Question Auto-Completion using a Typed LSTM Language Model

MASTER THESIS BY NATALIE PRANGE
Query Auto-Completion

Source: https://i.imgur.com/eObv3jI.jpg
Query Auto-Completion

Goals of Query Auto-Completion (QAC):

• Reduce typing effort
• Prevent spelling errors
• Assist in phrasing a query

A QAC system must therefore present completion predictions ...

• ... after a minimal amount of keystrokes
• ... in real time
• ... properly ranked
Motivation

Most QAC research focuses on QAC using query logs

**Problems with query log approaches:**

- Not available to search engines with a small user base or recently deployed search engines
- Publicly available query logs are outdated
- Queries that have never been asked before can not be predicted

→ Use a language-model-based approach
Tackling the Data Sparsity Problem

A common problem when working with language models:

Receiving an input that did not occur in the training data

Solution: Use a typed language model (LM)

Concrete entities are replaced by abstract types
  E.g. “Who played Gandalf in The Lord of the Rings?”
  → “Who played [fictional character] in [film]?”

Entities are later inserted using
  • an entity prominence score
  • co-occurrence
  • word vector similarity
Overview of the System

Pre-Computed Components
- Typed Language Model
- Entity Prominence Score
- Co-Occurrence Counts
- Word2Vec Model

Question Auto-Completion
- Pre-Process User Input
- Predict Words & Types
- Insert Entities
- Score & Rank
- Add & Remove Entities
- Completion Predictions

Question Prefix
Building the Typed Language Model

Create an entity-to-type mapping using Wikidata entities and classes

**Challenge: Types must neither be too general nor too specific**

→ Use two types: a more specific type (primary type) and a more general type (secondary type)
  
  • E.g. Gandalf → fictional character (primary)
  → creative work (secondary)

→ Use a hand-crafted and sorted list of preferred types
Training Data for the Language Model

• Dataset with 11,290,367 questions

• 97% of questions stem from the WikiQuestions dataset:
  • Wikipedia sentences with entity mentions transformed into questions

• The remaining 3% of questions stem from the ClueWeb12 corpus
  • Questions from English web pages with entity mentions

• Entities are replaced by their types
  • E.g. “Who is Gandalf?” → “Who is [fictional character/creative work]?”
Training the LSTM Language Model

LSTM network = Long Short Term Memory network

Architecture:
• Embedding layer of size 100
• Two stacked LSTM layers of size 512
• Softmax layer of size of the vocabulary
→ Given a question prefix, the network outputs a probability distribution over the vocabulary

Training:
• Batch size of 512
• 15 epochs
→ training time: ca. 9 days and 21 hours
The typed LM should predict the next word or type given a question prefix

E.g. “Who directed the Lord of t”

Standard LM:
• predict words for the current word prefix “t” given the context words C = ("who", "directed", "the", "Lord", "of")

Typed LM:
• predict words for all possible current word prefixes i.e. “t”, “of t”, “Lord of t” ...
→ predict “Who directed [film]”
Inserting Entities for Predicted Types

Extract candidate entities that...
  • ... have the predicted primary type (as primary or secondary type)
  • ... start with the current word prefix

Define **insertion context words** $I(C)$ as set of **entities** contained in a question prefix and **<type>** if question starts with “Which **<type>**”
  • E.g. for the context words $C = (“Which”, “country”, “did”, “J.R.R. Tolkien”)$$
    $\rightarrow I(C) = (“country”, “J.R.R. Tolkien”)$$
Inserting Entities for Predicted Types

If insertion context words $I(C) = \emptyset$:

→ Use an entity prominence score to score candidate entities
  • Based on an entity’s Wikibase sitelink count
  • Counts are normalized to a score between 0 and 1

Else:

→ Use the co-occurrence count between insertion context words and candidate entity
  • Co-occurrence is computed over a Wikipedia dump with entity mentions
  • Counts are normalized to a score between 0 and 1
Ranking Completion Predictions

Completion predictions are ranked according to a final score

Components of the final score:
• Language model probability
• Insertion score
• Penalty factors

Language model probability
• Incorporate probability to observe given context words
• For a completion prediction \( w \) and context words \( C = (w_1, w_2, ..., w_i) \)

\[
p_{lm}(w|C) = \hat{p}(C) \cdot p(w|C)
\]

\( \hat{p}(C) \): Discounted LM probability to observe the context words \( C \)
\( p(w|C) \): LM probability of the predicted word or type of \( w \) given the context words \( C \)
Ranking Completion Predictions

Entity insertion score
• Normalized sitelink count or normalized co-occurrence count for entities

Normal word insertion score
• Balance prediction of normal words vs. entities
• If $I(C) = \emptyset$ : Use constant score of 0.01
• Else: compute word vector similarity between non-stopwords in the question prefix and the predicted normal word
Ranking Completion Predictions

**Penalty factors**

- Penalize prediction of consecutive entities with penalty factor $g_{ce}$
  
  
  $g_{ce} = \begin{cases} 
  0.04 & \text{if completion prediction is an entity and previous word is an entity} \\
  1 & \text{else} 
  \end{cases}$

- Penalize prediction of the entity type [human] with penalty factor $g_h$
  
  $g_h = \begin{cases} 
  0.02 & \text{if LM predicts type [human]} \\
  1 & \text{else} 
  \end{cases}$

- Penalize alias-based completion predictions with penalty factor $g_a$
  
  $g_a = \begin{cases} 
  0.6 & \text{if completion prediction is based on entity alias instead of entity label} \\
  1 & \text{else} 
  \end{cases}$

**Final score:**

$$s = p_{lm} \times (s_{insert} \times g_a)^{0.3} \times g_{ce} \times g_h$$
Adding and Removing Entities

- Append entities when not enough completion predictions were generated using co-occurrence
  - Use product of word vector similarity and sitelink count to score candidate entities
- Append completely typed entities
- Remove double completion predictions
Evaluation

Multiple-True-Completions Evaluation
• Evaluate over set of 100 question prefixes along with reasonable completion predictions

• Measure precision at $k = 5$ (P@5)

$$P@k = \frac{|Q_{true} \cap Q_{results}^k|}{k}$$

$Q_{true}$ : set of true completions
$Q_{results}^k$ : set of top k completions predicted by the system

• Measure average precision (AP)

$$AP = \frac{\sum_{i=1}^{n} P@r_i}{n}$$

$r_1, \ldots, r_n$ : list of positions at which predictions from $Q_{true}$ appear in $Q_{results}$
Evaluation

- Measure **normalized discounted cumulative gain at k = 5** (nDCG@5)

Discounted cumulative gain at k:

$$\text{DCG@k} = rel_1 + \sum_{i=2}^{k} \frac{rel_i}{\log_2(i+1)}$$

$rel_i$: relevance score for the completion predicted at rank $i$

Normalized discounted cumulative gain at k:

$$\text{nDCG@k} = \frac{\text{DCG@k}}{\text{IDCG@k}}$$

$\text{IDCG@k}$: ideal discounted cumulative gain
Evaluation

Single-True-Completion Evaluation

- Base test set: 10,000 random questions
- 1ICW test set: 10,000 questions with one insertion context word
- >1ICW test set: 10,000 questions with more than one insertion context word

- Measure **mean reciprocal rank (MRR)**
  \[ RR = \frac{1}{r} \]
  - \( r \): rank of the correct completion prediction
  - Compute RR for each word in each question after its 1st letter has been typed
  - Report the mean over all computed RR scores

- Measure **required user interaction (RUI)**
  \[ RUI = \frac{\text{# user interactions needed given completion predictions}}{\text{# user interactions needed without completion predictions}} \]
  - Typing a letter and selecting a completion prediction each count as one interaction
Evaluated Versions

Three main versions that differ in the entity insertion method for $I(C) \neq \emptyset$

- sitelinks: Entity insertion based on sitelink count (Baseline)
- sitelinks + w2v: Entity insertion based on product of sitelink count and word vector similarity
- co-occurrence: Entity insertion based on co-occurrence

Additional versions:

- co-occurrence w/o $g_{ce}$: no penalty for consecutive entities
- co-occurrence w/o $g_h$: no penalty for prediction of type [human]
- co-occurrence w/o w2v fill-up: no filling up of entities using word vector similarity
## Multiple-True-Completions Evaluation Results

<table>
<thead>
<tr>
<th></th>
<th>P@5</th>
<th>AP</th>
<th>nDCG@5</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>sitelinks</td>
<td>0.286</td>
<td>0.375</td>
<td>0.422</td>
<td>0.64 s</td>
</tr>
<tr>
<td>sitelinks + w2v</td>
<td>0.322</td>
<td>0.464</td>
<td>0.524</td>
<td>0.77 s</td>
</tr>
<tr>
<td>co-occurrence w/o (g_{ce})</td>
<td>0.338</td>
<td>0.481</td>
<td>0.551</td>
<td>0.70 s</td>
</tr>
<tr>
<td>co-occurrence w/o (g_{h})</td>
<td>0.316</td>
<td>0.473</td>
<td>0.543</td>
<td>0.70 s</td>
</tr>
<tr>
<td>co-occurrence w/o w2v fill-up</td>
<td>0.340</td>
<td>0.489</td>
<td>0.564</td>
<td>0.65 s</td>
</tr>
<tr>
<td>co-occurrence</td>
<td>0.344</td>
<td>0.500</td>
<td>0.572</td>
<td>0.69 s</td>
</tr>
</tbody>
</table>
# Single-True-Completion Evaluation Results

<table>
<thead>
<tr>
<th></th>
<th>Base Test Set</th>
<th></th>
<th>1ICW Test Set</th>
<th></th>
<th>&gt;1ICW Test Set</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MRR</td>
<td>RUI</td>
<td>MRR</td>
<td>RUI</td>
<td>MRR</td>
<td>RUI</td>
</tr>
<tr>
<td>sitelinks</td>
<td>0.532</td>
<td>0.542</td>
<td>0.537</td>
<td>0.538</td>
<td>0.529</td>
<td>0.525</td>
</tr>
<tr>
<td>sitelinks + w2v</td>
<td><strong>0.533</strong></td>
<td>0.541</td>
<td><strong>0.540</strong></td>
<td>0.535</td>
<td><strong>0.533</strong></td>
<td>0.523</td>
</tr>
<tr>
<td>co-occurrence</td>
<td>0.531</td>
<td><strong>0.538</strong></td>
<td>0.537</td>
<td><strong>0.529</strong></td>
<td>0.531</td>
<td><strong>0.510</strong></td>
</tr>
</tbody>
</table>
Single-True-Completion Evaluation Results

<table>
<thead>
<tr>
<th>Correctly predicted entities for $I(C) \neq \emptyset$</th>
<th>entity $\in I(C)$</th>
<th>$&lt;type&gt; \in I(C)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>sitelinks</td>
<td>468</td>
<td>200</td>
</tr>
<tr>
<td>sitelinks + w2v</td>
<td>718</td>
<td>195</td>
</tr>
<tr>
<td>co-occurrence</td>
<td>1062</td>
<td>280</td>
</tr>
</tbody>
</table>

On the >1/CW test set after typing the first letter of the entity label
Future Work

• Experiment with different methods for language modeling
• Make system robust against spelling errors
• Create a larger ground truth for the evaluation
• Enhance completion predictions with contextual information