Interactive Attention Networks for Semantic Text Matching

Sendong Zhao*, Yong Huang†, Chang Su*, Yuantong Li‡, Fei Wang*
*Weill Cornell Medical College, Cornell University
†Cornell Tech, Cornell University
‡Department of Statistics, Purdue University
{sez4001, chs400, few2001}@med.cornell.edu, yh849@cornell.edu, li3551@purdue.edu

***Abstract*—Semantic text matching, which matches target texts to source texts, is a general problem in many areas, such as information retrieval, question answering, and recommendation. The challenges to existing research on this topic include 1) out-of-vocabulary and low-frequency keywords and 2) direct utilization of sparse matching matrix of source and target. The out-of-vocabulary and low-frequency keywords could lead to the mismatch of similar keywords in source and target texts. The sparse matching matrix cannot provide enough clues to match the source with the target. To address these challenges, we propose a novel deep neural semantic text matching model. Our model adopts an interactive attention network to achieve information exchange between the source text and the target text, and dynamically explores the matching matrix and learns new representations of source and target texts. Experimental results on three different text matching datasets demonstrate that our model can significantly outperform competitive baselines. Furthermore, our model demonstrates great advantage in alleviating the sparse matching problem and learning out-of-vocabulary words with the local context, which widely exists in a broad spectrum of NLP applications.**

***Index Terms*—text semantic matching, deep neural networks, interactive attention, sparse matching, out-of-vocabulary words, information retrieval, question answering, tweet linking**

	Approach	of	the	treatment	for	pneumothorax
How	0	0	0	0	0	0
do	0	0	0	0	0	0
you	0	0	0	0	0	0
treat	0	0	0	1	0	0
a	0	0	0	0	0	0
collapsed	0	0	0	0	0	0
lung	0	0	0	0	0	0
?	0	0	0	0	0	0

Fig. 1. An example of sparse matching between source and target texts. "Approach of the treatment for pneumothorax" is the title of a biomedical article. The source and target only share a common word "treat" (after stemming). "collapsed lung" in the source text is the synonym of "pneumothorax" in the target text. The uncommon disease name "pneumothorax" is likely to be an out-of-vocabulary word or low-frequency word. It is extremely hard to match these two texts if we cannot learn the similarly between "collapsed lung" and "pneumothorax".

I. Introduction

Semantic text matching learns the semantic similarities between the source and target text pieces. It plays an essential role in many areas, such as information retrieval, question answering, and recommendation. The challenges of semantic text matching mainly arise in three aspects. First, text pieces with the same meaning usually have different expressions (such as cancer and tumor). Second, semantically similar texts might have sparse matching on the exact keywords. For example, the text "this academic paper talks about bioinformatics" and the text "the topic of this research is bioinformatics" have almost the same meaning but share only one keyword "bioinformatics." Third, out-of-vocabulary and low-frequency keywords in source and target texts usually cause mismatches [1], [2]. For example, it is difficult to match the low-frequency keyword "pneumothorax" in the source text with its synonym "collapsed lung" in the target text [3]. All these difficulties make semantic text matching still a challenging problem.

The rise of deep learning approaches in recent years, such as recurrent neural networks [4], long short-term memory neural networks [5], convolutional neural networks [6], and Transformer [7], has firmly established the state-of-the-art performance for understanding the complex semantic relationships among texts. These approaches can be generally categorized into two classes: 1) representation-focused model, and 2) interaction-focused model. The representation-focused model learns representations of source and target texts, and measures relevance between them based on the learned representations [8]–[12]. The interaction-focused model constructs a matching matrix between the source and target texts first and estimates the matching score by analyzing the matching matrix [10], [13]–[17]. However, both models may fail if 1) keywords in texts are out-of-vocabulary (OOV) and low-frequency, which would cause the mismatch of semantically similar keywords; 2) there are few shared words in source and target texts, which could generate an extremely sparse matching matrix between source and target texts (see an example in Figure 1). Sparsity has been a challenge by itself for machine learning models, let alone the lack of sufficient information for linking source and target texts.

In view of these challenges, we propose a novel deep neural semantic text matching model named as **i**nteractive **a**ttention network for **s**emantic text **m**atching (IASM) in this paper. Our model builds an interactive attention network to

2374-8486/20/$31.00 ©2020 IEEE
DOI 10.1109/ICDM50108.2020.00095

achieve information exchange between source and target texts and updates the matching matrix during the model learning process. In this way, on the one hand, the representations of source and target texts will be enriched after information exchange, which makes the sparse matching matrix smoother. On the other hand, the contextual relevance in specific source and target text pairs will be encoded in the dynamic matching matrix. Consequently, our model could **alleviate** sparse matching problem with the enriched representations and the dynamic matching matrix. Experiments are conducted on three different semantic text matching datasets to demonstrate the effectiveness of our proposed model[1].

The main contributions of this paper are summarized as follows:

- This paper explores the matching matrix to enrich representations of source and target texts.
- The IASM model achieves information exchange between the source text and target text through interactive attention, which could alleviate the sparse matching problem through smoothing the sparse matching matrix.
- The IASM model takes advantage of interactive attention between source and target texts, which could alleviate the mismatch of text pairs, especially for those contain not well-learned keywords in pre-trained word embeddings, such as OOV and low-frequency words.
- We collect a new dataset for semantic text matching.

II. Interactive Attention Network

In this section, we present the interactive attention networks for text semantic matching. 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 I lists the symbols and their descriptions that are used throughout this paper.

The input of our model is a pair of source and target texts (q, d). The source text q is composed of a m words sequence $(q_1, q_2, ..., q_m)$ and the target text d is composed of a n words sequence $(d_1, d_2, ..., d_n)$. The pre-trained word embedding of each word $q_i \in q$ and $d_j \in d$ can be obtained via representation learning on external resources such as knowledge bases and large corpus. Then, we can get the representation of source text $\mathbf{Q}^{(0)} = \{\mathbf{q}_1^{(0)}, \mathbf{q}_2^{(0)}, ..., \mathbf{q}_m^{(0)}\}$ and the representation of target text $\mathbf{D}^{(0)} = \{\mathbf{d}_1^{(0)}, \mathbf{d}_2^{(0)}, ..., \mathbf{d}_n^{(0)}\}$. Thus, we can get an initial matching matrix $\mathbf{A}^{(0)}$ through the word-level similarity between the source text q and the target text d based on the pre-trained representations of source and target texts $\mathbf{Q}^{(0)}$ and $\mathbf{D}^{(0)}$.

$$\mathbf{A}_{ij}^{(0)} = Sim(\mathbf{q}_i^{(0)}, \mathbf{d}_j^{(0)}) \tag{1}$$

where we exploit cosine similarity as the Sim function. This matching matrix specifies the space of element-wise interactions between objects q and d. Also, the initial matching matrix $\mathbf{A}^{(0)}$ can be an adjacency matrix for a bipartite graph extracted from some external knowledge graphs.

[1]Our code is available on https://github.com/SendongZhao/IASM.git.

TABLE I
Symbols and descriptions.

Symbol	Description
q	the source text
d	the target text
$\mathbf{A}^{(n)}$	the matching matrix between q and d in the n-th layer, where $\mathbf{A}$ keeps the same in IASM-Static and changes dynamically in IASM-Dynamic
$\mathbf{Q}^{(n+1)}$	the new representation of source q after n layers of interactive attention
$\mathbf{D}^{(n+1)}$	the new representation of target d after n layers of interactive attention
$\mathbf{W}_q^{(n)}$	parameters of source-side the n-th layer of interactive attention network
$\mathbf{W}_d^{(n)}$	parameters of target-side n-th layer of interactive attention network
α, β	hyper-parameters to balance the local and global matching clues
γ, δ	hyper-parameters to balance two channel matching scores
Δ	a margin separating true pairs and corrupted pairs
$\mathcal{M}^+$	the set of true source and target text (q, d) pairs
$\mathcal{M}^-$	the set of corrupted pairs

The architecture of the interactive attention network consists of three components, which are illustrated in Figure 2.

- Interactive attention for the source text.
- Interactive attention for the target text.
- Two-channel distance measure.

They are designed to learn new representations of source and target texts, leverage interactive attention, and exchange information between these two through the matching matrix. This interactive process can update the matching matrix in dynamic mode. On the one hand, the matching matrix will be smoothed through information exchange. On the other hand, it is possible to incorporate the local relevance of a specific pair with the dynamic matching matrix.

A. Definition of static and dynamic matching matrix

Since the matching matrix is updated as the changing of representations of source and target texts, the dynamic matching matrix is defined as

$$\mathbf{A}_{ij}^{(n+1)} = \alpha Sim(\mathbf{q}_i^{(n+1)}, \mathbf{d}_j^{(n+1)}) + \beta \mathbf{A}_{ji}^{(n)} \tag{2}$$

where α and β are hyper-parameters to balance the local and global matching clues. $Sim(\mathbf{q}_i^{(n+1)}, \mathbf{d}_j^{(n+1)})$ is to calculate local matching information, while $\mathbf{A}_{ji}^{(n)}$ is to sustain the global matching information between source and target which derives from the initial matching matrix $\mathbf{A}^{(0)}$. $\mathbf{A}^{(0)}$ is obtained from global pre-trained word embeddings. In this way, the new matching matrix could incorporate the initial matching matrix, which is the crucial prior knowledge. We call the proposed model with the dynamic matching matrix as **IASM-Dynamic**.

If the matching matrix is defined as follows:

$$\mathbf{A}_{ij}^{(n+1)} = \mathbf{A}_{ji}^{(n)} \tag{3}$$

We call the model using static matching matrix as **IASM-Static**. The matching matrix $\mathbf{A}^{(n+1)}$ in the $(n+1)$th layer is the transpose of matching matrix $\mathbf{A}^{(n)}$ in the nth layer.

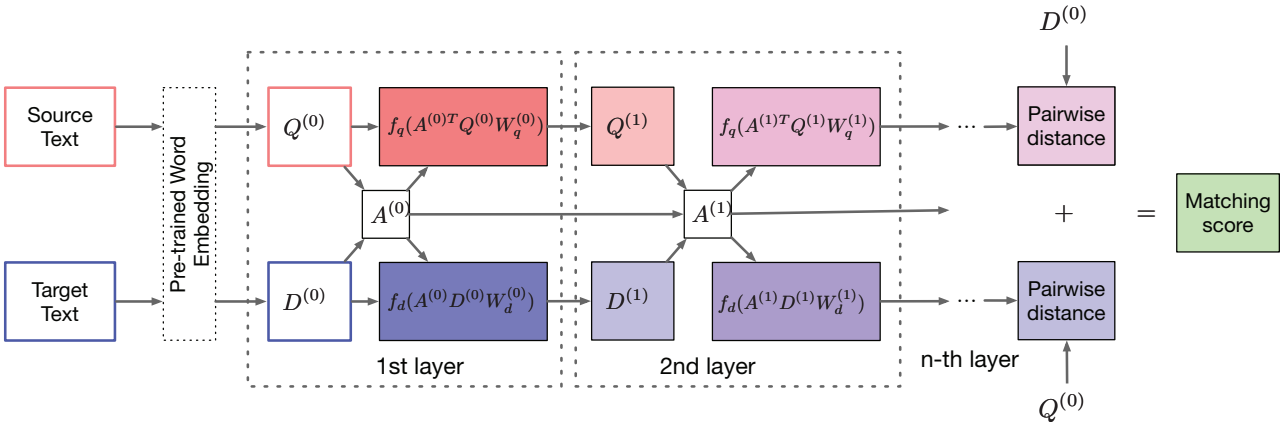

Fig. 2. The framework of the interactive attention network for semantic text matching. The dimension of $\mathbf{Q}^{(0)}$ is $m \times k$, $\mathbf{D}^{(0)}$ is $n \times k$, $\mathbf{A}^{(0)}$ is $m \times n$, $\mathbf{Q}^{(1)}$ is $n \times k$, $\mathbf{D}^{(1)}$ is $m \times k$. This framework could have multiple layers. In each layer, the representations of source $\mathbf{Q}$ and target $\mathbf{D}$ are fed into interactive attention with matching matrix $\mathbf{A}$ to get new representations for source and target. At last, distance measures are conducted in two channels to get a matching score of source and target. This is an interaction-focused method for calculating text similarity. Word embeddings are used to calculate the similarity kernel (i.e., $\mathbf{A}$). However, it is novel for changing the process in two ways. In each iteration: 1. kernel similarity is converted back to sentence embedding, and vice versa, resulting in a deep network; 2. kernel and final similarity are continuously updated using a moving average.

$\mathbf{A}^{(n+1)}$ keeps the information that derive from the very initial matching matrix $\mathbf{A}^{(0)}$.

B. Interactive Attention for Source Text

We take the initial matching matrix $\mathbf{A}^{(0)}$, which represents the word-level similarity between d and q, as the interactive matrix between target and source texts. For the initial representation of source text $\mathbf{Q}^{(0)}$, which is composed of pre-trained word embeddings $\{\mathbf{q}_i\}$, we conduct interactive attention with the matching matrix to learn a new representation of each word in source text q. More concretely, the interactive attention on source text q builds the message-passing from the target text d. In this way, the IASM model could integrate relevance information from all words in the target text to enrich the representation of each word in the source text. This interactive attention process could be multiple layers. For the 1st layer, we have

$$\mathbf{Q}^{(1)} = f_q(\mathbf{A}^{(0)T}\mathbf{Q}^{(0)}\mathbf{W}_q^{(0)}) \tag{4}$$

where the incoming interactive information is accumulated and passed through a neural network-like function f_q, such as a linear transformation plus ReLU. $\mathbf{Q}^{(0)}$ is the original representation of source text, which is composed of pre-trained word embeddings $\{\mathbf{q}_i\}$. For the $(n+1)$th layer, $n = 0, 1, 2, ...$, we have

$$\mathbf{Q}^{(n+1)} = f_q(\mathbf{A}^{(n)T}\mathbf{Q}^{(n)}\mathbf{W}_q^{(n)}) \tag{5}$$

where $\mathbf{W}_q^{(n)}$ is a weight matrix for nth layer.

C. Interactive Attention for Target Text

For the initial representation of target text $\mathbf{D}^{(0)}$, which is composed of pre-trained word embeddings $\{\mathbf{d}_j\}$, we conduct interactive attention with the initial matching matrix $\mathbf{A}^{(0)}$ to learn new representation of each word in target text d. Likewise, the interactive attention on target text d builds the message-passing from the source text q. In this way, the IASM model could integrate relevance information from all words in the source text to enrich the representation of each word in the target text. This interactive attention process could be multiple layers. For the 1st layer, we have

$$\mathbf{D}^{(1)} = f_d(\mathbf{A}^{(0)}\mathbf{D}^{(0)}\mathbf{W}_d^{(0)}) \tag{6}$$

where the incoming relevance information is accumulated and passed through a neural network-like function f_d, such as a linear transformation plus ReLU. $\mathbf{D}^{(0)}$ is the original representation of target text, which is composed of pre-trained word embeddings $\{\mathbf{d}_j\}$. For the $(n+1)$th layer, we have

$$\mathbf{D}^{(n+1)} = f_d(\mathbf{A}^{(n)}\mathbf{D}^{(n)}\mathbf{W}_d^{(n)}) \tag{7}$$

where $\mathbf{W}_d^{(n)}$ is a weight matrix for nth layer.

D. Two-channel Distance Measure

To measure the matching score between source and target texts, we conduct relevance measures on two different channels, i.e., source channel and target channel. We compare the difference between original source text representation $\mathbf{Q}^{(0)}$ and the new learned target representation $\mathbf{D}^{(n)}$ which is obtained after $n-1$ layers of interactive attention. The distance between these two representations is computed as $Dist(\mathbf{Q}^{(0)}, \mathbf{D}^{(n)})$. We compare the difference between original target text representation $\mathbf{D}^{(0)}$ and the new learned source representation $\mathbf{Q}^{(n)}$ which is obtained after $n-1$ layers of interactive attention. The distance between these two representations is computed as $Dist(\mathbf{D}^{(0)}, \mathbf{Q}^{(n)})$. Each $\mathbf{Q}$ and $\mathbf{D}$ should be normalized before conducting $Dist()$. There are different choices of the distance metric, e.g., Euclidean and cosine. In our experiments, the distance measurement we exploit is Euclidean distance. The scoring function is defined as follows.

$$s(q, d) = \gamma Dist(\mathbf{Q}^{(0)}, \mathbf{D}^{(n)}) + \delta Dist(\mathbf{D}^{(0)}, \mathbf{Q}^{(n)}) \tag{8}$$

where γ and δ are hyper-parameters and n is the number of layers.

E. Learning of IASM

To optimize the parameters of our IASM model, we consider a ranking criterion [18]. Intuitively, given an exact pair (q, d), if the target text d is missing, the model should be able to predict the correct target text. For each exact pair of source and target texts (q, d), we sample several negative samples. The objective of the training is to learn the proposed model so that it can successfully rank the exact pair (q, d) to precede all other possible negative samples. Therefore, we define a loss function to formalize this intuition:

$$\mathcal{L} = \sum_{(q,d)\in\mathcal{M}^+} \sum_{(q^{'},d^{'})\in\mathcal{M}^-} [\Delta + s(q,d) - s(q^{'},d^{'})]_+ \quad (9)$$

where $\mathcal{M}^+$ is the set of true source and target text (q, d) pairs, $\mathcal{M}^-$ contains corrupted pairs constructed by negative sampling which replaces the source text or the target text in the true (q, d), $\Delta > 0$ is a margin separating true pairs and corrupted pairs, and $[x]_+ = max(0, x)$ denotes the positive part of x.

III. Experiments

A. Data Sets

We exploit three datasets to evaluate our model for semantic text matching. Table II shows the statistics of three datasets. PubMed dataset is a biomedical article retrieval dataset. TNL is a tweet and news linking dataset. YodaQA is an answer sentence selection dataset. Each data set is tokenized, POS-tagged by Stanford POS tagger [19].

We collect the PubMed dataset through sampling PMID [2] on PubMed engine. Each biomedical article includes a title with brief-expression and abstract, which is the summary of this biomedical article. We exploit the title of each biomedical article as the source text and take the corresponding abstract to be the unique target text. With the ratio 8:1:1, we partition the biomedical articles into training, validation, and testing sets. In this data set, each query has only one relevant document, and the other documents are irrelevant, making it difficult to retrieve the correct document. Besides, too many low-frequency/OOV biomedical terms in query and document may cause the sparse matching problem.

PubMed dataset presents a specific application of text matching in biomedical literature retrieval. We take as input a source text and output the most relevant biomedical articles to the query. This specific application is widely needed in many real biomedical scenarios. For example, when a user types in some disease-related information in the text [20], [21] (e.g., symptoms, disease name, genetic information, personal characteristics, history, etc.) [22], the system can provide the most relevant articles with treatment, prevention, or prognosis of the corresponding disease [23]–[25].

TNL dataset is a publicly available tweet to news linking dataset, which is provided by Guo et al. [26]. This dataset contains explicit URL links from each tweet to a related news article. They crawled 6,312 CNN and NYTIMES news (title + snippets) from RSS feeds from Jan 11 to Jan 27, 2013. 34, 888 tweets that contain a single link to a CNN or NYTIMES news were collected during the same period.

TNL dataset introduces a specific task of linking a tweet to a news article that is relevant to the tweet. The task is to input the text in a tweet and find the most relevant news article. Therefore, it is natural to take a tweet as the source text and the most relevant news article as the target text. Linking news to tweets can enrich the context of tweets that are usually short and informal. It can benefit the analysis of tweets and topics. Moreover, it helps event discovery from the tweet. There are many real-world similar scenarios because people tend to discuss the same event and topic in different web spaces. For example, the reporting of the same event differs across different news media. Individuals tend to have different comments for the same event in different expressions, even in different languages.

YodaQA dataset is also a publicly available answer sentence selection dataset[3], which is generated from databases and text corpora using information extraction. In this dataset, the questions come from the YodaQA dataset and the YodaQA system generated the candidate sentences based on enwiki, using YodaQA sentence-selection branch. Sentences were generated by running fulltext solr search on enwiki for keywords extracted from the question, then considering all sentences from top N results containing at least a single keyword. Sentences that match the gold standard answer regex are labeled as 1, and the rest is 0.

YodaQA dataset is applied for the task of answer sentence selection. The task is to select the sentence that contains the correct answer with a question. The performance of the answer sentence task is not only crucial to non-factoid QA systems, where a question is expected to be answered with a sequence of descriptive text, but also very important to factoid QA systems, where the answer sentence selection step is also known as sentence scoring. As the definition of semantic text matching, we take the question as the source and the sentence carrying the corresponding answer as the target.

TABLE II
THE STATISTICS OF THREE DATASETS.

Data Set	#Source Text	#Target Text
PubMed	40,000	40,000
TNL	34,888	6,312
YodaQA	1,113	136,963

B. Experimental Settings

For these three different datasets of text semantic matching, we conduct ranking experiments, which are evaluated with

[2]A PMID is the unique identifier number used in PubMed for each article.

[3]https://github.com/brmson/dataset-sts/tree/master/data/anssel/yodaqa

three standard information retrieval metrics, including precision at N (such as $P@1$), mean reciprocal rank (MRR), and mean average precision (MAP). The precision at N is the retrieval precision at top N. The MRR is the mean of the multiplicative inverse of the rank of the first correct answer. The MAP is the mean of the average precision across samples in our testing sets.

We train the model using Adam with the initial learning rate of $1e-4$ for 50 epochs. The best setting for α is 0.75, β is 0.25, γ is 0.5, and δ is 0.5. The best setting for the number of layers is 3 for datasets PubMed and TNL, 5 for the dataset YodaQA. For deep neural model baselines, we use the same setting reported in their original papers.

C. Pre-trained Word Embeddings

We initialize the word embedding matrix with three types of pre-trained word embeddings, respectively. The first is trained by Word2Vec [27]. For PubMed, we choose the pre-trained 100-dimensional Word2Vec embeddings trained on all 27.5 million MEDLINE biomedical articles. For TNL and YodaQA, we choose Google's 300-dimensional Word2Vec embeddings trained on roughly 100 billion words from a Google News dataset. The second is trained by GloVe [28]. For PubMed, we choose 100-dimensional GloVe embeddings trained on 27.5 million MEDLINE biomedical articles. For TNL, we choose Stanford pre-trained 100-dimensional GloVe embeddings trained on 2 billion tweets. For YodaQA, we use Stanford pre-trained 100-dimensional GloVe embeddings trained on Wikipedia 2014 + Gigaword 5. 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 [29].

D. Compared Methods

For all three datasets, we compare with the following baselines to evaluate our IASM models.

- CNN-based approach (CNN) exploits textCNN [6] to learn representations of source and target texts. Cosine similarity is applied to measure the relevance between source and target texts.
- RNN-based approach (RNN) [11] exploits an RNN to learn representations of source and target texts. The siamese structure is applied to measure the relations between documents.
- SMASH RNN [12] exploits the text structure to improve the representation of long-form texts. Representations of source and target texts are concatenated and fed into an MLP to get the matching score.
- Simple BERT exploits pre-trained BERT [30] to model each document as a word sequence and output its vector representation. An MLP is utilized on the concatenation of source and target representations to get the matching score.
- ARC-I [10] finds the representation of the source and target texts and compares the representation of the two with an MLP. ARC-I is a CNN-based model. It takes advantage of the flexibility brought by the convolutional sentence model.
- ARC-II [10] utilizes CNN to learns hierarchical matching patterns from local interactions. There are related studies that share similar structures with this model, such as MV-LSTM [31].
- DRMM [13] builds a matching matrix between query and document through word-level similarity from word embeddings. Histogram pooling is used to summarize word-word interactions upon matching matrix to get a matching score.
- Delta [16] constructs a "modified" document matrix by replacing the words in the documents with the closest words in the query. Convolutions will be performed on this matrix to obtain a final relevance score.
- Conv-KNRM [15] is the n-gram version of kernel-based interaction-based neural ranker. It models the interactions between query and document with unigram and bigram matching matrices. In particular, it convolves on the interaction matrices and predicts whether two objects are related.
- Extended BERT [17] extends BERT by adding a neural ranking network upon BERT. It first constructs the matching matrix between query and document, using the cosine similarities between the projections of their embeddings from BERT.

Our model is different from cross attention models [32]. Our study is novel for calculating text similarity in two ways. In each iteration: 1. kernel similarity (i.e., matching matrix) is converted back to sentence embedding, and vice versa; 2. kernel and final similarity are continuously updated using a moving average. While cross attention models directly operate column-wise and row-wise summation/average upon matching matrix, which is very different from our model.

The existing attention-based reading comprehension model is not applicable for semantic text matching. The query2document attention and document2query attention are either dot-product to get the probability of the answer word in the document (like in attention-over-attention [32]) or combined to fed to the sequence of the document (like in BIDAF [33]). It is fine to select words or text pieces in the document as the answer. However, it is not applicable for matching query and document which needs to compute a matching score between a query and a document. Therefore, we propose the dynamic interactive attention mechanism to learn new representations and measure relevance between query and document in two channels.

E. Experimental Results

This section presents the performance results of different semantic text matching models over the three datasets. A summary of the results is displayed in Table III.

As we can see, the poor performances of representation-focused models demonstrate the unsuitability of these models for semantic text matching. Even with SOTA word embedding system BERT, the representation-focused model (i.e., Simple

TABLE III
THE OVERALL PERFORMANCE OF SEMANTIC TEXT MATCHING OVER THREE DATASETS. (P-VALUE ≤ 0.05)

Model Type	Model	PubMed			TNL			YodaQA		
		P@1	MRR	MAP	P@1	MRR	MAP	P@1	MRR	MAP
Representation	CNN	0.0005	0.0013	0.0013	0.0017	0.0041	0.0041	0.0004	0.0035	0.0052
	RNN	0.0005	0.0021	0.0021	0.0011	0.0052	0.0052	0.0004	0.0018	0.0074
	ARC-I	0.0008	0.0015	0.0015	0.0021	0.0047	0.0047	0.0004	0.0034	0.0053
	SMASH RNN	0.0014	0.0027	0.0027	0.0036	0.0079	0.0079	0.0011	0.0046	0.00103
	Simple BERT	**0.0050**	**0.0175**	**0.0175**	**0.0231**	**0.0340**	**0.0340**	**0.2279**	**0.1334**	**0.4310**
Interaction	ARC-II	0.1066	0.1934	0.1934	0.1385	0.2132	0.2132	0.2046	0.1230	0.3951
	DRMM	0.1246	0.2245	0.2245	0.1317	0.2366	0.2366	0.2247	0.1728	0.4935
	Conv-KNRM	0.1356	0.2389	0.2389	0.1435	0.2487	0.2487	0.2562	0.1934	0.5023
	Delta	**0.1527**	**0.2724**	**0.2724**	0.1503	0.2643	0.2643	0.2423	0.1665	0.4738
	Extended BERT	0.1133	0.2067	0.2067	**0.1908**	**0.3332**	**0.3332**	**0.2446**	**0.1979**	**0.5321**
Our Models	IASM-Static	0.2189	0.3439	0.3439	0.2534	0.3494	0.3494	0.2924	0.2345	0.5778
	IASM-Dynamic	**0.2474**	**0.3749**	**0.3749**	**0.2734**	**0.3678**	**0.3678**	**0.3233**	**0.2454**	**0.5936**

BERT) is not promising as well. These representation-focused models learn representations of the source text and target text separately with neural networks such as RNN, CNN, and BERT. RNN, CNN, and ARC-I have similar performance, and Simple BERT consistently outperforms the other representation-focused models over all datasets, which shows the superiority of BERT for text modeling over CNN and RNN.

When we look at the interaction-focused models, we find that the interaction-focused models, i.e., ARC-II, Extended BERT, DRMM, Delta, and Conv-KNRM, perform better than representation-focused models. This is consistent with previous studies [13], [34], [35]. Among these models, DRMM outperforms ARC-II due to the direct use of the matching matrix. Note that the matching matrix in ARC-II is obtained through the weighted sum of query and document term vectors rather than cosine similarity or dot product. Conv-KNRM outperforms ARC-II because of utilizing both unigram and bigram matching matrices. Extended BERT performs better than DRMM, Conv-KNRM, and Delta over datasets TNL and YodaQA, which presents its power of modeling dependency between the source text and target text. However, Extended BERT performs worse than DRMM, Conv-KNRM, and Delta on PubMed data. This worse performance happens due to 1) the target text in PubMed dataset is too long, which causes bad representations of target texts, 2) too many low-frequency biomedical terms in both source and target texts make it hard to learn good representations. Moreover, Delta achieves the best performance on PubMed dataset. This model operates on matching matrix and gets matching score as well, but the difference is that it highlights keywords in source texts by replacing the words in the target texts with the closest words in source texts.

As for our proposed IASM models, we have the following observations: (1) IASM-Static is comparable to the best baselines on PubMed, TNL and YodaQA; (2) IASM-Dynamic performs significantly better than all existing deep learning models; (3) the dynamic matching matrix is better than static matching matrix on semantic text matching. Note that IASM-Static and IASM-Dynamic are both interaction-focused models. Our IASM models exploit matching matrix to conduct interactive attention across the source and target texts to exchange information rather than the static input features (i.e., DRMM, ARC-II, and Conv-KNRM take the matching matrix as the static input features), which is different from the other interaction-focused models. Therefore, IASM models could empower the ability to smooth sparse matching matrix. IASM-Dynamic is better than IASM-Static because the dynamic matching matrix could learn the local relevance in the specific context of source and target texts.

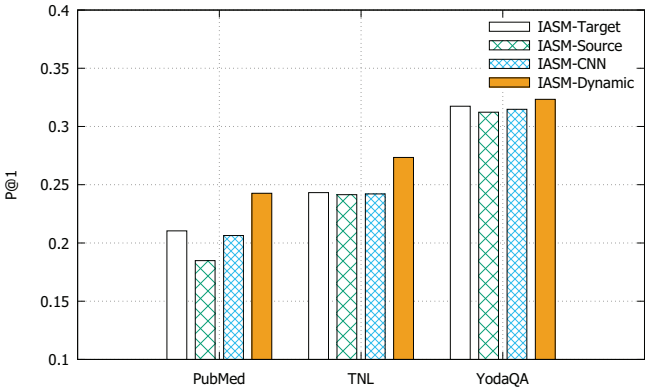

Fig. 3. The comparison results of different variants of IASM after removing some "feature".

F. Analysis of IASM model

1) Ablation Study: To further verify the effectiveness of our model, we alternately remove some "feature" of the model and see how that affects performance in three different data sets. In particular, we compare the original IASM-Dynamic model with several simpler versions of the model. IASM-Target removes the source-channel pairwise distance measure. IASM-Source removes the target-channel pairwise distance measure. IASM-CNN replaces the two-channel pairwise distance in our model with a CNN operation on the dynamic matching matrix. The comparison results of these variants over three datasets are depicted in Figure 3.

TABLE IV
COMPARISONS OVER THREE DATASETS WITH DIFFERENT PRE-TRAINED WORD EMBEDDINGS

Word Embedding	Model	PubMed		TNL		YodaQA	
		P@1	MAP	P@1	MAP	P@1	MAP
Random	ARC-II	0.0005	0.0009	0.0004	0.0007	0.0003	0.0012
	DRMM	0.0073	0.01249	0.00512	0.00830	0.0323	0.0879
	Conv-KNRM	0.0076	0.01325	0.0060	0.0094	0.0344	0.0879
	Delta	0.0145	0.0187	0.0177	0.0260	0.0513	0.1129
	IASM-Static	0.027	0.039	0.0194	0.0331	0.0742	0.1324
	IASM-Dynamic	0.0981	0.1164	0.0747	0.0974	0.1060	0.2154
Word2Vec	ARC-II	0.0823	0.1743	0.1184	0.1847	0.1763	0.3517
	DRMM	0.1127	0.1924	0.1103	0.2043	0.2123	0.4738
	Conv-KNRM	0.1145	0.1991	0.1245	0.2326	0.2485	0.4917
	Delta	0.1465	0.2631	0.1433	0.2579	0.1932	0.4156
	IASM-Static	0.1832	0.3166	0.2521	0.3432	0.2876	0.5721
	IASM-Dynamic	0.2059	0.3478	0.2643	0.3490	0.3078	0.5834
Glove	ARC-II	0.1066	0.1934	0.1385	0.2132	0.2046	0.3951
	DRMM	0.1246	0.2245	0.1317	0.2366	0.2247	0.4935
	Conv-KNRM	0.1356	0.2389	0.1435	0.2487	0.2562	0.5023
	Delta	0.1527	0.2724	0.1503	0.2643	0.2423	0.4738
	IASM-Static	0.2189	0.3439	0.2534	0.3494	0.2924	0.5778
	IASM-Dynamic	0.2474	0.3749	0.2734	0.3678	0.3233	0.5936

2) Impact of Different Word Embedding Systems: We also measure the effects of initialization with different strategies for pre-training the word embeddings described in Section "Pre-trained word embeddings". The results are shown in Table IV. From this 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 interaction-focused models.
- ARC-II, DRMM, Conv-KNRM, and Delta rely more on pre-trained word embeddings compared to our proposed IASM models.
- IASM models outperform ARC-II, DRMM, Conv-KNRM, and Delta even with randomly initialized word embeddings, which indicates its ability to learn relevance through the local context of source and target pair.

The reason why pre-trained word embeddings affect less on our IASM-Dynamic is that our model learns new representations of source and target texts through interactive attention with the matching matrix and updates the matching matrix to model local context. These two aspects provide training of the relevance between words through the local context of source and target texts.

3) OOV Analysis: To better understand the effectiveness of our model in dealing with OOV keywords, we perform analysis on the samples which contain Out-of-Vocabulary words (OOV). Specifically, we partition samples in the test set into four subsets: in-vocabulary words (IV), out-of-training-vocabulary words (OOTV), out-of-embedding-vocabulary words (OOEV) and out-of-both-vocabulary words (OOBV). A sample in the test set belongs to IV if all words in this sample (both source and target texts) appear in both the training and embedding vocabulary. A sample in the test set belongs to OOBV if there is at least one word in this sample (both source and target texts) neither appear in training or in the embedding vocabulary. OOTV samples are the ones which have words excluded in the training set but included in embedding vocabulary, while OOEV samples are the ones that have words included in embedding vocabulary but excluded in the training set. Note that we only count nouns and verbs in each sample for partition. Table V presents the statistics of the partition on each corpus. The embedding we used is pre-trained word embeddings described in Section "Pre-trained Word Embeddings".

TABLE V
STATISTICS OF THE PARTITION ON EACH TEST SET.

	PubMed	TNL	YodaQA
IV	1321	454	83
OOTV	1747	153	26
OOEV	1463	27	11
OOBV	531	3	9

Table VI illustrates the performance of our models on each subset of different corpora. The best performance appears on IV of both three corpora for IASM-Static and IASM-Dynamic. IV is the subset with samples that have all words appear in both the training and embedding vocabulary. This demonstrates that pre-trained word embedding and training are both critical for semantic text matching. The comparison of the performances on OOTV and OOEV shows that out-of-embedding-vocabulary words are more challenging for text semantic matching. By comparing performances of IASM-Static and IASM-Dynamic on different OOV subsets, it is apparent that IASM-Dynamic is more powerful to improve performance for samples with OOV words. It is reasonable that dynamic interactive attention in IASM-Dynamic can fully make use of very limited local information of OOV words to learn their semantics.

4) Alleviating Sparse Matching Matrix Problem: One advantage of our IASM-Dynamic comparing with interaction-

TABLE VI
COMPARISON OF PERFORMANCE OF OUR MODEL ON DIFFERENT SUBSETS OF OOV.

	PubMed		TNL		YodaQA	
	P@1	MAP	P@1	MAP	P@1	MAP
IASM-Static						
IV	0.3739	0.4961	0.2885	0.3843	0.3976	0.5336
OOTV	0.1723	0.1947	0.1895	0.2634	0.1923	0.3232
OOEV	0.052	0.074	0.037	0.076	0	0.011
OOBV	0.009	0.012	0	0.001	0	0.01
IASM-Dynamic						
IV	0.4005	0.5396	0.3061	0.5005	0.4096	0.5772
OOTV	0.1866	0.2134	0.2091	0.3724	0.2308	0.4127
OOEV	0.084	0.096	0.074	0.1141	0.091	0.127
OOBV	0.022	0.0311	0	0.001	0.1111	0.1132

TABLE VII
WHAT PERCENTAGE OF SAMPLES GET SMOOTHER MATCHING MATRIX AFTER ROUNDS OF INTERACTIVE ATTENTION.

	Training Samples	Testing Samples
PubMed	91.36%	82.43%
TNL	85.17%	75.62%
YodaQA	88.34%	84.67%

focused model baselines is that our IASM-Dynamic could alleviate the sparse matching problem. To verify the real effect of our model in smoothing the sparse matching matrix, we need to check the sparsity of the matching matrix before and after interactive attention. To this end, we separately compute the variance value of all entries in the initial matching matrix and the last updated matching matrix for each sample in training and testing sets. We compare the two variance values of these two matrices for each dataset and present the results in Table VII. From this table, we can see 91.36% of training samples and 82.43% of testing samples in PubMed, 85.17% of training samples and 75.62% of testing samples in TNL, 88.34% of training samples and 84.67% of testing samples in YodaQA have much smaller variances of the updated matching matrices than the initial matching matrices.

Besides, we divide PubMed testing samples into eleven groups according to the number of shared nonstop words in real source and target pairs. As shown in Figure 4, we check the performance of models on semantic text matching in these eleven groups to verify the power of our IASM-Dynamic in sparse matching cases. The results clearly indicate that our IASM-Dynamic has promising performance in those samples in which source and target texts share very few nonstop words.

G. Case Study

Last but not least, we conduct three case studies to better understand the power of the IASM-Dynamic. As a result of information exchange between source and target texts, the IASM-Dynamic could find those target texts which are semantically and implicitly related to the corresponding source text. Here are three examples in which the target text is placed at the top one by IASM-Dynamic but are placed at rank n ($n \gg 1$) by baselines. The bold words are keywords in texts, which are either low-frequency/OOV words or implicitly related to corresponding objects in their specific pairwise contexts.

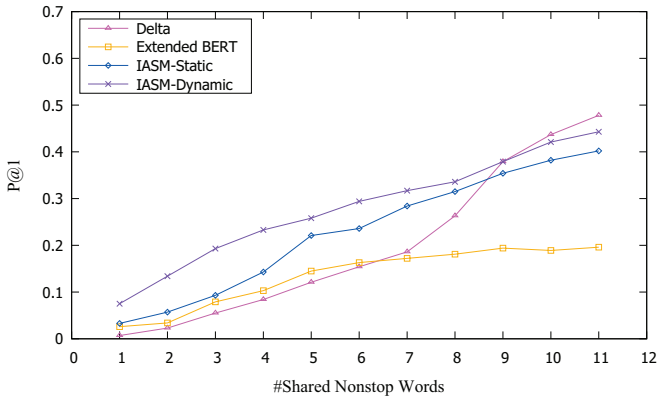

Fig. 4. Compassion of different models for different numbers of shared nonstop words between source and target texts.

- PubMed. *Query*: "**Neonatal chest** drain insertion–an animal model." *Document*: "Trainees rarely see a **pneumothorax** in the **newborn** because of the combination of decreased doctors' hours, the use of surfactant, and modern ventilator techniques ...."
- TNL. *Tweet*: "A modicum of progress: RT @cnnbrk: Saudi King Abdullah **decrees** currently **male-dominated** Council be at least 20% women." *News*: "Saudi Arabia's King Abdullah has appointed 30 women to the previously **all-male** consultative Shura Council."
- YodaQA. *Question*: "What does the **bugler play** at the end of the day on a US *military base*?" *Answer*: "The most widely circulated one states that a Union **Army** infantry officer, whose name often is given as Captain Robert Ellicombe, first ordered **"Taps" performed** at the funeral of his son, a **Confederate soldier** killed during the Peninsula Campaign."

From these examples, it is clear to see that IASM-Dynamic is very good at solving those sparse matching and implicit relevance between source and target texts.

IV. Related Work

Semantic text matching models the relevance/similarity of a pair of texts. It is intuitive for traditional approaches to measure the similarity by comparing words in texts. For example, Mihalcea et al. [36] computed the word similarity, while Wu et al. [37] exploited the vector space model with the term frequency-inverse document frequency (TF-IDF). However, the bag-of-words representation is usually sparse. Besides, the semantics between individual words are also hard to capture, so these approaches usually obtain unsatisfactory results. Although some studies attempted to leverage the semantics in external resources such as knowledge bases [38] and alleviate data sparseness by Latent Semantic Analysis [39], discrete words still limit traditional approaches.

The recent development of deep learning provides a new opportunity for semantic text matching [40], [41]. The deep learning-based semantic text matching models, such as siamese RNN [11], ARC-I [10], ARC-II [10], MV-LSTM [31], DRMM [13], PACRR [42], Delta [16], Conv-KNRM [15], SMASH RNN [12], BERT-based models [17], have dominated this field. In particular, these models have been focusing on 1) flexible representation learning of source and target texts and 2) measuring the similarity between the source and target texts at different levels. Correspondingly, there are two main categories of deep neural semantic text matching models. One is the representation-focused model, which tries to learn good representations for both source and target with deep neural networks, and then conducts matching between the learned representations. Examples include DSSM [8], C-DSSM [9], ARC-I [10], siamese RNN [11], MASH RNN [12], GRAPHENE [43]. The other is the interaction-focused model, which first builds interactions (i.e., matching matrix) between the source and target texts, and then uses deep neural networks to learn the overall matching score with the interactions. Examples include DeepMatch [44], ARC-II [10], DRMM [13], ESR [14], PACRR [42], Conv-KNRM [15] and Delta [16].

Interaction-focused models can alleviate the semantic gaps between words in source and target texts through encoding word-level similarities. Nevertheless, representation-focused models could not directly use the word-level similarities between source and target texts. Therefore, interaction-focused models usually perform better [13], [34], [35]. However, existing interaction-focused models are not aware of the sparse matching problem, which is very common in domain-sensitive text matching, such as biomedical literature retrieval. Besides, existing interaction-focused models neglect the local context of each specific source and target pair. The local context usually encodes the particular relevance between words in source and target texts, which is useful for OOV and low-frequency keywords. This paper proposes the IASM model and tries to explore these two problems.

Another type of method related to our proposed model is attention-based machine comprehension models [32], [33], [45]–[48]. However, our model has different operations on matching matrix compared to existing attention-based machine comprehension models. Attention mechanisms in existing machine comprehension model models (e.g., BIDAF [33], attention-over-attention [32], DrQA [48]) arrange weight distribution on query words and document words by operating column-wise and row-wise summation/average upon matching matrix. We apply interactive attention upon matching matrix to take as input original representations of source and target and output new representations. On the one hand, the interactive attention enables our model to encode fine-granularity interactive information between each word in the source text and each word in the target text. The fine-granularity interactive information is necessary to enrich representations of low-frequent and professional terms like biomedical concepts in biomedical literature [49]. On the other hand, the contribution of each word in the document for updating representations of source and target texts is unlearnable with the pre-trained word embeddings in existing attention-based reading comprehension models [32], [33], [45]–[48]. However, this is learnable in our model via weight matrices W_q in $f_q(A^T Q W_q)$ and W_d in $f_d(ADW_d)$ (see Figure 2). It is beneficial to enrich representations of those low-frequency and professional terms in source and target texts (e.g., biomedical concepts in biomedical articles and queries). Furthermore, the matching matrix in our model could be built from external knowledge, such as WordNet. Our model for semantic matching is designed to encode local and global information simultaneously. However, the attention-based machine comprehension model is designed only for modeling the local context.

V. Conclusion

In this paper, we proposed novel interactive attention network models, i.e., IASM-Static and IASM-Dynamic, for semantic text matching. Our models learn new representations of source and target texts through interactive attention with matching matrix between the two texts. IASM-Static utilized a static matching matrix and IASM-Dynamic updates the matching matrix accordingly. Therefore, IASM-Dynamic not only enriches the representation of source and target texts on top of their original representations, but also encodes the local word-level relevance in the dynamic matching matrix. All aspects benefit alleviating the sparse matching problem in semantic text matching. We conducted empirical evaluations of our models over three different data sets to analyze 1) the impact of different components; 2) the impact of different pre-trained word embedding systems; 3) the superiority of our IASM-Dynamic for OOV words; 4) the power of the IASM-Dynamic to alleviate the sparse matching problem. The experimental results indicated that our model: (i) dramatically improves on the previous state-of-the-art models by large margins on three data sets; (ii) is much more stable for the different pre-trained word embedding models; (iii) can better make use of very limited local information to learn semantics for OOV words; (iv) considerably alleviates the sparse matching problem, leading to better performance. The capability of our model in alleviating the sparse matching problem and learning OOV words through dynamic interac-

tive attention has considerable practical benefits for various interesting NLP applications, such as reading comprehension, machine translation and dialog, which can lead to a broad spectrum of future research.

ACKNOWLEDGEMENT

The work is partially supported by National Science Foundation under grant number 1750326 and 2027970, and Office of Naval Research under grant number N00014-18-1-2585.

REFERENCES

[1] S. Zhao, C. Su, Z. Lu, and F. Wang, "Recent advances in biomedical literature mining," *Briefings in Bioinformatics*, 2020.
[2] S. Zhao, M. Jiang, Q. Yuan, B. Qin, T. Liu, and C. Zhai, "Contextcare: Incorporating contextual information networks to representation learning on medical forum data." in *IJCAI*, 2017, pp. 3497–3503.
[3] S. Zhao and F. Wang, "Biomedical evidence generation engine," *arXiv preprint arXiv:1911.06146*, 2019.
[4] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Eleventh annual conference of the international speech communication association*, 2010.
[5] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[6] Y. Kim, "Convolutional neural networks for sentence classification," in *EMNLP*, Oct. 2014, pp. 1746–1751.
[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.
[8] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck, "Learning deep structured semantic models for web search using clickthrough data," in *CIKM*. ACM, 2013, pp. 2333–2338.
[9] J. Gao, P. Pantel, M. Gamon, X. He, and L. Deng, "Modeling interestingness with deep neural networks," in *EMNLP*, 2014, pp. 2–13.
[10] B. Hu, Z. Lu, H. Li, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," in *Advances in neural information processing systems*, 2014, pp. 2042–2050.
[11] J. Mueller and A. Thyagarajan, "Siamese recurrent architectures for learning sentence similarity," in *AAAI*, 2016.
[12] J.-Y. Jiang, M. Zhang, C. Li, M. Bendersky, N. Golbandi, and M. Najork, "Semantic text matching for long-form documents," 2019.
[13] J. Guo, Y. Fan, Q. Ai, and W. B. Croft, "A deep relevance matching model for ad-hoc retrieval," in *CIKM*. ACM, 2016, pp. 55–64.
[14] C. Xiong, R. Power, and J. Callan, "Explicit semantic ranking for academic search via knowledge graph embedding," in *WWW*, 2017, pp. 1271–1279.
[15] Z. Dai, C. Xiong, J. Callan, and Z. Liu, "Convolutional neural networks for soft-matching n-grams in ad-hoc search," in *WSDM*, 2018, pp. 126–134.
[16] S. Mohan, N. Fiorini, S. Kim, and Z. Lu, "A fast deep learning model for textual relevance in biomedical information retrieval," in *WWW*, 2018, pp. 77–86.
[17] Y. Qiao, C. Xiong, Z. Liu, and Z. Liu, "Understanding the behaviors of bert in ranking," *arXiv preprint arXiv:1904.07531*, 2019.
[18] S. Zhao, Q. Wang, S. Massung, B. Qin, T. Liu, B. Wang, and C. Zhai, "Constructing and embedding abstract event causality networks from text snippets," in *WSDM*, 2017, pp. 335–344.
[19] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *NAACL*, 2003, pp. 173–180.
[20] S. Zhao, T. Liu, S. Zhao, and F. Wang, "A neural multi-task learning framework to jointly model medical named entity recognition and normalization," in *AAAI*, vol. 33, 2019, pp. 817–824.
[21] J. Xu, C. Deng, X. Gao, D. Shen, and H. Huang, "Predicting alzheimer's disease cognitive assessment via robust low-rank structured sparse model," in *IJCAI: proceedings of the conference*, vol. 2017. NIH Public Access, 2017, p. 3880.
[22] J. Xu and F. Wang, "Federated learning for healthcare informatics," *arXiv preprint arXiv:1911.06270*, 2019.
[23] K. Roberts, D. Demner-Fushman, E. M. Voorhees, W. R. Hersh, S. Bedrick, A. J. Lazar, and S. Pant, "Overview of the trec 2017 precision medicine track," *NIST Special Publication*, pp. 500–324, 2017.
[24] M. Sun, S. Zhao, C. Gilvary, O. Elemento, J. Zhou, and F. Wang, "Graph convolutional networks for computational drug development and discovery," *Briefings in bioinformatics*, vol. 21, no. 3, pp. 919–935, 2020.
[25] J. Xu, Z. Xu, P. Walker, and F. Wang, "Federated patient hashing." in *AAAI*, 2020, pp. 6486–6493.
[26] W. Guo, H. Li, H. Ji, and M. Diab, "Linking tweets to news: A framework to enrich short text data in social media," in *ACL*, 2013, pp. 239–249.
[27] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.
[28] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *EMNLP*, 2014, pp. 1532–1543.
[29] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *ICCV*, 2015, pp. 1026–1034.
[30] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.
[31] S. Wan, Y. Lan, J. Guo, J. Xu, L. Pang, and X. Cheng, "A deep architecture for semantic matching with multiple positional sentence representations," in *AAAI*, 2016.
[32] Y. Cui, Z. Chen, S. Wei, S. Wang, T. Liu, and G. Hu, "Attention-over-attention neural networks for reading comprehension," in *ACL*, 2017, pp. 593–602.
[33] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, "Bidirectional attention flow for machine comprehension," *arXiv preprint arXiv:1611.01603*, 2016.
[34] Y. Zhang, M. M. Rahman, A. Braylan, B. Dang, H.-L. Chang, H. Kim, Q. McNamara, A. Angert, E. Banner, V. Khetan *et al.*, "Neural information retrieval: A literature review," *arXiv preprint arXiv:1611.06792*, 2016.
[35] B. Liu, T. Zhang, D. Niu, J. Lin, K. Lai, and Y. Xu, "Matching long text documents via graph convolutional networks," *arXiv preprint arXiv:1802.07459*, 2018.
[36] R. Mihalcea, C. Corley, C. Strapparava *et al.*, "Corpus-based and knowledge-based measures of text semantic similarity," in *AAAI*, vol. 6, no. 2006, 2006, pp. 775–780.
[37] H. C. Wu, R. W. P. Luk, K. F. Wong, and K. L. Kwok, "Interpreting tf-idf term weights as making relevance decisions," *ACM Transactions on Information Systems*, vol. 26, no. 3, p. 13, 2008.
[38] G. Tsatsaronis, I. Varlamis, and M. Vazirgiannis, "Text relatedness based on a word thesaurus," *Journal of Artificial Intelligence Research*, vol. 37, pp. 1–39, 2010.
[39] W.-t. Yih, K. Toutanova, J. C. Platt, and C. Meek, "Learning discriminative projections for text similarity measures," in *CoNLL*. Association for Computational Linguistics, 2011, pp. 247–256.
[40] S. Wan, Y. Lan, J. Xu, J. Guo, L. Pang, and X. Cheng, "Matchsrnn: Modeling the recursive matching structure with spatial rnn," *arXiv preprint arXiv:1604.04378*, 2016.
[41] L. Pang, Y. Lan, J. Guo, J. Xu, S. Wan, and X. Cheng, "Text matching as image recognition," in *AAAI*, 2016.
[42] K. Hui, A. Yates, K. Berberich, and G. de Melo, "PACRR: A position-aware neural IR model for relevance matching," in *EMNLP*, Sep. 2017, pp. 1049–1058.
[43] S. Zhao, C. Su, A. Sboner, and F. Wang, "Graphene: A precise biomedical literature retrieval engine with graph augmented deep learning and external knowledge empowerment," in *CIKM*, 2019, pp. 149–158.
[44] Z. Lu and H. Li, "A deep architecture for matching short texts," in *Advances in neural information processing systems*, 2013, pp. 1367–1375.
[45] A. W. Yu, D. Dohan, M.-T. Luong, R. Zhao, K. Chen, M. Norouzi, and Q. V. Le, "Qanet: Combining local convolution with global self-attention for reading comprehension," *arXiv preprint arXiv:1804.09541*, 2018.
[46] C. Xiong, V. Zhong, and R. Socher, "Dynamic coattention networks for question answering," *arXiv preprint arXiv:1611.01604*, 2016.
[47] Y. Shen, P.-S. Huang, J. Gao, and W. Chen, "Reasonet: Learning to stop reading in machine comprehension," in *KDD*, 2017, pp. 1047–1055.
[48] D. Chen, A. Fisch, J. Weston, and A. Bordes, "Reading Wikipedia to answer open-domain questions," in *ACL*, 2017, pp. 1870–1879.
[49] S. Zhao, "Mining medical causality for diagnosis assistance," in *WSDM*, 2017, pp. 841–847.