Pretrained Transformers for Simple Question Answering over Knowledge Graphs

D. Lukovnikov, A. Fischer, J. Lehmann
A Simple Question

Name the **studio** which made **Iron man**?

```
SELECT ?uri WHERE {
}
```
**Question Answering is Hard!**

Name all the movies in which Robert Downey Jr acted?
Which movies have RDJ?
Flicks where I can see Robert DJ?
Find me all the films casting Robert Downey Jr?
List all the movies starring Robert Downey Junior?
RDJ has acted in which movies?

```
SELECT ?uri WHERE {
}
```
Background
Transformers[^20]

- Alternative to RNNs and CNNs for sequences
- Uses (mostly) linear layers and attention
  - Multi-head attention!
- Every word attends to every other word, several times
- Can be used for encoding a sequence
BERT\textsuperscript{[5]}

- A Transformer model
  - Different position encodings (learned)
  - Adds sequence type embeddings
- Works on WordPiece level (instead of word level)
- Pretrained on masked LM task and sentence pair task
  - MLM: mask out some words, learn to predict masked words
  - Sentence pair: does this sentence follow another?
- Can be finetuned on other NLP tasks
  - Shows great performance
Approach

Overview  Span prediction  Relation classification  Entity candidate generation  Reranking
Overview

1. Predict entity span
   a. Single entity assumed $\rightarrow$ one contiguous span
2. Predict relation
3. Generate entity candidates
   a. Based on string match with predicted span
4. Re-rank the queries
   a. Take predictions from (2) and (3)
   b. Remove non-existing triples

similar to [13, 16]
Overview

1. Predict entity span
   a. **Sequence tagging/...**

2. Predict relation
   a. **Sequence classification**

3. Generate entity candidates
   a. Based on string match with predicted span

4. Re-rank the queries
   a. Take predictions from (2) and (3)
   b. Remove non-existing triples

Can be done together
Entity span and relation prediction

Model:
1. Encode with BERT
2. Entity span:
   a. Use start-of-sequence classifier over sequence
   b. Use end-of-sequence classifier over sequence
3. Relation:
   a. Use classifier over possible relations
   b. [CLS] token representation
Entity span prediction

Start-of-sequence and end-of-sequence classifiers:

\[ p(i = \text{START}|x_1, \ldots, x_N) = \frac{e^{x_i^{L+1}^T w_{\text{START}}}}{\sum_{j=1}^N e^{x_j^{L+1}^T w_{\text{START}}}} , \]

⇒ softmax over sequence

Adds parameter vectors \( w_{\text{START}} \) and \( w_{\text{END}} \)

Trained on pseudo-gold spans obtained by aligning sentences with entity labels (similar to previous work)
Relation prediction

Simple classifier over relations:

\[ p(r = R_i | x_1, \ldots, x_N) = \frac{e^{x_{CLS}^T w_{R_i}}}{\sum_{k=1}^{N_R} e^{x_{CLS}^T w_{R_k}}} , \]

Adds parameter vectors \( w_{R_i} \), for every relation

1. Takes [CLS] token representation
   a. = sequence level representation for BERT
2. Applies softmax classifier

Trained from data given in dataset
Entity Span and Relation prediction, together

\[ p(r = R_i|x_1, \ldots, x_N) = \frac{e^{x^{L+1}_i w_{R_i}}}{\sum_{k=1}^{N_R} e^{x^{L+1}_i w_{R_k}}} , \]

\[ p(i = \text{START}|x_1, \ldots, x_N) = \frac{e^{x^{L+1}_{\text{START}} w_{\text{START}}}}{\sum_{j=1}^{N} e^{x^{L+1}_j w_{\text{START}}}} , \]

Entity span and relation are predicted together

- question pattern didn’t improve in our setup
- Single model
  - Simple
  - BERT is expensive
Entity candidate generation

Given question and predicted entity span

Who wrote It?

Retrieve entities whose label matches the span

⇒ 1. :lt_(2017 film)
2. :lt_(novel)
3. :lt_(miniseries)
4. ...
Reranking

- Taking best entity and best relation might not be best
  → might be incompatible ("area" of "Michael Jackson")
  ⇒ discard entity-relation pairs that don’t occur in KG

- Ranking criteria:
  1. String similarity of entity
  2. Predicates with higher prediction probability under model
  3. Entity in-degree
Experiments
Experimental Setup

- **Dataset**: SimpleQuestions[3] - single-fact questions over subset of Freebase
  - 75k+ training examples
- **Optimizer**: Adam
- **Metrics**:
  - Spans: F1 and accuracy
  - Relations: Accuracy
Baseline

- Simple BiLSTM, similar to [13]
- Glove embeddings
- One BiLSTM for relation classification
  - Not doing question pattern
- One BiLSTM with start/end classifiers for entity span
  - Similar to BERT-based model
- Our re-implemented BiLSTM baseline is on par with [13]
Questions

Q1. How does BERT-based model compare to the baseline for subtasks individually?
   a. Not affected by choice of re-ranking/candidate generation

Q2. How does BERT-based model compare to previous works on whole task

Q3. How does BERT-based model degrade with fewer data compared to baseline (and how does fewer data affect baseline performance)

Q4. Does BERT-based model actually learn meaningful patterns?
# Results: Q1 - subtask results

<table>
<thead>
<tr>
<th>Accuracy Avg. F1 F1*</th>
<th>Accuracy</th>
<th>R@N BiLSTM BiLSTM BERT</th>
</tr>
</thead>
<tbody>
<tr>
<td>BiLSTM [13] 93.8 97.0 97.1</td>
<td>BiGRU [13] 82.3</td>
<td>1 67.8 76.45 77.17</td>
</tr>
<tr>
<td>CRF [13] 90.2</td>
<td>CNN [13] 82.8</td>
<td>5 82.6 87.46 88.18</td>
</tr>
<tr>
<td>BiLSTM (ours) 95.6 97.8 97.9</td>
<td>BiLSTM (ours) 82.8</td>
<td>20 88.7 91.47 92.13</td>
</tr>
<tr>
<td>BERT (ours)</td>
<td>BERT (ours) 83.6</td>
<td>50 91.0 93.07 93.71</td>
</tr>
<tr>
<td></td>
<td></td>
<td>150 –</td>
</tr>
</tbody>
</table>

(a) Entity span prediction.

(b) Relation prediction.

Table 2: Entity recall on validation set.

- validation set numbers
Results: Q1 - subtask results

<table>
<thead>
<tr>
<th>Model</th>
<th>Entity Span Accuracy</th>
<th>Entity Span Avg. F1</th>
<th>Relation Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>BiLSTM</td>
<td>93.2</td>
<td>96.7</td>
<td>82.4</td>
</tr>
<tr>
<td>BERT</td>
<td>95.2</td>
<td>97.5</td>
<td>83.5</td>
</tr>
</tbody>
</table>

Table 4: Component results on test set.
# Results: Q2 - end results

- **Effect of re-ranking on relation accuracy**
  - 83.5% → 86.6%

- **Error cases:**
  - 35%: both entity and relation wrong
  - 41%: only entity wrong
  - 24%: only relation wrong

- **Wrong entity cases:**
  - 28.6%: entity not among candidates

<table>
<thead>
<tr>
<th>Approach</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>MemNN [3]</td>
<td>61.6</td>
</tr>
<tr>
<td>Attn. LSTM [6]</td>
<td>70.9</td>
</tr>
<tr>
<td>GRU [11]</td>
<td>71.2</td>
</tr>
<tr>
<td>BuboQA [13]</td>
<td>74.9</td>
</tr>
<tr>
<td>BiGRU [4]</td>
<td>75.7</td>
</tr>
<tr>
<td>Attn. CNN [23]</td>
<td>76.4</td>
</tr>
<tr>
<td>HR-BiLSTM [24]</td>
<td>77.0</td>
</tr>
<tr>
<td>BiLSTM-CRF [16]</td>
<td>78.1</td>
</tr>
<tr>
<td>BERT (ours)</td>
<td>77.3</td>
</tr>
</tbody>
</table>
Results: Q3 - fewer data

Setup:

- Retain only fraction of training data and train
  - Maximize number of covered relations
- $\Rightarrow$ BERT degrades better than baseline

<table>
<thead>
<tr>
<th>Entity Span</th>
<th>0.03%</th>
<th>0.2%</th>
<th>1%</th>
<th>2.5%</th>
<th>5%</th>
<th>10%</th>
<th>25%</th>
<th>50%</th>
<th>75%</th>
<th>100%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(22)</td>
<td>(151)</td>
<td>(757)</td>
<td>(1k9)</td>
<td>(3k8)</td>
<td>(7k6)</td>
<td>(18k9)</td>
<td>(37k9)</td>
<td>(56k8)</td>
<td>(75k7)</td>
</tr>
<tr>
<td>BiLSTM</td>
<td>33.1</td>
<td>64.5</td>
<td>74.0</td>
<td>78.1</td>
<td>82.5</td>
<td>85.5</td>
<td>90.1</td>
<td>92.0</td>
<td>93.4</td>
<td>93.8</td>
</tr>
<tr>
<td>BERT</td>
<td>62.5</td>
<td>79.1</td>
<td>85.4</td>
<td>88.9</td>
<td>90.8</td>
<td>92.4</td>
<td>94.2</td>
<td>94.9</td>
<td>95.5</td>
<td>95.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Relation</th>
<th>BiLSTM</th>
<th>BERT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>26.5</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td>56.3</td>
<td>72.4</td>
</tr>
<tr>
<td></td>
<td>79.0</td>
<td>81.3</td>
</tr>
<tr>
<td></td>
<td>82.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>29.6*</td>
<td>48.6</td>
</tr>
<tr>
<td></td>
<td>67.5</td>
<td>76.5</td>
</tr>
<tr>
<td></td>
<td>80.1</td>
<td>82.6</td>
</tr>
<tr>
<td></td>
<td>83.6</td>
<td></td>
</tr>
</tbody>
</table>
Results: Q4 - internal behavior

BERT-small: 12 layers, 12 heads/layer

⇒ 144 attentions per token

[CLS] token attentions:

● Indicate which words contributed more to relation classification

Compute average of all 144 attentions for [CLS] token:

\[ \beta_{i,j} = \frac{\sum_{l=1}^{L} \sum_{h=1}^{M} \alpha_{l,h,i,j}}{L \cdot M}, \]
Results: Q4 - internal behavior

(a) Before fine-tuning

(b) After fine-tuning
Results: Q4 - internal behavior
Thank You
References

SAME reference numbers as in paper
References (Icons)

Sherlock holmes by Matthew Davis from the Noun Project
Empire State Building by Jake Dunham from the Noun Project
Golden gate bridge by icon 54 from the Noun Project
Statue of Liberty by Berkah Icon from the Noun Project
MARVEI is from flaticon
Sherlock Holmes is from FlatIcon
Iron Man by Tatyana Kyul from the Noun Project
Ferguson by priyanka from the Noun Project