BAG: Bi-directional Attention Entity Graph Convolutional Network for Multi-hop Reasoning Question Answering

Overview and aim of this report

This writing is an analysis and elaboration of the paper [CFT19], referred to as BAG .First, I have tried to give an intuition and motivation for the paper , and then different parts of the model is explained in simple words and when appropriate visualized. Furthermore, I have compared BAG with the top model for question answering on wikihop ,namely BigBird [Zah+20] , and the other top graph based models,namely MHQA [Dhi+18] . Additionally , I have highlighted the pros and cons of top models in comparison with BAG.At the end , there is a final conclusion and research directions.

Authored by:Ramazan Ali Bahrami, Supervised by :Laura Hansel

0.1 Introduction

One interesting task to do with machine learning is question answering . It is quiet a challenging task , and in advanced forms requires good amount of reasoning. However, due to unavailability of data , this field of machine learning has started later than other branches. The Stanford question answering data set; Squad1.0 and Squad2.0 [Raj+16; RZL18] being released in recent years , are considered to be among primary attempts to generate a data-sets with which ML models for question answering or question generation can be trained on. Data-sets such as squad are simple and is considered one hop(given a question, the answer could be found in a single document , paragraph or in a single sentence). In another more advanced forms of question answering , called multi-hop Question Answering , finding the answer requires reasoning across multiple documents.WIKIHOP [WSR17] is one of the many recently published data sets , that aims to provide machine learning trainable data for multi hop question answering. In the next chapter , we will have an overview of WIKIHOP.

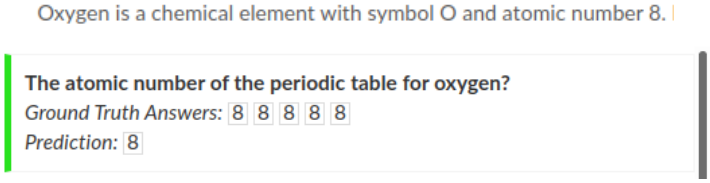

Figure 1 – Single Hop Question Answering [Raj+16]

0.1.1 The Data Set :WIKIHOP

Each instance of the data in WIKIHOP consists of " a query q, a set of supporting documents S_q, and a set of candidate answers C_q – all of which are mentioned in S_q. The goal is to identify

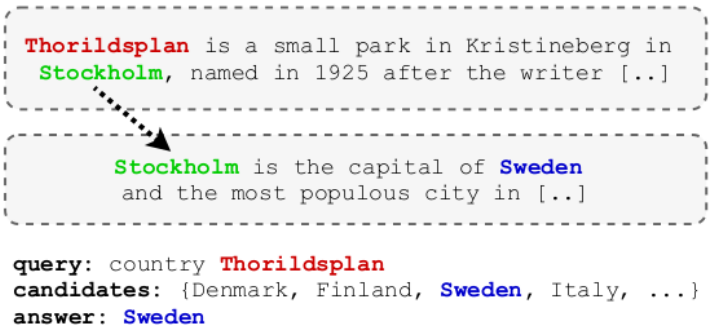

Figure 2 – MultiHop Question Answering[DAT19]

the correct answer $a^{\star} \in C_q$ by drawing on the support documents S_q [WSR17]".It assumes that there is a corpus and a knowledge base. more specifically it assumes Wikipedia as corpus and Wikidata [Vra12] as the knowledge base. "The KB contains triples $< s, r, o >$ where s is a subject entity, o an object entity, and r a unidirectional relation between them" [DAT19].It is important to note that the Knowledge base is used for constructing the data , but it is not available to the model and is not present in the data instance. Additionally not all supporting documents are relevant and some of them are random . Furthermore queries are in the form of triples $< s, r, ? >$ where the model has to find the correct object that satisfies the triple.

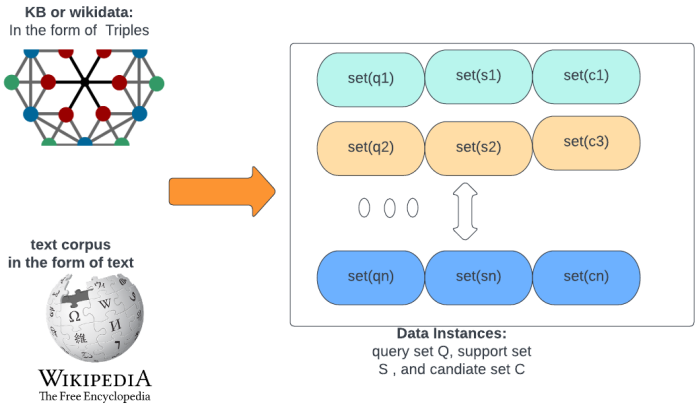

Figure 3 – For generating WikiHop, wikipedia is used as text corpus and wikidata as the knowledge base.

0.1.2 Graph Convolutional Neural Network

Convolutinal neural network is mostly associated with tasks such as image segmentation , object recognition , and ...etc . The principal idea behind convolution is similar to self attention , except that it exploit a build-in structure that is present in images.

We can notice that graphs too have a build in structure, can't we use its build in structure for a convolution ? the answer is yes, and that leads to graph convolutional neural network. In

Figure 4 – Similarity between a convolution and self attention. In self attention all entities are attended to, while in convolution only neighbouring entities are attended to.[Abb23]

both cases, value of a node could be influenced by the neighboring nodes as can be seen from the figure.

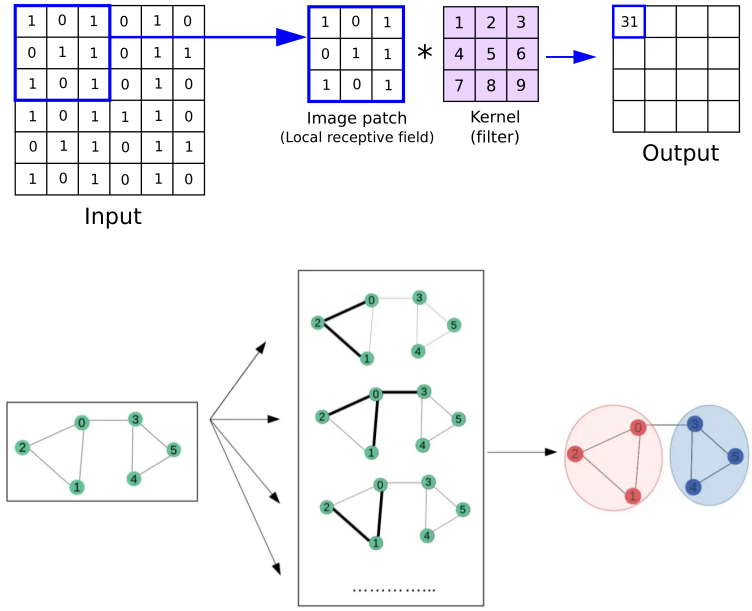

Figure 5 – Similarity between image Convolution and Graph convolution .There is structure to be considered in both images and graph.In images, the structure is implicit , while in graph it could come in many forms, different edges, different incoming edges ...etc. [Rey23; May23]

To formally present graph convolution , consider that a node v at time l has state $h_v^{(l)}$ that is defined by the following equation :

$$h_v^{(l)} = W_l \cdot h_v^{(l-1)} + W_r \cdot AGG(\{h_u^{(l-1)}, \forall u \in N(v)\}) \tag{1}$$

Where $h_v^{(l-1)}$ is its state at time $l-1$, $u \in N(v)$ are neighbors of v , W_l and W_r are learn-able parameters . AGG is an aggregation function such as max pooling or sum.

0.2 Related Work

There are several works that uses graph for solving multi-hop reasoning , the most notable graph based model at the time of BAG publication being [DAT19]. However according to the author of BAG[CFT19] , they "care less about input features and the attention between queries and graph nodes"[CFT19]. Additionally the authors think , attention is very important for the performance of NLP task and especially bidirectional attention [Seo+16] "shows its superiority to vanilla mutual attention because it provides complementary information to each other for both contexts and queries. However, little work exploits the attention between graphs and queries"[CFT19].Furthermore,as it can be seen from figure 17, at the time of writing of this report , transformers are having the best performance for multi-hop question answering. Bigbird [Zah+20] , and Longformer [BPC20] having the best accuracy are both transformers . Among top models , only MHQA [Son+18] is based on graph. Additionally [Dhi+18] proposes a recurrent layer in RNN to overcome RNN's bias toward short term dependency . "They uses an external co-reference annotations extracted from an external library to connect entity mentions belonging to the same cluster"[Dhi+18].

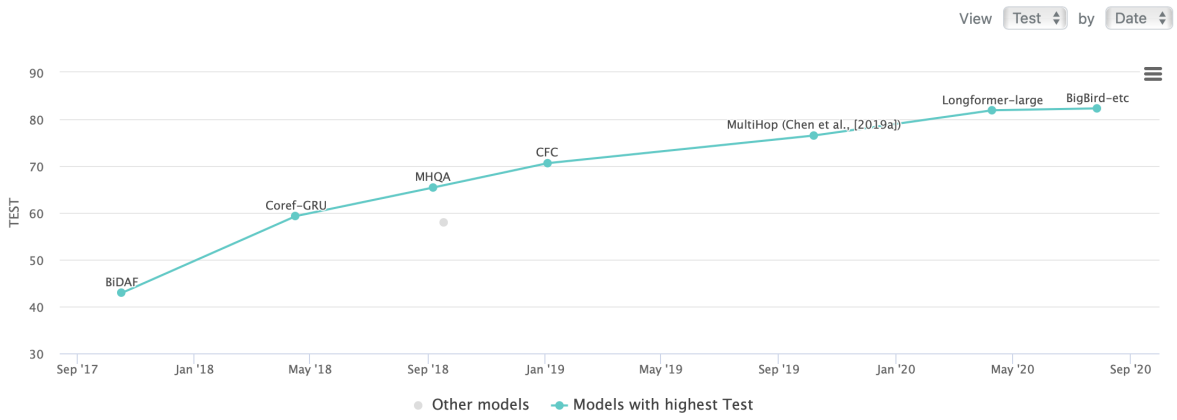

Figure 6 – https://paperswithcode.com/dataset/wikihop

0.3 BAG

As it can be seen from the figure 7 the model first construct the entity graph . The input for entity graph construction are the supporting documents set S and candidate set C respectively .In doing so, for every node a rich set of features : [PSM14] , [Pet+18], POS and NER are are considered. The same types of features are concatenated for the query.Feature and edge information are feed into the GCN network, and then the output is passed into the bidirectional attention , where it is processed alongside features for query that has passed from a 2 layer bi-lstm network. The output from bi-directional attention is then feed into a 2 layer feed forward network that generates the output probabilities . We will have a detailed look into each of these phases.

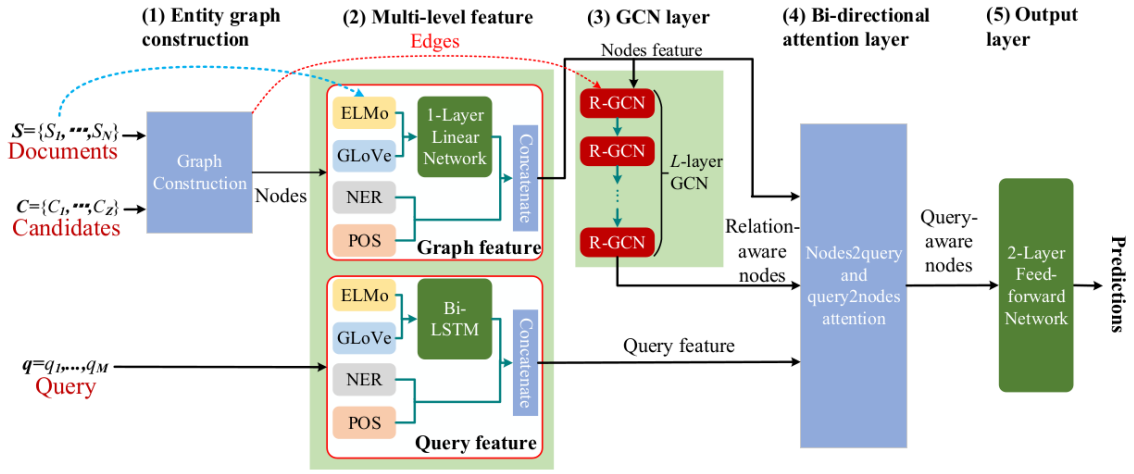

Figure 7 – BAG Architecture [CFT19]

0.3.1 Entity Graph Construction :

The Graph construction in this paper is similar to that of [DAT19].Every document is checked against the candidate set c and all mentions of candidates are considered as nodes . As can be seen from the Figure 11 , exact matches are connected both inside a document and between documents. Furthermore , nodes that are in the same co-reference chain, are connected too (an external library has been used to identify co-reference chains). [CFT19].

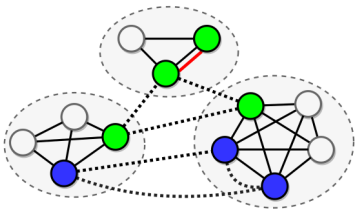

Figure 8 – Entity Graph Overview,Every circle in this document represent a document. [DAT19]

TO illustrate the process, lets consider the documents in figure 9 , and construct the corresponding sample graph of entities 10. As it can be seen , there are 3 documents , and our candidate set, options= $\{Iran, India, Pakistan, Somalia, ...\}$ has 4 entities. Searching for these entities in documents, we find all exact matches of these entities, plus their co-reference chains(They have an external library for this purpose). Here, "India" and "Arabian Sea" both are connected with "Mumbai" in tow different sentences , hence are having the same co-reference, and therefor linked with co-reference edge.

0.3.2 Multi-level feature

The author uses a concatenation of multiple features in order to benefit from each .Features from Glove [PSM14] (pre-trained) ,and Elmo [Pet+18] (calculated based on original documents) is concatenated and feed into a one layer encoder . if a node contains more than one token ,

Figure 9 – Documents, Query and Options [DAT19]

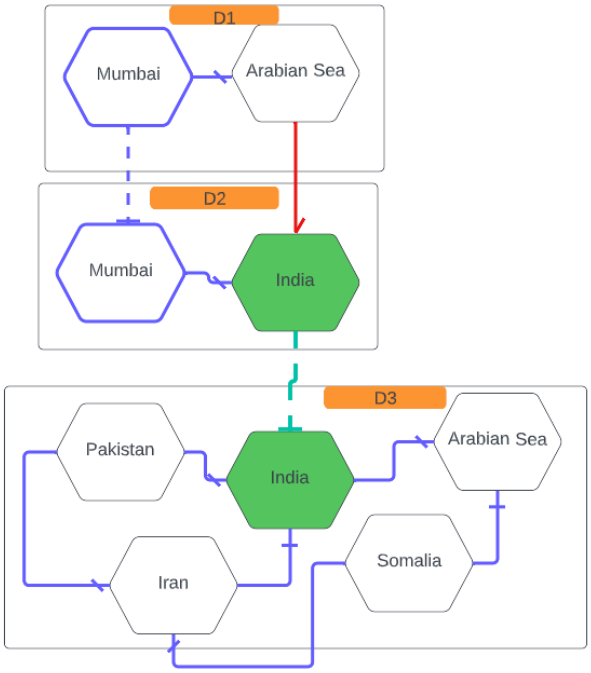

Figure 10 – Constructed Graph

node's feature is the average of tokens(for example 'Arabian Sea' has two tokens in figure 10). For query, features from Glove [PSM14] (pre-trained) ,and ELmo [Pet+18] are concatenated and feed into a bi-lstm network. The output dimension of the one layer encoder and the bi-lstm are both $\tilde{d}$ [CFT19].

The output from the one-layer encoder and the bi-lstm is concatenated with the corresponding POS and NER information, and thus resulting in an output of dimension $d = \tilde{d} + d_{pos} + d_{NER}$[CFT19].

0.3.2.1 Glove and ELMO

Glove [PSM14] ,and ELmo [Pet+18] both are word embedding models.In other words, given a word , they will give you a vector such that similar words will be in vicinity . However ,there

is also a difference between them , and that is in the way they obtain the corresponding vector of a word . One notable consequence of different approach of calculating the mentioned vector is that Elmo gives a contextualized vector, while Gloves gives a vector that points to word's general concept or meaning . This is important as in many cases , the meaning of a word depends on the context.

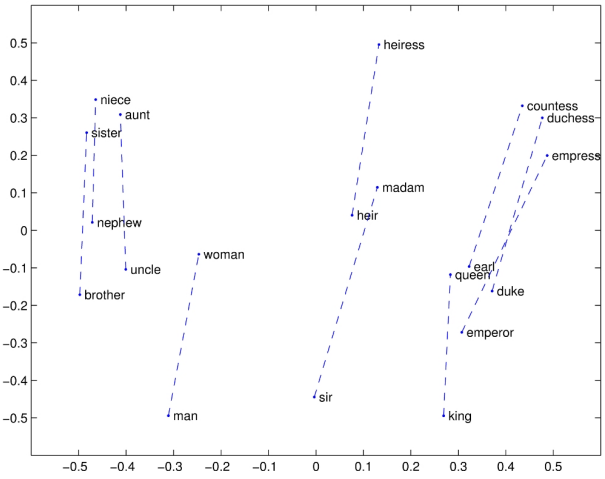

Figure 11 – Given a word , Glove will give you a meaningful vector [PSM14]

0.3.3 RGCN

If every node is initialized with features obtained from multi-feature module previously explained, then at each layer of GCN network, the hidden state $h_i^{(l+1)} \in R^d$ of node i, is defined as below [CFT19] :

$$h_v^{(l+1)} = \sigma(\sum_{r \in R_{N_i}} \sum_{j \in N_i} \frac{1}{C_{i,r}} W_r^l h_j^l + W_0^l h_i^l) \tag{2}$$

Here $h_j^{(l)} \in R^d$ for all $j \in N_j$, accounts for hidden states of neighbours.$C_{i,r}$ is the normalization constant.Additionally R_{N_i} are the corresponding relations[CFT19]. Additionally ,a gate on update vector $u_i^{(l)}$ and hidden state $h_i^{(l)}$ of current node by a linear transformation f_s is used to control the flow of information[CFT19].

$$W_i^{(l)} = \sigma(f_s(concat(u_i^l, h_i^l))) \tag{3}$$

Here $u_i^{(l)}$ can be obtained via eq (2) without sigmoid function.$W_i^{(l)}$ will then be used as updating weights for hidden state $h_v^{(l+1)}$

$$h_i^{(l+1)} = W_i^{(l)} \bigodot tanh(u_i^{(l)}) + (1 - W_i^{(l)}) \bigodot h_i^{(l)}. \tag{4}$$

To put it simply , given a node i, its relations with neighbouring nodes , and the neighbouring nodes hidden state, in every gcn layer , an update vector that is a relation depended weighted sum of node i's hidden state and node i's neighbours hidden state is calculated. The update vector is then summed with hidden state $h_j^{(l)}$ through a gated function that controls influence of update vector u_i^l and hidden state $h_j^{(l)}$ on the new hidden state $h_j^{(l+1)}$[CFT19].

0.3.4 Bi-directional Attention Between a Graph and a Query

0.3.4.1 Attention and Bidirectional Attention

In order to understand bidirectional attention , lets first build the intuition for single directional attention , and then extend it to bidirectional attention used in the paper.lets use the annotated transformer [Ala23] that describe the attention very well . As such we will need keys ,queries and values as described there. The terms key, query and value comes from the database. Given a table, it has primary key, and value , which can be any of the columns in that particular row. Query of course is nothing but query, or that data we are looking for. Now if we put this abstraction into machine learning , we want to learn the query , key and value for each entities . As such , we consider a learn-able weight matrix for each of query , key and value and multiply it with the feature vector to get query , key and value respectively . This can be seen in the figure 12 .

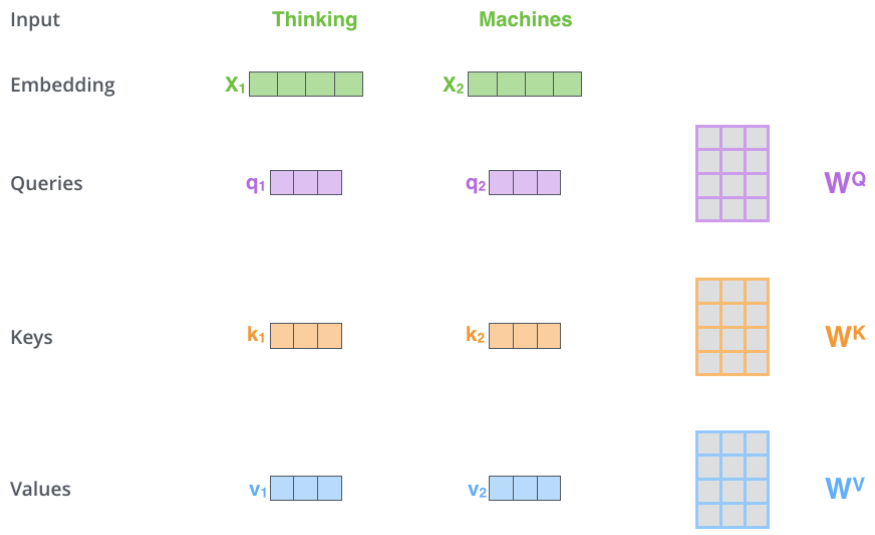

Figure 12 – key , query and value [Ala23]

Next , given two entities x and y , in order to get attention from x to y , we multiply query of x into key and value for y , and put in the equation of the figure 13 , and as such obtain the desired attention.

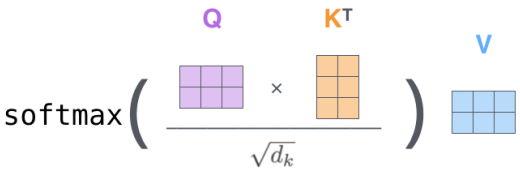

Figure 13 – How attention is calculated with the given key, query and value [Ala23]

To further elaborate , given two entities X and Y , for the attention from Y to X and for attention from X to Y respective , the input will be as in the figure 14 .

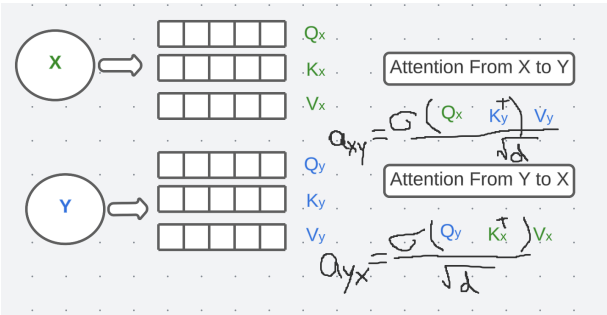

Figure 14 – Attention from X to Y and from Y to X.

Now, we can introduce the bidirectional attention, which is nothing but a_{xy} and a_{yx} for two entities X and Y respectively.

0.3.4.2 Bidirectional Attention in the Paper

In BAG , they have first defined a similarity matrix $S \in R^{T \times M}$ [DAT19] as :

$$S = avg_{-i} f_a(concat(h_n, f_q, h_n \star f_q)) \tag{5}$$

Where h_n is the output from last GCN layer , f_q is the query feature from bi-lstm , $\star$ is element wise multiplication, and avg_{-1} is the average function in last dimension. Furthermore the node-to-query and query-to-node attention is defined as follows [DAT19] :

$$\tilde{a}_{n2q} = softmax_{col}(S).f_q \tag{6}$$

$$\tilde{a}_{q2n} = dup(softmax_{col}(S))^T.f_q) \tag{7}$$

Where $softmax_{col}$ is the softmax function across the column , the dot "." is matrix multiplication and dup is duplication function that turns $\tilde{a_{q2n}} \in R^{1 \times M}$ into $R^{T \times M}$ [DAT19] .

Finally the output for this layer is defined as :

$$\tilde{a} = concat(f_n, \tilde{a}_{n2q}, f_n \star \tilde{a}_{n2q}, f_n \star \tilde{a}_{q2n}) \tag{8}$$

0.4 Output layer

"A 2 fully connected layer is used to generate the final predictions" [DAT19].

0.4.1 Result

As it can be seen from the figure 15 , BAG outperforms all other models at the time of publication . Both MHQA-GRN and Entity GCN , that are graph based model, has lower accuracy than BAG.

Additionally ablation study shows 16 that ELMO has a significant rule in the better performance of the model in comparison with other factors.

Models	Unmasked			Masked	
	dev	test	test[1]	dev	test[1]
FastQA	27.2*	-	38.5	38.0*	48.3
BiDAF	49.7*	-	45.2	59.8*	57.5
Coref-GRU†	56.0*	59.3	57.2	-	-
MHQA-GRN‡	62.8*	65.4	-	-	-
Entity-GCN	64.8*	67.6	63.1	70.5*	68.1
BAG	**66.5**	**69.0**	**65.7**	**70.9**	**68.9**

* Means results reported in original papers, others are obtained by running official code.
† Masked data is not suitable for coreference parsing.
‡ Some results are missing due to unavailability of source code.

Figure 15 – Comparative Performance of BAG , that outperforms state of the art at the time of publication of [CFT19]

Models	Unmasked
Without Attention	63.1
Using Single Attention	64.5
Without GCN	63.3
Without edge type	63.9
Without NER, POS	66.0
+Without ELMo	60.5
Full Model	**66.5**

Figure 16 – Ablation study showing the significance of each parts of BAG. ELmo has the highest influence on accuracy [CFT19]

0.5 Comparison

What I understood is that " rich feature vectors and bidirectional attention is secrete sauce of BAG" , and the reason for its comparatively good performance . As such, I will compare it with top two models, while focusing on the way they have drives their feature vector, and the way they deal with attention.
The top model as of the time of writing of this report is BigBird[Zah+20]. Although BAG [CFT19] has a better performance than MHQA-GRN [Son+18] , but it is not clear why it is not showing in the diagram, and MHQA-GRN can be seen as the best graph based model (Perhaps BAG is not submitted to the platform).

0.5.1 BigBird

BigBird addresses the sequence length limitation(in terms of memory) of full attention mechanism with a sparse attention mechanism that reduces the quadratic dependency of memory on sequence length to linear[Zah+20] .Bigbird treats question answering problem as sequence problem and proves that the proposed solution is "a universal approximator of sequence function and is Turing complete"[Zah+20].

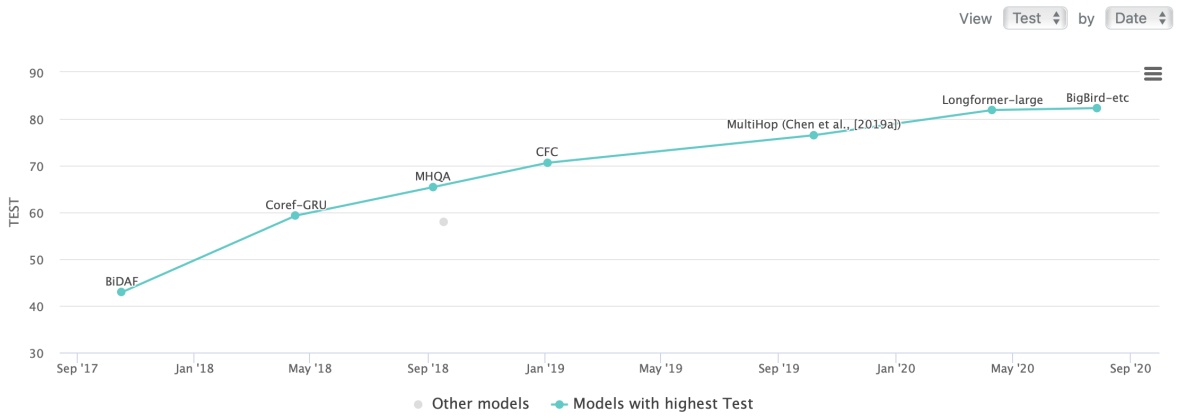

Figure 17 – https://paperswithcode.com/dataset/wikihop

0.5.1.1 Attention

Given the input sequence $X = (x_1, ..., x_n) \in R^{nd}$, the attention mechanism can be described with the help of a graph,whose vertex is the set [n] = 1, . . . , n, the set of directed edges from a vertex of the graph represent entities the node will attend to. If $N(i)$ are nodes to which node i has an edge , then the output vector of the attention mechanism is defined as[Zah+20] :

$$Att_D(X)_i = x_i + \sum_{h=1}^{H} \sigma(Q_h(x_i)K_h(X_{N_i})^T)).V_h(X_{N_i})$$

(9)

"where Q_h, $K_h : R^d \to R^m$ are query and key functions respectively, $V_h : R^d \to R^d$ is a value function, σ is a scoring function (e.g. softmax or hardmax) and H denotes the number of heads. Also note X_{N_i} corresponds to the matrix formed by only stacking $\{x_j : j \in N(i)\}$ and not all the inputs"[Zah+20].

If A is the adjacency matrix of graph D , in cell (i,j) it has value 1 if node i attends to node

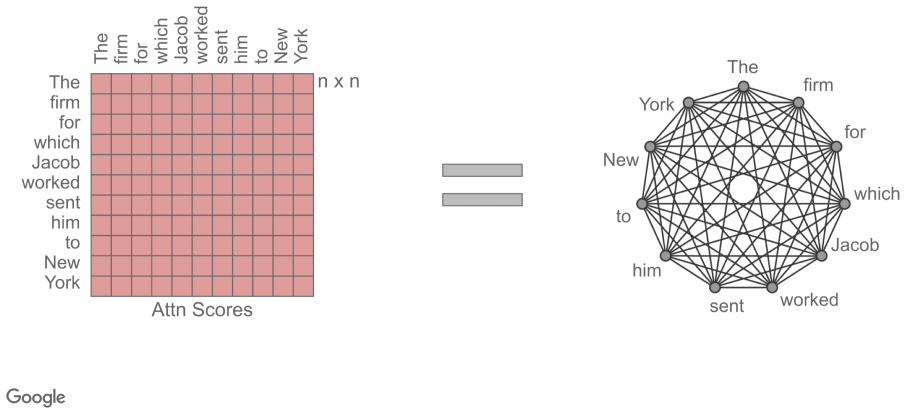

Figure 18 – Attention and Graph [Goo23].

j , and zero otherwise. In case of full attention every node attends to every other nodes , and all entry of this matrix will be one . In BigBird , they take advantage of this analogy and

simply try to reduce the matrix D to a sparse one that can approximate the complete graph.Given that from graph theoretic point of view, it is possible, BigBird reduces the quadratic complexity into a linear one. They also take help locality of reference (Window Attention) and global tokens (Global Attention) 20. I am not going into details of their approach , and just accept the intuition behind the sparse attention as a bases for comparing it with BAG .

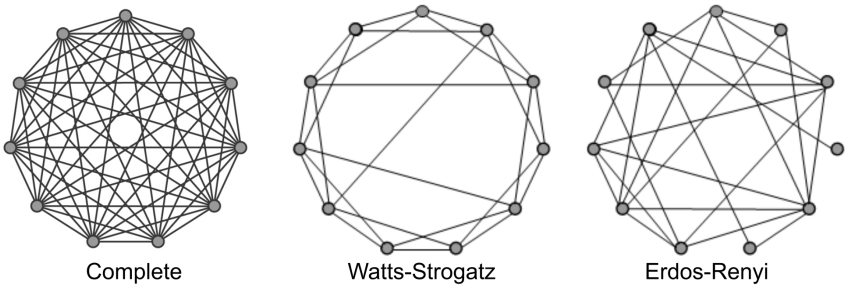

Figure 19 – A Motivation for Sparse Attention [Goo23].

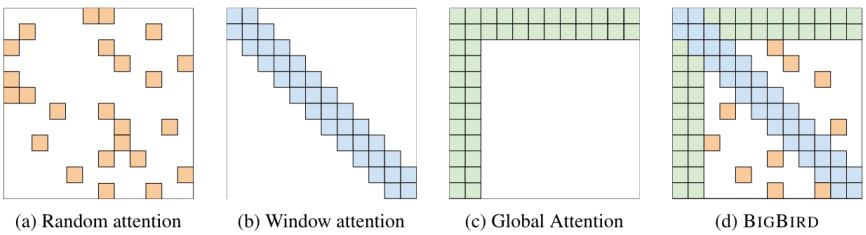

Figure 20 – ” Building blocks of the attention mechanism used in BIGBIRD. White color indicates absence of attention. (a) random attention with r = 2, (b) sliding window attention with w = 3 (c) global attention with g = 2. (d) the combined BIGBIRD model”[Goo23].

Insights and Comparison with BAG

Obviously the attention calculated as here in BigBird is revolutionary . This is indeed a very good idea , for sequence to sequence task, and might be even possible to be used for graph convolutional neural network. however, in my view logical question answering or multi-hop question answering for that matter is not a sequence to sequence problem, the type of task which is addressed by BigBird. One obvious reason for that ,is that there is no order or sequence for output.As such, given a set of passages D_i , and a query q_j , and set of candidates C_k , the order of candidates does not matter. In other words , logical question answering have an underlying topology that best fits with graph structures. Moreover , by treating the problem as that of reasoning over graph and hence GCN , BAG already has reduced the complexity , as only concerned entities are involved. Furthermore , the amount of training of BigBird and BAG is not comparable. In short , although BigBird has higher accuracy than BAG, but in logical question answering , a graph based model such as BAG seem to have more potential for improvements than BigBird .

0.5.2 MHQA

MHQA [Dhi+18] combines global evidence obtained by encoding the passage into a hidden state with local information from the topology of the entities and the way the entities are connected in each passage.In other words, MHQA creates a graph similar to BAG, but in addition uses the hidden state of the passage in calculating the feature vectors for each entity in the graph. In doing so, first a DAGs is created,where entities are connected with three different types of edges namely , exact match in the same passage, co-reference edges and within context window [Son+18].

0.5.2.1 Attention

Features obtained for entities as mentioned above, is then feed into "additive attention" model. in doing so , the model, "additive attention" treats all entity mentions as memory and question entities as query [Son+18]. "The probability for an entity ϵ being the answer is calculated by summing up all the occurrences of ϵ across the input passages"[Son+18].

$$Pr_e = \frac{\sum_{k \in N_\epsilon} \alpha_k}{\sum_{\bar{k} \in N} \alpha_{\bar{k}}} \tag{10}$$

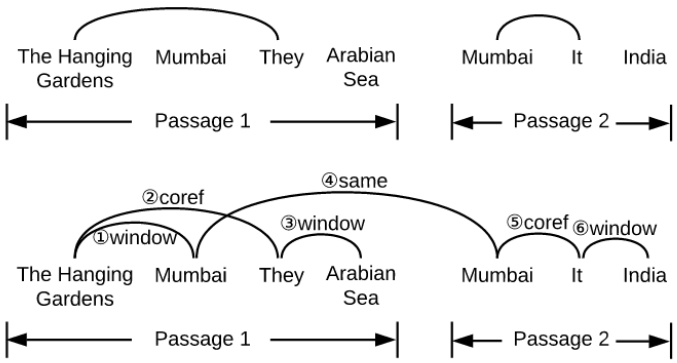

Figure 21 – A DAG generated by [Dhi+18] (top) and a graph by considering all three types of edges (bottom) on the example

"where N_ϵ and N represent all occurrences of entity ϵ and all occurrences of all entities, respectively. Previous work [Wan+17] shows that summing the probabilities over all occurrences of the same entity mention is important for the multi-passage scenario. α_k is the attention score for the entity mention ϵ_k , calculated by an additive attention model shown below"

$$e_0^{(k)} = v_a^{(k)} tanh(W_a h_\epsilon^{(k)} * + U_a h_q + b_a) \tag{11}$$

$$\alpha_k = \frac{exp(e_0^{(k)})}{\sum_{\bar{k} \in N_\epsilon} exp(e_0^{\bar{k})}} \tag{12}$$

where V_a, W_a, U_a and b_a are model parameters.

To summarize , with the use of a BILSTM network, in MHQA , the node's hidden state h_ϵ contains information of node's different contexts (an entity might appear in several passage and location) and also the information from the neighbouring nodes in the constructed DAG. In terms of attention,the three edge types mentioned above, enable the model to attend to local context of entities in different passages and also to co-referenced entities .

Insights and Comparison with BAG

MHQA and BAG are very much alike. First they are both graph based models. Second , they both use three different types of edges in the their graph. However the edge types are not completely similar. In MHQA , we have context edge, while in BAG we have cross document edge. furthermore , BAG uses a rich set of features namely embeddings from GLOVE, ELMO , and information from POS and NER taggers. MHQA in the other hand, uses only embedding from GLOVE. Additionally, for BAG, ablation study shows multi-feature have a significant influence on accuracy. As such the reason for better performance of BAG could lie in rich feature vectors . Moreover, the attention calculated in MHQA is inspiring, but the bi-directional attention in BAG does not have much significance as it is shown in their ablation study . Another interesting similarity , is that between global token in BigBird , and global Evidence in MHQA.In addition the local attention in BigBird is analogous to context window in MHQA.

0.6 Conclusion and Result

Considering the low training time of BAG (14 hours on GTX1080Ti GPUs) , its performance seems convincing. In addition, logical question answering shows some structures or topology between entities that is best resembled by graph . Hence for logical question answering , a graph based model is more suitable than a sequence to sequence model. As such, BAG and other graph based models , might have more potential to improve than a sequence to sequence model.In the other hand , the rich set of features in BAG although significantly improves performance , but doubles feature dimension; and makes BAG costlier in terms of computation and memory. Additionally the type of edges in BAG could be improved with the context window edges , as is used in MHQA.

Last, but not least is that ;it would be interesting to study how or weather sparse attention in BigBird can or could be used in graph convolutional neural network.

Bibliography

[Vra12] Denny Vrandečić. "Wikidata: A New Platform for Collaborative Data Collection". In: *Proceedings of the 21st International Conference on World Wide Web*. WWW '12 Companion. Lyon, France: Association for Computing Machinery, 2012, pp. 1063–1064. ISBN: 9781450312301. DOI: 10.1145/2187980.2188242. URL: https://doi.org/10.1145/2187980.2188242.

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher Manning. "GloVe: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. DOI: 10.3115/v1/D14-1162. URL: https://aclanthology.org/D14-1162.

[Raj+16] Pranav Rajpurkar et al. "SQuAD: 100,000+ Questions for Machine Comprehension of Text". In: *arXiv e-prints*, arXiv:1606.05250 (2016), arXiv:1606.05250. arXiv: 1606.05250.

[Seo+16] Min Joon Seo et al. "Bidirectional Attention Flow for Machine Comprehension". In: *CoRR* abs/1611.01603 (2016). arXiv: 1611.01603. URL: http://arxiv.org/abs/1611.01603.

[Wan+17] Shuohang Wang et al. "R^3: Reinforced Reader-Ranker for Open-Domain Question Answering". In: *CoRR* abs/1709.00023 (2017). arXiv: 1709.00023. URL: http://arxiv.org/abs/1709.00023.

[WSR17] Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. "Constructing Datasets for Multi-hop Reading Comprehension Across Documents". In: *CoRR* abs/1710.06481 (2017). arXiv: 1710.06481. URL: http://arxiv.org/abs/1710.06481.

[Dhi+18] Bhuwan Dhingra et al. "Neural Models for Reasoning over Multiple Mentions using Coreference". In: *CoRR* abs/1804.05922 (2018). arXiv: 1804.05922. URL: http://arxiv.org/abs/1804.05922.

[Pet+18] Matthew E. Peters et al. "Deep contextualized word representations". In: *CoRR* abs/1802.05365 (2018). arXiv: 1802.05365. URL: http://arxiv.org/abs/1802.05365.

[RZL18] Pranav Rajpurkar, Jian Zhang, and Percy Liang. "Know What You Don't Know: Unanswerable Questions for SQuAD". In: *ACL 2018*. 2018.

[Son+18] Linfeng Song et al. "Exploring Graph-structured Passage Representation for Multi-hop Reading Comprehension with Graph Neural Networks". In: *CoRR* abs/1809.02040 (2018). arXiv: 1809.02040. URL: http://arxiv.org/abs/1809.02040.

[CFT19] Yu Cao, Meng Fang, and Dacheng Tao. "BAG: Bi-directional Attention Entity Graph Convolutional Network for Multi-hop Reasoning Question Answering". In: *CoRR* abs/1904.04969 (2019). arXiv: 1904.04969. URL: http://arxiv.org/abs/1904.04969.

[DAT19] Nicola De Cao, Wilker Aziz, and Ivan Titov. "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 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 2306–2317. DOI: 10.18653/v1/N19-1240. URL: https://aclanthology.org/N19-1240.

[BPC20] Iz Beltagy, Matthew E. Peters, and Arman Cohan. "Longformer: The Long-Document Transformer". In: *CoRR* abs/2004.05150 (2020). arXiv: 2004.05150. URL: https://arxiv.org/abs/2004.05150.

[Zah+20] Manzil Zaheer et al. "Big Bird: Transformers for Longer Sequences". In: *CoRR* abs/2007.14062 (2020). arXiv: 2007.14062. URL: https://arxiv.org/abs/2007.14062.

[Abb23] Pieter Abbeel. *Deep Unsupervised Learning*. https://sites.google.com/view/berkeley-cs294-158-sp20/home. 2023.

[Ala23] Jay Alammar. *The Illustrated Transformer*. https://jalammar.github.io/illustrated-transformer/. 2023.

[Goo23] Google. *BigBird Presentation*. https://docs.google.com/presentation/d/1FdMNqG2b8XYc89_v7-_2sba7Iz6YAlXXWuMxUbrKFK0/preview?resourcekey=0-KHcdpCx83g7a2JNz0h0-6w&slide=id.g9dee1ebbe6_0_200. 2023.

[May23] Inneke Mayachita. *Understanding Graph Convolutional Networks for Node Classification*. https://towardsdatascience.com/understanding-graph-convolutional-networks-for-node-classification-a2bfdb7aba7b. 2023.

[Rey23] Anh H Reynolds. *Convolutional Neural Networks CNNs*. https://paperswithcode.com/sota/question-answering-on-wikihop. 2023.