Socialformer: Social Network Inspired Long Document Modeling for Document Ranking

Yujia Zhou[2], Zhicheng Dou[1,4*], Huaying Yuan[3], and Zhengyi Ma[2]
[1]Gaoling School of Artificial Intelligence, Renmin University of China
[2]School of Information, Renmin University of China
[3]College of Computer Science, Nankai University
[4]Beijing Key Laboratory of Big Data Management and Analysis Methods
{zhouyujia,*dou}@ruc.edu.cn

ABSTRACT

Utilizing pre-trained language models has achieved great success for neural document ranking. Limited by the computational and memory requirements, long document modeling becomes a critical issue. Recent works propose to modify the full attention matrix in Transformer by designing sparse attention patterns. However, most of them only focus on local connections of terms within a fixed-size window. How to build suitable remote connections between terms to better model document representation remains underexplored. In this paper, we propose the model Socialformer, which introduces the characteristics of social networks into designing sparse attention patterns for long document modeling in document ranking. Specifically, we consider several attention patterns to construct a graph like social networks. Endowed with the characteristic of social networks, most pairs of nodes in such a graph can reach with a short path while ensuring the sparsity. To facilitate efficient calculation, we segment the graph into multiple subgraphs to simulate friend circles in social scenarios. Experimental results confirm the effectiveness of our model on long document modeling.

CCS CONCEPTS

• **Information systems** → **Retrieval models and ranking**.

KEYWORDS

Document ranking; Social network; Long document modeling

ACM Reference Format:
Yujia Zhou, Zhicheng Dou, Huaying Yuan, and Zhengyi Ma. 2022. Socialformer: Social Network Inspired Long Document Modeling for Document Ranking. In *Proceedings of the ACM Web Conference 2022 (WWW '22), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3485447.3511962

1 INTRODUCTION

Document ranking is a crucial task in information retrieval. It focuses on generating an ordered document list in response to the user's query. In recent years, pre-trained language models, such as BERT [9], have made impressive progress in natural language processing and information retrieval. Its powerful contextual representation learning ability is suitable for modeling documents in semantic space and has been widely applied in information retrieval [7, 24, 46]. However, due to the memory constraints of quadratic attention matrix, the input sequence of the BERT-based model is limited to 512 tokens [2]. Therefore, how to apply BERT over long documents remains a challenge for document ranking.

To deal with this problem, some early studies [7, 15, 46] divide a document into multiple passages with fixed-size windows, match the query with each passage, and aggregate passage-level relevance signals for document ranking. These works concentrate on the semantic modeling in each passage, but ignore the word level interactions between passages. This prevents the model from learning a global document representation. Subsequently, another group of works proposes handling long document with sparse attention patterns in Transformer [33], such as sliding window attention [3, 16, 44, 47], dilated window attention [3, 50], and global attention [3, 44]. These patterns enlarge the receptive field of each term to interact with more distant terms while ensuring the sparsity.

Previous methods have made great progress in reducing complexity of self-attention layer with sparse connections. Most of them mainly concentrate on building local connections of terms to model semantic dependencies inside a fixed-size sliding window. However, in these methods, remote connections between terms are ignored or captured by simple patterns [3, 44]. In fact, based on small-world theory [21, 34], suitable remote connections in a sparse social network can shorten the path between most pairs of nodes. Bigbird [44] first introduces the small-world graph in building remote connections with random attention. However, in the real social network, the formation of remote connection between people is not random, but is related to the distance between them and their status in social networks. Inspired by it, **we attempt to leverage the characteristics of social networks to build well designed remote connections of terms within a long document**. With the social network inspired graph, we are able to achieve better information propagation ability and finally yield better document representations for Web document ranking.

Social networks have been thoroughly investigated by many researchers. There are three main characteristics to ensure the effective transmission of information [11, 17, 32]. **(1) Randomness**. Any two people have a certain probability to establish a new contact. **(2) Distance-aware**. In a small-world network, the probability of constructing connection between two people should follow

WWW '22, April 25–29, 2022, Virtual Event, Lyon, France

ACM ISBN 978-1-4503-9096-5/22/04…$15.00
https://doi.org/10.1145/3485447.3511962

the inverse square law with distance [21]. **(3) Centrality**. Some celebrities possess more influence over social networks and are usually connected with more people. These characteristics ensure that even if the graph is sparse, most pairs of nodes can reach with a short path. This will assure the efficient information exchange over the sparse social network. Inspired by these characteristics, **we propose a similar paradigm to form the sparse attention matrix in long document modeling**. Different from traditional fixed attention patterns, all connections between words are sampled according to the probability. This enables us to dynamically adjust the edge distribution based on the document length and content. The calculation of the probability follows the characteristics of social networks, which take the word distance and word centrality into account. Under such a strategy, the graph we construct imitating social networks can enhance the information transmission in the document while ensuring the sparsity.

Due to the randomness of our probability sampled graph, how to achieve fast calculation becomes a new challenge. The reason why previous sparse attention matrices can handle long documents is that they can be easily split into multiple small self-attention blocks. To facilitate calculation, **we propose to segment the graph into multiple subgraphs**, while retaining as much information as possible. In social networks, the relationships between people usually depend on friend circles [10, 13], and there is often a central person in each circle [52]. Based on this observation, we propose a graph partition algorithm which focuses on finding central nodes in the graph. According to these central nodes, we are able to form multiple subgraphs simulating the friend circles in social scenarios.

In general, members within a friend circle are connected through strong ties with frequent interactions [13, 18]. By contrast, the interactions between friend circles through weak ties are relatively infrequent [35]. To model such interactions, we propose a two-stage method for information transmission. At the first stage, the **intra-circle interaction** is applied to model semantic dependency between terms within each subgraph. Second, for information transmission between circles, we carry out the **inter-circle interaction** on central nodes of multiple subgraphs. Under such a strategy, most pairs of words in the document can transmit information via a direct connection or multiple iterative stacking blocks.

More specifically, we propose Socialformer, a social network inspired long document modeling method for document ranking. Socialformer is composed of four steps. **First**, based on the characteristics of social networks, we design four sparse attention patterns to construct a graph with probability sampling. **Second**, we present two friend circle based strategies of graph partition to reduce the memory and computational complexity. **Third**, we devise a two-stage information transmission model to capture the interactions between terms with the augmented transformer structure. **Finally**, by aggregating the representations of passages and subgraphs, a comprehensive document representation is formed for document ranking. We conduct experiments on the widely used TREC DL and MS MARCO dataset [5] for document ranking. Experimental results show that our proposed model Socialformer significantly outperforms existing document ranking models.

The contributions are summarized as follows. (1) We introduce the social network theories into long document modeling, which provides a theoretical basis for enhancing information transmission in long documents. (2) Inspired by the characteristics of social networks, we devise several social-aware sparse attention patterns to build the graph with probability sampling. (3) To reduce complexity, a graph partition algorithm is proposed referring to the concept of friend circles in social networks. (4) We apply a two-stage information transmission model to achieve intra-circle and inter-circle interactions with the augmented transformer.

2 RELATED WORK

Passage-level Document Ranking. A major disadvantage of the Transformer [33] based models is that they cannot handle long documents due to the quadratic memory complexity. Inspired by using passage-level evidence for document ranking [4], an intuitive idea is to segment the document text into multiple small chunks, compare the query to all passages [1, 7, 15, 25, 27, 31], then aggregate the information of each passage [22, 41]. Some early studies focus on combining passage ranking scores with different strategies. Dai and Callan [7] devised three ways (MaxP, FirstP, SumP) to get the document ranking scores from all passage-level scores. Hofstätter et al. [15] proposed an intra-document cascade ranking model with knowledge distillation to speed up selecting passages. However, these methods ignore the information transfer between passages. This will prevent the model from learning a global document representation. To deal with this problem, several representation aggregation methods were proposed to learn the global document embeddings. Wu et al. [37] used LSTM [14] to model the sequential information hidden in passages. Li et al. [22] tried a series of representation aggregation strategies with max pooling, attention pooling, transformer, etc. In order to further strengthen the information transfer between passages, some hierarchical transformer structures were proposed to model intra-passage and inter-passage interactions [36, 39, 40, 48]. To further learn the document embedding with a global view, some studies use iterative attention blocks to enlarge the receptive field of each term layer by layer, such as Transformer-XL [8] and Transformer-XH [49]. Although these efforts have achieved certain success, how to guide information propagation in a reasonable manner still remains underexplored.

Long-Document Transformers. Another idea to solve the problem of long document representation is to design sparse attention patterns [3, 20, 30, 51], so as to avoid computing the full quadratic attention matrix multiplication. One of the most intuitive attention patterns is the sliding window attention [3, 16, 28, 44, 47], which only keeps links to surrounding terms. Moreover, dilated sliding window [3, 47, 50] was devised to further increase the receptive field without additional computational costs. To fit the specific tasks, several works proposed to use the global attention [2, 3, 12, 44] to highlight the influence of certain tokens. In the field of information retrieval, query terms are usually set as global tokens to attend to all tokens [19]. To model the document structure, some graph-based transformer methods [42, 43] were presented to lower computational costs. However, any two words in the document should have a probability to be connected [34]. To implement this idea, Zaheer et al. [44] applied random attention to construct the sparse attention matrix, which brought significant improvement compared to structured patterns. In this paper, we integrate the social network theory to build remote edges for long document

modeling. Our model can enhance the information transmission and learn comprehensive representations for document ranking.

3 METHODOLOGY

Document ranking has become an indispensable component in many search engines. Recently, BERT-based models are applied to encode documents with deeper text understanding. For longer document texts, previous studies devise various long-document transformers with sparse attention patterns to reduce the complexity. However, most of them only pay more attention to local connections of terms. Inspired by social networks, we argue that remote edges between terms are crucial for effective information transmission across the whole document. The overview of Socialformer is shown in Figure 1. To build a graph with reasonable remote edges, we incorporate the characteristics of social networks considering the influence of word distance and word centrality. To facilitate calculations, we segment the whole graph into multiple subgraphs according to the characteristics of friend circles. Then, we design a two-stage information transmission method to simulate the information flow in social scenarios. In the remaining part of this section, we will introduce the details.

3.1 Social Network based Graph Construction

As we stated in Section 1, the characteristics of social networks (*i.e.*, randomness, distance-aware, and centrality) ensure that most pairs of nodes in the sparse graph can reach with a short path. Efficient information transmission and sparsity are in line with our needs of designing attention patterns. In this section, we will introduce how to combine social networks to construct a graph.

Inspired by the randomness of social networks, we abandon the traditional fixed attention patterns. Instead, we sample the edges according to the social-aware probability. This enables us to construct diverse social networks for documents. To calculate the probability, we take the word distance and word centrality into account. In addition to the static probability which is only related to the document, we also consider the dynamic probability in response to the specific query. In fact, facing different queries, the contribution of each word in the document should not be the same. As shown in Figure 1, there are four social-aware attention patterns we designed to compute the probability matrix. They are:

Static Distance. In addition to local connections, Watts and Strogatz [34] believed that remote edges are necessary for information transmission. They proposed Watts-Strogatz model to randomly sample remote edges, which is applied to long document modeling by BigBird [44]. However, Kleinberg [21] pointed out that the random strategy does not match the real social scenarios. They claimed the probability two people are connected usually follows the inverse square law with their distance. Inspired by this, we argue that this rule is also in line with the long document modeling. The further the distance between two words, the lower the probability that they have semantic dependence. Formally, given a document d with length l, denoted as $d = \{t_1, \cdots, t_l\}$, The static distance based probability of establishing an edge between tokens t_i and t_j is:

$$P_{\mathrm{sd}}(i,j) = \frac{1}{(1+|i-j|/p)^2}, \tag{1}$$

where p is a hyper-parameter to control the probability range and is set to 50 in experiments.

Static Centrality. In social networks, some celebrities usually have connections with more people and have greater influence. Similarly, each word in the document has a different contribution to expressing the semantics of the document. We attempt to extract the "celebrities" in the document and highlight their influence. We choose a common indicator, TF-IDF weights, to indicate the static centrality of each word, denoted as $\{w_1^{\mathrm{sc}}, \cdots, w_l^{\mathrm{sc}}\}$. The static centrality based probability $P_{\mathrm{sc}}(i,j)$ is related to the weights of tokens t_i and t_j. We have:

$$P_{\mathrm{sc}}(i,j) = f(w_i^{\mathrm{sc}} \cdot w_j^{\mathrm{sc}}), \tag{2}$$

where the function $f(\cdot)$ is used to map the weight product to probability. It consists of a smoothing layer and a normalization layer:

$$\begin{aligned} s(i,j) &= \mathrm{smooth}(w_i \cdot w_j), \\ f(i,j) &= \frac{s(i,j) - \min s(i,j)}{\max s(i,j) - \min s(i,j)}, \end{aligned} \tag{3}$$

where $\mathrm{smooth}(\cdot)$ is the smoothing function, which is implemented by $\mathrm{sqrt}(\cdot)$ in experiments. It can be replaced by more sophisticated methods in the future.

Dynamic Distance. Given the query q, we assume that query terms contained in the document are more critical for document modeling, and their surrounding words in the document are usually more informative for the query. Formally, we extract the words that exactly match the query in the document as a set, denoted as $\{t_1^q, t_2^q, \cdots, t_n^q\}$. The weight of i-th document word w_i^{dd} is related to the distance to these head words. We have:

$$w_i^{\mathrm{dd}} = \frac{1}{n}\sum_{j=1}^{n} \frac{1}{1+|i-\mathrm{pos}(t_j^q)|/p}, \tag{4}$$

where $\mathrm{pos}(\cdot)$ is to compute the original position in the document, and p is the same hyper-parameter as in Eq. (1). The computing of dynamic distance based probability is similar to above:

$$P_{\mathrm{dd}}(i,j) = f(w_i^{\mathrm{dd}} \cdot w_j^{\mathrm{dd}}). \tag{5}$$

Dynamic Centrality. Some infrequent words will play an important role in semantics when matching with the query. Concretely, BERT-based model has performed well on document ranking task, and attention weights at the special token '[CLS]' position can reflect the contribution of each word. However, due to length limitation, we cannot feed all the words of a long document into BERT. To handle this issue, we propose using simple model to select several relevant words and applying BERT model to compute accurate weights of them. At the first stage, cosine similarity is used to determine the relevance of each word in the document to the query. Then, we select top 512 relevant words and feed them into BERT model with following input format:

$$\texttt{[CLS] query [SEP] rel}_1\ \texttt{rel}_2 \cdots\ \texttt{rel}_n\ \texttt{[SEP]}. \tag{6}$$

We replace the cosine similarity weights of relevant words with BERT weights. Similarly, the dynamic centrality based probability is related to the weight of each term:

$$P_{\mathrm{dc}}(i,j) = f(w_i^{\mathrm{dc}} \cdot w_j^{\mathrm{dc}}). \tag{7}$$

Finally, based on these four strategies, we take the weighted average of the four probability matrices $P = \lambda_1 P_{\mathrm{sd}} + \lambda_2 P_{\mathrm{sc}} + \lambda_3 P_{\mathrm{dd}} +$

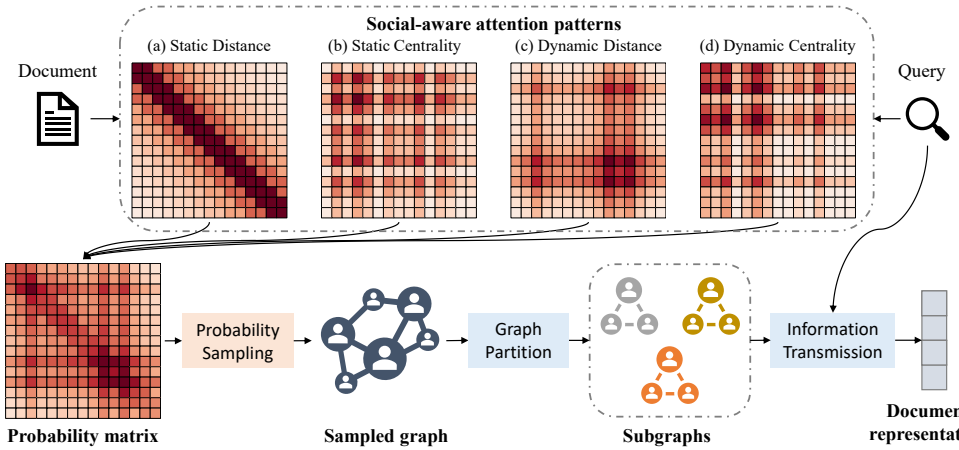


Figure 1: The overview of Socialformer. For a long document, we combine four social-aware attention patterns to sample a token-level graph. Darker color indicates higher probability. The graph partition module and the information transmission module are designed to facilitate calculation. Finally, the global document representation is obtained for ranking.

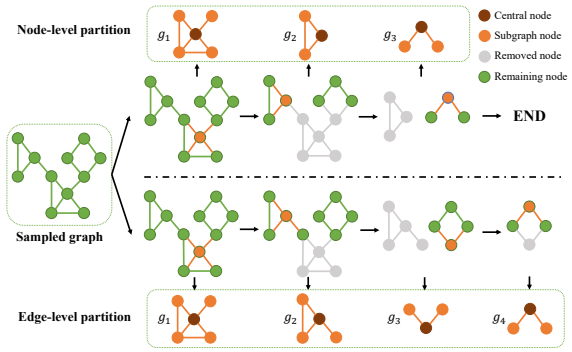


Figure 2: The overview of graph partition. Inspired by friend circles, node-level partition and edge-level partition are devised to segment the graph into multiple subgraphs.

$\lambda_4 P_{\text{dc}}$ for sampling. To control the sparsity of generated graph, we set a hyper-parameter μ to scale probability following:

$$\sum_{i,j} \frac{P_{ij}}{\mu} = l^2 * (1 - sparsity), \tag{8}$$

where l is the max length of documents. The adjacency matrix M of the graph is sampled on the scaled probability matrix P by:

$$M_{ij} = \begin{cases} 1, & \text{if random}(0,1) < P_{ij}; \\ 0, & \text{otherwise}, \end{cases}$$

where random(0,1) means getting a random number from 0 to 1. However, due to the randomness of our sampling strategy, the edges of the sampled graph are unstructured. It is hard to compute the self-attention matrix like traditional attention patterns. To handle this issue, we attempt to divide the whole graph into multiple subgraphs while retaining as much information as possible.

3.2 Graph Partition

The reason why the previous sparse attention patterns can reduce the complexity is that they can be easily split into multiple small self-attention blocks [3, 44]. However, due to the randomness of our sampled graph, the edge distribution is unstructured. We propose to segment it into multiple subgraphs for calculation. We expect these subgraphs to retain as many nodes and edges as possible to minimize the loss of information. To determine the way of graph partition, we refer to another feature of social networks: the relationships between people are usually formed based on friend circles. In social scenarios, the friend circle is a common relationship structure, such as classmates and relatives. One of the characteristics of the friend circle is that there is often one person at the core who is responsible for connecting people in the entire circle [52]. This feature provides us with a way to extract friend circles.

Specifically, we devise two partition strategies as shown in Figure 2: node-level partition and edge-level partition. The former assumes that one node only appears in one subgraph, while the latter allows each node to belong to different subgraphs. In a limited number of subgraphs, node-level strategy can record more node information, while the edge-level strategy retains more edges.

Formally, given the whole graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$, where $\mathcal{N}$ is the set of nodes containing the document words and $\mathcal{E}$ represents connections between words, our goal is to find out top k informational subgraphs. Specifically, we first select the node with the highest degree as the central node of the first subgraph. Then, the central node and its neighboring nodes form the first subgraph g_1. To ensure the distinction between different subgraphs, there are two strategies of partition. For node-level partition, we remove all nodes in the subgraph g_1 from $\mathcal{G}$ and repeat the above process to form other subgraphs. For edge-level partition, we only remove edges in g_1 from $\mathcal{G}$, and some nodes still have a chance to appear in other subgraphs. Finally, we obtain k subgraphs $G = \{g_1, ...g_k\}$, which will act on information transmission in the next section.

3.3 Iterative Information Transmission

In social networks, the connections within a friend circle are usually dense, which are called strong ties [13, 18]. They contribute to person-level interactions and information transmitted through strong ties tends to be redundant. By contrast, weak ties [35] have a

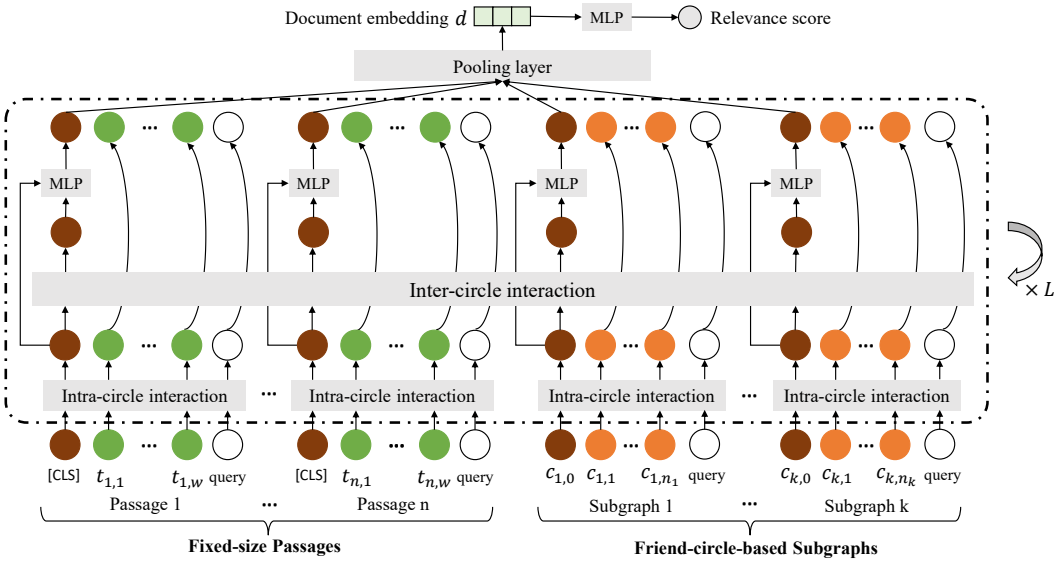


Figure 3: The architecture of information transmission model. Integrating fixed-size passages and social-aware subgraphs, intra-circle and inter-circle interactions are applied to enhance the information transmission. Finally, by aggregating the information of central nodes, we obtain the comprehensive document embedding to compute the relevance score.

greater impact on the group-level spread of information. To imitate this pattern in long documents, as shown in Figure 3, we devise a two-stage information transmission model over all subgraphs and passages, which consists of the intra-circle interaction and the inter-circle interaction. With L iterative stacking blocks, the information between most pairs of nodes can be transmitted to learn a global document embedding. The details are introduced as follows.

Intra-circle Interaction. People who belong to the same friend circle often have certain similarities. Triadic closure theory [10] shows two people in the same circle have higher probability of becoming friends. Based on such observation, we use a fully-connected transformer layer to achieve information transmission in each subgraph. Formally, for the subgraph g_i, assuming it consists of a central node $c_{i,0}$ and n_i neighboring nodes, *i.e.*, $C_i = \{c_{i,0}, c_{i,1}, \cdots, c_{i,n_i}\}$, the intra-circle interaction with low-level transformer is defined as:

$$\mathbf{C}_i^{\text{low}} = \text{Trm}(\{c_{i,0}, c_{i,1}, \cdots c_{i,n_i}, \texttt{query}\}), \tag{9}$$

where $\text{Trm}(\cdot)$ is the transformer encoder. The output of this layer is denoted as $\mathbf{C}_i^{\text{low}} = \{\mathbf{c}_{i,0}^{\text{low}}, \mathbf{c}_{i,1}^{\text{low}}, \cdots, \mathbf{c}_{i,n_i}^{\text{low}}\}$. In the remaining of this section, we use $\mathbf{c}_i^{\text{low}}$ to denote the central node $\mathbf{c}_{i,0}^{\text{low}}$, which represents this circle for inter-circle interaction.

Inter-circle Interaction. Connections between different friend circles can help the information to be transmitted to further places. To promote the semantics of each word in the document to be transmitted to all positions, we design an inter-circle interaction layer over central nodes with a high-level transformer. To preserve the sentence structure information, we integrate fixed-size passages with subgraphs together for information transmission. Assuming that there are m passages and k subgraphs, we take the central nodes of each subgraph and passage (regarding "[CLS]" as the central node) as the input, *i.e.*, $\mathbf{C}^{\text{low}} = \{\mathbf{c}_1^{\text{low}}, \cdots, \mathbf{c}_{k+m}^{\text{low}}\}$. We have:

$$\mathbf{C}^{\text{high}} = \text{Trm}(\{\mathbf{c}_1^{\text{low}}, \cdots, \mathbf{c}_{k+m}^{\text{low}}\}). \tag{10}$$

The output $\mathbf{C}^{\text{high}} = \{\mathbf{c}_1^{\text{high}}, \cdots, \mathbf{c}_{k+m}^{\text{high}}\}$ considers the information transmission across all subgraphs and passages, and will take the global information to its neighboring nodes in the next iteration.

Iterative Stacking Blocks. In order to promote global information to be spread to every node, it is more reasonable to alternate the intra-circle and inter-circle interactions. The overall structure is composed of L stacking blocks. Each block contains an intra-circle and an inter-circle interaction layer. The outputs of two transformer layers will be aggregated as the inputs to the next block:

$$\{\mathbf{c}_{i,j}\}_L = \begin{cases} [\{\mathbf{c}_i^{\text{low}}, \mathbf{c}_i^{\text{high}}\}_{L-1}] \cdot \mathbf{W}^C, & \text{if } c_{i,j} \text{ is the central node;} \\ \{\mathbf{c}_{i,j}^{\text{low}}\}_{L-1}, & \text{otherwise,} \end{cases}$$

where $\mathbf{W}^C \in \mathbb{R}^{2E \times E}$ is the projection matrix. In the iteration of the L layers, central nodes serve as the bridge for global information transmission. The whole process is highly consistent with the exchange of information in social networks.

Aggregation. After L stacking blocks, we aggregate all passages and subgraphs to learn the document embedding with global information. Following prior works, such as PARADE [22], we aggregate the representations corresponding to central nodes by a pooling layer to get the document embedding $\mathbf{d}$, defined as:

$$\mathbf{d} = \text{Pooling}(\{\mathbf{c}_1^{\text{high}}, \cdots, \mathbf{c}_{k+m}^{\text{high}}\}_L), \tag{11}$$

where $\text{Pooling}(\cdot)$ is the aggregation function, which can be implemented by Mean, Max, Attention, Transformer, etc. Since the document embedding has encoded the query information, we can directly compute the relevance score by feeding the document embedding into a linear layer:

$$\text{score}(\mathbf{d}) = \mathbf{v}^{\text{T}}\mathbf{d}, \tag{12}$$

where $\mathbf{v} \in \mathbb{R}^E$ is a linear function to project the document embedding into a scalar score.

3.4 Training

For each query q and a group of documents G_q, we choose the listwise cross entropy as the loss function following [7]:

$$\mathcal{L}^q = -\log \frac{\exp(\text{score}(\mathbf{d}^+))}{\sum_{\mathbf{d} \in G_q} \exp(\text{score}(\mathbf{d}))} \tag{13}$$

where $\mathbf{d}^+$ is the document embedding of positive sample, and $\mathbf{d}$ is the document representation for a document $\mathbf{d} \in G_q$.

4 EXPERIMENTAL SETTINGS

4.1 Datasets and Evaluation Metrics

To prove the effectiveness of our proposed Socialformer, we conduct experiments on the 2019 TREC Deep Learning Track Document Collection [5]. This collection is a large-scale benchmark dataset for web document retrieval. It contains 3.2 million documents with a mean document length of 1,600 words. We conduct experiments on two representative query sets widely used in existing works.

- **MS MARCO Document Ranking (MS MARCO)** [26]: It consists of 367 thousand training queries, and 5 thousand development queries for evaluation. The relevance is rated in 0/1.
- **TREC 2019 Deep Learning Track (TREC DL)** [6]: It replaces the test queries in MS MARCO with a novel set of 43 queries. Although it is smaller than MS MARCO, it has more comprehensive notations with the relevance scored in 0/1/2/3.

We use the official metrics to evaluate the top-ranking results, such as MRR@100 and nDCG@10. Besides, we also report MRR@10 and nDCG@100 for MS MARCO and TREC DL, respectively.

4.2 Baselines

We evaluate the performance of our approach by comparing it with three groups of methods for modeling long documents:

(1) *Traditional IR Models.* **BM25** [29] is a highly effective probabilistic retrieval model based on IDF-weighted counting. **QL** [45] is another famous model which measures the query likelihood of query with Dirichlet prior smoothing.

(2) *Passage-based Models.* These methods firstly split the long documents into multiple passages with the fixed-size window, then use the standard Transformer architecture to predict the relevance of each small passage. **BERT-FirstP** [7] predicts the relevance of each passage with BERT model independently, and uses the score of the first passage to represent the relevance of the whole document. **BERT-MaxP** [7] combines the independent score of each passage with a max-pooling layer to ensemble the global relevance information. **IDCM** [15] is an intra-document cascade ranking model with an efficient passage selection strategy. **PARADE** [22] proposes strategies for aggregating representations of document's passages into a global document embedding and computes the final score.

(3) *Long-document Transformer Models.* These methods handle long document ranking by designing sparse attention patterns in Transformer. **Longformer** [3] combines a local windowed attention with a task motivated global attention. We experiment with its both variants, *i.e.*, standard **Longformer** and **Longformer$_{\text{Global}}$** with global attention. **QDS-Transformer** [19] designs IR-axiomatic structures in transformer self-attention. **BigBird** [44] combines global attention, local attention and random attention together for building a universal framework of sequence encoders.

Our method, which is called **Socialformer**[1], combines the advantages of passage-based models and long-document transformer models. We use **Socialformer$_{\text{node}}$** and **Socialformer$_{\text{edge}}$** to represent the model with two different graph partition strategies.

4.3 Implementation Details

We re-rank the documents from Top100 results retrieved by the advanced retrieval model ANCE [38]. During training, for each query, we choose positive samples and negative samples in a 1:7 ratio. Negative samples are randomly selected from the candidate documents. Considering the balance of time cost and effect, all models are trained for one epoch with a batch size of 8. We use AdamW [23] to optimize the parameters with learning rate of 1e-5. For the model, we use the hyper-parameter μ to control the sparsity of the social graph at about 0.93 level by Eq. (8). The document length and window size are set to 2048 and 128 for experiments, and larger window size does not bring more improvement [15, 19]. To control memory complexity, the max number of subgraphs k is set to 16, which could retain critical nodes or edges for information transmission. We set the number of layers L to 12 and intra-circle interaction layers are initialized by BERT-base model. Considering the time cost, the pooling layer is set to max pooling operation, which is also applied to the baseline model PARADE.

5 EXPERIMENTAL RESULTS

5.1 Overall Results

Experimental results on the MS MARCO and the TREC-DL 2019 datasets are shown in Table 1. Some observations are as follows.

(1) Among all models, our social-aware models outperform all baselines with the same settings in terms of all evaluation metrics. Compared with the best baseline models, our models have significant improvements in both datasets with paired t-test at $p < 0.05$ level. Concretely, for MS MARCO dataset, our best model Socialformer$_{\text{edge}}$ outperforms PARADE by over 2.37% improvement on MRR@100, while the improvement over BigBird is 4.70% on nDCG@10 for TREC DL dataset. These results indicate that introducing the characteristics of social networks into attention patterns can improve the ranking quality.

(2) Comparing different model types, we find that information transmission among passages is effective in learning global document representations. Specifically, PARADE, which aggregates the representations of all passages, outperforms score aggregation methods such as BERT-MaxP. This indicates that the aggregated document representation can alleviate the problem of lack of global information in document embeddings. Moreover, long-document transformer models devise different attention patterns to achieve information transmission between passages, which shows comparable performance. Our model Socialformer refers to the social networks when designing attention patterns, so as to achieve more effective information transmission within the document.

(3) Comparing different versions of Socialformer, it can be observed that longer document input (2048 vs. 512) brings an obvious

[1]The code is available on https://github.com/smallporridge/Socialformer

Table 1: Results of all models on two document ranking benchmarks. "†" denotes the result is significantly better than other models from the same setting in t-test with $p < 0.05$ level. The best results are in bold and the second best results are underlined.

Model Type	Model Name	Doc. Length	Window Size	MS MARCO		TREC DL	
				MRR@100	MRR@10	nDCG@100	nDCG@10
Traditional IR Models	BM25	-	-	0.2538	0.2383	0.4692	0.5411
	QL	-	-	0.2457	0.2295	0.4644	0.5370
Passage-based Models	BERT-FirstP	512	512	0.4321	0.4268	0.4949	0.6202
	BERT-MaxP	512	128	0.4173	0.4088	0.4835	0.6014
	BERT-MaxP	2048	128	0.4326	0.4272	0.4952	0.6215
	IDCM	2048	128	0.4367	0.4280	0.4960	0.6235
	PARADE	2048	128	0.4386	0.4312	0.4975	0.6280
Long-Document Transformer Models	Longformer	2048	128	0.4263	0.4192	0.4942	0.6208
	$\text{Longformer}_{\text{Global}}$	2048	128	0.4381	0.4302	0.4982	0.6292
	QDS-Transformer	2048	128	0.4379	0.4300	0.4988	0.6315
	BigBird	2048	128	0.4385	0.4311	0.4985	0.6318
Our Models	$\text{Socialformer}_{\text{node}}$	512	128	$0.4290^{\dagger}$	$0.4231^{\dagger}$	$0.4902^{\dagger}$	$0.6084^{\dagger}$
	$\text{Socialformer}_{\text{edge}}$	512	128	$0.4313^{\dagger}$	$0.4258^{\dagger}$	$0.4950^{\dagger}$	$0.6212^{\dagger}$
	$\text{Socialformer}_{\text{node}}$	2048	128	$0.4483^{\dagger}$	$0.4402^{\dagger}$	$0.5087^{\dagger}$	$0.6534^{\dagger}$
	$\text{Socialformer}_{\text{edge}}$	2048	128	$\mathbf{0.4490}^{\dagger}$	$\mathbf{0.4411}^{\dagger}$	$\mathbf{0.5119}^{\dagger}$	$\mathbf{0.6615}^{\dagger}$

improvement in results. This conclusion can also be drawn on BERT-MaxP. This reveals longer context contains more useful information to understand the semantics of documents. Moreover, the performances of $\text{Socialformer}_{\text{node}}$ and $\text{Socialformer}_{\text{edge}}$ are similar for 2048 document length, but when we limit the input length to 512, $\text{Socialformer}_{\text{edge}}$ demonstrates greater superiority. This indicates that when the number of nodes in the social graph is relatively small, edge-level partition retains much information.

In summary, the experimental results show that **introducing the characteristics of social networks into designing sparse attention patterns is conducive to refinement of document representations in long document modeling**.

5.2 Effects of Social-aware Attention Patterns

In the process of generating the graph, we calculate the probability matrix from the dynamic-static and distance-centrality dimensions respectively. To verify the necessity of each of our attention patterns, we explore the role of each strategy, including probability matrices of static distance, static centrality, dynamic distance, and dynamic centrality. In order to directly observe the effect of each attention pattern, we visualize the adjacency matrix generated by each strategy with 0.9 sparsity. As shown in Figure 4, each strategy highlights different parts of the adjacency matrix to model relations between words. For further analysis, we remove one strategy at a time to observe the impact on the MS MARCO dataset. In addition, we use random sampling that the probability of establishing each edge is equal for comparison.

The results are shown in Table 2. We find that the removal of each attention patterns will damage the results on all evaluation metrics. Concretely, deleting the dynamic patterns causes the most obvious impact on performance. This indicates that building the semantic dependencies of documents based on query is more helpful for learning global document representation. Meanwhile, the static patterns also make some contributions to the results. The four strategies work together to build a graph like social networks in the document. Additionally, using the strategy of randomly building edges instead of our attention patterns causes a severe drop on the results. This shows that using the characteristics of social networks can promote the transfer of information in documents. After removing the social features, the model mainly carries out two-stage information transmission through passages, which has similar performance to PARADE.

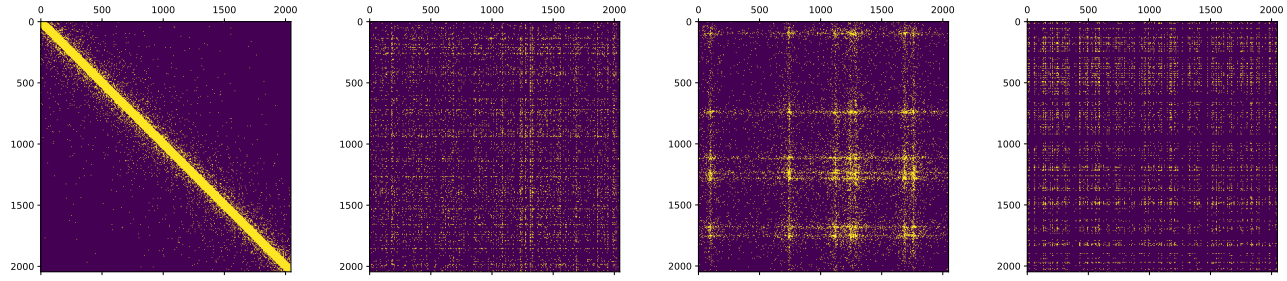
Table 2: Performance of ablation studies of attention patterns on MS MARCO dataset.

Model	MRR@100		MRR@10	
PARADE	0.4382	-2.41%	0.4302	-2.47%
$\text{Socialformer}_{\text{edge}}$	0.4490	-	0.4411	-
w/o. static distance	0.4469	-0.47%	0.4380	-0.70%
w/o. static centrality	0.4478	-0.27%	0.4392	-0.43%
w/o. dynamic distance	0.4447	-0.96%	0.4359	-1.18%
w/o. dynamic centrality	0.4450	-0.89%	0.4364	-1.06%
random edges	0.4398	-2.05%	0.4320	-2.06%

5.3 The Effect of Sparsity on Graph Partition

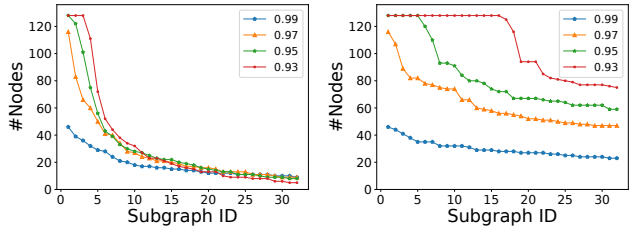
Sparsity is an important hyper-parameter in the process of building the graph. Lower sparsity can enhance information transmission. However, it will also cause higher computational complexity, which leads to more subgraphs in graph partition. In order to compare the impact of different sparsity on graph partition, we select a document with 2,000 tokens, and set the sparsity of the graph at 0.99, 0.97, 0.95, 0.93 level respectively following Eq. (8). We observe the relationship between the number of nodes (with maximum value of 128) of top 32 subgraphs and the sparsity.

The results of two graph partition strategies are shown in Figure 5. We observe that as the sparsity increases, the number of nodes in the top 32 subgraphs will also increase. When the sparsity reaches 0.93, the number of nodes in top 16 subgraphs of edge-level partition reaches the upper limit. Lower sparsity cannot bring more information if we set the max number of subgraphs to 16. This



(a) Static Distance (b) Static Centrality (c) Dynamic Distance (d) Dynamic Centrality

Figure 4: The adjacency matrix of using each attention pattern, and the yellow part means there is an edge.



(a) Node-level partition (b) Edge-level partition

Figure 5: The relationship between the number of subgraph nodes and the sparsity of the graph.

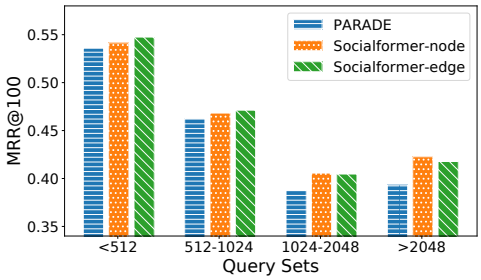


Figure 6: Performance with different query sets related to document length.

is why we choose 0.93 as the sparsity in experiments. Comparing the two strategies, we find that the number of nodes in node-level partition drops quickly. The reason is that there is no overlap between the nodes of each subgraph. In edge-level partition, more connections are retained, but there are many non-central nodes that cannot be included in top 32 subgraphs. In order to further explore the pros and cons of the two strategies, we explore the effect of different document lengths in the next section.

To observe what kind of query set the model is suitable for, we divide the whole query set on MS MARCO to four subsets based on the length l of corresponding positive documents: (a) <512; (b) 512-1024; (c) 1024-2048; (d) >2048. We choose a baseline model PARADE and our two models for comparison.

5.4 Experiment with Document Lengths

From Figure 6, we find that our social-aware models perform better than the baseline model on all query sets. Specifically, the gap between Socialformer and PARADE is widening as the document length grows. This indicates that building direct remote edges based on social networks enables the model to understand long documents better. Moreover, comparing two graph partition strategies, edge-level partition shows superiority when the document length is short, while the node-level partition performs better for longer texts. A possible reason is that top k subgraphs of Socialformer$_{\text{edge}}$ can keep more edge information for short texts than Socialformer$_{\text{node}}$. When the document length grows, more node information is abandoned in top k subgraphs. But for node-level partition, the majority of nodes can be retained regardless of the length of the document.

6 CONCLUSION

In this paper, we propose a social network inspired method for long document modeling. Concretely, we devise four attention patterns related to social networks and use probability sampling to construct a graph like social networks. To limit the computational complexity, the graph is divided into multiple subgraphs by two partition strategies. Then, to promote the full transmission of semantics in long documents, we present an iterative information transmission method which consists of inter-circle and intra-circle interactions. Finally, we can get a global document representation by an aggregation layer to re-rank the results. We conduct extensive experiments to verify the effectiveness of Socialformer. In the future, we will explore more sophisticated attention patterns and graph partition strategies according to features of webpage texts.

ACKNOWLEDGMENTS

Thanks for reviewers' valuable comments. Zhicheng Dou is the corresponding author. This work was supported by National Natural Science Foundation of China No. 61872370, Beijing Outstanding Young Scientist Program NO. BJJWZYJH012019100020098, China Unicom Innovation Ecological Cooperation Plan, the Outstanding Innovative Talents Cultivation Funded Programs 2020 of Renmin University of China, and Intelligent Social Governance Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China. We also acknowledge the support provided and contribution made by Public Policy and Decision-making Research Lab of RUC.

REFERENCES

[1] Qingyao Ai, Brendan O'Connor, and W. Bruce Croft. 2018. A Neural Passage Model for Ad-hoc Document Retrieval. In *ECIR (Lecture Notes in Computer Science)*, Vol. 10772. Springer, 537–543.

[2] Joshua Ainslie, Santiago Ontañón, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. 2020. ETC: Encoding Long and Structured Inputs in Transformers. In *EMNLP (1)*. Association for Computational Linguistics, 268–284.

[3] Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer. *CoRR* abs/2004.05150 (2020).

[4] James P. Callan. 1994. Passage-Level Evidence in Document Retrieval. In *SIGIR*. ACM/Springer, 302–310.

[5] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. Overview of the TREC 2019 deep learning track. *CoRR* abs/2003.07820 (2020). arXiv:2003.07820 https://arxiv.org/abs/2003.07820

[6] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. Overview of the TREC 2019 deep learning track. (2020).

[7] Zhuyun Dai and Jamie Callan. 2019. Deeper Text Understanding for IR with Contextual Neural Language Modeling. In *SIGIR*. ACM, 985–988.

[8] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc Viet Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In *ACL (1)*. Association for Computational Linguistics, 2978–2988.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT (1)*. Association for Computational Linguistics, 4171–4186.

[10] David Easley, Jon Kleinberg, et al. 2012. Networks, crowds, and markets: Reasoning about a highly connected world. *Significance* 9, 1 (2012), 43–44.

[11] Linton Freeman. 2004. The development of social network analysis. *A Study in the Sociology of Science* 1, 687 (2004), 159–167.

[12] Ankit Gupta and Jonathan Berant. 2020. GMAT: Global Memory Augmentation for Transformers. *CoRR* abs/2006.03274 (2020).

[13] Cecilia Henning and Mats Lieberg. 1996. Strong ties or weak ties? Neighbourhood networks in a new perspective. *Scandinavian Housing and planning research* 13, 1 (1996), 3–26.

[14] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* 9, 8 (1997), 1735–1780.

[15] Sebastian Hofstätter, Bhaskar Mitra, Hamed Zamani, Nick Craswell, and Allan Hanbury. 2021. Intra-Document Cascading: Learning to Select Passages for Neural Document Ranking. In *SIGIR*. ACM, 1349–1358.

[16] Sebastian Hofstätter, Hamed Zamani, Bhaskar Mitra, Nick Craswell, and Allan Hanbury. 2020. Local Self-Attention over Long Text for Efficient Document Retrieval. In *SIGIR*. ACM, 2021–2024.

[17] Ronald E Holtzman, George W Rebok, Jane S Saczynski, Anthony C Kouzis, Kathryn Wilcox Doyle, and William W Eaton. 2004. Social network characteristics and cognition in middle-aged and older adults. *The Journals of Gerontology Series B: Psychological Sciences and Social Sciences* 59, 6 (2004), P278–P284.

[18] Hai-hua Hu, Le Wang, Lining Jiang, and Wei Yang. 2019. Strong ties versus weak ties in word-of-mouth marketing. *BRQ Business Research Quarterly* 22, 4 (2019), 245–256.

[19] Jyun-Yu Jiang, Chenyan Xiong, Chia-Jung Lee, and Wei Wang. 2020. Long Document Ranking with Query-Directed Sparse Transformer. In *EMNLP (Findings) (Findings of ACL)*, Vol. EMNLP 2020. Association for Computational Linguistics, 4594–4605.

[20] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The Efficient Transformer. In *ICLR*. OpenReview.net.

[21] Jon M Kleinberg. 2000. Navigation in a small world. *Nature* 406, 6798 (2000), 845–845.

[22] Canjia Li, Andrew Yates, Sean MacAvaney, Ben He, and Yingfei Sun. 2020. PARADE: Passage Representation Aggregation for Document Reranking. *CoRR* abs/2008.09093 (2020).

[23] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *ICLR (Poster)*. OpenReview.net.

[24] Zhengyi Ma, Zhicheng Dou, Wei Xu, Xinyu Zhang, Hao Jiang, Zhao Cao, and Ji-Rong Wen. 2021. Pre-training for Ad-hoc Retrieval: Hyperlink is Also You Need. In *CIKM*. ACM, 1212–1221.

[25] Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. 2019. CEDR: Contextualized Embeddings for Document Ranking. In *SIGIR*. ACM, 1101–1104.

[26] Tri Nguyen, Mir Rosenberg, Xia Song, et al. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. In *NIPS 2016*.

[27] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *CoRR* abs/1901.04085 (2019).

[28] Jiezhong Qiu, Hao Ma, Omer Levy, Wen-tau Yih, Sinong Wang, and Jie Tang. 2020. Blockwise Self-Attention for Long Document Understanding. In *EMNLP (Findings) (Findings of ACL)*, Vol. EMNLP 2020. Association for Computational Linguistics, 2555–2565.

[29] Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389.

[30] Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. 2021. Efficient Content-Based Sparse Attention with Routing Transformers. *Trans. Assoc. Comput. Linguistics* 9 (2021), 53–68.

[31] Koustav Rudra and Avishek Anand. 2020. Distant Supervision in BERT-based Adhoc Document Retrieval. In *CIKM*. ACM, 2197–2200.

[32] John Scott. 1988. Social network analysis. *Sociology* 22, 1 (1988), 109–127.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NIPS 2017*. 5998–6008.

[34] Duncan J Watts and Steven H Strogatz. 1998. Collective dynamics of 'small-world'networks. *nature* 393, 6684 (1998), 440–442.

[35] Tamar Wilson. 1998. Weak ties, strong ties: Network principles in Mexican migration. *Human Organization* 57, 4 (1998), 394–403.

[36] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2021. Hi-Transformer: Hierarchical Interactive Transformer for Efficient and Effective Long Document Modeling. In *ACL/IJCNLP (2)*. Association for Computational Linguistics, 848–853.

[37] Zhijing Wu, Jiaxin Mao, Yiqun Liu, Jingtao Zhan, Yukun Zheng, Min Zhang, and Shaoping Ma. 2020. Leveraging Passage-level Cumulative Gain for Document Ranking. In *WWW*. ACM / IW3C2, 2421–2431.

[38] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, et al. 2020. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. *CoRR* abs/2007.00808 (2020).

[39] Liu Yang, Mingyang Zhang, Cheng Li, Michael Bendersky, and Marc Najork. 2020. Beyond 512 Tokens: Siamese Multi-depth Transformer-based Hierarchical Encoder for Long-Form Document Matching. In *CIKM*. ACM, 1725–1734.

[40] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *HLT-NAACL*. The Association for Computational Linguistics, 1480–1489.

[41] Zeynep Akkalyoncu Yilmaz, Shengjin Wang, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Applying BERT to Document Retrieval with Birch. In *EMNLP/IJCNLP (3)*. Association for Computational Linguistics, 19–24.

[42] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. 2021. Do Transformers Really Perform Bad for Graph Representation? *CoRR* abs/2106.05234 (2021).

[43] HongChien Yu, Zhuyun Dai, and Jamie Callan. 2021. PGT: Pseudo Relevance Feedback Using a Graph-Based Transformer. In *ECIR (2) (Lecture Notes in Computer Science)*, Vol. 12657. Springer, 440–447.

[44] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big Bird: Transformers for Longer Sequences. In *NeurIPS*.

[45] Chengxiang Zhai and John D. Lafferty. 2017. A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval. *SIGIR Forum* 51, 2 (2017), 268–276.

[46] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. An Analysis of BERT in Document Ranking. In *SIGIR*. ACM, 1941–1944.

[47] Hang Zhang, Yeyun Gong, Yelong Shen, Weisheng Li, Jiancheng Lv, Nan Duan, and Weizhu Chen. 2021. Poolingformer: Long Document Modeling with Pooling Attention. In *ICML (Proceedings of Machine Learning Research)*, Vol. 139. PMLR, 12437–12446.

[48] Xingxing Zhang, Furu Wei, and Ming Zhou. 2019. HIBERT: Document Level Pre-training of Hierarchical Bidirectional Transformers for Document Summarization. In *ACL (1)*. Association for Computational Linguistics, 5059–5069.

[49] Chen Zhao, Chenyan Xiong, Corby Rosset, Xia Song, Paul N. Bennett, and Saurabh Tiwary. 2020. Transformer-XH: Multi-Evidence Reasoning with eXtra Hop Attention. In *ICLR*. OpenReview.net.

[50] Guangxiang Zhao, Junyang Lin, Zhiyuan Zhang, Xuancheng Ren, Qi Su, and Xu Sun. 2019. Explicit Sparse Transformer: Concentrated Attention Through Explicit Selection. *CoRR* abs/1912.11637 (2019).

[51] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In *AAAI*. AAAI Press, 11106–11115.

[52] Yujia Zhou, Zhicheng Dou, Bingzheng Wei, Ruobing Xie, and Ji-Rong Wen. 2021. Group based Personalized Search by Integrating Search Behaviour and Friend Network. In *SIGIR 2021*. ACM, 92–101.