Question Answering

CS293S, 2017. Tao Yang

(some of these slides were adapted from Giuseppe Attardi, Girish K)
Table of Content

• Question answering in search engines
• Natural language processing techniques for QA
  – Word embedding
Question Answering

• Earlier IR systems focus on queries with short keywords
  – Most of search engine queries are short queries.

• QA systems focus in natural language question answering.

• Outline
  – What is QA
  – Examples of QA systems/algorithms.
People *want* to ask questions...

Examples from Ask.com query log
how much should i weigh
what does my name mean
how to get pregnant
where can i find pictures of hairstyles
who is the richest man in the world
what is the meaning of life
why is the sky blue
what is the difference between white eggs and brown eggs
can you drink milk after the expiration date
what is true love
what is the jonas brothers address

Around 10-20% of query logs
General Search Engine

- Include question words etc. in stop-list with standard IR
- Sometime it works. Sometime it requires users to do more investigation (a study in 2008)
  - Question: Who was the prime minister of Australia during the Great Depression?
    - Ask.com gives an explicit answer.
    - Google’s top 1-2 results are also good.

- what is phone number for united airlines
  - Ask.com gives a direct answer
  - Google gives no direct answers in top 10.

- How much money did IBM spend on advertising in 2006?
  - No engine can answer
Why QA?

• QA engines attempt to let you ask your question the way you'd normally ask it.
  – More specific than short keyword queries
    • Orange chicken
    • what is orange chicken
    • how to make orange chicken
  – Inexperienced search users

• What is involved in QA?
  – Natural Language Processing
    • Question type analysis and answer patterns
    • Semantic Processing
    • Syntactic Processing and Parsing
  – Knowledge Base to store candidate answers
  – Candidate answer search and answer processing
AskJeeves (now Ask.com)

- Earlier AskJeeves is probably most well-known QA site
  - It largely does pattern matching to match your question to their own knowledge base of questions
  - Has own knowledge base and uses partners to answer questions
  - Catalogues previous questions
  - Answer processing engine
    - Question template response
    - If that works, you get template-driven answers to that known question
    - If that fails, it falls back to regular web search

- Ask.com:
  - Search answers from a large web database
  - Deep integration with structured answers
Question Answering at TREC

• Question answering competition at TREC consists of answering a set of 500 fact-based questions, e.g., “When was Mozart born?”.

• For the first three years systems were allowed to return 5 ranked answer snippets (50/250 bytes) to each question.
  – IR think
  – Mean Reciprocal Rank (MRR) scoring:
    • 1, 0.5, 0.33, 0.25, 0.2, 0 for 1, 2, 3, 4, 5, 6+ doc
  – Mainly Named Entity answers (person, place, date, ...)

• From 2002 the systems were only allowed to return a single *exact* answer and the notion of confidence has been introduced.
The TREC Document Collection

- The current collection uses news articles from the following sources:
  - AP newswire,
  - New York Times newswire,
  - Xinhua News Agency newswire,

- In total there are 1,033,461 documents in the collection. 3GB of text

- Clearly this is too much text to process entirely using advanced NLP techniques so the systems usually consist of an initial information retrieval phase followed by more advanced processing.

- Many supplement this text with use of the web, and other knowledge bases
Sample TREC questions

1. Who is the author of the book, "The Iron Lady: A Biography of Margaret Thatcher"?
2. What was the monetary value of the Nobel Peace Prize in 1989?
3. What does the Peugeot company manufacture?
4. How much did Mercury spend on advertising in 1993?
5. What is the name of the managing director of Apricot Computer?
6. Why did David Koresh ask the FBI for a word processor?
7. What debts did Qintex group leave?
8. What is the name of the rare neurological disease with symptoms such as: involuntary movements (tics), swearing, and incoherent vocalizations (grunts, shouts, etc.?)?
Web Question Answering: Is More Always Better?
– Dumais, Banko, Brill, Lin, Ng, SIGIR 2002

Q: “Where is the Louvre located?”

Want “Paris” or “France” or “75058 Paris Cedex 01” or a map

Don’t just want URLs
AskMSR: Shallow approach

- In what year did Abraham Lincoln die?
- Ignore hard documents and find easy ones

**Abraham Lincoln, 1809-1865**

*LINCOLN, ABRAHAM* was born near Hodgenville, Kentucky, on February 12, 1809. In 1816, the Lincoln family moved to Pigeon Creek in Perry (now Spencer) County. Two years later, Abraham Lincoln's mother died and his father married a woman that was his "angel" mother. Lincoln attended a formal school for only a few months but acquired knowledge through the reading of books. In 1830, he lost his attempts at the state legislature, but two years later he tried again, was successful, and Lincoln was admitted to the bar and became noteworthy as a witty, honest, competent circuit attorney, at which time he opposed the war with Mexico. By 1855, he was the leading figure in his party. By 1856, when the election he became a significant figure in his party. In his inauguration on March 4, seven southern states had seceded, for a total of 11. Lincoln immediately took action. The situation would eventually be the central difference in maintaining the Union. Lincoln's Emancipation Proclamation which expanded the purpose of the war beyond preserving the Union by freeing slaves and leading to the end of slavery in the Kingdom.

**Sixteenth President of the United States**

Born in 1809 - Died in 1865

**Abraham Lincoln**

16th President of the United States (March 4, 1861 to April 15, 1865)
Born: February 12, 1809, in Hardin County, Kentucky
Died: April 15, 1865, at Petersen's Boarding House in Washington, D.C.

"I was born February 12, 1809, in Hardin County, Kentucky. My parents were both born in Virginia, of undistinguished families, perhaps I should say. My mother, who died in my tenth year, was of a family of the name of Hanks."
AskMSR: Details

Where is the Louvre Museum located?

in Paris France 59%
museums 12%
hostels 10%

N-Best Answers

1. Rewrite Query
2. <Search Engine>
3. Collect Summaries, Mine N-grams
4. Filter N-Grams
5. Tile N-Grams
Step 1: Rewrite queries

• Intuition: The user’s question is often syntactically quite close to sentences that contain the answer
  – Where is the Louvre Museum located?
  – The Louvre Museum is located in Paris
  – Who created the character of Scrooge?
  – Charles Dickens created the character of Scrooge.
Query rewriting

- Classify question into seven categories
  - **Who** is/was/are/were...?
  - **When** is/did/will/are/were ...?
  - **Where** is/are/were ...?

  a. Category-specific transformation rules

  eg “For Where questions, move ‘is’ to all possible locations”
  “Where **is** the Louvre Museum located”
  → “**is** the Louvre Museum located”
  → “the **is** Louvre Museum located”
  → “the Louvre **is** Museum located”
  → “the Louvre Museum **is** located”
  → “the Louvre Museum located **is**”

  b. Expected answer “Datatype” (eg, Date, Person, Location, ...)

  **When** was the French Revolution? → DATE

- Hand-crafted classification/rewrite/datatype rules
  (Could they be automatically learned?)
Query Rewriting - weights

- One wrinkle: Some query rewrites are more reliable than others

**Where is the Louvre Museum located?**

- **Weight 1**
  Lots of non-answers could come back too

- **Weight 5**
  if we get a match, it’s probably right

  + “the Louvre Museum is located”

  + Louvre + Museum + located
Step 2 and Step 3

• Step 2: Query Search engine
  – Send all rewrites to a Web search engine
  – Retrieve top N answers (100?)
  – For speed, rely just on search engine’s “snippets”, not the full text of the actual document

• Step 3: Mining N-grams
  – Unigram, bigram, trigram, ... N-gram: list of N adjacent terms in a sequence
    • Unigrams: Web, Question, Answering, Is, More, Always, Better
    • Bigrams: Web Question, Question Answering, Answering Is, Is More, More Always, Always Better
    • Trigrams: Web Question Answering, Question Answering Is, Answering Is More, Is More Always, More Always Betters
Mining N-Grams

• Simple: Enumerate all N-grams (N=1,2,3 say) in all retrieved snippets
  • Use hash table and other fancy footwork to make this efficient

• Weight of an n-gram: occurrence count, each weighted by “reliability” (weight) of rewrite that fetched the document

• Example: “Who created the character of Scrooge?”
  – Dickens - 117
  – Christmas Carol - 78
  – Charles Dickens - 75
  – Disney - 72
  – Carl Banks - 54
  – A Christmas - 41
  – Christmas Carol - 45
  – Uncle - 31
Step 4: Filtering N-Grams

- Each question type is associated with one or more “data-type filters” = regular expression
  - When...
  - Where...
  - What ...
  - Who ...

- Boost score of n-grams that do match regexp
- Lower score of n-grams that don’t match regexp
Step 5: Tiling the Answers

Scores

<table>
<thead>
<tr>
<th>Score</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>Charles Dickens</td>
</tr>
<tr>
<td>15</td>
<td>Dickens</td>
</tr>
<tr>
<td>10</td>
<td>Mr Charles</td>
</tr>
</tbody>
</table>

Score 45

merged, discard old n-grams

N-Grams

tile highest-scoring n-gram

Repeat, until no more overlap

N-Grams
Results

- Standard TREC contest test-bed: 
  ~1M documents; 900 questions

- Technique doesn’t do too well (though would have placed in top 9 of ~30 participants!)
  - MRR = 0.262 (ie, right answered ranked about #4-#5)

- Using the Web as a whole, not just TREC’s 1M documents... MRR = 0.42 (ie, on average, right answer is ranked about #2-#3)
  - Why? Because it relies on the enormity of the Web!
NLP with Word Embedding Techniques

- Tutorial and Visualization tool by Xin Rong ([http://www-personal.umich.edu/~ronxin/pdf/w2vexp.pdf](http://www-personal.umich.edu/~ronxin/pdf/w2vexp.pdf))
## Word Representations

<table>
<thead>
<tr>
<th>Traditional Method - Bag of Words Model</th>
<th>Word Embeddings</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Uses one hot encoding</td>
<td>• Stores each word in as a point in space, where it is represented by a vector of fixed number of dimensions (generally 300)</td>
</tr>
<tr>
<td>• Each word in the vocabulary is represented by one bit position in a HUGE vector.</td>
<td>• Unsupervised, built just by reading huge corpus</td>
</tr>
<tr>
<td>• For example, if we have a vocabulary of 10000 words, and “Hello” is the 4th word in the dictionary, it would be represented by: 0 0 0 1 0 0 . . . . . . . 0 0 0 0</td>
<td>• For example, “Hello” might be represented as: [0.4, -0.11, 0.55, 0.3 . . . 0.1, 0.02]</td>
</tr>
<tr>
<td>• Context information is not utilized</td>
<td>• Dimensions are basically projections along different axes, more of a mathematical concept.</td>
</tr>
</tbody>
</table>
Examples

\[
\text{vector}[\text{Queen}] = \text{vector}[\text{King}] - \text{vector}[\text{Man}] + \text{vector}[\text{Woman}]
\]
The Power of Word Vectors

• They provide a fresh perspective to *ALL* problems in NLP, and not just solve one problem.

• Technological Improvement
  ▪ Rise of deep learning since 2006 (Big Data + GPUs + Work done by Andrew Ng, Yoshua Bengio, Yann Lecun and Geoff Hinton)
  ▪ Application of Deep Learning to NLP – led by Yoshua Bengio, Christopher Manning, Richard Socher, Tomas Mikalov

• The need for unsupervised learning. (Supervised learning tends to be excessively dependant on hand-labelled data and often does not scale)
Applications of Word Vectors

1. **Word Similarity**

Classic Methods: Edit Distance, WordNet, Porter’s Stemmer, Lemmatization using dictionaries

- Easily identifies similar words and synonyms since they occur in similar contexts
- Stemming (thought -> think)
- Inflections, Tense forms
- *eg. Think, thought, ponder, pondering*,
- *eg. Plane, Aircraft, Flight*
Applications of Word Vectors

2. Machine Translation

Classic Methods: Rule-based machine translation, morphological transformation
Applications of Word Vectors

3. Part-of-Speech and Named Entity Recognition

Classic Methods: Sequential Models (MEMM, Conditional Random Fields), Logistic Regression

<table>
<thead>
<tr>
<th></th>
<th>POS WSJ (acc.)</th>
<th>NER CoNLL (F1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>State-of-the-art*</td>
<td>97.24</td>
<td>89.31</td>
</tr>
<tr>
<td>Supervised NN</td>
<td>96.37</td>
<td>81.47</td>
</tr>
<tr>
<td>Unsupervised pre-training followed by supervised NN**</td>
<td>97.20</td>
<td>88.87</td>
</tr>
<tr>
<td>+ hand-crafted features***</td>
<td>97.29</td>
<td>89.59</td>
</tr>
</tbody>
</table>

Noun: book/books, nature, eat, wrote
Verb: can, should, have
Adjective: new, newer, newest
Adverb: well, urgently
Numbers: 872, two, first
Article/Determiner: the, some
Conjunction: and, or
Pronoun: he, my
Preposition: to, in
Particle: off, up
Interjection: Ow, Eh
Applications of Word Vectors

4. Relation Extraction

Classic Methods: OpenIE, Linear Programming models, Bootstrapping

<table>
<thead>
<tr>
<th>Relationship</th>
<th>Example 1</th>
<th>Example 2</th>
<th>Example 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>France - Paris</td>
<td>Italy: Rome</td>
<td>Japan: Tokyo</td>
<td>Florida: Tallahassee</td>
</tr>
<tr>
<td>big - bigger</td>
<td>small: larger</td>
<td>cold: colder</td>
<td>quick: quicker</td>
</tr>
<tr>
<td>Miami - Florida</td>
<td>Baltimore: Maryland</td>
<td>Dallas: Texas</td>
<td>Kona: Hawaii</td>
</tr>
<tr>
<td>Einstein - scientist</td>
<td>Messi: midfielder</td>
<td>Mozart: violinist</td>
<td>Picasso: painter</td>
</tr>
<tr>
<td>Sarkozy - France</td>
<td>Berlusconi: Italy</td>
<td>Merkel: Germany</td>
<td>Koizumi: Japan</td>
</tr>
<tr>
<td>copper - Cu</td>
<td>zinc: Zn</td>
<td>gold: Au</td>
<td>uranium: plutonium</td>
</tr>
<tr>
<td>Berlusconi - Silvio</td>
<td>Sarkozy: Nicolas</td>
<td>Putin: Medvedev</td>
<td>Obama: Barack</td>
</tr>
<tr>
<td>Microsoft - Windows</td>
<td>Google: Android</td>
<td>IBM: Linux</td>
<td>Apple: iPhone</td>
</tr>
<tr>
<td>Microsoft - Ballmer</td>
<td>Google: Yahoo</td>
<td>IBM: McNealy</td>
<td>Apple: Jobs</td>
</tr>
<tr>
<td>Japan - sushi</td>
<td>Germany: bratwurst</td>
<td>France: tapas</td>
<td>USA: pizza</td>
</tr>
</tbody>
</table>
Applications of Word Vectors

5. Sentiment Analysis

Classic Methods: Naive Bayes, Random Forests/SVM

- Classifying sentences as positive and negative
- Building sentiment lexicons using seed sentiment sets
- No need for classifiers, we can just use cosine distances to compare unseen reviews to known reviews.
Applications of Word Vectors

6. Clustering

- Words in the same class naturally occur in similar contexts, and this feature vector can directly be used with any conventional clustering algorithms (K-Means, agglomerative, etc). Human doesn’t have to waste time hand-picking useful word features to cluster on.

7. Question answering.