1 Research Question

Research question: Finding answers to open domain questions in natural language.

Difficulties:
- Large amount of data
- Hard to combine the information from knowledge bases and text articles

2 Overview Figure

![Overview Figure]

Figure 1: **Left:** 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. **Right:** Embeddings are propagated in the graph for a fixed number of layers (L) and the final node representations are used to classify answers.

3 Method Overview

- Overview:
  - Use graph structures to represent Knowledge base, and combine text articles into the graph
  - Effectively select a subgraph for a particular input.
  - Learn a heterogeneous graph network to represent the graph
• Before learning: Prepare a knowledge base. Also, create an article dataset from Wikipedia.

• Train a pair of question $q$ and answer $a$ in two steps:
  1. Extract a subgraph $G_q \subset G$ which contains the answer to the question with high probability.
  2. Use proposed model GRAFT-Net to learn graph representations in $G_q$, and do the classification.

4 GRAFT-Net

• In one sentence: A large Inductive Graph Recurrent model with multiple heterogeneous embedded inputs.

• Setting:
  – Knowledge Base is a graph where $\mathcal{K} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$. $\mathcal{V}$ is the set of vertices. $\mathcal{E}$ is the set of edges. $\mathcal{R}$ is the set of relations, which is a property of every edge.
  – Articles $\mathcal{D} = \{d_1, d_2, \ldots, d_D\}$, where each document is a sequence of words $d_i = w_{d_i1}, w_{d_i2}, \ldots, w_{d_i|d_i|}$.
  – Query $q$: a sequence of words $w_{q1}, w_{q2}, \ldots, w_{q|q|}$.
  – Answer $\{a\}$: a set of entities.
  – Problem: Give $(\mathcal{K}, \mathcal{D}, q)$, find $a$.

• Subgraph (Input to the GRAFT-Net):
  A graph represents by triplet $G_q = (\mathcal{V}_q, \mathcal{E}_q, \mathcal{R}^+)$. $\mathcal{V}_q$ includes two parts:
  1. A subset of $\mathcal{V}$, retrieved from top-K PageRank in $\mathcal{K}$.
  2. A subset of $\mathcal{D}$, retrieved by bag-of-word model and search engine.
  $\mathcal{E}_q$ is generally edges between $\mathcal{V}_q$:
  1. Subgraph of $\mathcal{V}_q$ in $\mathcal{K}$. That is, $(s, o, r) \in \mathcal{E} : s, o \in \mathcal{V}_q, r \in \mathcal{R}$.
  2. Links between documents and entities in $\mathcal{V}_q$, represents by $(v, d, r_L) : v \in \mathcal{K} \cap \mathcal{V}_q, d \in \mathcal{V}_q$, and $r_L$ is a special relation represents a link between document and entity.
  $\mathcal{R}^+ = \mathcal{R} \cup \{r_L\}$.

• Key equation: Updating rule for entity vertices in the graph $G_q$

$$ h_v^{(l)} = \text{FFN} \left( \begin{bmatrix} h_v^{(l-1)} \\ h_q^{(l-1)} \\ \sum_{v' \in N(v)} \alpha_{v'v} \psi_v(h_v^{(l-1)}) \\ \sum_{(d,p) \in M(v)} H_{d,p}^{(l-1)} \end{bmatrix} \right) \quad (1) $$
• Explanation of Equation 1:
  
  - **FFN**: feed-forward network, i.e., $\text{FFN}(x) = \sigma(Wx + b)$ similar to other recurrent structure. The weights are shared among vertices.
  
  - $l$: A number indicate current step in recurrent neural network. It keeps increasing from 0 to $L$.
  
  - $h_v$: The hidden state of entity $v$. $h_v^{(0)}$ is initialized either randomly or with some knowledge base embedding.
  
  - $h_q$: The hidden state of whole query. $h_q^{(0)}$ is initialized with an LSTM. $h_q$ is then updated with the combination of entities mentioned in the question.
    
    $h_q^{(l)} = \text{FFN}(\sum_{v \in S_q} h_v^{(l)})$  

  - Words in documents $H_{d,p}$: The hidden state of a document is the combination over hidden states of every words in the document, and is updated using a different recurrent network synchronously. Each word is updated with the hidden state of linking entities.
    
    $H_{d,p}^{(l)} = \text{FFN}(H_{d,p}^{(l-1)}, \sum_{v \in L(d,p)} h_v^{(l-1)})$  

  - Aggregate Neighbors: The third term in Equation 1 aggregates the neighbor hidden states.
    
    * Relations $r$: Each relation is modeled using a relation vector $x_r$
    * $a_{r'}^v$ is an attention weight, which is calculated by the query hidden state $h_q$ and relation embeddings $a_{r'} = \text{softmax}(x_r^T h_q^{(l-1)})$
    * $\psi_r$ is a weighted function:
      
      $\psi_r(h_v^{(l-1)}) = pr_v^{(l-1)} \text{FFN}(x_r, h_v^{(l-1)})$  

    * $pr_v$ is a scalar weight calculated using PageRank.
      
      $pr_v^{(0)} = \begin{cases} \frac{1}{|S_q|}, & \text{if } v \in S_q \\ 0, & \text{otherwise} \end{cases}$
      
      $pr_v^{(l)} = (1 - \lambda)pr_v^{(l-1)} + \lambda \sum_{r} \sum_{v' \in N_r(v)} a_{r'}^v pr_v^{(l-1)}$

  - **Output of GRAFT-Net**:
    
    After $l$ iterations, a probability over all entities is calculated by
    
    $Pr(v \in \{a\}_q | G_q, q) = \sigma(w^T h_v^{(l)} + b)$
5 Evaluation:

5.1 Dataset

WikiMovies-10K and WebQuestionsSP.

- WikiMovies-10K is a subset of dataset WikiMovies[1] containing questions on movies. Knowledge Base and Document corpus is gathered from Wikipedia by [1].

5.2 Models in comparison

- KV-KB: Key Value Memory Networks[1] on only KB input.
- KV-EF: Key Value Memory Networks with both KB and text.
- GN-KB: GRAFT-Net only on KB.
- GN-LF: late-fusion GRAFT-Net, trained on text/KB seperately and then combine.
- GN-EF: Main model.
- GN-EF+LF: Ensemble over GN-EF and GN-LF.

5.3 Result

1. Get best result with full knowledge base on GN-EF+LF model, as in Table 2.
2. Get comparable result to state-of-the-art with only text or KB.
3. Without heterogeneous update the performance is worse.

<table>
<thead>
<tr>
<th>Model</th>
<th>Text Only</th>
<th>10%</th>
<th>30%</th>
<th>50%</th>
<th>100%</th>
</tr>
</thead>
<tbody>
<tr>
<td>WikiMovies-10K</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KV-KB</td>
<td>-</td>
<td>15.8</td>
<td>9.8</td>
<td>44.7</td>
<td>30.4</td>
</tr>
<tr>
<td>KV-EF</td>
<td>50.4/40.9</td>
<td>53.6</td>
<td>44.0</td>
<td>60.6</td>
<td>48.1</td>
</tr>
<tr>
<td>GN-KB</td>
<td>-</td>
<td>19.7</td>
<td>17.3</td>
<td>48.4</td>
<td>37.1</td>
</tr>
<tr>
<td>GN-LF</td>
<td>75.3/65.4</td>
<td>78.7</td>
<td>65.4</td>
<td>83.3</td>
<td>74.2</td>
</tr>
<tr>
<td>GN-EF</td>
<td>75.4/66.3</td>
<td>82.6</td>
<td>71.3</td>
<td>87.6</td>
<td>76.2</td>
</tr>
<tr>
<td>GN-EF+LF</td>
<td>79.8/66.3</td>
<td>84.6</td>
<td>74.2</td>
<td>88.4</td>
<td>76.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>Text Only</th>
<th>10%</th>
<th>30%</th>
<th>50%</th>
<th>100%</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebQuestionsSP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KV-KB</td>
<td>-</td>
<td>12.5</td>
<td>4.3</td>
<td>25.8</td>
<td>13.8</td>
</tr>
<tr>
<td>KV-EF</td>
<td>23.2/13.0</td>
<td>24.6</td>
<td>14.4</td>
<td>27.0</td>
<td>17.7</td>
</tr>
<tr>
<td>GN-KB</td>
<td>-</td>
<td>15.5</td>
<td>8.5</td>
<td>34.9</td>
<td>20.4</td>
</tr>
<tr>
<td>GN-LF</td>
<td>29.8/17.0</td>
<td>39.1</td>
<td>25.9</td>
<td>46.2</td>
<td>35.6</td>
</tr>
<tr>
<td>GN-EF</td>
<td>31.5/17.7</td>
<td>40.7</td>
<td>25.2</td>
<td>49.9</td>
<td>34.7</td>
</tr>
<tr>
<td>GN-EF+LF</td>
<td>33.3/19.3</td>
<td>42.5</td>
<td>26.7</td>
<td>52.3</td>
<td>37.4</td>
</tr>
</tbody>
</table>

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.

2. Get comparable result to state-of-the-art with only text or KB.
3. Without heterogeneous update the performance is worse.
References
