Faculty for Computer Science, Electrical Engineering and Mathematics
Department of Computer Science
Seminar I

Open Domain Question Answering Using Early Fusion of Knowledge Bases and Text

by
Garima Mudgal

submitted to
Prof. Dr. Axel Ngonga

Supervisor
and
Second Examiner
Diego Moussallem

Paderborn, August 2, 2019

Report Seminar Natural Language Processing by Garima Mudgal
Supervisor: Diego Moussallem

First examiner: Prof. Dr. Axel Ngonga
Second examiner: Diego Moussallem

Abstract

Open Domain Question Answering has come a long way over the past years. Many advance models have been developed that use KBs and Text to extract the answers to the questions posed in natural language. Most of the models limit themselves to just one of the two information sources i.e. either KB or Text. It hinders the model to use its potential and make use of the relations between KBs and Text. GRAFT-Nets model is introduced to this area and it makes use of the graph representation learning to hold both types of data sources as nodes. Graph representation also helps the model to consider multiple answers to one question. It is proved using the statistics that GRAFT-Nets outperforms most of the state of the art models.

Open Domain Question Answering Using Early Fusion of Knowledge Bases and Text

Garima Mudgal

August 2, 2019

Contents

1 Introduction

Open Domain Question Answering, a paradigm of Natural Language Processing that deals with extraction of answers w.r.t. a question posed in natural language. Open Domain deals with questions from any topic that exists in the world. Since there is no constrain about the domain, the information sources for the answer extraction are huge. The models developed for Question Answering use two types of sources :- (a) Knowledge Bases and (b) Text. Information sources have their own pros and cons. Text is rich in relations and has high coverage whereas KBs have a defined schema and that makes it easier to extract the answers from it. Text suffers as it doesn't have a fix schema and that makes it difficult to extract answers. On other hand, KBs suffer from low coverage due to evident reasons.KBs have their restrictions in terms of completeness as we build the KBs for a specific purpose and train our model just for our purpose. Most of the models developed by the researchers in this paradigm work with either text alone or KB alone. Further more ,models in which both text and

KBs were considered didn't use the full potential of the knowledge. Das et. al. used both the sources but they didn't make use of the rich relational structure offered by text in the Key Value Neural Network Model (KV-MemNN). They used two separate models, one for text and other for KB, the results from the two were merged in the later stages of the model and using some heuristics function the answer is extracted. This model fuses the two source information in the later stages , hence this technique is called Late Fusion.
The Question Answering models earlier used complicated pipelined systems for answer extraction[Ferrucci et al., 2010]. Eventually we moved to en-to-end deep neural networks that used either KB or text. Currently, graph representation learning is used in the model. Late Fusion model used the graph representation learning.
To make use of text in places where we don't have complete KBs, a model called GRAFT-Nets is developed. It has a single model that is trained to extract answers from the question subgraph containing KB facts and text sentences.[Sun et al., 2018]
GRAFT-Nets can be seen as an enhancement of the KV-MemNN model. The model takes care of both type of nodes i.e. KB entities and text. Text can be variable in length as it is a document retrieved from the information source. The model learns the node representation and assigns score to each node. The node with highest probability is chosen as the answer to the question. The technique used in the model is called Early Fusion. GRAFT-Nets makes use of the rich relational structure offered by the text documents.
GRAFT-Nets (Graphs of Relations Among Facts and Text Networks) is a convolution based neural network(CNN) in which we train the network using neurons and control the overfitting of data by means of controlled weights . GRAFT-Net is based on the work of graph representation learning [Kipf and Welling, 2016], [Schlichtkrull et al., 2018]. However, there are two main updates in the model :-

1. Heterogeneous update rules
2. Directed propagation method

Heterogeneous update will make sure that text nodes are updated differently than the KB nodes. Directed propagation method [Haveliwala, 2002],constrains the propagation of embeddings in the graph to follow paths starting from seed nodes linked to the question. The graph maintains a scalar PageRank score for each node in the graph and it spreads out from the seed node. More about the model is explained in the next section.

2 Formal Notation

GRAFT-Nets has a natural question (q) as the input which is represented as $q = w_1, w_2, ..., w_{|}q_{|}$. First information source is : The knowledge base (*V, E, R*) . It is represented in terms of entities(V), edges that connect these entities (E) and the relations among these entities(R). The edge is represented as a triplet

(s,r,o), where s is the subject which is related to the object o with a relation r.

$$s \in V, o \in V, r \in R$$

Second Information source is : The text corpus D, which is a collection of documents and each document is a collection of words. Documents can variable in size and the difference in their size can be huge. We can represent the collection of documents as : $D = \{d_1, d_2, ..., d_{|D|}\}$. After entity linkage is done over the documents, we get a set of linked entities, which is represented asL_d .L_d denotes the set of all linked entities in the document d, where d is represented as a sequence of words $d_i = (w_1, w_2, ..., w_{|d_i|})$. In order to get these links, each of the documents is fed to the program that links all the entities present in the document. Output of such a run is the set L of links (v, d_p), that informs that the entity v is connected to the word at position p in the document d. Once we have the question q and the entity set L, we are ready to extract the answer $\{a\}_q$ from the subgraph G = (K, D,L). The answers are extracted either from the text or KB. We might end up with a no of answers depending on the documents and the text that we have. The models presented in this paper experiments with text and KB in a number of different settings. The models use different fusions in order to demonstrate which settings give the most promising recall and precision values. [Sun et al., 2018] The experiment can be divided into two major sub tasks for training the model and extracting the subgraphs that contain the answers. These techniques are used by Dhingra et al., 2017 in the search-and-read paradigm for text-based QA . First step is extracting G_q with high probability from the entity linked graph G that we already have. Second step is used for training the model with the node representation used in G_q conditioned on q. Distant supervision is used to train the model , which mimics the training data that we have and helps in generalising the text , in cases when the model has not seen the exact same text before but a similar relationship has been used to train the data.

2.1 Retrieving question graphs

Different methods are used for searching and extracting (a) relevant documents from the Text , and (b) entities from the KB K. Once the documents and entities are retrieved, they are combined using entity links and a final graph G is produced which is completely connected.

2.1.1 Text Retrieval

Graph representation needs retrieval of the text documents from the various information sources and in this model, it is Wikipedia. The text retrieved is called as the document, which is a sequence of words. A document can be of variable length. The text retrieval process can be divided into 2 steps :-
(a) Extraction of top 5 documents from Wikipedia along with their titles using the DrQA model [Chen et al., 2017].

(b) Producing the top documents that we extracted in step (a) along with their titles in the form of Lucene index. The title is important to be added to the index because most sentences in the article are about the entities that are mentioned in the title.
The top entities from the Lucene index are selected along with any linked entities and are populated in the form of a question subgraph G_q.
Graph is represented as :: G_q : (V_q, E_q, R^+)[Sun et al., 2018], where
V_q : consists of all retrieved entities and documents.

$$V_q = \{v_1, v_2, ..., v_E\} \cup \{d_1, d_2, .., d_D\}$$

E_q : all relations from K among these entities, plus the entity links between documents and entities.

$$E_q = \{(s, o, r)\} \in E : s, o \in V_q, r \in R\} \cup \{v, d_p, r_L) : (v, d_p) \in L_d, d \in V_q\}$$

R^+ : Set of all edge types in the subgraph.

$$R^+ = R \cup \{r_L\}$$

2.1.2 KB Retrieval

To retrieve the entities from the KB , we need to define the seed entities that are the words from the question. We use the methods presented by Haveliwala [Haveliwala, 2002], a set of PageRank vectors, each biased with a different topic to create for each page a set of importance scores with respect to particular topics. The PPR is run around the seeds to get other entities that are important for the answer extraction. Initially only the seed entities are assigned the weights. When the PPR model is run, the weights of the neighbouring entities are updated. The outgoing edges that are of the same type get the same score. The entities that more relevant to the question are assigned higher score than the rest. In this process, two vectors are calculated, one by averaging the word vector in order to compute the relation vector v(r) and second is the question vector calculated from the words in the question v(q). Using the cosine similarity between v(r) and v(q), we can have the edge weights. PPR returns a top E entities and the edges between them which can be added to the graph G_q.

2.2 Datasets

The system is modelled using 2 main Datasets :-

1. WikiMovies Dataset

The dataset uses the KB and Text corpus alongside to extract the answers to the question. It is created by Miller et al. from the Wikimovies domain. They have used Key-Value Memory Network that makes the reading viable. WikiMovies is a analysis Dataset to bridge the gaps between incomplete KBs and unstructured text. For the GRAFT-Nets question subgraph, 50 top entities are retrieved

around the seeds using simple surface level matches. In addition to that 50 top sentences are added to the subgraph using Lucene search over the text corpus.This results in an overall recall of 99.6 % for the subgraph.[Sun et al., 2018]

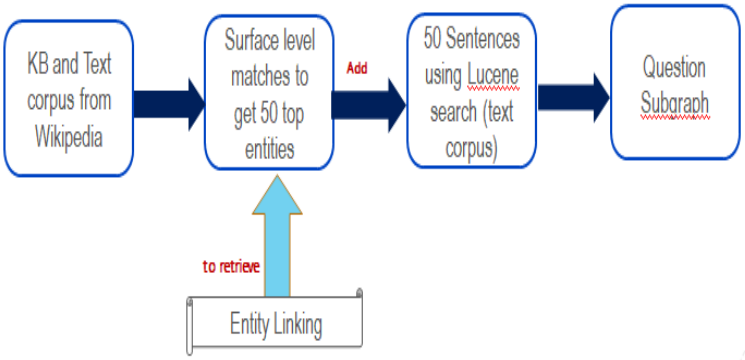

Figure 1: Illustration of the subgraph generation using Dataset WikiMovies

2. WebQuestionsSP

This dataset is created by Yih et al. , it focuses on labelled semantic parsing of the freebase text. Their model collected one of the largest datasets for QA and also showcased how parsing can be done at a low cost . Semantic parsing improves the accuracy, consistency and efficiency of obtaining answers.
For GRAFT-Nets, 250 questions from the dataset were used for training and early stopping. The entity linking outputs were obtained from S-MART and 500 entities from the neighbourhood around the question seeds to populate the question subgraph.

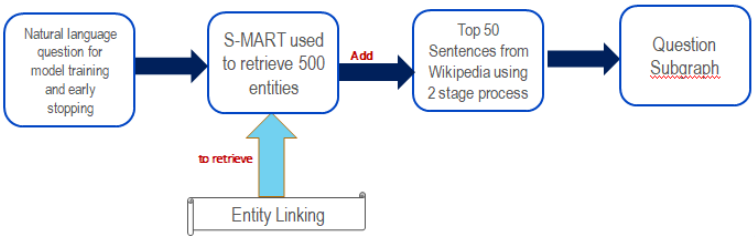

Figure 2: Illustration of the subgraph generation using WebQuestionsSP Dataset

Since the real domains can have different levels of completeness, GRAFT-Nets is tested under 10%, 30% and 50% KB completeness settings. This is done to check the robustness of the model and see how it performs in these settings. We can observe the statistics from both these datasets in the below table. We will compare the results in the Comparison section.
It is evident from the data that WebQuestionsSP has more types of relations than the WikiMovies dataset. The entities present in WebQuestionsSP is also far more than those present in WikiMovies.

Dataset	# train / dev / test	# entity nodes	# edge types	# document nodes	# question vocab
WikiMovies-10K	10K / 10K / 10K	43,233	9	79,728	1759
WebQuestionsSP	2848 / 250 / 1639	528,617	513	235,567	3781

table 1 : Statistics of all the retrieved subgraphs $\cup_q G_q$ for WikiMovies-10K and WebQuestionsSP.[Sun et al., 2018]

2.3 GRAFT-Nets Implementation

GRAFT-Nets model uses the basic gather-apply-scatter paradigm . The graph generated using the method described in the previous section , is used to extract the answers. The nodes in the subgraph are labeled as 1 if the score of the node is above a certain threshold and 0 otherwise. After this point, it is just a binary classification task to extract the answer. First, the basic model is explained in two steps:-

1. Initialise node representations $h_v^{(0)}$.

$$H_d^{(0)} = LSTM(w_1, w_2, ...),$$

and to access the pth word, we can denote it as $H_{d,p}^{(l)}$.

2. Update the node representations for each layer.

$$h_v^{(l)} = \phi\left(h_v^{(l-1)}, \sum_{v' \in N_r(v)} h_{v'}^{(l-1)}\right)$$

The nodes are updated w.r.t. the relation with their neighbour nodes and the previous layer. Maximum no of layers are L.

But, GRAFT-Nets has two differences from the basic models :-

a) **Heterogeneous Updates :**
The nodes represented in the question subgraph can be either the entities that are extracted from the KBs or they can be the documents that have been extracted from the relevant articles. Both these nodes have different features and need to be handled differently. The process is explained in the figure 3.

1. Entity Update : Here single layer Feed Forward Network is used for entity update. The FFN has 4 states. (a) the entity representations ,(b) question representations ,(c) third state aggregates the states from the neighbours of the current node v, that is being handled and (d) fourth state aggregates the states of all tokens , where the entity v is part of some document d. [Sun et al., 2018]

2. Document Update : Document update happens in two steps : (a) aggregation over the entity states coming in at each position, (b) aggregating the states within the document using LSTM.

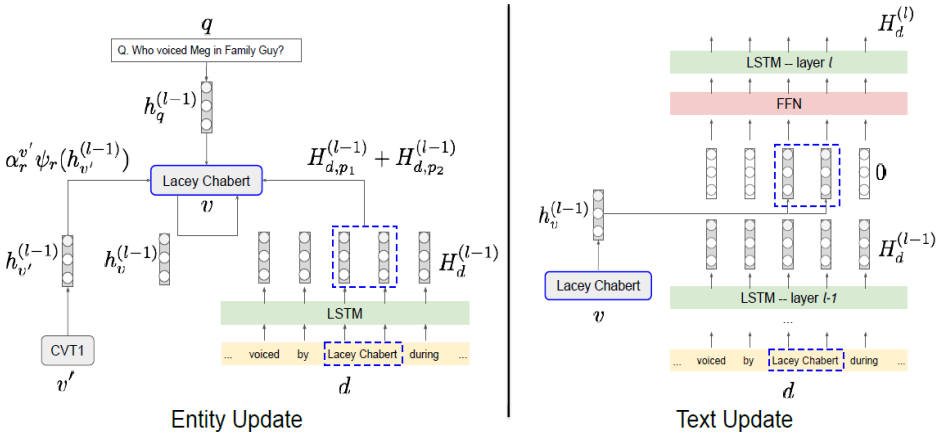

Figure 3 : Illustration of the heterogeneous update rules for entities (left) and text documents (right) [Sun et al., 2018]

b) **Conditioning on the Question :**

Conditioning over questions improves the performance of the model, the effect is explained the Comparison section. There are two main components for conditioning, first is Attention over Relations and second is Directed Propagation.

1. Attention over Relations makes sure that embeddings are propagated more along the edges that are relevant to the question.

2. Directed Propagation : The idea here is to get the entities from the graph that are related to the question. Therefore, in the first layer , the weight is assigned to only the seed entities. Once the propagation happens, it goes in the outwards direction from the seed entities and updates the score of each node that is relevant. The PageRank score method [Haveliwala, 2002], keeps on updating the value of each node as the model propagates outwards.

Once we have the final representation $h_v^{(L)} \in R^n$, we can select the answer using binary classification :

$$Pr(v \in \{a\}_q | G_q, q) = \sigma(w^T h_v^{(L)} + b)$$

Here, σ is the sigmoid function.

3 State of the art

A lot of progress has been done in the Open Domain Question Answering paradigm of Natural Language Processing over the last few decades. Many researchers have worked with answer extraction from the text alone and some

have worked with Knowledge Bases. Very few models have been developed that combine the two resources for answer extraction. KB have desired structure for the task and text has much more facts and information that can lead to a rich relational structure when combined together.
GRAFT-Nets is an extension of the work done by [Das et al., 2017] , who have worked to combine the text and incomplete KB in order to achieve the desired performance in Question Answering. They used the Key-Value Memory Neural Networks for storing the triplets in the memory. The question is asked in the form of blank space. While searching the answer, only key value is mapped and the answer is retrieved from the KV-MemNets. In the KV-MNN, only KB is used to train the model,this limitation from their system has been tried to cover using GRAFT-Nets because KV model ignore the rich relational structure between the facts and text-snippets.

[Gardner and Krishnamurthy, 2017] presented a semantic parser in their paper which is not just limited to KB that has a fixed schema. Their model could build compositional and executable representation of the language. They showed how KB information can be incorporated into open vocabulary semantic parsing models.

Other research is done by [Ryu et al., 2014] in which they have taken Wikipedia as the knowledge base that is semi structured and has information in different formats. Therefore, it is important to first analyse the question that the user is posing. They have collected data and analysed about 600 questions to know the structure and type of questions posed by the users. The model used pipelined system aggregating evidence from both structured and unstructured sources for the open-domain QA.

Verga et al., 2016 [Verga et al., 2015] tried to expand the coverage abilities of universal schema relation extraction. They also made an attempt of multilingual transfer learning, which could cover the learning of a language that is not present in the KB. Common representations of the entities and relations is used in their Model.

Knowledge Base Completion (KBC) is another field that has been explored by many researchers.This is not directly linked with Question Answering but holds an important place in the paradigm. In QA, the search and extract process is based on the question, whether the source is text or KB or both, the question entities made the knowledge graph look different with every new question. The structure for the graph keeps changing. KBC deals with the fixed Schema of the Knowledge Base that is fixed and has very less or no flexibility in the information representation. Lao et al. worked on completing the knowledge bases which were left incomplete by the human curators as they were not confident about the status of some putative fact [Lao et al., 2012]. It uses Path-Ranking Algorithm (PRA)[Lao et al., 2012], which learns such rules on heterogeneous graphs for link prediction tasks.

Scarselli proposed graph Neural Network Model in his paper [Scarselli et al., 2008], the Neural Network used in his model can take care of both graph and node

focused specific application. The model is capable of handling cyclic, acyclic and directed graphs therefore it extends the random walk models and recursive neural networks.

Another interesting work in the field of semi-supervised classification using graph convolution network is presented by Kipf and Welling [Kipf and Welling, 2016]. By using layer-wise propagation rule for the neural network they demonstrated that such a model can be fast and efficient for classifying nodes in a graph.

In 2016 , SQuAD was created by [Rajpurkar et al., 2016] which contains over 100,000 questions posed over Wikipedia. In the dataset that is freely available, the answers to the questions are in the form of segments of text that is taken from the corresponding reading passage.

GraphSAGE model uses the aggregator function to collect the information about a node from its neighbourhood nodes(node profile, node degree and text attributes).Hamilton proposed the inductive node embedding function to embed the information about the unseen nodes [Hamilton et al., 2017].By embedding the node features, it becomes easy to identify the node attributes and its topological structure. They used the structural information of the graph to generalise the unseen nodes. The limitation that GraphSAGE has is that it randomly samples the nodes , whereas, GRAFT-Nets uses heterogeneous mixture of nodes and uses retrieval for getting a subgraph from the combination of text or KB.

Walk-steered convolutions has been used by Jiang for graph classification [Jiang et al., 2018] recently. They used Random walk to construct the local receptive fields of graph filters as they can preserve the graph structure.

GRAFT-Nets is influenced by these work of using graph structured data, where the answer is not presented in one structured form. Rather, the model and the graph network is used iteratively to extract the most relevant documents based on the question posed. Personalisation technique used by GRAFT-Nets is also influenced by Walk-Steered convolution and is used for localising propagation of embeddings. Working with KB and memory networks or reinforcement learning has been achieved to some extent in the scientific world as KB has a pre-defined structure.There are various advancements from the deep learning side using KBs. The main problem comes when we are dealing with the text which is in natural language and unstructured. Lu et al. have implemented Object Oriented Neural Programming (OONP) that semantically parses the documents. OONP can be trained with reinforcement learning and supervised learning for capturing the ontology of the text documents properly.

GRAFT-Net doesn't use any neural network for the textual data, only a simple sequential representation is implemented that is augmented with the entity links to KB.GRAFT-Nets doesn't have the text documents pre-processed, collected before hand. It works with large information sources. However, it has combined KBs also with the text to have a better relational structure.

4 Comparison

There are three major comparisons that can be made with respect to the GRAFT-Nets model.

Model	Text Only	KB + Text			
		10 %	30%	50%	100%
WikiMovies-10K					
KV-KB	–	15.8 / 9.8	44.7 / 30.4	63.8 / 46.4	94.3 / 76.1
KV-EF	50.4 / 40.9	53.6 / 44.0	60.6 / 48.1	75.3 / 59.1	93.8 / 81.4
GN-KB	–	19.7 / 17.3	48.4 / 37.1	67.7 / 58.1	**97.0 / 97.6**
GN-LF	**73.2 / 64.0**	74.5 / 65.4	78.7 / 68.5	83.3 / 74.2	96.5 / 92.0
GN-EF		75.4 / 66.3	82.6 / 71.3	87.6 / 76.2	96.9 / 94.1
GN-EF+LF		**79.0 / 66.7**	**84.6 / 74.2**	**88.4 / 78.6**	**96.8 / 97.3**
WebQuestionsSP					
KV-KB	–	12.5 / 4.3	25.8 / 13.8	33.3 / 21.3	46.7 / 38.6
KV-EF	23.2 / 13.0	24.6 / 14.4	27.0 / 17.7	32.5 / 23.6	40.5 / 30.9
GN-KB	–	15.5 / 6.5	34.9 / 20.4	47.7 / 34.3	66.7 / 62.4
GN-LF	**25.3 / 15.3**	29.8 / 17.0	39.1 / 25.9	46.2 / 35.6	65.4 / 56.8
GN-EF		31.5 / 17.7	40.7 / 25.2	49.9 / 34.7	67.8 / 60.4
GN-EF+LF		**33.3 / 19.3**	**42.5 / 26.7**	**52.3 / 37.4**	**68.7 / 62.3**

Table 2: Hits@1 / F1 scores of GRAFT-Nets (GN) compared to KV-MemNN (KV) in KB only (-KB), early fusion (-EF), and late fusion (-LF) settings.

At First, the base model i.e. Key Value Network Model and the GRAFT-Nets model are compared using both the datasets. The datasets used in the comparisons are generated with different levels of KB completeness to check the robustness of the model and the effect on the overall performance matrix. The experiment is performed with 3 different settings i.e. 10% , 30% and 50%. The performance can be seen from table 2. WikiMovies had all the answers retrievable from one KB, whereas the WebQuestionsSP had 2 KBs from which around 30% of the answers had to be extracted.

The models compared are Key-Value MNN and GRAFT-Nets with many variants such as :-
1. KV - with KB only { varying the completeness of the KB to 10%, 30% , 50 % and 100%}
2. KV - with Early Fusion {text + varying the completeness of the KB to 10%, 30% , 50 % and 100%}
3. GN - with KB only { varying the completeness of the KB to 10%, 30% , 50 % and 100%}
4. GN with Late Fusion { varying the completeness of the KB to 10%, 30% , 50 % and 100%}
5. GN with Early Fusion { varying the completeness of the KB to 10%, 30% , 50 % and 100%}
6. GN with an ensemble over the (4) and (5) model described above

Three major points can be analysed from this comparison :-

a) GN model outperforms KV model in all settings.
b) Early Fusion settings always outperforms the Late Fusion Settings.
c) Last and most important observation is regarding the difference in the Hits@1 and F1 score. The difference is larger in case of KV model than the GN model. KV model normalises over the memories, therefore model is not able to assign high probabilities to multiple answers at the same time. In GRAFT-Nets model, the attention over the types of relations outgoing from a node is normalised, and this makes possible to assign high probabilities to multiple nodes.

Method	WikiMovies (full)		WebQuestionsSP	
	kb	doc	kb	doc
MINERVA	**97.0 / –**	–	–	–
R2-AsV	–	85.8 / –	–	–
NSM	–	–	**– / 69.0**	–
DrQA*	–	–	–	21.5 / –
R-GCN#	**96.5 / 97.4**	–	37.2 / 30.5	–
KV	93.9 / –	76.2 / –	– / –	– / –
KV#	95.6 / 88.0	80.3 / 72.1	46.7 / 38.6	23.2 / 13.0
GN	**96.8 / 97.2**	**86.6 / 80.8**	67.8 / 62.8	**25.3 / 15.3**

Table 3: Hits@1 / F1 scores compared to SOTA models using only KB or text: MINERVA (Das et al., 2017a), R2-AsV (Watanabe et al., 2017), Neural Symbolic Machines (NSM) (Liang et al., 2017), DrQA (Chen et al., 2017), RGCN (Schlichtkrull et al., 2017) and KV-MemNN (Miller et al., 2016). *DrQA is pretrained on SQuAD. # = Re-implemented

Second important comparison is done with the State of the art models as indicated in Table 3.

We can see the performance matrix in the table and the statistics prove that the GRAFT-Nets model either performs similar or outperforms the existing QA models. However, there is one exception to it. In the WebQuestionsSP dataset KB setting, NSM (Neural Symbolic Machines) performs better than the GN model by 6.2 F1 points. There are three probable reasons for this :
(a) In the KB only setting, the recall value for the subgraph retrieval is 90.2%.This limits the performance of the model.
(b) same threshold is used in the GN model for all type of questions. If separate thresholds are set for different questions, the performance can be better.
(c) If the question has certain constraints assigned to it , the GN model performs poorly. [Sun et al., 2018]

Third important type of observation is made by dropping various components of the Model and capturing the data for further analysis :-

- Heterogeneous Updates : If we update both KB entities and test documents in the same manner, the model performs consistently worse than the heterogeneous update model.
- Conditioning on the question : Test is performed by first dropping the directed propagation method from the model and later dropping the attentions over relations. The performance degrades in the first case and worsens in the second case when both the important components are dropped. The effect is same in complete and incomplete KB settings.
- Fact Dropout : Fact dropping improves the performance of the model. But as we keep dropping the facts, after a certain point, the model is not able to learn the inference chain from KB and the performance starts decreasing rapidly.

5 Discussion

GRAFT-Nets performs similar or outperforms most of the state of the art models. There are many factors that enhance the model's performance.

1. In GRAFT-Nets normalisation happens over types of relations outgoing from a node hence can assign high weights to all correct answers. This is the reason why GN has lower difference in Hits@1 and F1 score than the KV models. KV models are normalised over memories, hence no of facts considered at one point of time are less.

2. GRAFT-Nets conditions on the question and assigns a PageRank score to each node. Since the data is not randomly sampled, the accuracy of the answer extraction increases and the relevant nodes of the graph get a higher score.

However, there are certain limitations as well in the GRAFT-Nets model :

1. The subgraph retrieval process can be improved especially the text retrieval. Current state is that it uses the surface search for the entities. We can perform some heuristic in order to fetch more appropriate answers.

2. Current model is capable of returning entities as the answers, when the question is posed. Further enhancement can be done to retrieve text span as the answer for detailed answer.

3. GRAFT-Nets model can be improved if it makes use of certain heuristic function while selecting the answer node from the subgraph.

4. While comparing the state of the art model, we came across NSM model that performs better than GRAFT-Nets. There are three probable reasons of it :-

a). GRAFT-Nets perform poorly when questions contain certain constraints. For example if we ask a question , "Who is the president of USA ?" and a

constrained question " Who is the first Afro-American President of USA ?" . GRAFT-Nets performs poorly for the second example. This can be improved by using certain heuristics [Yu et al., 2017].

b). GRAFT-Nets uses the same probability threshold for all types of questions. There are different types of questions whose answers can be found from various type of relations in the Graph. The PPR method assigns a PR score to each node and the answer extraction is based on a fix threshold value irrespective of the type of question. Experiment can be performed to see if setting different threshold can improve the performance of the model.

c). Recall value for the subgraph retrieval is 90.2%. In the current model, there is no guarantee that the answer to the question posed lies in the subgraph that has been generated. If we make the experiment in an oracle setting, where it is made sure before hand that the answer is part of the subgraph, the performance can increase.

GRAFT-Nets takes the current state of the art and tries to implement the new advancements done in the field of reinforcement learning and neural nets.

We can conclude from the Experiment that combining text and KBs in an Early Fusion setting is better than any other setting described in the paper. However, more work needs to be done in the extraction part of the experiment. A no of different settings were tested using the KBs and Text along with "Early Fusion" and "Late Fusion". The results are as expected and it is proved that the relational structure between text and KB can be leveraged by using heterogeneous graph structure for the entities.

References

[Chen et al., 2017] Chen, D., Fisch, A., Weston, J., and Bordes, A. (2017). Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*.

[Das et al., 2017] Das, R., Zaheer, M., Reddy, S., and McCallum, A. (2017). Question answering on knowledge bases and text using universal schema and memory networks. *arXiv preprint arXiv:1704.08384*.

[Dhingra et al., 2017] Dhingra, B., Mazaitis, K., and Cohen, W. W. (2017). Quasar: Datasets for question answering by search and reading. *arXiv preprint arXiv:1707.03904*.

[Ferrucci et al., 2010] Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A. A., Lally, A., Murdock, J. W., Nyberg, E., Prager, J., et al. (2010). Building watson: An overview of the deepqa project. *AI magazine*, 31(3):59–79.

[Gardner and Krishnamurthy, 2017] Gardner, M. and Krishnamurthy, J. (2017). Open-vocabulary semantic parsing with both distributional statistics and formal knowledge. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[Hamilton et al., 2017] Hamilton, W., Ying, Z., and Leskovec, J. (2017). Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034.

[Haveliwala, 2002] Haveliwala, T. H. (2002). Topic-sensitive pagerank. In *Proceedings of the 11th international conference on World Wide Web*, pages 517–526. ACM.

[Jiang et al., 2018] Jiang, J., Cui, Z., Xu, C., Li, C., and Yang, J. (2018). Walk-steered convolution for graph classification. *arXiv preprint arXiv:1804.05837*.

[Kipf and Welling, 2016] Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

[Lao et al., 2012] Lao, N., Subramanya, A., Pereira, F., and Cohen, W. W. (2012). Reading the web with learned syntactic-semantic inference rules. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1017–1026. Association for Computational Linguistics.

[Rajpurkar et al., 2016] Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

[Ryu et al., 2014] Ryu, P.-M., Jang, M.-G., and Kim, H.-K. (2014). Open domain question answering using wikipedia-based knowledge model. *Information processing & management*, 50(5):683–692.

[Scarselli et al., 2008] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. (2008). The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80.

[Schlichtkrull et al., 2018] Schlichtkrull, M., Kipf, T. N., Bloem, P., Van Den Berg, R., Titov, I., and Welling, M. (2018). Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer.

[Sun et al., 2018] Sun, H., Dhingra, B., Zaheer, M., Mazaitis, K., Salakhutdinov, R., and Cohen, W. W. (2018). Open domain question answering using early fusion of knowledge bases and text. *arXiv preprint arXiv:1809.00782*.

[Verga et al., 2015] Verga, P., Belanger, D., Strubell, E., Roth, B., and McCallum, A. (2015). Multilingual relation extraction using compositional universal schema. *arXiv preprint arXiv:1511.06396*.

[Yu et al., 2017] Yu, M., Yin, W., Hasan, K. S., Santos, C. d., Xiang, B., and Zhou, B. (2017). Improved neural relation detection for knowledge base question answering. *arXiv preprint arXiv:1704.06194*.