Breadth First Reasoning Graph for Multi-hop Question Answering

Yongjie Huang[1] and **Meng Yang**[1,2,*]
[1]School of Computer Science and Engineering, Sun Yat-sen University / Guangzhou, China
[2]Key Laboratory of Machine Intelligence and Advanced Computing,
Sun Yat-sen University, Ministry of Education / China
huangyj229@mail2.sysu.edu.cn, yangm6@mail.sysu.edu.cn

Abstract

Recently Graph Neural Network (GNN) has been used as a promising tool in multi-hop question answering task. However, the unnecessary updations and simple edge constructions prevent an accurate answer span extraction in a more direct and interpretable way. In this paper, we propose a novel model of Breadth First Reasoning Graph (BFR-Graph), which presents a new message passing way that better conforms to the reasoning process. In BFR-Graph, the reasoning message is required to start from the question node and pass to the next sentences node hop by hop until all the edges have been passed, which can effectively prevent each node from over-smoothing or being updated multiple times unnecessarily. To introduce more semantics, we also define the reasoning graph as a weighted graph with considering the number of co-occurrence entities and the distance between sentences. Then we present a more direct and interpretable way to aggregate scores from different levels of granularity based on the GNN. On HotpotQA leaderboard, the proposed BFR-Graph achieves state-of-the-art on answer span prediction.

1 Introduction

Typical Question Answering (QA) or Reading Comprehension (RC) task aims at exploring a desired answer through a single evidence document or paragraph. Recently, a more challenging multi-hop QA task, where we need to reason over multiple paragraphs to find the answer, is gradually catching attention. One example from HotpotQA dataset (Yang et al., 2018) is shown in Fig. 1.

One method for achieving multi-hop QA is to concatenate all the paragraphs together and treat it as a typical single-hop QA task (Yang et al., 2018), then existing QA techniques can be applied. Although multi-hop QA can be solved to some extent, this method lacks interpretation of the reasoning process from one hop to the next hop.

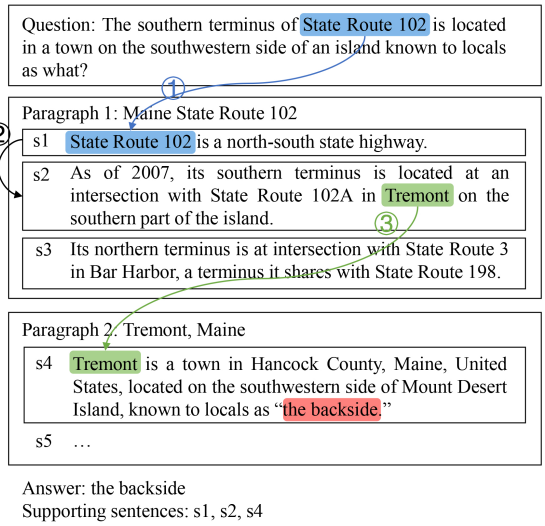


Figure 1: One example from HotpotQA dataset. "s1", "s2", ... denote the sentences in paragraphs. The model needs to find the answer and supporting sentences by reasoning over multiple sentences and paragraphs. It's obvious that the reasoning is in a ordered process from the question to "s1", "s2" and finally to "s4".

Graph Neural Networks (GNN) is a natural way to represent the solving procedure of multi-hop QA. For instance, nodes in GNN represent sentences/entities in the paragraphs, and from the updation through edges we can get interactive message between them, which is similar to the process of reasoning. Thus, a more reasonable method is to construct GNN to simulate the reasoning process among multiple paragraphs (Ding et al., 2019; Qiu et al., 2019; Tu et al., 2020). Promising performance has been reported in methods that designed different type of nodes or edges for GNN(De Cao et al., 2019; Tu et al., 2019, 2020; Fang et al., 2020) and the features generated from GNN has also been combined with those from the context encoder in a latent way (Qiu et al., 2019; Fang et al., 2020).

*Corresponding author.

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5810–5821
June 6–11, 2021.

Despite of the success that GNN achieves in multi-hop QA, new problems associated to GNN arise. Firstly, current approaches update all the nodes, including some unnecessary ones, together within each layer, which may lead the nodes to converge to similar values and lose the discriminating ability for GNN with more layers (Kipf and Welling, 2017). Secondly, although different types of edges have been designed for GNN, there is no more fine-grained distinction between edges of the same type, without considering the other relational information between sentences. Thirdly, existing methods only latently fuse the hidden representations of GNN and context encoder, without contributing to the answer span extraction in a direct and interpretable way.

To solve the aforementioned issues, we proposed a novel model of Breadth First Reasoning Graph (BFR-Graph) to effectively adapt GNN to multi-hop QA. The proposed BFR-Graph is a weighted graph in which the weight of an edge is computed based on other relational information (e.g., co-occurrence entities and distance) of the connected sentences. Inspired by the Human reasoning mechanism and the Breadth First Search algorithm, in BFR-Graph the reasoning message starts from the question and passes to the next sentence nodes hop by hop until all the edges have been passed, effectively preventing each node from updating multiple times or being updated unnecessarily. Then the reasoning result from BFR-Graph is converted to the sentence scores and paragraph scores, contributing to the answer span extraction. Specifically, the final answer span probability is the sum of the score of answer span, the sentence and the paragraph, in both of which the answer is located. Experiment results shows that our methods make GNN more powerful in multi-hop QA and achieves state-of-the-art on answer span prediction of HotpotQA.

The contributions of this paper are summarized as follows:

- We propose BFR-Graph for multi-hop QA, which is more in line with reasoning process than existing GNNs. The reasoning message starts at the question and then reasons to the next sentences hop by hop.
- Our BFR-Graph is a weighted graph, considering the number of co-occurrence entities and the distance between sentences.
- To take advantage of the reasoning result from BFR-Graph, multi-score mechanism is used for answer span extraction in a more direct and interpretable way.

2 Related Work

2.1 Multi-hop QA

Serval multi-hop QA datasets have been proposed such as WikiHop (Welbl et al., 2018) and HotpotQA (Yang et al., 2018). WikiHop provides candidate answers for selection while HotpotQA needs to find an answer span over all paragraphs. Based on these datasets, several categories of multi-hop QA approaches were proposed.

Yang et al. (2018) proposed a baseline method based on RNNs and Min et al. (2019) decomposed the multi-hop question into simpler single-hop subquestion that can be answered by existing single-hop RC models. To better utilize multiple paragraphs, Nishida et al. (2019) proposed Query Focused Extractor to sequentially summarize the context and Asai et al. (2020) used a recurrent retrieval approach that learns to sequentially retrieve evidence paragraphs. Moreover, reasoning has also been conducted in multi-hop QA. Jiang and Bansal (2019) designed a neural modular network to perform unique types of reasoning; Chen et al. (2020) presented extra hop attention that can naturally hops across the connected text sequences. Qiu et al. (2019) regards the task as a two-stage task including paragraph selection and downstream model. and Tu et al. (2020) further proposed a pairwise learning-to-rank loss for better interaction between paragraphs. Although the aforementioned methods are specifically designed for multi-hop QA with different structures, they lack an explicit scheme to show the reasoning process.

2.2 GNNs for Multi-hop QA

Recently GNNs such as Graph Convolution Networks (Kipf and Welling, 2017) and Graph Attention Networks (Veličković et al., 2018) show enhancement in multi-hop QA because the GNN-based methods are more intuitive and explicit.

Entity-GCN (De Cao et al., 2019) considered different type of edges and Tu et al. (2019) further built a heterogeneous graph with multiple types of nodes and edges for different granularity levels of information. Besides, Ding et al. (2019) coordinated implicit extraction and explicit reasoning through a GNN inspired by the dual process theory in cognitive science, and Tu et al. (2020) built a

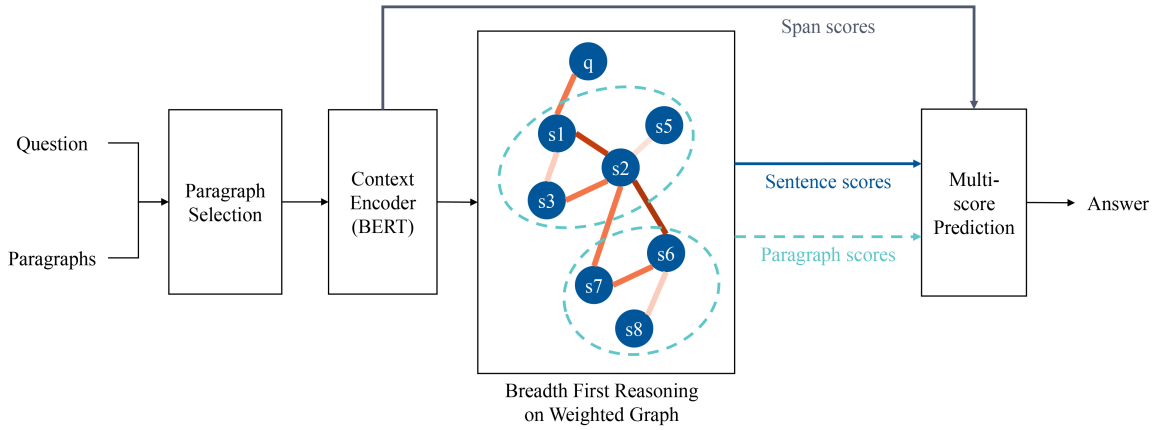


Figure 2: Diagram of our system. Each node in the graph represents a sentence. The dotted line in light blue means the sentences are from the same paragraph.

GNN model for reasoning over sentence, which is summarized over token representations based on a mixed attentive pooling mechanism. Furthermore, more complex graphs is also designed. Qiu et al. (2019) proposed a Dynamically Fused Graph Network to explore along the entity graph dynamically and finds supporting entities from the context. Fang et al. (2020) created a hierarchical graph for different levels of granularity to aggregate clues from scattered texts across multiple paragraphs. However, GNNs in these methods update all the nodes together, including some unnecessary ones.

3 Model

To solve the aforementioned issues, we propose a novel model of Breadth First Reasoning Graph (BFR-Graph) for multi-hop QA. Different from existing GNN-based methods, BFR-Graph introduces new restrictions on the message passing: the message only starts from the question and then passes to the latter sentence nodes hop by hop. Besides, our graph is constructed as a weighted graph considering the co-occurrence entities and distance between sentences. Moreover, multi-score answer prediction is designed to take advantage of the reasoning result from BFR-Graph. In short, we propose breadth first reasoning on the weighted graph and then combine multi-level scores for answer prediction in the framework of multi-task joint training.

The diagram of our system is shown in Fig. 2. Given multiple paragraphs, we first filter out irrelevant paragraph with paragraph selection (Sec. 3.1) and then use a BERT for context encoding (Sec. 3.2). A weighted graph is constructed (Sec. 3.3) to reason over sentences (Sec. 3.4) and calculate the sentence score and paragraph score. Finally, we use multi-score mechanism to predict the answer span (Sec. 3.5).

3.1 Paragraph Selection

Although multiple candidate paragraphs are given for answering the question, not all of them are useful (i.e., relevant to the question). Following Qiu et al. (2019), we retrieve N useful paragraphs for each question through a straightforward way. Each candidate paragraph is concatenated with the question ("[CLS]" + question + "[SEP]" + paragraph + "[SEP]") and fed into a BERT (Devlin et al., 2019) for binary classification. After training procedure, we select paragraphs with top-N score as the useful paragraphs, which are then concatenated together as context C.

3.2 Context Encoding

Following Qiu et al. (2019), we concatenate each question Q and its corresponding context C, and feed them into a BERT followed by a bi-attention layer (Seo et al., 2017) to obtain the encoded representations of question and context. The output is denoted as:

$$\mathbf{H} = \{\mathbf{h}_0, \cdots, \mathbf{h}_{L-1}\} \in \mathbb{R}^{L \times d}, \quad (1)$$

where L is the length of the input sequence (concatenating question and context), and d is the output dimension of bi-attention layer (also the dimension of BERT).

To achieve sentence-level representations, we first obtain token-level representation of each sen-

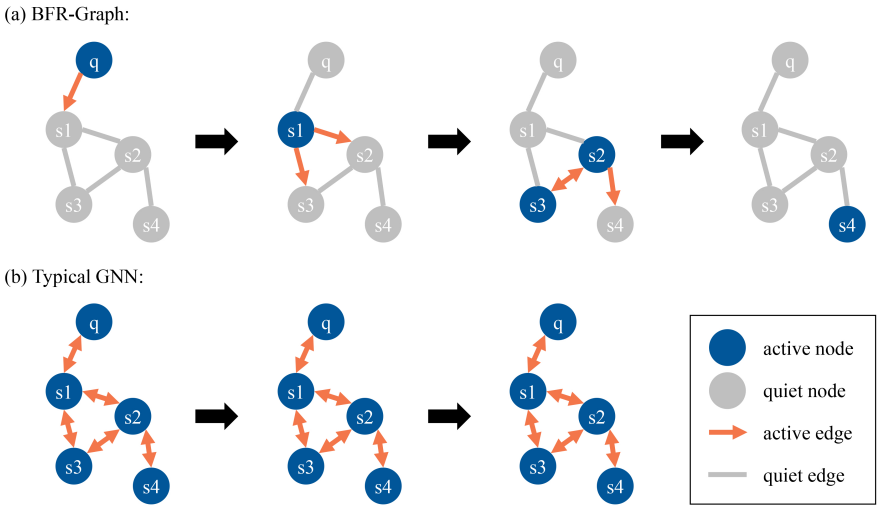


Figure 3: Message passing procedure of BFR-Graph and typical GNN. Active node is the node that is reachable for its neighbors while the quiet one is on the contrary. Active edge is the passable edge while the quiet one is on the contrary.

tence:

$$\mathbf{S}_i^{seq} = \mathbf{H}[s_i^{start} : s_i^{end}, :] \in \mathbb{R}^{L_{s_i} \times d}, \quad (2)$$

where s_i^{start}, s_i^{end} are the start and end position of the sentence i respectively, L_{s_i} is the length of sentence i. Note that the question is also a sentence. Then using the method in Rei and Søgaard (2019), we get sentence representation:

$$\mathbf{s}_i = \sum_{k=0}^{L_s} \alpha_k^i \mathbf{S}_i^{seq}[k, :] \in \mathbb{R}^d, \quad (3)$$

where α_k^i is the weight on the k-th token of sentence i, obtained from a two-layer MLP (Multi-Layer Perceptron) with output size = 1.

3.3 Weighted Graph Construction

The nodes in our weighted graph represent question Q and sentences in context C. To better exploit complex relational information between sentences, two types of correlation are defined: positive correlation and negative correlation. Although they can be designed in many ways, now we illustrate our design:

(1) Positive correlation: an edge is added if the nodes representing the sentences i and j have $n (n \geq 1)$ of the same named entities, and the weight of the edge is:

$$w_{ij} = \frac{1}{1 + e^{-n+K_1}}. \quad (4)$$

(2) Negative correlation: otherwise, an edge is added if the two nodes are originally from the same paragraph, and the weight of the edge is:

$$w_{ij} = \frac{1}{1 + e^{d+K_2}}, \quad (5)$$

where d is the distance of the two sentences (e.g., $d = 1$ if the sentence is immediately followed by the other sentence in a paragraph, $d = 2$ if there is a sentence between them, etc.). K_1 and K_2 are hyperparameters.

To simplify our design, we treat our graph as a homogeneous graph, which contains single type of nodes and edges.

3.4 Breadth First Reasoning

When we reason over paragraphs to answer a question, we start from the question and find the next sentence hop by hop. For a GNN where nodes represent sentences, the following message passing is unnecessary and may suppress the disturbance from useless nodes: (1) from the latter node to the former node, (2) a node haven't received the message from question but it updates other nodes.

To prevent each node from being updated multiple times unnecessarily, the reasoning message in our BFR-Graph starts from the question node and passes to the next nodes hop by hop until all the edges have been passed. Note that a node is allowed to update multiple times, depending on whether the connected edges have all been passed.

Algorithm 1: Algorithm of BFR-Graph

$\mathcal{E}$ represents the set of edges that haven't been passed yet (dynamic);
$\mathcal{A}$ represents the set of active nodes (dynamic);
$\mathcal{N}_i$ represents neighbors of node i (static);
$\mathcal{N}'_i$ represents reachable neighbors of node i (dynamic).
Input: Initial node representations S, the set of neighbors $\mathcal{N}$.
Output: Node representations S'.

$\mathcal{E} \leftarrow$ all edges
$\mathcal{A} \leftarrow$ question node
while True **do**
  // To update node i
  $\mathcal{N}'_i \leftarrow \emptyset$
  forall $j \in \mathcal{N}_i$ **do**
    if $j \in \mathcal{A}$ ***and*** $(i,j) \in \mathcal{E}$ **then**
      Add (i,j) to $\mathcal{N}'_i$
    end
  end
  if $\mathcal{N}'_i == \emptyset$ **then**
    break
  end
  Update node i with $\mathcal{N}'_i$
  forall $j \in \mathcal{N}'_i$ **do**
    Remove (i,j) from $\mathcal{E}$
    Remove j from $\mathcal{A}$
  end
  Add i to $\mathcal{A}$
end

Fig. 3 visually shows the difference between BFR-Graph and typical GNN.

Specifically, a node i is updated by node j when the following conditions are met simultaneously: (1) node i and node j are neighbors, (2) node j is active, i.e., it is updated last layer, (3) the edge between node i and node j haven't been passed previously. The overall message passing procedure of BFR-Graph is illustrated in Algorithm 1.

Inspired by Graph Attention Networks (Veličković et al., 2018), the updating function (or message passing function) is defined as:

$$\mathbf{s}'_i = \text{LeakyRelu}(\sum_{j \in \mathcal{N}'_i} \beta_{ij} \mathbf{s}_j \mathbf{W}), \quad (6)$$

$$\beta_{ij} = \frac{\exp(f(\mathbf{s}_i, \mathbf{s}_j)) \cdot w_{ij}}{\sum_{k \in \mathcal{N}'_i} \exp(f(\mathbf{s}_i, \mathbf{s}_k)) \cdot w_{ik}}, \quad (7)$$

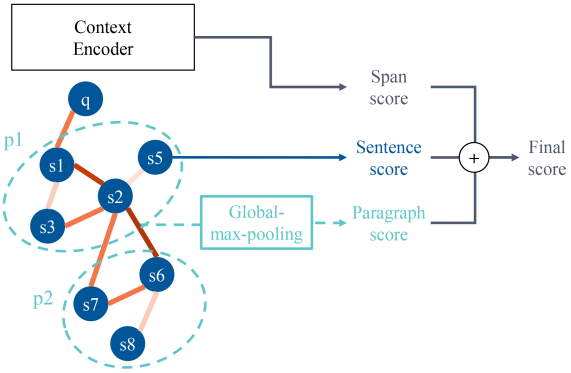


Figure 4: Multi-score answer prediction. The example is calculating the score for an answer span located in paragraph 1 ("p1") and sentence 5 ("s5").

where $\mathcal{N}'_i$ is the set of reachable neighbors for node i, calculated with Algorithm 1. $f(\mathbf{s}_i, \mathbf{s}_j) = \mathbf{s}_i \mathbf{W_1} \mathbf{W_2} \mathbf{s}_j$ is for calculating the attention score between node i and j. $\mathbf{W}$, $\mathbf{W_1}$ and $\mathbf{W_2}$ are learnable parametres. w_{ij} is the weight of the edge (i,j), described in the Sec. 3.3. For clarity, $\mathbf{s}'$ is written as $\mathbf{s}$ in following contents.

3.5 Multi-score Answer Prediction

The answer in HotpotQA dataset is a span from the context. Existing works only calculate the span probability on the output of encoder (e.g., BERT) or additionally concatenate the GNN's hidden output. Differently, we use a more interpretable method by calculating the sentence score and paragraph score obtained from the GNN. An example is shown in Fig. 4.

Conventionally, the score of y-th word in context being the start / end of the answer span is calculated by:

$$\phi_{start}(y) = \text{MLP}_1(\mathbf{H}[y,:]), \quad (8)$$

$$\phi_{end}(y) = \text{MLP}_2(\mathbf{H}[y,:]), \quad (9)$$

where MLP is a two-layer MLP with output size = 1 to obtain the score value.

Then, we calculate the sentence score corresponding to each node in GNN:

$$\phi_{sent}(s_i) = \text{MLP}_3(\mathbf{s}_i). \quad (10)$$

Similarly, we calculate the paragraph score through a global-max-pooling:

$$\phi_{para}(p_j) = \text{MLP}_4(\text{Max}(\{\mathbf{s}_0^{p_j}, ..., \mathbf{s}_{L_{p_j}-1}^{p_j}\})), \quad (11)$$

where $\mathbf{s}_i^{p_j}$ is the representation of the i-th sentence in paragraph p_j, L_{p_j} is the number of sentences in paragraph p_j. Max$(\cdot)$ is a max-pooling layer with pooling size = $L_{p_j} \times 1$, which can also be done by taking the maximum hidden value on each dimension over all the sentence nodes.

Finally, the probability of y-th word in context being the start of the answer span is determined by:

$$p_{start}(y) = \text{softmax}(\phi'_{start}(y)), \quad (12)$$

$$\phi'_{start}(y) = \phi_{start}(y) + \phi_{sent}(s_i) + \phi_{para}(p_j), \quad (13)$$

where the y-th word is located in sentence s_i and paragraph p_j. And the probability of y-th word in context being the end of the answer span can be calculated similarly.

In other words, if a sentence or paragraph has a higher score, the words located in it are more likely to be the answer.

3.6 Multi-task Joint Training

In addition to the answer span prediction, there are other two training tasks in HotpotQA. One is the answer type prediction task: some answers cannot be retrieved from the context, but are "Yes" or "No", so finally there are three type of answers (e.g., span, "Yes" and "No"). We use a global-max-pooling similar with Eq.(11) to compress all the nodes in the GNN and predict the answer type through a two-layer MLP.

The other task is to predict whether a sentence in the context is a support sentence (or called supporting fact in some papers) that is an evidence to the answer. Following previous works (Tu et al., 2020), we use the output of the GNN to predict the supporting sentences with a two-layer MLP.

The tasks in HotpotQA are jointly performed through multi-task learning, and the loss function is:

$$\begin{aligned}\mathcal{L} =& \mathcal{L}_{CE}(\hat{y}^{start}, y^{start}) + \mathcal{L}_{CE}(\hat{y}^{end}, y^{end}) + \\ & \lambda_1 \cdot \mathcal{L}_{CE}(\hat{y}^{type}, y^{type}) + \lambda_2 \cdot \mathcal{L}_{BCE}(\hat{y}^{sp}, y^{sp}),\end{aligned} \quad (14)$$

where $\mathcal{L}_{CE}$ and $\mathcal{L}_{BCE}$ denote the cross entropy and binary cross entropy loss respectively. $\hat{y}^{start}$ denotes the logits of start position from Eq.(12) and y^{start} is the label. Similarly, $\hat{y}^{type}$ and $\hat{y}^{sp}$ are the logits of answer type prediction and supporting sentence prediction respectively.

4 Experiments

4.1 Dataset

The HotpotQA dataset (Yang et al., 2018) is the first explainable multi-hop QA dataset with sentence-level evidence supervision. Each sample in the dataset contains 2 gold paragraphs and 8 distracting paragraphs. Three tasks are included for evaluation: (1) answer span prediction (denoted as "Ans") that extracts a span in the paragraphs or generate "Yes"/"No"; (2) supporting sentences prediction (denoted as "Sup") that determines which sentences are evidences to the answer; (3) joint prediction (denoted as "Joint"). We submit our model to HotpotQA official leaderboard[1] and carry out ablation studies on the dev-set.

We also apply the main idea of BFR-Graph to the WikiHop dataset (Welbl et al., 2018), which provides candidate answers for selection while HotpotQA dataset needs to find an answer span over all paragraphs.

Implementation details can be found in Appendix A.

4.2 Results

The experimental result on HotpotQA dataset is shown in Table 1. As a reading comprehension task, the performance of answer prediction should be emphasized. Our model improves 0.84% Ans-EM (Exact Match) than HGN-large, becoming the first model to break through 70% and achieving state-of-the-art on answer span prediction. On supporting sentence prediction and joint prediction, our model shows a close performances to HGN-large, possibly because this paper is based on the standard GNN (homogeneous graph) for simple clarification, and we just plan to prove that our algorithm can improve the performance of GNN. Existing GNN methods mostly constructed elaborate graphs for more granular expression of nodes, while our BFR-Graph solve the problem from another novel perspective. Thus, BFR-Graph is universal and can be easily applied to existing promising models (e.g., HGN) to get better results, which provides a promising direction for future research.

We also compare our model with two state-of-the-art GNN models (i.e., SAE and HGN), shown in Table 2. Both of them need to set the number of GNN layers manually while BFR-Graph can pass through all the connected nodes automatically with

[1] https://hotpotqa.github.io/

Model	Ans		Sup		Joint	
	EM	F1	EM	F1	EM	F1
Official baseline (Yang et al., 2018)	45.60	59.02	20.32	64.49	10.83	40.16
QFE (Nishida et al., 2019)	53.86	68.06	57.75	84.49	34.63	59.61
DFGN (Qiu et al., 2019)	56.31	69.69	51.50	81.62	33.62	59.82
LQR-Net (Grail et al., 2020)	60.20	73.78	56.21	84.09	36.56	63.68
SAE-large (Tu et al., 2020)	66.92	79.62	61.53	86.86	45.36	71.45
C2F-reader (Shao et al., 2020)	67.98	81.24	60.81	87.63	44.67	72.73
HGN-large (Fang et al., 2020)	69.22	82.19	**62.76**	**88.47**	**47.11**	**74.21**
BFR-Graph	**70.06**	**82.20**	61.33	88.41	45.92	74.13

Table 1: Results on HotpotQA leaderboard. "Ans", "Sup" and "Joint" denote answer span prediction, supporting sentence prediction and joint prediction, respectively.

	Layers	Edges	Intuitive
SAE	manual	3 types	false
NGN	manual	7 types	false
BFR-Graph	adaptive	fine-grained	true

Table 2: Comparison with state-of-the-art GNN models. "Layers" denotes the number of GNN layers, "Edges" denotes how fine-grained the edges are, and "Intuitive" denotes whether the output of GNN can be intuitively observed.

Model	Accuracy
HDE (Tu et al., 2019)	68.1
DynSAN (Zhuang and Wang, 2019)	70.1
Path-based GCN (Tang et al., 2020)	70.8
ChainEx (Chen et al., 2019)	72.2
Longformer*	73.8
Longformer+BFR	**74.4**

Table 3: Results on WikiHop dev-set. Model annotated with "$*$" is our re-implementation.

an extremely low risk of over-smoothing (Kipf and Welling, 2017). SAE and HGN set a fixed types of edges, which is still not fine-grained enough, while BFR-Graph define different weights (can up to ∞ different weights depends on the dataset) to distinguish nodes in a finer granularity. Furthermore, we can easily observe scores from GNN in an intuitive way in BFR-Graph.

Besides, Table 3 shows the results on WikiHop dev-set. When we add the breadth first reasoning graph and weights to Longformer (Beltagy et al., 2020), the performance is slightly improved, showing that our method have the ability for better reasoning.

5 Ablations and Analysis

In this section, we carry out ablation studies on HotpotQA dev-set. Table 4 shows the results of our full model and that without breadth first reasoning, weights, and multi-score. It indicates that our methods obviously improve the performance of GNN.

5.1 Evaluation on Breadth First Reasoning

Table 5 shows the result by gradually replace the BFR-Graph layers with standard GNN layers. In detail, "r/p 1 layer" denotes replacing the first layer with a standard GNN layer, "r/p 2 layers" denotes the same operation for the first and second layers, etc.. We observe that the more layers to be replaced, the more severely the result drops. And when we replace 4 layers, the joint F1 drops at about 6%, meaning that it causes over-smoothing. It also reflects the severe problem of typical GNN: if it have more layers, over-smoothing is caused; if it have less layers, it cannot achieve long-path reasoning.

To further analyze why this particular approach of message passing in a breadth first reasoning fashion should result in better reasoning, we propose to calculate how many useful messages the answer sentence node received from supporting senteences: $precision = \frac{N_{sp\&rcv}}{N_{rcv}}, recall = \frac{N_{sp\&rcv}}{N_{sp}}$, where N_{rcv} denotes how many nodes' massages the answer sentence node received, N_{sp} denotes the number of supporting sentence (containing the question sentence here), and $N_{sp\&rcv}$ denotes how many supporting nodes' massages the answer sentence node received.

The above-mentioned precision, recall and corresponding F1 on dev-set is shown in Table 6, where the typical GNN is a 2-layer GNN following previous works. With breadth first reasoning, the answer

	Ans F1	Sup F1	Joint F1
full model	**81.82**	**88.80**	**73.98**
- bfr&ws&ms	80.72	87.77	72.20

Table 4: General ablation study for our full model. "-bfr" denotes a typical GNN without breadth first reasoning; "ws" and "ms" denote the weights and multi-score respectively.

	Ans F1	Sup F1	Joint F1
full model	81.82	**88.80**	**73.98**
r/p 1 layer	**81.86**	88.49	73.80
r/p 2 layers	81.57	88.50	73.62
r/p 3 layers	80.66	87.63	72.04
r/p 4 layers	77.08	86.50	67.97

Table 5: Ablations on breadth first reasoning.

sentence could receive messages from supporting sentences with a higher precision, meaning that it can focus on useful sentences and eliminate invalid distractions. Since the restrictions on message passing in breadth first reasoning, it leads to a decrease in recall. However, it is hard to draw a PR curve or get different precision-recall results because this is not a binary classification task as we generally understand. But fortunately, BFR-Graph shows a higher F1 than the typical GNN.

5.2 Evaluation on Weights and Multi-score

Table 7 (top) presents the results with and without the weights in the GNN. "-ent" denotes removing the weights (we set the weights = 0.5 rather than simply remove them) and "- dist" denotes removing the distance weights. When we remove the weights, although the answer F1 rises slightly, the supporting F1 falls to a greater extent. This shows that the proposed weights is beneficial to the supporting sentences prediction, which is directly predicted from the GNN nodes.

To our understanding, our model enhances the discrimination of edges by setting weights for them, and inevitably reduces the robustness of model. Fortunately, by designing Eqs.(4) and (5), the quantitative error will not cause the weight to increase or decrease sharply, and is still able to distinguish the difference between sentences.

	Precision	Recall	F1
typical GNN	37.62	**95.61**	52.89
BFR-Graph	**59.44**	83.49	**63.08**

Table 6: Message passing in different style.

	Ans F1	Sup F1	Joint F1
full model	81.82	**88.80**	**73.98**
- ent	81.91	88.53	73.90
- dist	**81.98**	88.55	73.91
- ent&dist	81.90	88.51	73.75
full model	**81.82**	**88.80**	**73.98**
- sent	81.73	88.75	73.97
- para	81.81	88.64	73.95
- sent&para	81.73	88.56	73.68

Table 7: Ablations on weights and multi-score.

	Complexity
typical GNN	$K * N * M * d$
BFR-Graph	$K * N_{update} * M_{reach} * d$

Table 8: Complexities for different message passing ways on K-layer GNN with N nodes and representation dimension d, and M is the average number of neighbors for each node. For BFR-Graph, N_{update} is the number of nodes to be updated in current layer and M_{reach} is the number of neighbors for current node in current layer. For clarity, we ignore the difference between different layers and different nodes.

For multi-score, we evaluate how the result changes if this particular way of exploiting GNN's output is replaced by traditional way. In Table 7 (bottom), "-sent" and "-para" denote removing multi-score for sentence and paragraph respectively. It indicates that both the addition of sentence scores and paragraph scores are beneficial to the performance.

5.3 Complexity Analysis

We also analyze the complexities of BFR-Graph and typical GNN, which is simply shown in Table 8. Firstly, in each layer of our BFR-Graph, only several nodes are updated by active nodes, so the number of nodes to be updated in a BFR-Graph layer is less than or equal to that in a typical GNN ($N_{update} \leq N$). Secondly, for a node in a layer of BFR-Graph, it is only updated by its reachable nodes (i.e., active neighbors), so the number of reachable nodes for a node in a BFR-Graph layer is also less than or equal to that in typical GNN ($M_{reach} \leq M$). Therefore, breadth first reasoning leads to lower complexity.

For GPU parallel training, we also show the actual cost of time per epoch. BFR-Graph cost 158.6 minutes per epoch, while a 2-layer and 3-layer typical GNN costs 157.5 and 165.6 minutes respec-

tively. We find that BFR-Graph is always 4 layers in HotpotQA dataset, and it can even cost less time than a 3-layer typical GNN and is close to a 2-layer typical GNN.

6 Conclusion

In this paper, we proposed a novel GNN model of BFR-Graph. Specifically, the reasoning message starts from the question node and passes to the next sentences node hop by hop until all the edges have been passed. We also construct the reasoning graph as a weighted graph and present a more interpretable way to aggregate scores of different levels from GNN. On HotpotQA leaderboard, BFR-Graph achieved state-of-the-art on answer span prediction.

Acknowledgements

This work is partially supported by National Natural Science Foundation of China (Grants no. 61772568), Guangdong Basic and Applied Basic Research Foundation (Grant no. 2019A1515012029), and Youth science and technology innovation talent of Guangdong Special Support Program.

References

Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2020. Learning to retrieve reasoning paths over wikipedia graph for question answering. In *Proceedings of International Conference on Learning Representations*.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Jifan Chen, Shih-ting Lin, and Greg Durrett. 2019. Multi-hop question answering via reasoning chains. *arXiv preprint arXiv:1910.02610*.

Zhao Chen, Xiong Chenyan, Rosset Corby, Song Xia, Bennett Paul, and Tiwary Saurabh. 2020. Transformer-xh: Multi-evidence reasoning with extra hop attention. In *Proceedings of International Conference on Learning Representations*.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2019. Question answering by reasoning across documents with graph convolutional networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*, pages 2306–2317.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*, pages 4171–4186.

Ming Ding, Chang Zhou, Qibin Chen, Hongxia Yang, and Jie Tang. 2019. Cognitive graph for multi-hop reading comprehension at scale. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2694–2703.

Yuwei Fang, Siqi Sun, Zhe Gan, Rohit Pillai, Shuohang Wang, and Jingjing Liu. 2020. Hierarchical graph network for multi-hop question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 8823–8838.

Quentin Grail, Julien Perez, and Eric Gaussier. 2020. Latent question reformulation and information accumulation for multi-hop machine reading. Https://openreview.net/forum?id=S1x63TEYvr.

Dirk Groeneveld, Tushar Khot, Mausam, and Ashish Sabharwal. 2020. A simple yet strong pipeline for HotpotQA. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 8839–8845.

Yichen Jiang and Mohit Bansal. 2019. Self-assembling modular networks for interpretable multi-hop reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 4474–4484.

Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of International Conference on Learning Representations*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019. Multi-hop reading comprehension through question decomposition and rescoring. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6097–6109.

Kosuke Nishida, Kyosuke Nishida, Masaaki Nagata, Atsushi Otsuka, Itsumi Saito, Hisako Asano, and Junji Tomita. 2019. Answering while summarizing: Multi-task learning for multi-hop QA with evidence extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2335–2345.

Lin Qiu, Yunxuan Xiao, Yanru Qu, Hao Zhou, Lei Li, Weinan Zhang, and Yong Yu. 2019. Dynamically fused graph network for multi-hop reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6140–6150.

Marek Rei and Anders Søgaard. 2019. Jointly learning to label sentences and tokens. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6916–6923.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *Proceedings of International Conference on Learning Representations*.

Nan Shao, Yiming Cui, Ting Liu, Shijin Wang, and Guoping Hu. 2020. Is Graph Structure Necessary for Multi-hop Question Answering? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 7187–7192.

Zeyun Tang, Yongliang Shen, Xinyin Ma, Wei Xu, Jiale Yu, and Weiming Lu. 2020. Multi-hop reading comprehension across documents with path-based graph convolutional network. pages 3905–3911.

Ming Tu, Kevin Huang, Guangtao Wang, Jing Huang, Xiaodong He, and Bowen Zhou. 2020. Select, answer and explain: Interpretable multi-hop reading comprehension over multiple documents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9073–9080.

Ming Tu, Guangtao Wang, Jing Huang, Yun Tang, Xiaodong He, and Bowen Zhou. 2019. Multi-hop reading comprehension across multiple documents by reasoning over heterogeneous graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2704–2713.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *Proceedings of International Conference on Learning Representations*.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

Yimeng Zhuang and Huadong Wang. 2019. Token-level dynamic self-attention network for multi-passage reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2252–2262.

A Implementation Details

We select $N = 3$ useful paragraphs in paragraph selection, which achieves 98.7% recall in dev-set. We use RoBerta-large (Liu et al., 2019) for context encoding, with a maximum length of 512 tokens. We also fine-tune the model on SQuAD dataset similar as Groeneveld et al. (2020). We use spaCy[2] for named entity recognition and we found the balance factor $K_1 = 0$, $K_2 = -2$ lead to better result. The manual weights of the loss function are $\lambda_1 = 1, \lambda_2 = 5$ in this work. The sentences number is limited to 30 and the max sentence length is set to 512 (same with BERT). We use Adam with learning rate of 1e-5, L2 weight decay of 0.01, learning rate warm-up over the first 1,000 steps and linear decay to 0. Other hyperparameters mainly follow previous works (Fang et al., 2020). We implement our model using PyTorch[3] and train it on RTX 2080ti GPUs.

The whole task consists of two stage training: the first stage is the paragraph selection and the second stage is the following. For the second stage, we train the model using annotated gold paragraphs, and take the predicted paragraphs from the first stage during evaluation.

More details of the dataset and metrics can be found in Yang et al. (2018). For WikiHop dataset, we migrate the breadth first reasoning and weights to a baseline model (we reimplement Longformer-base (Beltagy et al., 2020) as the baseline) and evaluate the models on the dev-set.

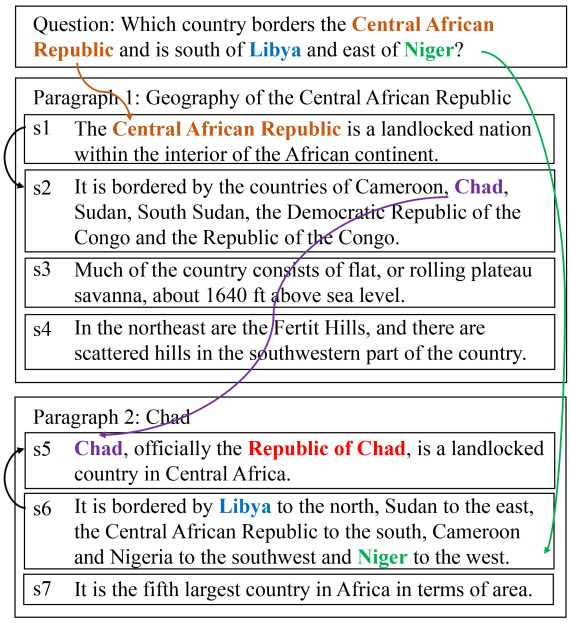
B Case Study and Error Analysis

In Fig.5, we provide an example for case study. The reasoning chain in this case should be divided into two part: Q→s1→s2→s5 and Q→s6→s5, and finally the two part of the chain is combined together and contribute to the final answer. The complex and long reasoning chain make the question hard to answer.

As reported in Fang et al. (2020), HGN retrieved another incorrect answer span. But fortunately, our BFR-Graph can effectively deal with complex reasoning and extract a better answer through the long reasoning chain.

[2]https://spacy.io/

[3]https://pytorch.org/



Reference Answer: Republic of Chad
BFR-Graph: Chad (correct)
HGN: Sudan (incorrect)

Figure 5: Case study.

Category	Percentage(%)
Annotation	10
Multiple Answers	22
Discrete Reasoning	16
External Knowledge	20
Multi-hop	16
MRC	16

Table 9: Error Analysis of our BFR-Graph.

To provide in-depth understanding of the weaknesses of our model, we carry out error analysis. Following Fang et al. (2020), we randomly sample 100 examples in the dev-set with the answer F1 as 0. Then we group the error cases into 6 categories: (1) Annotation: the reference answer is incorrect; (2) Multiple Answers: multiple correct answers can answer the question, but only one is provided in the dataset; (3) Discrete Reasoning: this type of error often appears in "comparison" questions, where discrete reasoning is required to answer the question; (4) External Knowledge: commonsense, external knowledge or mathematical operation is required; (5) Multi-hop: the model fails to perform multi-hop reasoning, and finds the final answer from wrong paragraphs; (6) MRC: the model extracts the wrong answer span but correctly finds the supporting paragraphs and sentences.

Table 9 shows the percentages of the 6 error categories of our BFR-Graph. We find that many errors are due to the wrong reference answer (10%) or multiple answers (22%), which actually should not be considered as the error cases. Among other error cases, the major category of errors comes from the questions that need external knowledge (20%, including commonsense and mathematical operation), which is hard to handle without a knowledge base.

C A Case for Multi-score Prediction

Fig. 6 shows an example with specific scores when calculating multi-scores. The RoBerta-style tokens have already been converted to the BERT-style tokens for better reading.

"Token-idx" denotes the index for each token. "Para-score" and "Sent-score" denote paragraph scores and sentences scores respectively. "Start-score" and "End-score" are the scores that be the start and end of the answer span.

There are 3 paragraphs in this case (token index: 0-54, 55-108, 109-204), and the second paragraph achieve the highest paragraph score. Similarly, we can find the highest sentence score (token index: 55-79). Both (token index: 66-67) and (token index: 88-89) lead to the correct answer, with high span scores.

Question: Which German project recorded a song that featured vocals by a duo from Silverdale, England?

Target answer: Enigma

Token-idx	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Paragraphs	A	##qu	##ilo	is	an	alternative	musical	duo	from	Silver	##dale	,	Lanc	##ash	##ire	,	England	,	consisting
Para-score	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47
Sent-score	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89
Start-score	-5.71	-9.40	-10.38	-14.01	-13.32	-8.07	-7.30	-11.96	-14.11	-12.13	-13.39	-14.19	-9.55	-10.54	-10.87	-13.10	-9.81	-12.81	-9.94
End-score	-10.92	-11.60	-6.38	-11.54	-11.80	-10.78	-9.09	-9.68	-12.01	-12.50	-11.40	-11.07	-11.55	-11.48	-11.48	-11.19	-8.66	-10.07	-10.68

19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
of	Tom	High	##am	and	Ben	Fletcher	.	They	began	gaining	recognition	in	2013	for	their	singles	such	as	"
-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47
0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	-5.42	-5.42	-5.42	-5.42	-5.42	-5.42	-5.42	-5.42	-5.42	-5.42	-5.42	-5.42
-9.48	-7.82	-8.35	-9.29	-8.83	-7.75	-8.69	-6.75	-6.03	-8.32	-7.87	-7.90	-7.76	-7.09	-8.24	-7.70	-8.03	-8.43	-8.96	-7.16
-10.67	-10.02	-10.45	-9.05	-11.00	-10.11	-8.91	-2.39	-7.80	-7.87	-9.18	-8.63	-9.41	-7.69	-9.27	-9.65	-8.61	-8.15	-8.64	-9.95

39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58
Calling	Me	"	and	"	You	There	"	in	addition	to	their	five	E	##Ps	.	"	A	##men	"
-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.19	-1.19	-1.19	-1.19
-5.42	-5.42	-5.42	-5.42	-5.42	-5.42	-5.42	-5.42	-5.42	-5.42	-5.42	-5.42	-5.42	-5.42	-5.42	-5.42	1.49	1.49	1.49	1.49
-6.98	-7.99	-8.68	-8.30	-7.36	-6.80	-8.06	-8.44	-8.57	-8.96	-8.93	-7.69	-7.19	-6.81	-9.01	-8.43	-3.39	-0.72	-5.84	-7.33
-9.89	-8.53	-7.74	-9.71	-10.23	-9.62	-9.08	-7.57	-9.20	-8.32	-9.26	-9.58	-9.05	-9.73	-7.80	-2.49	-9.11	-9.51	-5.16	-7.57

59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78
is	a	song	by	German	musical	project	En	##igma	,	featuring	vocals	by	English	dream	-	pop	duo	Aqu	##ilo
-1.19	-1.19	-1.19	-1.19	-1.19	-1.19	-1.19	-1.19	-1.19	-1.19	-1.19	-1.19	-1.19	-1.19	-1.19	-1.19	-1.19	-1.19	-1.19	-1.19
1.49	1.49	1.49	1.49	1.49	1.49	1.49	1.49	1.49	1.49	1.49	1.49	1.49	1.49	1.49	1.49	1.49	1.49	1.49	1.49
-7.08	-6.43	-8.43	-7.22	0.79	-4.55	-8.23	7.74	-4.57	-13.01	-9.64	-9.92	-11.04	-6.94	-8.39	-10.40	-9.45	-8.63	-3.14	-9.65
-11.40	-11.34	-10.64	-13.69	-8.36	-9.88	-9.32	-7.24	7.12	-6.61	-12.59	-10.05	-11.03	-8.69	-9.75	-11.18	-9.59	-8.45	-10.13	-2.71

79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98
.	It	was	released	as	the	second	single	from	En	##igma	's	eighth	studio	album	,	"	The	Fall	of
-1.19	-1.19	-1.19	-1.19	-1.19	-1.19	-1.19	-1.19	-1.19	-1.19	-1.19	-1.19	-1.19	-1.19	-1.19	-1.19	-1.19	-1.19	-1.19	-1.19
1.49	-4.46	-4.46	-4.46	-4.46	-4.46	-4.46	-4.46	-4.46	-4.46	-4.46	-4.46	-4.46	-4.46	-4.46	-4.46	-4.46	-4.46	-4.46	-4.46
-8.58	-4.95	-8.57	-7.63	-8.00	-6.64	-6.28	-7.90	-7.12	6.28	-5.06	-8.68	-6.18	-8.04	-8.13	-9.33	-6.91	-5.61	-7.09	-8.00
-2.40	-7.96	-8.64	-7.72	-9.49	-9.11	-7.25	-6.69	-9.23	-6.96	5.15	-4.34	-7.39	-8.12	-5.00	-6.39	-8.51	-8.36	-9.25	-9.38

99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118
a	Rebel	Angel	"	on	November	18	,	2016	.	T	##rou	##lm	is	a	German	project	of	drone	music
-1.19	-1.19	-1.19	-1.19	-1.19	-1.19	-1.19	-1.19	-1.19	-1.19	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97
-4.46	-4.46	-4.46	-4.46	-4.46	-4.46	-4.46	-4.46	-4.46	-4.46	-8.85	-8.85	-8.85	-8.85	-8.85	-8.85	-8.85	-8.85	-8.85	-8.85
-7.69	-6.95	-7.16	-7.91	-7.58	-7.09	-6.20	-9.98	-7.64	-9.23	-3.15	-8.03	-9.08	-7.10	-6.37	-5.28	-7.71	-7.54	-6.71	-7.35
-8.74	-8.54	-5.69	-4.77	-9.57	-8.59	-7.16	-8.11	-6.61	-3.33	-9.36	-8.86	-3.10	-9.21	-9.57	-7.85	-7.42	-9.26	-8.98	-7.49

119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138
,	ambient	music	,	noise	music	,	and	experimental	music	.	It	was	founded	in	the	late	1990	s	by
-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97
-8.85	-8.85	-8.85	-8.85	-8.85	-8.85	-8.85	-8.85	-8.85	-8.85	-8.85	-8.52	-8.52	-8.52	-8.52	-8.52	-8.52	-8.52	-8.52	-8.52
-8.33	-7.33	-7.64	-8.38	-7.28	-7.79	-10.19	-7.70	-7.74	-7.61	-6.75	-5.48	-7.73	-7.32	-7.22	-6.46	-7.09	-7.75	-8.87	-7.79
-8.62	-9.39	-7.91	-8.55	-9.39	-7.61	-6.89	-9.45	-9.20	-7.26	-2.39	-8.72	-8.80	-8.75	-9.52	-8.87	-8.98	-8.77	-6.87	-8.88

139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158
Stefan	Kn	##app	##e	(	a	.	k	.	a	.	Bar	##aka	[	H	])	and	Martin	G	##its
-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97
-8.52	-8.52	-8.52	-8.52	-8.52	-8.52	-8.52	-8.52	-8.52	-8.52	-8.52	-8.52	-8.52	-8.52	-8.52	-8.52	-8.52	-8.52	-8.52	-8.52
-6.70	-7.48	-8.55	-8.84	-7.46	-7.07	-9.54	-7.96	-9.51	-7.86	-9.25	-6.58	-8.57	-8.18	-8.45	-8.68	-7.45	-6.05	-6.68	-8.55
-8.93	-9.98	-8.61	-7.48	-9.33	-9.11	-4.73	-9.26	-4.54	-9.06	-5.45	-9.75	-6.88	-8.69	-8.54	-8.01	-9.74	-9.09	-9.92	-8.34

159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178
##chel	(	a	.	k	.	a	.	Gl	##it	[	S	]	ch	).	It	is	sometimes	considered	to
-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97
-8.52	-8.52	-8.52	-8.52	-8.52	-8.52	-8.52	-8.52	-8.52	-8.52	-8.52	-8.52	-8.52	-8.52	-8.52	-8.58	-8.58	-8.58	-8.58	-8.58
-8.90	-7.15	-7.03	-6.75	-8.56	-6.75	-8.30	-6.75	-7.14	-8.73	-8.47	-7.75	-9.60	-8.22	-8.77	-5.34	-7.80	-6.85	-7.23	-7.80
-6.39	-9.59	-9.40	-2.39	-8.97	-2.39	-8.60	-2.39	-10.09	-7.95	-7.03	-9.09	-7.41	-8.07	-5.04	-8.94	-8.98	-9.60	-9.71	-9.04

179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198
be	the	follow	-	up	project	to	Ma	##er	##or	Tri	.	Stefan	Kn	##app	##e	is	also	the	founder
-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97	-1.97
-8.58	-8.58	-8.58	-8.58	-8.58	-8.58	-8.58	-8.58	-8.58	-8.58	-8.58	-8.58	-8.64	-8.64	-8.64	-8.64	-8.64	-8.64	-8.64	-8.64
-7.61	-7.05	-6.44	-8.06	-8.57	-7.55	-7.59	-4.88	-8.25	-8.46	-7.72	-6.75	-5.05	-6.89	-8.24	-8.76	-7.39	-7.35	-6.96	-6.89
-9.23	-9.73	-9.78	-9.29	-7.68	-8.28	-9.28	-9.50	-8.53	-7.17	-7.09	-2.39	-9.33	-10.12	-8.68	-6.71	-9.05	-8.99	-9.30	-8.87

199	200	201	202	203	204
and	owner	of	Drone	Records	.
-1.97	-1.97	-1.97	-1.97	-1.97	-1.97
-8.64	-8.64	-8.64	-8.64	-8.64	-8.64
-7.59	-6.92	-7.59	-5.69	-7.84	-6.75
-9.36	-9.34	-9.18	-8.01	-6.69	-2.39

Figure 6: A Case for Multi-score Prediction.