Open Domain Question Answering Using Early Fusion Knowledge Bases and Text

H. Sun, B. Dhingra, M. Zaheer, K. Mazaitis, R. Salakhutdinov, W. Cohen

Carnegie Mellon University School of Computer Science

Presenter: William Zhang
https://qdata.github.io/deep2Read

Outline

1. Introduction
2. Task Setup
3. GRAFT-Nets
4. Related Work
5. Experiments & Results
6. Conclusion

Introduction

Motivation

- Existing QA models generally answer questions using a single information source, usually either a knowledge base (KB) or text corpora
- Large text corpora have high coverage, but too many text patterns, making it harder for models to generalize beyond training domains
- KBs have low coverage but are easier to extract questions from because of their predefined structure
- Some questions are better answered using text, some using KBs; how can we combine both types of information?

Introduction

Basic Premise

- Naive option would be to take state-of-the-art QA models developed for each source of interest and aggregate their predictions using some heuristic
- This approach is called late fusion, and can be shown to be suboptimal
- Instead, focus on early fusion, where a single model is trained on a question subgraph
- To solve this problem, they develop GRAFT-Net (Graphs of Relations Among Facts and Text Networks)

Task Setup

Notation

- 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) denoting 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 $\{d_1, ..., d_{|\mathcal{D}|}\}$ where each document is a sequence of words $d_i = (w_1, ..., w_{|d_i|})$

Task Setup

Notation

- Further assume that an (imperfect) entity linking system has been run whose output is a set $\mathcal{L}$ of links (v, d_p) connecting $v \in \mathcal{V}$ with a word at position p in document d
- $\mathcal{L}_d$ represents the set of all entity links in document d and entity mentions spanning multiple words has links to all words in mention in d

Task Setup

Task Description

- Given a natural language question $q = (w_1, ..., w_{|q|})$, extract its answers $\{a\}_q$ from $\mathcal{G} = (\mathcal{K}, \mathcal{D}, \mathcal{L})$
- There may be multiple correct questions for a question
- This paper also assumes that the answers are entities from either the documents or KB
- Interested in this question for KBs of varying completeness

Task Setup

Approach

- First, extract a subgraph $\mathcal{G}_q \subset \mathcal{G}$ which contains answer to q with high probability
    - $\mathcal{G}_q$ is generated using two parallel pipelines: one over KB, which returns a set of entities, and one over documents, which returns a set of documents
- This ensures high recall for answers and produces graph small enough to fit on GPUs for gradient learning
- Next, use proposed GRAFT-Net to learn node representations in $\mathcal{G}_q$, conditioned on q, to classify each node as an answer or not
- Training data for this step is done using distant supervision

Task Setup

Question Subgraph Retrieval: KB Retrieval

- Perform entity linking on q to produce set of seed entities S_q
- Run Personalized PageRank (PPR) around these seeds to identify which could be answer to question
- Edge-weights around S_q are distributed equally among edges of same type
- Average word vectors to compute $v(r)$ (relation vector) and $v(q)$ (question vector) and use cosine similarity; more similar $\Rightarrow$ higher weight
- After PPR, take entities with top E PPR scores

Task Setup

Question Subgraph Retrieval: Document Retrieval

- Use Wikipedia as corpus; retrieve text at sentence level (”documents” are single sentences)
- First, choose top 5 most relevant articles using DrQA (Chen et al., 2017)
- Populate a Lucene index with sentences from these articles and retrieve top ranked ones $d_1, d_2, ..., d_D$ based on words in question
- Title of article also included in Lucene index (this way, sentences not explicitly mentioning article topic are still considered)
- Add documents and any entities linked to them to $\mathcal{G}_q$

Task Setup

Question Subgraph Retrieval: Final Steps

- Final subgraph is
$$\mathcal{G}_q = (\mathcal{V}_q, \mathcal{E}_q, \mathcal{R}^+)$$
- $\mathcal{V}_q = \{v_1, ..., v_E\} \cup \{d_1, ..., d_D\}$ is all of the documents and entities
- $\mathcal{E}_q = \{(s, o, r) \in \mathcal{E} : s, o \in \mathcal{V}_q, r \in \mathcal{R}\} \cup \{(v, d_p, r_L) : (v, d_p) \in \mathcal{L}_d, d \in \mathcal{V}_q\}$ is all relations from $\mathcal{K}$ among the entities and all the entity-links between documents and entities
- r_L is a special linking relation; $\mathcal{R}^+ = \mathcal{R} \cup \{r_L\}$ is the set of all edge types in this subgraph

GRAFT-Nets

Binary Classification of Nodes

- Question q and answers $\{a\}_q$ induce labeling: $y_v = 1$ if $v \in \{a\}_q$ and 0 otherwise for all $v \in \mathcal{V}_q$
- Reduces problem to binary classification problem
- Several models which tackle this problem follow the gather-apply-scatter paradigm to learn node representation with homogeneous updates

GRAFT-Nets

Typical Models

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

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

  where $N_r(v)$ denotes the neighbors of v along incoming edges of type r, ϕ is a neural network layer, L is the number of layers in model, corresponding to maximum path length information should be propagated in graph
- $h_v^{(L)}$ is used for final classification task
- **Key Differences:** (1) $\mathcal{G}_q$ consists of *heterogeneous* nodes (some are objects, some are sequences of words), (2) need to condition representation of nodes on q

GRAFT-Nets

Node Initialization

- For entities, nodes are initialized with fixed-size vectors $h_v^{(0)} = x_v \in \mathbb{R}^n$ where x_v can be pre-trained embedding or random and n is the embedding size
- Document nodes are variable length sequences of words, thus maintain a variable-length representation in each layer $H_d^{(l)} \in \mathbb{R}^{|d| xn}$
- Initialize hidden representation by $H_d^{(0)} = \text{LSTM}(w_1, ...)$
- $H_{d,p}^{(l)}$ is the pth row at layer l

GRAFT-Nets

Heterogeneous Updates: Entities

- Let $M(v) = \{(d, p)\}$ be the set of positions p in documents d which correspond to a mention of entity v
- Update for entity nodes involves a single-layer feed-forward network (FFN) over the concatenation of 4 states

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

GRAFT-Nets

Heterogeneous Updates: Entities

- The first two terms correspond to the entity representation and question representation, respectively, from the previous layer
- The fourth term aggregates states of all tokens that correspond to mentions of entities v in documents of the subgraph

GRAFT-Nets

Heterogeneous Updates: Entities

- The third term aggregates states from the entity neighbors of current node after scaling with attention weight $\alpha_r^{v'}$ and applying relation specific transformations ψ_r
  - Previous work has ψ_r as a linear projection, but for batch implementations, results in prohibitively large matrices of size $O(B|\mathcal{R}_q||\mathcal{E}_q|n)$ where B is batch size
  - Instead of relation matrices, use relation vectors x_r for $r \in \mathcal{R}_q$ such that

    $$\psi_r(h_v^{(l-1)}) = pr_{v'}^{(l-1)}\text{FFN}(x_r, h_{v'}^{(l-1)})$$

    where $pr_{v'}^{(l-1)}$ is the page rank score
  - Results in memory complexity of $O(B(|\mathcal{F}_q| + |\mathcal{E}_q|)n)$, where $|\mathcal{F}_q|$ is the number of facts in the subgraph

GRAFT-Nets

Heterogeneous Updates: Documents

- Let $L(d, p)$ represent the set of all entities linked to word p of document d
- First, aggregate over entity states at each position

$$\tilde{H}_{d,p}^{(l)} = \text{FFN}(H_{d,p}^{(l-1)}, \Sigma_{v \in L(d,p)} h_v^{(l-1)})$$

  where $h_v^{(l-1)}$ is normalized by number of outgoing edges at v
- Then, aggregate states within document

$$H_d^{(l)} = \text{LSTM}(\tilde{H}_d^{(l)})$$

GRAFT-Nets

Heterogeneous Updates: Diagram

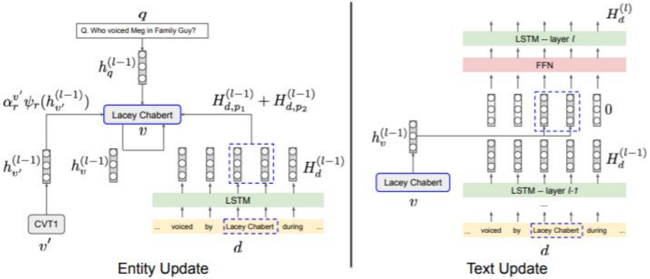


Figure 2: Illustration of the heterogeneous update rules for entities (**left**) and text documents (**right**)

GRAFT-Nets

Conditioning on Question

- Represent q as

$$h_q^{(0)} = \mathrm{LSTM}(w_1^q, ..., w_{|q|}^q)_{|q|} \in \mathbb{R}^n$$

  where the final state is extracted from the output of the LSTM

- Subsequent updates:

$$h_q^{(l)} = \mathrm{FFN}(\Sigma_{v \in S_q} h_v^{(l)})$$

GRAFT-Nets

Conditioning: Attention over Relations

- Attention weight is computed using the following

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

- x_r is the relation vector of relation r
- Ensures that embeddings are propagated more along edges relevant to question

GRAFT-Nets

Conditioning: Directed Propagation

- Want to encourage multi-hop reasoning, which is required by some questions
- Use technique inspired by PPR in IR and maintain scalar $pr_v^{(l)}$ score which measures total weight of path from seed entity to the current node

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

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

GRAFT-Nets

Answer Selection

- The final representation of each node is used in binary classification

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

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

GRAFT-Nets

Regularization via Dropout

- To encourage model to exploit all available sources of information, randomly drop edges in training with probability p_0
- This is called *fact-dropout*
- This is necessary because model tends to prefer using information from KBs because its easier to do so

Related Work

- Das et al. (2017) attempts an early fusion strategy for QA over facts and texts based on Key-Value Memory Networks (KV-MemNNs) coupled with a universal schema; does not take advantage of relational structure between text and facts ⇒ this is used as a baseline comparison later on
- Traditional feature extraction methods of open vocabulary semantic parsing
- Knowledge Base Completion (KBC); does not account for question conditioning
- Many more in paper

Experiments & Results

Datasets

- WikiMovies-10K: 10K randomly sampled training questions from WikiMovies, along with original testing and validation sets
  - Sample traning set to create more difficult setting
  - Includes both KB and text corpus; links made using surface level matches
  - Retrieve top 50 entities around seeds and top 50 sentences from articles to create subgraph
  - 99.6% answer recall from subgraphs
- WebQuestionsSP: 4737 natural language questions posed over Freebase entities
  - Take top 500 entities around seeds and top 50 articles from Wikipedia
  - 94% answer recall from subgraphs
- Repeat above processes with KBs downsampled to 10%, 30%, and 50% or original KB to simulate incomplete KBs

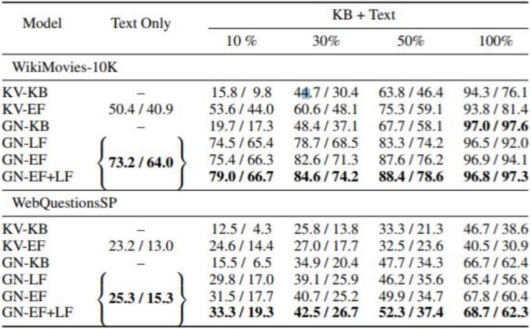
Experiments & Results

Compared Models

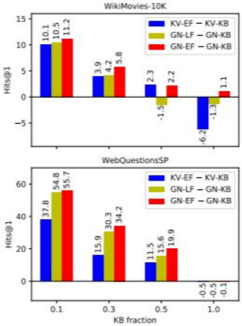
- **KV-KB**: Key Value Memory Networks using only KB and ignoring text
- **KV-EF**: Key Value Memory Network but with access to text
- **GN-KB**: GRAFT-Net model ignoring text
- **GN-LF**: Late fusion version of GRAFT-Net (train two separate models only on KB and only on text, then ensemble the two)
- **GN-EF**: Main GRAFT-Net model with early fusion
- **GN-EF+LF**: Ensemble over GN-EF and GN-LF models
- Compare using Hits@1 score (accuracy of top-predicted answer) and F1 score

Experiments & Results

Main Results



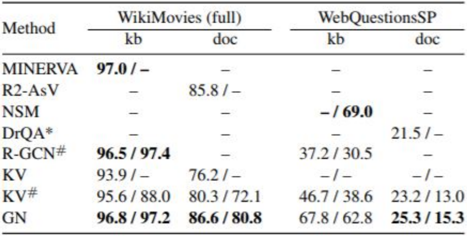
Model	Text Only	KB + Text			
		10 %	30%	50%	100%
WikiMovies-10K					
KV-KB	–	15.8 / 9.8	44.7 / 30.4	63.8 / 46.4	94.3 / 76.1
KV-EF	50.4 / 40.9	53.6 / 44.0	60.6 / 48.1	75.3 / 59.1	93.8 / 81.4
GN-KB	–	19.7 / 17.3	48.4 / 37.1	67.7 / 58.1	**97.0 / 97.6**
GN-LF	**73.2 / 64.0**	74.5 / 65.4	78.7 / 68.5	83.3 / 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**



Experiments & Results

Comparison to Specialized Methods

- Compare to models specifically tuned for QA using either only KB or only text
- Use full WikiMovies dataset for direct comparison; also train GRAFT-Nets only on KBs or only on text

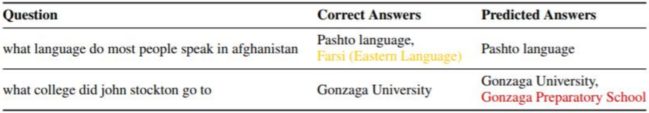


Method	WikiMovies (full)		WebQuestionsSP	
	kb	doc	kb	doc
MINERVA	**97.0 / –**	–	–	–
R2-AsV	–	85.8 / –	–	–
NSM	–	–	**– / 69.0**	–
DrQA*	–	–	–	21.5 / –
R-GCN#	**96.5 / 97.4**	–	37.2 / 30.5	–
KV	93.9 / –	76.2 / –	– / –	– / –
KV#	95.6 / 88.0	80.3 / 72.1	46.7 / 38.6	23.2 / 13.0
GN	**96.8 / 97.2**	**86.6 /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

Experiments & Results

Comparison to Specialized Methods: Analysis

- GRAFT-Net outperforms other state-of-the-art models in 3 out of 4 cases; it is worse than Neural Symbolic Machines
- Possible reasons:
  - In KB-only setting, the recall of the subgraph is only 90.2%; in oracle setting (where answer always in subgraph), F1 increases by 4.8%
  - Probability threshold is same for all questions, even though number of answers may vary significantly
  - GRAFT-Nets perform poorly when constraints are given by the question (i.e. who *first* voiced Meg in Family Guy?)



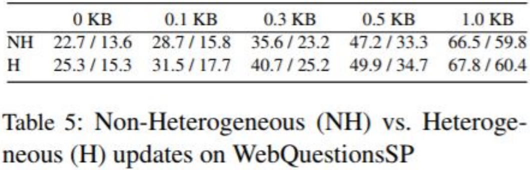
Question	**Correct Answers**	**Predicted Answers**
what language do most people speak in afghanistan	Pashto language, Farsi (Eastern Language)	Pashto language
what college did john stockton go to	Gonzaga University	Gonzaga University, Gonzaga Preparatory School

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

Experiments & Results

Effect of Model Components

- Heterogeneous Updates
  - If we use homogeneous updates, all entities $v \in L(d, \bullet)$ will receive the same update from d; different entities in same document cannot be differentiated $\Rightarrow$ significant loss of performance



	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

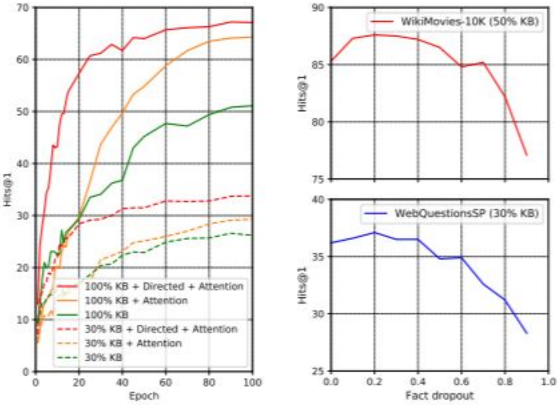
Experiments & Results

Effects of Model Components

- Conditioning on the Question
    - Ablation test on both attention and directed propagation methods; both components improved performance
- Fact Dropout
    - Performance increases as dropout increase until model unable to learn inference chain from KB

Experiments & Results

Effects of Model Components



Conclusion

- Investigated QA using text sources and incomplete KBs, a problem that has received limited attention in the past
- Introduced benchmarks for task by modifying existing QA datasets, comparing early and late fusion techniques
- Introduced GRAFT-Net for classifying nodes in question subgraph with both KB and text entities
- Achieved comparable performance with state-of-the-art models trained only on KBs or texts and outperformed baselines trained on text and incomplete KBs
- Possible future work: extending GRAFT-Nets to pick spans of texts as answers (rather than only entities), improving subgraph retrieval process