Knowledge Bases

- A knowledge base allows for rapid search, retrieval, and reuse
- Stores information as answers to questions or solutions to problems
- Can be fed into a language model
Examples of Knowledge bases
• Concepts like *classes* and *individuals* are modeled as nodes

• *Relations* as edges of graphs

• *Classes* - concepts like documents, events, or subjects

• *Individuals* - instances of a class or an object

• *Relations* - capture relationships between classes and individuals
  • *is-type-of*, *is-instance-of*, and *has-attribute*

---

**WordNet Search - 3.1**

- [WordNet home page](#) - [Glossary](#) - [Help](#)

**Word to search for:** smile

**Display Options:** [Select option to change] ▼ [Change]

Key: "S." = Show Synset (semantic) relations, "W." = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

**Noun**

- S: (n) smile, smiling, grin, grinning (a facial expression characterized by turning up the corners of the mouth, usually shows pleasure or amusement)

**Verb**

- S: (v) smile (change one's facial expression by spreading the lips, often to signal pleasure)
- S: (v) smile (express with a smile) "She smiled her thanks"
How knowledge bases are used in NLP models:

• Entity extraction - replace or augment entity occurrences in text
Coreference resolution:

“I voted for Nader because he was most aligned with my values,” she said.

Entity Linking:

“Barack Obama was born in Hawaii.”
Proposed Solution:

• Ask the model to fill in masked tokens
• “Alex was born in [MASK]”
• Pre-trained high-capacity models such as ELMo and BERT store vast amounts of linguistic knowledge useful for downstream tasks

The Pros:
- Requires no schema engineering
- No need for human annotations
- Supports a more diverse/open set of inquiries
Questions this paper addresses:

• How much relational knowledge do they store?
• How does this differ for different types of knowledge such as facts about entities, common sense, and general question answering?
• How does their performance without fine-tuning compare to symbolic knowledge bases automatically extracted from text?
LAMA (Language Model Analysis) Probe

• consisting of a set of knowledge sources, each comprised of a set of facts (subject, relation, object)
• Success depends on predicting masked objects such as “Dante was born in ___”
• tested for a variety of types of knowledge: relations between entities stored in Wikidata, common sense relations between concepts from ConceptNet, and knowledge necessary to answer natural language questions in SQuAD.

• Key Steps:
  • Query each model for a missing token
  • Evaluate each model based on how highly they rank the ground truth token against every word in a fixed candidate vocabulary
Knowledge Sources Used:

- Google-RE - contains ~60K facts manually extracted from Wikipedia
  - Only utilized 3 relations: “place of birth”, “date of birth” and “place of death”
  - manually defined a template for each considered relation, e.g., “[Adam] was born in [Illinois]” for “place of birth”
- T-Rex - is a subset of Wikidata triples
  - Much larger than Google-RE with broader relations
  - Facts were automatically aligned to Wikipedia (can be noisy)
- SQuAD
  - Question-answering dataset
  - a subset of 305 context-insensitive questions with single token answers
  - rewriting “Who developed the theory of relativity?” as “The theory of relativity was developed by __”.
- ConceptNet
  - Multilingual knowledge base, initially built on top of Open Mind Common Sense sentences
  - English parts that have single-token objects covering 16 relations
Language Models evaluated:

• **Unidirectional Language Models:**
  
  • Given a string of input tokens $w = [w_1, w_2, \ldots, w_n]$, assign probability $p(w)$

  
  $p(w) = \prod_t p(w_t | w_{t-1}, \ldots, w_1)$

  • Using neural language models:

  $p(w_t | w_{t-1}, \ldots, w_1) = \text{softmax}(W h_t + b)$

  • $h_t =$ output vector at position $t$
  • $W =$ learned parameter matrix
Fairseq-fconv

- Multiple layers of gated convolutions
- Trained on the WikiText-103 corpus

Transformer-XL

- Large-scale LM based on the Transformer
- Takes into account a longer history
- Used relative instead of absolute positional encoding
- Trained on the WikiText-103 corpus
• **Bidirectional Language Models:**
  
  • **ELMO:**
    
    • Given a string of input tokens $w = [w_1, w_2, …, w_n]$ and position $1 \leq i \leq N$, estimate
    
    $$p(w_i) = p(w_i \mid w_1, \ldots, w_{i-1}, w_{i+1}, \ldots, w_N)$$
    
    • ELMo: Forward and backward LSTM, resulting in $\overrightarrow{h_i}$ and $\overleftarrow{h_i}$
    
    • Trained on the Google Billion Word dataset
    
    • ELMo 5.5B
      
      • Trained on English Wikipedia and monolingual news crawl from WMT 2008-2012
• **BERT**:  
  - Transformer architecture  
  - Trained on the BookCorpus and English Wikipedia  
  - *language modelling* (15% of tokens were masked and BERT was trained to predict them from context) and *next sentence prediction* (if a chosen next sentence was probable or not given the first sentence)  
  - BERT-base (12 encoders with 12 bidirectional self-attention heads)  
  - BERT-large (24 encoders with 16 bidirectional self-attention heads)
Methodology

- **ELMo**: averaged forward and backward probabilities from the corresponding softmax layers

- **BERT**: masked the token at position $t$, fed output to vector corresponding to masked token ($h_t$) into softmax layer

\[
p(w) = \prod p(w_t | w_{t-1}, \ldots, w_1).
\]

\[
p(w_t | w_{t-1}, \ldots, w_1) = \text{softmax}(W h_t + b)
\]

$W = ['\text{compare}', '\text{language}', '\text{models}', 'to', '\text{canonical}', '\text{ways}']$

\[
p('\text{ways}') = \text{softmax}(W h_{\text{ways}} + b)
\]

**Uni-directional**:

\[h_{t+1} = \text{output vector at 'canonical'}\]

**Bi-directional**:

- **ELMo**: ($t = 2 \Rightarrow '\text{models}')$
  \[\overrightarrow{h_{t-1}} = \text{output vector at 'language'}\]
  \[\overleftarrow{h_{t+1}} = \text{output vector at 'to'}\]
Baselines

• **Freq**
  - subject and relation pair, this baseline ranks words based on how frequently they appear as objects for the given relation in the test data

• **Relation Extraction (RE)**
  - extracts relation triples from a given sentence using an LSTM-based encoder and an attention mechanism
  - constructs a knowledge graph of triples
  - At test time, they queried this graph by finding the subject entity and then rank all objects in the correct relation based on the confidence scores by the RE

• **DrQA**
  - a popular system for open-domain question answering
  - Two-step pipeline:
    • First, a TF/IDF information retrieval step is used to find relevant articles from a large store of documents (e.g. Wikipedia)
    • Secondly, on the retrieved top k articles, a neural reading comprehension model then extracts answers
Metrics

• Rank-based metrics

• For multiple valid objects for Subject-Relation pair, removed all other valid objects from the candidates when ranking at test time other than the ones they were testing

• Mean precision at k (P@k)
  • For a given fact, this value is 1 if the object is ranked among the top k results, 0 otherwise
Considerations in LAMA

• Manually Define Templates:
  • Manually defined a template that queries for the object slot for each relation
  • For example, for a relation ID “works-for”, and the user asks for “is-working-for”, the accuracy would be 0
  • e.g., “[S] was born in [O]” for “place of birth”.

• Single Token

• Object Slots
  • Only in triples (subject, relation, object)

• Intersection of Vocabularies
  • ELMO uses ~800K tokens compared to BERT’s ~30K tokens
  • Intersection of 2 vocabularies yielding ~21K tokens
### Results

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Relation</th>
<th>Statistics</th>
<th>Baselines</th>
<th>KB</th>
<th>LM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>#Facts</td>
<td>#Rel</td>
<td>Freq</td>
<td>DrQA</td>
</tr>
<tr>
<td>Google-RE</td>
<td>birth-place</td>
<td>2937</td>
<td>1</td>
<td>4.6</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>birth-date</td>
<td>1825</td>
<td>1</td>
<td>1.9</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>death-place</td>
<td>765</td>
<td>1</td>
<td>6.8</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>5527</td>
<td>3</td>
<td>4.4</td>
<td>-</td>
</tr>
<tr>
<td>T-REx</td>
<td>1-1</td>
<td>937</td>
<td>2</td>
<td>1.78</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>N-1</td>
<td>20006</td>
<td>23</td>
<td>23.85</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>N-M</td>
<td>13096</td>
<td>16</td>
<td>21.95</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>34039</td>
<td>41</td>
<td>22.03</td>
<td>-</td>
</tr>
<tr>
<td>ConceptNet</td>
<td>Total</td>
<td>11458</td>
<td>16</td>
<td>4.8</td>
<td>-</td>
</tr>
<tr>
<td>SQuAD</td>
<td>Total</td>
<td>305</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 2: Mean precision at one (P@1) for a frequency baseline (Freq), DrQA, a relation extraction with naïve entity linking (RE$_n$), oracle entity linking (RE$_o$), fairseq-fconv (Fs), Transformer-XL large (Txl), ELMo original (Eb), ELMo 5.5B (E5B), BERT-base (Bb) and BERT-large (Bl) across the set of evaluation corpora.
Discussion of Results

- From earlier example, “Adam was born in [MASK]”
- BERT-Large (last column) outperformed all models by a substantial margin
- REn - naïve entity linking, i.e. exact string matching
- REo - uses an oracle for entity-linking, i.e. any given (s, r, o) in sentence x, if any other (s', r, o') has been extracted in the same sentence, s will be linked to s’, and o to o’
• More facts and relations than Google-RE
• BERT-Large performed better on 1-to-1 relations, i.e. “capital-of”
• N-1: Multiple valid subjects-relations→ 1 correct object
• N-M relations: multiple objects for a subject-relation pair. i.e. “Brian owns [car, laptop, iPhone, etc]”
• BERT-Large achieved best performance for ConceptNet
• Able to retrieve commonsense knowledge at a similar level to factual knowledge
- Open domain cloze-style (fill in the blanks)
- Huge performance gap between BERT-Large and supervised DrQA
- Note: BERT and ELMo were both unsupervised and not fine-tuned for this task
- In terms of P@10 (Top-10 best answers), gap is remarkably small (57.1 for Bl and 63.5 for DrQA)

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Relation</th>
<th>Statistics</th>
<th>Baselines</th>
<th>KB</th>
<th>LM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>#Facts</td>
<td>Freq</td>
<td>DrQA</td>
<td>RE_1</td>
</tr>
<tr>
<td>SQuAD</td>
<td>Total</td>
<td>305</td>
<td>-</td>
<td>37.5</td>
<td>-</td>
</tr>
</tbody>
</table>
Conclusions

• For an unsupervised, not fine-tuned, pre-trained model BERT-Large, it is possible to recall knowledge better than its competitors, comparable to that of a knowledge base extracted with an off-the-shelf relation extractor and an oracle-based entity linker from a corpus known to express the relevant knowledge.

• Factual knowledge can be recovered surprisingly well from pre-trained language models, however, for some relations (particularly N-to-M relations) performance is very poor.

• This paper focused on the as-is knowledge inherent in the weights of existing pre-trained models which are often used as starting points for most research works.

• Language models trained on ever-growing corpora might become a viable alternative to traditional knowledge bases extracted from text in the future.
Limitations

- Only used Single-Token objects as prediction targets
- Chose only query objects in triples
- Still spent time manually defining templates for each relation
Questions/Thoughts?