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

Haitian Sun* **Bhuwan Dhingra*** **Manzil Zaheer** **Kathryn Mazaitis**
Ruslan Salakhutdinov **William W. Cohen**
School of Computer Science
Carnegie Mellon University
{haitians,bdhingra,manzilz,krivard,rsalakhu,wcohen}@cs.cmu.edu

Abstract

Open Domain Question Answering (QA) is evolving from complex pipelined systems to end-to-end deep neural networks. Specialized neural models have been developed for extracting answers from either text alone or Knowledge Bases (KBs) alone. In this paper we look at a more practical setting, namely QA over the combination of a KB and entity-linked text, which is appropriate when an *incomplete* KB is available with a large text corpus. Building on recent advances in graph representation learning we propose a novel model, GRAFT-Net, for extracting answers from a question-specific subgraph containing text and KB entities and relations. We construct a suite of benchmark tasks for this problem, varying the difficulty of questions, the amount of training data, and KB completeness. We show that GRAFT-Net is competitive with the state-of-the-art when tested using either KBs or text alone, and vastly outperforms existing methods in the combined setting.

1 Introduction

Open domain Question Answering (QA) is the task of finding answers to questions posed in natural language. Historically, this required a specialized pipeline consisting of multiple machine-learned and hand-crafted modules (Ferrucci et al., 2010). Recently, the paradigm has shifted towards training end-to-end deep neural network models for the task (Chen et al., 2017; Liang et al., 2017; Raison et al., 2018; Talmor and Berant, 2018; Iyyer et al., 2017). Most existing models, however, answer questions using a *single* information source, usually either text from an encyclopedia, or a single knowledge base (KB).

Intuitively, the suitability of an information source for QA depends on both its *coverage* and the *difficulty* of extracting answers from it. A large text corpus has high coverage, but the information is expressed using many different text patterns. As a result, models which operate on these patterns (e.g. BiDAF (Seo et al., 2017)) do not generalize beyond their training domains (Wiese et al., 2017; Dhingra et al., 2018) or to novel types of reasoning (Welbl et al., 2018; Talmor and Berant, 2018). KBs, on the other hand, suffer from low coverage due to their inevitable incompleteness and restricted schema (Min et al., 2013), but are easier to extract answers from, since they are constructed precisely for the purpose of being queried.

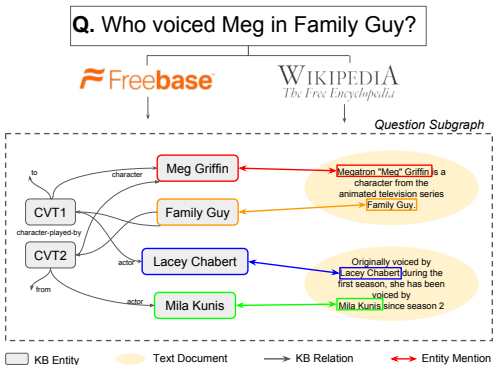

Figure 1: To answer a question posed in natural language, GRAFT-Net considers a heterogeneous graph constructed from text and KB facts, and thus can leverage the rich relational structure between the two information sources.

In practice, some questions are best answered using text, while others are best answered using KBs. A natural question, then, is how to effectively combine both types of information. Surprisingly little prior work has looked at this problem. In this paper we focus on a scenario in which a large-scale KB (Bollacker et al., 2008; Auer et al., 2007) and a text corpus are available, but neither is sufficient alone for answering all questions.

*Haitian Sun and Bhuwan Dhingra contributed equally to this work.

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 4231–4242
Brussels, Belgium, October 31 - November 4, 2018.

A naïve option, in such a setting, is to take state-of-the-art QA systems developed for each source, and aggregate their predictions using some heuristic (Ferrucci et al., 2010; Baudiš, 2015). We call this approach *late fusion*, and show that it can be sub-optimal, as models have limited ability to aggregate evidence across the different sources (§ 5.4). Instead, we focus on an *early fusion* strategy, where a single model is trained to extract answers from a *question subgraph* (Fig 1) containing relevant KB facts as well as text sentences. Early fusion allows more flexibility in combining information from multiple sources.

To enable early fusion, in this paper we propose a novel graph convolution based neural network, called GRAFT-Net (Graphs of Relations Among Facts and Text Networks), specifically designed to operate over heterogeneous graphs of KB facts and text sentences. We build upon recent work on graph representation learning (Kipf and Welling, 2016; Schlichtkrull et al., 2017), but propose two key modifications to adopt them for the task of QA. First, we propose *heterogeneous update rules* that handle KB nodes differently from the text nodes: for instance, LSTM-based updates are used to propagate information into and out of text nodes (§ 3.2). Second, we introduce a *directed propagation method*, inspired by personalized Pagerank in IR (Haveliwala, 2002), which constrains the propagation of embeddings in the graph to follow paths starting from seed nodes linked to the question (§ 3.3). Empirically, we show that both these extensions are crucial for the task of QA.

We evaluate these methods on a new suite of benchmark tasks for testing QA models when both KB and text are present. Using WikiMovies (Miller et al., 2016) and WebQuestionsSP (Yih et al., 2016), we construct datasets with a varying amount of training supervision and KB completeness, and with a varying degree of question complexity. We report baselines for future comparison, including Key Value Memory Networks (Miller et al., 2016; Das et al., 2017c), and show that our proposed GRAFT-Nets have superior performance across a wide range of conditions (§ 5). We also show that GRAFT-Nets are competitive with the state-of-the-art methods developed specifically for text-only QA, and state-of-the art methods developed for KB-only QA (§ 5.4)[1].

[1]Source code and data are available at https://github.com/OceanskySun/GraftNet

2 Task Setup

2.1 Description

A knowledge base is denoted as $\mathcal{K} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$, where $\mathcal{V}$ is the set of entities in the KB, and the edges $\mathcal{E}$ are triplets (s, r, o) which denote that relation $r \in \mathcal{R}$ holds between the subject $s \in \mathcal{V}$ and object $o \in \mathcal{V}$. A text corpus $\mathcal{D}$ is a set of documents $\{d_1, \ldots, d_{|\mathcal{D}|}\}$ where each document is a sequence of words $d_i = (w_1, \ldots, w_{|d_i|})$. We further assume that an (imperfect) entity linking system has been run on the collection of documents whose output is a set $\mathcal{L}$ of links (v, d_p) connecting an entity $v \in \mathcal{V}$ with a word at position p in document d, and we denote with $\mathcal{L}_d$ the set of all entity links in document d. For entity mentions spanning multiple words in d, we include links to all the words in the mention in $\mathcal{L}$.

The task is, given a natural language question $q = (w_1, \ldots, w_{|q|})$, extract its answers $\{a\}_q$ from $\mathcal{G} = (\mathcal{K}, \mathcal{D}, \mathcal{L})$. There may be multiple correct answers for a question. In this paper, we assume that the answers are entities from either the documents or the KB. We are interested in a wide range of settings, where the KB $\mathcal{K}$ varies from highly incomplete to complete for answering the questions, and we will introduce datasets for testing our models under these settings.

To solve this task we proceed in two steps. First, we extract a subgraph $\mathcal{G}_q \subset \mathcal{G}$ which contains the answer to the question with high probability. The goal for this step is to ensure high recall for answers while producing a graph small enough to fit into GPU memory for gradient-based learning. Next, we use our proposed model GRAFT-Net to learn node representations in $\mathcal{G}_q$, conditioned on q, which are used to classify each node as being an answer or not. Training data for the second step is generated using distant supervision. The entire process mimics the search-and-read paradigm for text-based QA (Dhingra et al., 2017).

2.2 Question Subgraph Retrieval

We retrieve the subgraph $\mathcal{G}_q$ using two parallel pipelines – one over the KB $\mathcal{K}$ which returns a set of entities, and the other over the corpus $\mathcal{D}$ which returns a set of documents. The retrieved entities and documents are then combined with entity links to produce a fully-connected graph.

KB Retrieval. To retrieve relevant entities from the KB we first perform entity linking on the ques-

tion q, producing a set of *seed entities*, denoted S_q. Next we run the Personalized PageRank (PPR) method (Haveliwala, 2002) around these seeds to identify other entities which might be an answer to the question. The edge-weights around S_q are distributed equally among all edges of the same type, and they are weighted such that edges relevant to the question receive a higher weight than those which are not. Specifically, we average word vectors to compute a relation vector $v(r)$ from the surface form of the relation, and a question vector $v(q)$ from the words in the question, and use cosine similarity between these as the edge weights. After running PPR we retain the top E entities $v_1, \ldots, v_E$ by PPR score, along with any edges between them, and add them to $\mathcal{G}_q$.

Text Retrieval. We use Wikipedia as the corpus and retrieve text at the sentence level, i.e. documents in $\mathcal{D}$ are defined along sentences boundaries[2]. We perform text retrieval in two steps: first we retrieve the top 5 most relevant Wikipedia articles, using the weighted bag-of-words model from DrQA (Chen et al., 2017); then we populate a Lucene[3] index with sentences from these articles, and retrieve the top ranking ones $d_1, \ldots, d_D$, based on the words in the question. For the sentence-retrieval step, we found it beneficial to include the title of the article as an additional field in the Lucene index. As most sentences in an article talk about the title entity, this helps in retrieving relevant sentences that do not explicitly mention the entity in the question. We add the retrieved documents, along with any entities linked to them, to the subgraph $\mathcal{G}_q$.

The final question subgraph is $\mathcal{G}_q = (\mathcal{V}_q, \mathcal{E}_q, \mathcal{R}^+)$, where the vertices $\mathcal{V}_q$ consist of all the retrieved entities and documents, i.e. $\mathcal{V}_q = \{v_1, \ldots, v_E\} \cup \{d_1, \ldots, d_D\}$. The edges are all relations from $\mathcal{K}$ among these entities, plus the entity-links between documents and entities, i.e.

$$\begin{aligned}\mathcal{E}_q = &\{(s, o, r) \in \mathcal{E} : s, o \in \mathcal{V}_q, r \in \mathcal{R}\} \\ &\cup \{(v, d_p, r_L) : (v, d_p) \in \mathcal{L}_d, d \in \mathcal{V}_q\},\end{aligned}$$

where r_L denotes a special "linking" relation. $\mathcal{R}^+ = \mathcal{R} \cup \{r_L\}$ is the set of all edge types in the subgraph.

[2]The term *document* will always refer to a sentence in the rest of this paper.

[3]https://lucene.apache.org/

3 GRAFT-Nets

The question q and its answers $\{a\}_q$ induce a labeling of the nodes in $\mathcal{V}_q$: we let $y_v = 1$ if $v \in \{a\}_q$ and $y_v = 0$ otherwise for all $v \in \mathcal{V}_q$. The task of QA then reduces to performing binary classification over the nodes of the graph $\mathcal{G}_q$. Several graph-propagation based models have been proposed in the literature which learn node representations and then perform classification of the nodes (Kipf and Welling, 2016; Schlichtkrull et al., 2017). Such models follow the standard gather-apply-scatter paradigm to learn the node representation with homogeneous updates, i.e. treating all neighbors equally.

The basic recipe for these models is as follows:

1. Initialize node representations $h_v^{(0)}$.
2. For $l = 1, \ldots, L$ update node representations

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

where $N_r(v)$ denotes the neighbours of v along incoming edges of type r, and ϕ is a neural network layer.

Here L is the number of *layers* in the model and corresponds to the maximum length of the paths along which information should be propagated in the graph. Once the propagation is complete the final layer representations $h_v^{(L)}$ are used to perform the desired task, for example link prediction in knowledge bases (Schlichtkrull et al., 2017).

However, there are two differences in our setting from previously studied graph-based classification tasks. The first difference is that, in our case, the graph $\mathcal{G}_q$ consists of *heterogeneous* nodes. Some nodes in the graph correspond to KB entities which represent symbolic objects, whereas other nodes represent textual documents which are variable length sequences of words. The second difference is that we want to condition the representation of nodes in the graph on the natural language question q. In §3.2 we introduce heterogeneous updates to address the first difference, and in §3.3 we introduce mechanisms for conditioning on the question (and its entities) for the second.

3.1 Node Initialization

Nodes corresponding to entities are initialized using fixed-size vectors $h_v^{(0)} = x_v \in \mathbb{R}^n$, where

x_v can be pre-trained KB embeddings or random, and n is the embedding size. Document nodes in the graph describe a variable length sequence of text. Since multiple entities might link to different positions in the document, we maintain a variable length representation of the document in each layer. This is denoted by $H_d^{(l)} \in \mathbb{R}^{|d| \times n}$. Given the words in the document $(w_1, \ldots, w_{|d|})$, we initialize its hidden representation as:

$$H_d^{(0)} = \text{LSTM}(w_1, w_2, \ldots),$$

where LSTM refers to a long short-term memory unit. We denote the p-th row of $H_d^{(l)}$, corresponding to the embedding of p-th word in the document d at layer l, as $H_{d,p}^{(l)}$.

3.2 Heterogeneous Updates

Figure 2 shows the update rules for entities and documents, which we describe in detail here.

Entities. Let $M(v) = \{(d, p)\}$ be the set of positions p in documents d which correspond to a mention of entity v. The update for entity nodes involves a single-layer feed-forward network (FFN) over the concatenation of four states:

$$h_v^{(l)} = \text{FFN}\left(\begin{bmatrix} h_v^{(l-1)} \\ h_q^{(l-1)} \\ \sum_r \sum_{v' \in N_r(v)} \alpha_r^{v'} \psi_r(h_{v'}^{(l-1)}) \\ \sum_{(d,p) \in M(v)} H_{d,p}^{(l-1)} \end{bmatrix}\right). \quad (1)$$

The first two terms correspond to the entity representation and question representation (details below), respectively, from the previous layer.

The third term aggregates the states from the entity neighbours of the current node, $N_r(v)$, after scaling with an attention weight $\alpha_r^{v'}$ (described in the next section), and applying relation specific transformations ψ_r. Previous work on Relational-Graph Convolution Networks (Schlichtkrull et al., 2017) used a linear projection for ψ_r. For a batched implementation, this results in matrices of size $O(B|\mathcal{R}_q||\mathcal{E}_q|n)$, where B is the batch size, which can be prohibitively large for large subgraphs[4]. Hence in this work we use *relation vectors* x_r for $r \in \mathcal{R}_q$ instead of matrices, and compute the update along an edge as:

$$\psi_r(h_{v'}^{(l-1)}) = pr_{v'}^{(l-1)} \text{FFN}\left(x_r, h_{v'}^{(l-1)}\right). \quad (2)$$

[4]This is because we have to use adjacency matrices of size $|\mathcal{R}_q| \times |\mathcal{E}_q| \times |\mathcal{E}_q|$ to aggregate embeddings from neighbours of all nodes simultaneously.

Here $pr_{v'}^{(l-1)}$ is a PageRank score used to control the propagation of embeddings along paths starting from the seed nodes, which we describe in detail in the next section. The memory complexity of the above is $O(B(|\mathcal{F}_q| + |\mathcal{E}_q|)n)$, where $|\mathcal{F}_q|$ is the number of facts in the subgraph $\mathcal{G}_q$.

The last term aggregates the states of all tokens that correspond to mentions of the entity v among the documents in the subgraph. Note that the update depends on the positions of entities in their containing document.

Documents. Let $L(d, p)$ be the set of all entities linked to the word at position p in document d. The document update proceeds in two steps. First we aggregate over the entity states coming in at each position separately:

$$\tilde{H}_{d,p}^{(l)} = \text{FFN}\left(H_{d,p}^{(l-1)}, \sum_{v \in L(d,p)} h_v^{(l-1)}\right). \quad (3a)$$

Here $h_v^{(l-1)}$ are normalized by the number of outgoing edges at v. Next we aggregate states within the document using an LSTM:

$$H_d^{(l)} = \text{LSTM}(\tilde{H}_d^{(l)}). \quad (3b)$$

3.3 Conditioning on the Question

For the parts described thus far, the graph learner is largely agnostic of the question. We introduce dependence on question in two ways: by attention over relations, and by personalized propagation.

To represent q, let $w_1^q, \ldots, w_{|q|}^q$ be the words in the question. The initial representation is computed as:

$$h_q^{(0)} = \text{LSTM}(w_1^q, \ldots, w_{|q|}^q)_{|q|} \in \mathbb{R}^n, \quad (4)$$

where we extract the final state from the output of the LSTM. In subsequent layers the question representation is updated as $h_q^{(l)} = \text{FFN}\left(\sum_{v \in S_q} h_v^{(l)}\right)$, where S_q denotes the seed entities mentioned in the question.

Attention over Relations. The attention weight in the third term of Eq. (1) is computed using the question and relation embeddings:

$$\alpha_r^{v'} = \text{softmax}(x_r^T h_q^{(l-1)}),$$

where the softmax normalization is over all outgoing edges from v', and x_r is the relation vector for relation r. This ensures that embeddings are propagated more along edges relevant to the question.

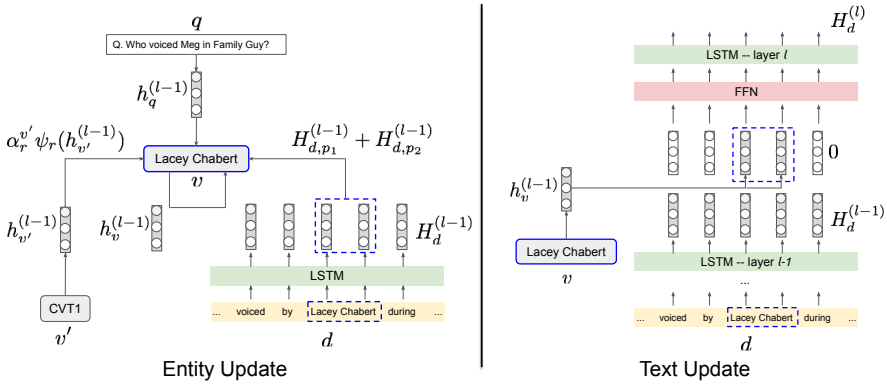

Figure 2: Illustration of the heterogeneous update rules for entities (**left**) and text documents (**right**)

Directed Propagation. Many questions require multi-hop reasoning, which follows a path from a seed node mentioned in the question to the target answer node. To encourage such a behaviour when propagating embeddings, we develop a technique inspired from personalized PageRank in IR (Haveliwala, 2002). The propagation starts at the seed entities S_q mentioned in the question. In addition to the vector embeddings $h_v^{(l)}$ at the nodes, we also maintain scalar "PageRank" scores $pr_v^{(l)}$ which measure the total weight of paths from a seed entity to the current node, as follows:

$$pr_v^{(0)} = \begin{cases} \frac{1}{|S_q|} & \text{if} \quad v \in S_q \\ 0 & \text{o.w.} \end{cases},$$

$$pr_v^{(l)} = (1-\lambda) pr_v^{(l-1)} + \lambda \sum_r \sum_{v' \in N_r(v)} \alpha_r^{v'} pr_{v'}^{(l-1)}.$$

Notice that we reuse the attention weights $\alpha_r^{v'}$ when propagating PageRank, to ensure that nodes along paths relevant to the question receive a high weight. The PageRank score is used as a scaling factor when propagating embeddings along the edges in Eq. (2). For $l = 1$, the PageRank score will be 0 for all entities except the seed entities, and hence propagation will only happen outward from these nodes. For $l = 2$, it will be non-zero for the seed entities and their 1-hop neighbors, and propagation will only happen along these edges. Figure 3 illustrates this process.

3.4 Answer Selection

The final representations $h_v^{(L)} \in \mathbb{R}^n$, are used for binary classification to select the answers:

$$\Pr(v \in \{a\}_q | \mathcal{G}_q, q) = \sigma(w^T h_v^{(L)} + b), \qquad (5)$$

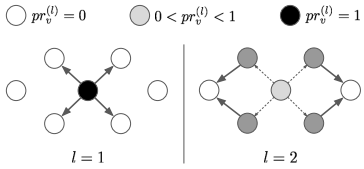

Figure 3: Directed propagation of embeddings in GRAFT-Net. A scalar *PageRank* score $pr_v^{(l)}$ is maintained for each node v across layers, which spreads out from the seed node. Embeddings are only propagated from nodes with $pr_v^{(l)} > 0$.

where σ is the sigmoid function. Training uses binary cross-entropy loss over these probabilities.

3.5 Regularization via Fact Dropout

To encourage the model to learn a robust classifier, which exploits all available sources of information, we randomly drop edges from the graph during training with probability p_0. We call this *fact-dropout*. It is usually easier to extract answers from the KB than from the documents, so the model tends to rely on the former, especially when the KB is complete. This method is similar to DropConnect (Wan et al., 2013).

4 Related Work

The work of Das et al. (2017c) attempts an early fusion strategy for QA over KB facts and text. Their approach is based on Key-Value Memory Networks (KV-MemNNs) (Miller et al., 2016) coupled with a universal schema (Riedel et al., 2013) to populate a memory module with representations of KB triples and text snippets independently. The key limitation for this model is that it ignores the rich relational structure between the

facts and text snippets. Our graph-based method, on the other hand, explicitly uses this structure for the propagation of embeddings. We compare the two approaches in our experiments (§5), and show that GRAFT-Nets outperform KV-MemNNs over all tasks.

Non-deep learning approaches have been also attempted for QA over both text assertions and KB facts. Gardner and Krishnamurthy (2017) use traditional feature extraction methods of open-vocabulary semantic parsing for the task. Ryu et al. (2014) use a pipelined system aggregating evidence from both unstructured and semi-structured sources for open-domain QA.

Another line of work has looked at learning combined representations of KBs and text for relation extraction and Knowledge Base Completion (KBC) (Lao et al., 2012; Riedel et al., 2013; Toutanova et al., 2015; Verga et al., 2016; Das et al., 2017b; Han et al., 2016). The key difference in QA compared to KBC is that in QA the inference process on the knowledge source has to be conditioned on the question, so different questions induce different representations of the KB and warrant a different inference process. Furthermore, KBC operates under the fixed schema defined by the KB before-hand, whereas natural language questions might not adhere to this schema.

The GRAFT-Net model itself is motivated from the large body of work on graph representation learning (Scarselli et al., 2009; Li et al., 2016; Kipf and Welling, 2016; Atwood and Towsley, 2016; Schlichtkrull et al., 2017). Like most other graph-based models, GRAFT-Nets can also be viewed as an instantiation of the Message Passing Neural Network (MPNN) framework of Gilmer et al. (2017). GRAFT-Nets are also *inductive* representation learners like GraphSAGE (Hamilton et al., 2017), but operate on a heterogeneous mixture of nodes and use retrieval for getting a subgraph instead of random sampling. The recently proposed Walk-Steered Convolution model uses random walks for learning graph representations (Jiang et al., 2018). Our personalization technique also borrows from such random walk literature, but uses it to localize propagation of embeddings.

Tremendous progress on QA over KB has been made with deep learning based approaches like memory networks (Bordes et al., 2015; Jain, 2016) and reinforcement learning (Liang et al., 2017; Das et al., 2017a). But extending them with text, which is our main focus, is non-trivial. In another direction, there is also work on producing parsimonious graphical representations of textual data (Krause et al., 2016; Lu et al., 2017); however in this paper we use a simple sequential representation augmented with entity links to the KB which works well.

For QA over text only, a major focus has been on the task of reading comprehension (Seo et al., 2017; Gong and Bowman, 2017; Hu et al., 2017; Shen et al., 2017; Yu et al., 2018) since the introduction of SQuAD (Rajpurkar et al., 2016). These systems assume that the answer-containing passage is known apriori, but there has been progress when this assumption is relaxed (Chen et al., 2017; Raison et al., 2018; Dhingra et al., 2017; Wang et al., 2018, 2017; Watanabe et al., 2017). We work in the latter setting, where relevant information must be retrieved from large information sources, but we also incorporate KBs into this process.

5 Experiments & Results

5.1 Datasets

WikiMovies-10K consists of $10K$ randomly sampled training questions from the WikiMovies dataset (Miller et al., 2016), along with the original test and validation sets. We sample the training questions to create a more difficult setting, since the original dataset has $100K$ questions over only 8 different relation types, which is unrealistic in our opinion. In § 5.4 we also compare to the existing state-of-the-art using the full training set.

We use the KB and text corpus constructed from Wikipedia released by Miller et al. (2016). For entity linking we use simple surface level matches, and retrieve the top 50 entities around the seeds to create the question subgraph. We further add the top 50 sentences (along with their article titles) to the subgraph using Lucene search over the text corpus. The overall answer recall in our constructed subgraphs is 99.6%.

WebQuestionsSP (Yih et al., 2016) consists of 4737 natural language questions posed over Freebase entities, split up into 3098 training and 1639 test questions. We reserve 250 training questions for model development and early stopping. We use the entity linking outputs from S-MART[5] and retrieve 500 entities from the neighbourhood around the question seeds in Freebase to populate the

[5] https://github.com/scottyih/STAGG

question subgraphs[6]. We further retrieve the top 50 sentences from Wikipedia with the two-stage process described in §2. The overall recall of answers among the subgraphs is 94.0%.

Table 1 shows the combined statistics of all the retreived subgraphs for the questions in each dataset. These two datasets present varying levels of difficulty. While all questions in WikiMovies correspond to a single KB relation, for WebQuestionsSP the model needs to aggregate over two KB facts for ∼30% of the questions, and also requires reasoning over constraints for ∼7% of the questions (Liang et al., 2017). For maximum portability, QA systems need to be robust across several degrees of KB availability since different domains might contain different amounts of structured data; and KB completeness may also vary over time. Hence, we construct an additional 3 datasets each from the above two, with the number of KB facts downsampled to 10%, 30% and 50% of the original to simulate settings where the KB is incomplete. We repeat the retrieval process for each sampled KB.

5.2 Compared Models

KV-KB is the Key Value Memory Networks model from Miller et al. (2016); Das et al. (2017c) but using only KB and ignoring the text. **KV-EF** (early fusion) is the same model with access to both KB and text as memories. For text we use a BiLSTM over the entire sentence as keys, and entity mentions as values. This re-implementation shows better performance on the text-only and KB-only WikiMovies tasks than the results reported previously[7] (see Table 4). **GN-KB** is the GRAFT-Net model ignoring the text. **GN-LF** is a late fusion version of the GRAFT-Net model: we train two separate models, one using text only and the other using KB only, and then ensemble the two[8]. **GN-EF** is our main GRAFT-Net model with early fusion. **GN-EF+LF** is an ensemble over the GN-EF and GN-LF models, with the same ensembling method as GN-LF. We report Hits@1, which is the accuracy of the top-predicted answer from the model, and the F1 score. To compute the F1 score we tune a threshold on the development set to select answers based on binary probabilities for each node in the subgraph.

5.3 Main Results

Table 2 presents a comparison of the above models across all datasets. GRAFT-Nets (GN) shows consistent improvement over KV-MemNNs on both datasets in all settings, including KB only (-KB), text only (-EF, Text Only column), and early fusion (-EF). Interestingly, we observe a larger relative gap between the Hits and F1 scores for the KV models than we do for our GN models. We believe this is because the attention for KV is normalized over the memories, which are KB facts (or text sentences): hence the model is unable to assign high probabilities to multiple facts at the same time. On the other hand, in GN, we normalize the attention over *types of relations* outgoing from a node, and hence can assign high weights to all the correct answers.

We also see a consistent improvement of early fusion over late fusion (-LF), and by ensembling them together we see the best performance across all the models. In Table 2 (right), we further show the improvement for KV-EF over KV-KB, and GN-LF and GN-EF over GN-KB, as the amount of KB is increased. This measures how effective these approaches are in utilizing text plus a KB. For KV-EF we see improvements when the KB is highly incomplete, but in the full KB setting, the performance of the fused approach is worse. A similar trend holds for GN-LF. On the other hand, GN-EF with text improves over the KB-only approach in all settings. As we would expect, though, the benefit of adding text decreases as the KB becomes more and more complete.

5.4 Comparison to Specialized Methods

In Table 4 we compare GRAFT-Nets to state-of-the-art models that are specifically designed and tuned for QA using either only KB or only text. For this experiment we use the full WikiMovies dataset to enable direct comparison to previously reported numbers. For DrQA (Chen et al., 2017), following the original paper, we restrict answer spans for WebQuestionsSP to match an entity in Freebase. In each case we also train GRAFT-Nets using only KB facts or only text sentences. In three out of the four cases, we find that GRAFT-Nets ei-

[6] A total of 13 questions had no detected entities. These were ignored during training and considered as incorrect during evaluation.

[7] For all KV models we tuned the number of layers $\{1, 2, 3\}$, batch size $\{10, 30, 50\}$, model dimension $\{50, 80\}$. We also use fact dropout regularization in the KB+Text setting tuned between $\{0, 0.2, 0.4\}$.

[8] For ensembles we take a weighted combination of the answer probabilities produced by the models, with the weights tuned on the dev set. For answers only in text or only in KB, we use the probability as is.

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 $\cup_q \mathcal{G}_q$ for WikiMovies-10K and WebQuestionsSP.

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**

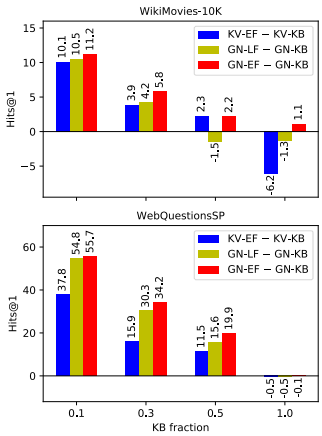

Table 2: **Left:** Hits@1 / F1 scores of GRAFT-Nets (GN) compared to KV-MemNN (KV) in KB only (-KB), early fusion (-EF), and late fusion (-LF) settings. **Right:** Improvement of early fusion (-EF) and late fusion (-LF) over KB only (-KB) settings as KB completeness increases.

ther match or outperform the existing state-of-the-art models. We emphasize that the latter have no mechanism for dealing with the fused setting.

The one exception is the KB-only case for WebQuestionsSP where GRAFT-Net does 6.2% F1 points worse than Neural Symbolic Machines (Liang et al., 2017). Analysis suggested three explanations: (1) In the KB-only setting, the recall of subgraph retrieval is only 90.2%, which limits its overall performance. In an oracle setting where we ensure the answers are part of the subgraph, the F1 score increases by 4.8%. (2) We use the same probability threshold for all questions, even though the number of answers may vary significantly. Models which parse the query into a symbolic form do not suffer from this problem since answers are retrieved in a deterministic fashion. If we tune separate thresholds for each question the F1 score improves by 7.6%. (3) GRAFT-Nets perform poorly in the few cases where there is a constraint involved in picking out the answer (for example, "who *first* voiced Meg in Family Guy"). If we ignore such constraints, and consider all entities with the same sequence of relations to the seed as correct, the performance improves by 3.8% F1. Heuristics such as those used by Yu et al. (2017) can be used to improve these cases. Figure 3 shows examples where GRAFT-Net fails to predict the correct answer set exactly.

5.5 Effect of Model Components

Heterogeneous Updates. We tested a non-heterogeneous version of our model, where instead of using fine-grained entity linking information for updating the node representations ($M(v)$ and $L(d, p)$ in Eqs. 1, 3a), we aggregate the document states across all its positions as $\sum_p H^{(l)}_{d,p}$ and use this combined state for all updates. Without the heterogeneous update, all entities $v \in L(d, \cdot)$ will receive the same update from document d. Therefore, the model cannot disambiguate different entities mentioned in the same document. The result in Table 5 shows that this version is consistently worse than the heterogeneous model.

Conditioning on the Question. We performed an ablation test on the directed propagation method and attention over relations. We observe that both components lead to better performance. Such effects are observed in both complete and incomplete KB scenarios, e.g. on WebQuestionsSP dataset, as shown in Figure 4 (left).

Fact Dropout. Figure 4 (right) compares the performance of the early fusion model as we vary

Question	Correct Answers	Predicted Answers
what language do most people speak in afghanistan	Pashto language, Farsi (Eastern Language)	Pashto language
what college did john stockton go to	Gonzaga University	Gonzaga University, Gonzaga Preparatory School

Table 3: Examples from WebQuestionsSP dataset. **Top:** The model misses a correct answer. **Bottom:** The model predicts an extra incorrect answer.

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 4: 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), R-GCN (Schlichtkrull et al., 2017) and KV-MemNN (Miller et al., 2016). *DrQA is pretrained on SQuAD. #Re-implemented.

	0 KB	0.1 KB	0.3 KB	0.5 KB	1.0 KB
NH	22.7 / 13.6	28.7 / 15.8	35.6 / 23.2	47.2 / 33.3	66.5 / 59.8
H	25.3 / 15.3	31.5 / 17.7	40.7 / 25.2	49.9 / 34.7	67.8 / 60.4

Table 5: Non-Heterogeneous (NH) vs. Heterogeneous (H) updates on WebQuestionsSP

the rate of fact dropout. Moderate levels of fact dropout improve performance on both datasets. The performance increases as the fact dropout rate increases until the model is unable to learn the inference chain from KB.

6 Conclusion

In this paper we investigate QA using text combined with an incomplete KB, a task which has received limited attention in the past. We introduce several benchmark problems for this task by modifying existing question-answering datasets, and discuss two broad approaches to solving this problem—"late fusion" and "early fusion". We show that early fusion approaches perform better.

We also introduce a novel early-fusion model, called GRAFT-Net, for classifying nodes in subgraph consisting of both KB entities and text documents. GRAFT-Net builds on recent advances in graph representation learning but includes several innovations which improve performance on this task. GRAFT-Nets are a single model which achieve performance competitive to state-of-the-art methods in both text-only and KB-only settings, and outperform baseline models when using text combined with an incomplete KB. Current directions for future work include – (1) extending GRAFT-Nets to pick spans of text as answers, rather than only entities and (2) improving the subgraph retrieval process.

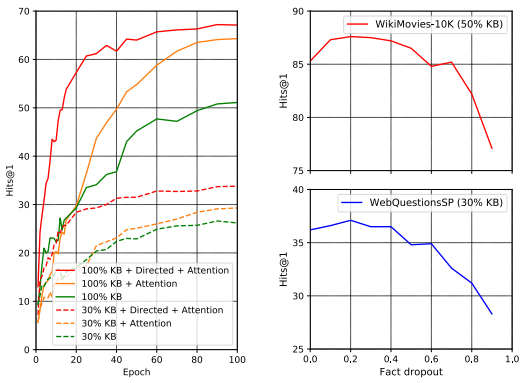

Figure 4: **Left:** Effect of directed propagation and query-based attention over relations for the WebQuestionsSP dataset with 30% KB and 100% KB. **Right:** Hits@1 with different rates of fact-dropout on and WikiMovies and WebQuestionsSP.

Acknowledgments

Bhuwan Dhingra is supported by NSF under grants CCF-1414030 and IIS-1250956 and by grants from Google. Ruslan Salakhutdinov is supported in part by ONR grant N000141812861, Apple, and Nvidia NVAIL Award.

References

James Atwood and Don Towsley. 2016. Diffusion-convolutional neural networks. In *Advances in Neu-*

ral Information Processing Systems, pages 1993–2001.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer.

Petr Baudiš. 2015. Yodaqa: a modular question answering system pipeline. In *POSTER 2015-19th International Student Conference on Electrical Engineering*, pages 1156–1165.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM.

Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Association for Computational Linguistics (ACL)*.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. 2017a. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. *arXiv preprint arXiv:1711.05851*.

Rajarshi Das, Arvind Neelakantan, David Belanger, and Andrew McCallum. 2017b. Chains of reasoning over entities, relations, and text using recurrent neural networks. *EACL*.

Rajarshi Das, Manzil Zaheer, Siva Reddy, and Andrew McCallum. 2017c. Question answering on knowledge bases and text using universal schema and memory networks. *ACL*.

Bhuwan Dhingra, Kathryn Mazaitis, and William W Cohen. 2017. Quasar: Datasets for question answering by search and reading. *arXiv preprint arXiv:1707.03904*.

Bhuwan Dhingra, Danish Pruthi, and Dheeraj Rajagopal. 2018. Simple and effective semi-supervised question answering. *NAACL*.

David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J William Murdock, Eric Nyberg, John Prager, et al. 2010. Building watson: An overview of the deepqa project. *AI magazine*, 31(3):59–79.

Matt Gardner and Jayant Krishnamurthy. 2017. Open-vocabulary semantic parsing with both distributional statistics and formal knowledge. In *AAAI*, pages 3195–3201.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. *ICML*.

Yichen Gong and Samuel R Bowman. 2017. Ruminating reader: Reasoning with gated multi-hop attention. *arXiv preprint arXiv:1704.07415*.

William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *CoRR*, abs/1706.02216.

Xu Han, Zhiyuan Liu, and Maosong Sun. 2016. Joint representation learning of text and knowledge for knowledge graph completion. *arXiv preprint arXiv:1611.04125*.

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

Minghao Hu, Yuxing Peng, and Xipeng Qiu. 2017. Mnemonic reader for machine comprehension. *arXiv preprint arXiv:1705.02798*.

Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. 2017. Search-based neural structured learning for sequential question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1821–1831.

Sarthak Jain. 2016. Question answering over knowledge base using factual memory networks. In *Proceedings of the NAACL Student Research Workshop*, pages 109–115.

Jiatao Jiang, Zhen Cui, Chunyan Xu, Chengzheng Li, and Jian Yang. 2018. Walk-steered convolution for graph classification. *arXiv preprint arXiv:1804.05837*.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Sebastian Krause, Leonhard Hennig, Andrea Moro, Dirk Weissenborn, Feiyu Xu, Hans Uszkoreit, and Roberto Navigli. 2016. Sar-graphs: A language resource connecting linguistic knowledge with semantic relations from knowledge graphs. *Web Semantics: Science, Services and Agents on the World Wide Web*, 37:112–131.

Ni Lao, Amarnag Subramanya, Fernando Pereira, and William W Cohen. 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.

Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. 2016. Gated graph sequence neural networks. *ICLR*.

Chen Liang, Jonathan Berant, Quoc Le, Kenneth D Forbus, and Ni Lao. 2017. Neural symbolic machines: Learning semantic parsers on freebase with weak supervision. *ACL*.

Zhengdong Lu, Haotian Cui, Xianggen Liu, Yukun Yan, and Daqi Zheng. 2017. Object-oriented neural programming (oonp) for document understanding. *arXiv preprint arXiv:1709.08853*.

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. *EMNLP*.

Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. 2013. Distant supervision for relation extraction with an incomplete knowledge base. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 777–782.

Martin Raison, Pierre-Emmanuel Mazaré, Rajarshi Das, and Antoine Bordes. 2018. Weaver: Deep co-encoding of questions and documents for machine reading. *arXiv preprint arXiv:1804.10490*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–84.

Pum-Mo Ryu, Myung-Gil Jang, and Hyun-Ki Kim. 2014. Open domain question answering using wikipedia-based knowledge model. *Information Processing and Management*, 50(5):683 – 692.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2017. Modeling relational data with graph convolutional networks. *arXiv preprint arXiv:1703.06103*.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. *ICLR*.

Yelong Shen, Po-Sen Huang, Jianfeng Gao, and Weizhu Chen. 2017. Reasonet: Learning to stop reading in machine comprehension. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1047–1055. ACM.

A. Talmor and J. Berant. 2018. The web as a knowledge-base for answering complex questions. In *North American Association for Computational Linguistics (NAACL)*.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509.

Patrick Verga, David Belanger, Emma Strubell, Benjamin Roth, and Andrew McCallum. 2016. Multilingual relation extraction using compositional universal schema. *NAACL*.

Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. 2013. Regularization of neural networks using dropconnect. In *International Conference on Machine Learning*, pages 1058–1066.

Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerald Tesauro, Bowen Zhou, and Jing Jiang. 2018. R^3: Reinforced reader-ranker for open-domain question answering. *AAAI*.

Shuohang Wang, Mo Yu, Jing Jiang, Wei Zhang, Xiaoxiao Guo, Shiyu Chang, Zhiguo Wang, Tim Klinger, Gerald Tesauro, and Murray Campbell. 2017. Evidence aggregation for answer re-ranking in open-domain question answering. *arXiv preprint arXiv:1711.05116*.

Yusuke Watanabe, Bhuwan Dhingra, and Ruslan Salakhutdinov. 2017. Question answering from unstructured text by retrieval and comprehension. *arXiv preprint arXiv:1703.08885*.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *TACL*.

Georg Wiese, Dirk Weissenborn, and Mariana Neves. 2017. Neural domain adaptation for biomedical question answering. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 281–289, Vancouver, Canada. Association for Computational Linguistics.

Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 201–206.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. *ICLR*.

Mo Yu, Wenpeng Yin, Kazi Saidul Hasan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2017. Improved neural relation detection for knowledge base question answering. *ACL*.