Auto-completion for Question Answering Systems at Bloomberg

SIGIR Symposium on IR in Practice (SIRIP 2018)
July 9, 2018

Konstantine Arkoudas, Senior Research Scientist
Mohamed Yahya, Research Scientist
Outline

• Background: Question Answering Systems at Bloomberg

• Why Auto-completion for QA?

• Desiderata & Challenges

• Approaches
Outline

• Background: Question Answering Systems at Bloomberg

• Why Auto-completion for QA?

• Desiderata & Challenges

• Approaches
QA at Bloomberg

• Bloomberg Professional Service (The Terminal)
  (325,000+ subscribers, 12,000+ functions)
QA at Bloomberg

Terminal

News  Company filings  Stock markets  Forex  Supply chain info  Geospatial data  Biographies

Query  Plot  Map  Alert  Analyze

Ask for a demo at our booth!

Raw Data:
textual, relational, graph, time-series, …
QA at Bloomberg

• What are the top 10 Asian tech **companies** with EPS of at least 4?

• **Who** are the UMich alumni with a net worth of more than $50M?

• **Plot** Apple’s quarterly profits over the past four years against those of Samsung and Google.

• Find corporate **bonds** rated A or better and with coupon higher than 7%.

• Show me **news** about oil from the Financial Times over the last two months.

Bloomberg has the data and Terminal functions to answer all of the above, but there are some **usability issues**…

**TechAtBloomberg.com**
QA at Bloomberg

• Functions
  — 12,000+ functions, exposed mostly through form-filling interfaces
  — **Function discovery**: Which function answers my information need?
  — **Function usability**: What’s the right input to the function?

• Data
  — Tons of semi-structured & structured data
  — Diversity everywhere: data models, back-ends, ...
  — Relations & joins very (very) important in this setting
QA at Bloomberg

The vision: natural language is the ultimate declarative query language –
• Single entry point to the Terminal (and individual functions)
• Expressive questions

QA systems for several domains are already in place

Our flavor of QA relies on Semantic Parsing

QA interfaces introduce their own set of usability issues;
We tackle these with Question Auto-completion…
Outline

• Background: Question Answering Systems at Bloomberg
• Why Auto-completion for QA?
• Desiderata & Challenges
• Approaches
Outline

• Background: Question Answering Systems at Bloomberg

• Why Auto-completion for QA?

• Desiderata & Challenges

• Approaches
Why Auto-completion for QA?

- **Discovery**: AC as a vehicle to introduce users to QA capabilities

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Completion</th>
<th>Discovery</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Google sha”</td>
<td>“Google shares plotted against Apple shares”</td>
<td>Plotting!</td>
</tr>
<tr>
<td>“Chinese bonds m”</td>
<td>“Chinese bonds maturing in the next 5 years”</td>
<td>Temporal reasoning!</td>
</tr>
<tr>
<td>“Trum”</td>
<td>“Trump election tweets ordered by likes”</td>
<td>Tweet search!</td>
</tr>
<tr>
<td>“P”</td>
<td>“Positive news about Facebook”</td>
<td>News sentiment!</td>
</tr>
</tbody>
</table>

- **Expectation management**: All QA systems have limitations, expose them early on to avoid user frustration

<table>
<thead>
<tr>
<th>Where are my the keys of m...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sorry, no question suggestions!</td>
</tr>
</tbody>
</table>

- **Put users in the loop to help us help them**: Some inherent ambiguity can be resolved by guiding users to the appropriate formulation

<table>
<thead>
<tr>
<th>Assets of Lloyd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assets of Lloyd Bank</td>
</tr>
<tr>
<td>Assets of Lloyd’s of London</td>
</tr>
<tr>
<td>Assets of Lloyd Blankfein</td>
</tr>
</tbody>
</table>

- **Faster typing 😊**
Outline

• Background: Question Answering Systems at Bloomberg

• Why Auto-completion for QA?

• Desiderata & Challenges

• Approaches
Outline

• Background: Question Answering Systems at Bloomberg
• Why Auto-completion for QA?
• Desiderata & Challenges
• Approaches
Desiderata

We want auto-completion systems that are:

- **Sound**
  Completions that we provide should be *understandable* & *answerable*

- **Complete**
  If a prefix can be extended to something we can understand & answer, then we should complete it

- **Diverse**
  Expose multiple capabilities of the underlying QA system

- **Propositional**
  Complete to next *full “subquestion”*

- **Predictive**
  High-ranked completion capture user’s intended information need

<table>
<thead>
<tr>
<th>News QA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show m</td>
</tr>
<tr>
<td>Show me news about Mexican elections</td>
</tr>
<tr>
<td>Show me news from yesterday</td>
</tr>
<tr>
<td>Show me news by the NYT</td>
</tr>
<tr>
<td>Show me news about Angela Merkel</td>
</tr>
<tr>
<td>Show me news about Facebook</td>
</tr>
</tbody>
</table>
Challenges

Our setting is very different than that of AC for more traditional keyword-based search

• AC solves major usability issues for QA → Simultaneous deployment of QA and corresponding AC system

• **Cold-start problem**: No, or very meager, query logs to make use of

• In a semantic setting like ours, especially in the financial sector, infinite sets (e.g., numbers, dates, …) are fundamental, but a major challenge for completeness

• **Grammaticality**: We must complete and produce potentially complex, multi-clause natural language utterances that have to sound, well, natural
But, it’s not all bad

Semantics allows us to do some things much better:

“German non-tech bonds maturing in the next 5 years”

(CNTRY_OF_RISK = GERMANY) AND
(NOT (INDUSTRY_SECTOR = SECTOR_TECHNOLOGY)) AND
(MATURITY_DATE = Interval(Relative_Time_Reference(1,year,now) through
Relative_Time_Reference(5,year,now)))
But, it’s not all bad

Semantics allows us to do some things much better:

“German non-tech bonds maturing in the next 5 years”

(CNTRY_OF_RISK = GERMANY) AND
(NOT (INDUSTRY_SECTOR = SECTOR_TECHNOLOGY)) AND
(MATURITY_DATE = Interval(Relative_Time_Reference(1,year,now) through Relative_Time_Reference(5,year,now)))

- Diversity & Deduping
  Completions can be binned into different classes based on “categories” determined by underlying semantics (which we know)
  \[
  \text{semantics("Alphabet") = semantics("GOOGL")}
  \]

- Metadata
  Utilize whatever metadata other teams collect to inform AC (e.g., entity popularity, better lexicons, …)
Outline

• Background: Question Answering Systems at Bloomberg

• Why Auto-completion for QA?

• Desiderata & Challenges

• Approaches
Outline

• Background: Question Answering Systems at Bloomberg

• Why Auto-completion for QA?

• Desiderata & Challenges

• Approaches
Approaches – Query Logs

• Where query logs exist, this is a good approach

• But, the **world changes**
  
  o AC has to keep up
  
  o Access to semantics very helpful in **log-normalization**

<table>
<thead>
<tr>
<th>Last month</th>
<th>This month</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trump’s supreme court</td>
<td>Trump’s supreme court</td>
</tr>
<tr>
<td>Trump’s supreme court travel ban ruling</td>
<td>Trump’s supreme court nomination news</td>
</tr>
</tbody>
</table>

• **Compositionality** is a major issue!!

Query log

- German bonds maturing in 2023
- Siemens bonds **denominated in Euros**
- Danish Krone bonds

- German bonds d

**Sorry, no question suggestions!**
Approaches – Query Log Atoms

Let’s switch to log atoms, utilizing semantics “*German bonds maturing in 2023*” *(CNTRY_OF_RISK = GERMANY) AND (MATURITY_DATE = Year(2023))*

- Great mileage, but requires tackling coherence and well-formedness issues when stitching atoms together. We tackle this as a ranking problem using statistical techniques

- Fixes the fat head that appears in logs, but there is a long tail that rarely appears in the logs …

Query log
German bonds maturing in 2023
Siemens bonds denominated in Euros
Danish Krone bonds

Query log atoms

<table>
<thead>
<tr>
<th>German bonds</th>
<th>Siemens bonds</th>
<th>Danish Krone bonds</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>German bonds d</td>
<td></td>
<td></td>
</tr>
<tr>
<td>German bonds d</td>
<td>Siemens bonds</td>
<td></td>
</tr>
</tbody>
</table>

Sorry, no question suggestions!
Approaches – Templates

- **Grammars** are used for semantic parsing (question understanding). Why not use them for **generation** as well?
- Based on **recursive, reusable components**
- This way we achieve **completeness**
- Again, **coherence & well-formedness** are major issues here
- AC has to happen, end-to-end in **<50ms**

---

**Guatemalan bonds that will mature within 3 years**

**Asian issued 2 years ago**
Conclusion

• QA is an important step in enhancing usability & discovery of information systems
  AC is an important step in enhancing usability & discovery of QA systems!

• AC for QA facilitates discovery, expectation management, and ambiguity resolution

• Simultaneous deployment of QA & AC → cold-start problem

• We’ve shown some of our complementary approaches to AC, with semantics playing a crucial role!

• Still, much more left to do: We’re hiring!
  Chat with us, visit our booth (and get a demo)!
  Get in touch: careers.bloomberg.com
  Learn more: www.TechAtBloomberg.com

TechAtBloomberg.com

© 2018 Bloomberg Finance L.P. All rights reserved.