing systems*. 2042–2050.

[13] 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 22nd ACM international conference on Conference on information & knowledge management*. ACM, 2333–2338.

[14] Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 133–142.

[15] Minsuk Lee, James Cimino, Hai Ran Zhu, Carl Sable, Vijay Shanker, John Ely, and Hong Yu. 2006. Beyond information retrieval : medical question answering. In *AMIA annual symposium proceedings*, Vol. 2006. American Medical Informatics Association, 469.

[16] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *International Conference on Learning Representations (ICLR)* (2017).

[17] Gang Luo, Chunqiang Tang, Hao Yang, and Xing Wei. 2008. MedSearch: a specialized search engine for medical information retrieval. In *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 143–152.

[18] Yuanhua Lv and ChengXiang Zhai. 2011. When documents are very long, BM25 fails!. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 1103–1104.

[19] Wenlei Mao and Wesley W Chu. 2002. Free-text medical document retrieval via phrase-based vector space model.. In *Proceedings of the AMIA Symposium*. American Medical Informatics Association, 489.

[20] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[21] Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to match using local and distributed representations of text for web search. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1291–1299.

[22] Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab Ward. 2016. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)* 24, 4 (2016), 694–707.

[23] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text matching as image recognition. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI Press, 2793–2799.

[24] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[25] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 2383–2392.

[26] Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, Vol. 242. 133–142.

[27] Stephen E Robertson and K Sparck Jones. 1976. Relevance weighting of search terms. *Journal of the American Society for Information science* 27, 3 (1976), 129–146.

[28] Stephen E Robertson, Steve Walker, Susan Jones, et al. 1995. Okapi at TREC-3. (1995).

[29] Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management* 24, 5 (1988), 513–523.

[30] G. Salton, A. Wong, and C. S. Yang. 1975. A Vector Space Model for Automatic Indexing. *Commun. ACM* 18, 11 (Nov. 1975), 613–620. https://doi.org/10.1145/361219.361220

[31] Rudolf Schneider, Sebastian Arnold, Tom Oberhauser, Tobias Klatt, Thomas Steffek, and Alexander Löser. 2018. Smart-MD: Neural Paragraph Retrieval of Medical Topics. In *Companion of the The Web Conference 2018 on The Web Conference 2018*. International World Wide Web Conferences Steering Committee, 203–206.

[32] Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45, 11 (1997), 2673–2681.

[33] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. *International Conference on Learning Representations (ICLR)* (2017).

[34] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. Learning semantic representations using convolutional neural networks for web search. In *Proceedings of the 23rd International Conference on World Wide Web*. ACM, 373–374.

[35] Ellen M. Voorhees and Dawn M. Tice. 2000. Building a Question Answering Test Collection. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '00)*. ACM, New York, NY, USA, 200–207. https://doi.org/10.1145/345508.345577

[36] Shengxian Wan, Yanyan Lan, Jiafeng Guo, Jun Xu, Liang Pang, and Xueqi Cheng. 2016. A Deep Architecture for Semantic Matching with Multiple Positional Sentence Representations.. In *AAAI*, Vol. 16. 2835–2841.

[37] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. 2014. Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1386–1393.

[38] Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 189–198.

[39] Dirk Weissenborn, Georg Wiese, and Laura Seiffe. 2017. Making Neural QA as Simple as Possible but not Simpler. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*. 271–280.

[40] 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 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 55–64.

[41] Jun Xu and Hang Li. 2007. Adarank: a boosting algorithm for information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 391–398.

[42] Liu Yang, Qingyao Ai, Jiafeng Guo, and W Bruce Croft. 2016. aNMM: Ranking short answer texts with attention-based neural matching model. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 287–296.

[43] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 1480–1489.

[44] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. 2018. QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension. *International Conference on Learning Representations (ICLR)* (2018).

[45] Hong Yu, Minsuk Lee, David Kaufman, John Ely, Jerome A Osheroff, George Hripcsak, and James Cimino. 2007. Development, implementation, and a cognitive evaluation of a definitional question answering system for physicians. *Journal of biomedical informatics* 40, 3 (2007), 236–251.

[46] Matthew D Zeiler. 2012. ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701* (2012).