Question Answering and Dialog Systems

Natural Language Processing: Lecture 15

22.12.2017

Kairit Sirts
Factoid questions

• Who founded Virgin Airlines?
• What is an average age of the onset of autism?
• Where is Apple based?
Factoid questions

• Who founded Virgin Airlines?
  • Richard Branson

• What is an average age of the onset of autism?
  • In the United States, the average age of diagnosis with an autism spectrum disorder (ASD) is around 4 years of age. For many years, doctors and scientists alike thought the disorder couldn't be identified much earlier than age 3

• Where is Apple based?
  • Apple Campus, 1 Infinite Loop, Cupertino, California, U.S.
Task-oriented dialog agents

- Designed for a particular task
- Short dialogues to get information from the user to complete the task
- Digital assistants: Siri, Cortana, Alexa, etc
Siri 2014

(a)

(b)

I'm sorry, Dan, I'm afraid I can't do that.
Siri 2017
Chatbots

• Systems designed for extended dialogues
• Attempt to mimic the unstructured conversations or chats
• www.cleverbot.com
User: Do you like Doctor Who?
Cleverbot: Yes, so funny.
User: What’s your favorite show?
Cleverbot: I don’t have one.
User: What’s your name?
Cleverbot: Jason I already told you.
User: No your real name.
Cleverbot: Stephanie IS my real name.
User: I thought it was Jason?
Cleverbot: How did you come to that conclusion.
User: Because you said so.
Cleverbot: I have to do my homework.
Plan

• Factoid QA systems
• Task-oriented dialog agents
• Chatbots
Factoid QA methods
Factoid QA methods

• IR-based question answering or text-based question answering
  • Relies on huge amounts of texts
  • Given a question, IR techniques extract passages directly from these documents

• Knowledge-based question answering
  • Build a semantic representation of the query
  • Use this representation to query databases of facts

• Hybrid systems combining IR and knowledge-based QA
IR-based Factoid Question Answering

- Answer a user’s question by finding short text segments from a document collection.

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Where is the Louvre Museum located?</td>
<td>in Paris, France</td>
</tr>
<tr>
<td>What’s the abbreviation for limited partnership?</td>
<td>L.P.</td>
</tr>
<tr>
<td>What are the names of Odin’s ravens?</td>
<td>Huginn and Muninn</td>
</tr>
<tr>
<td>What currency is used in China?</td>
<td>the yuan</td>
</tr>
<tr>
<td>What kind of nuts are used in marzipan?</td>
<td>almonds</td>
</tr>
<tr>
<td>What instrument does Max Roach play?</td>
<td>drums</td>
</tr>
<tr>
<td>What’s the official language of Algeria?</td>
<td>Arabic</td>
</tr>
<tr>
<td>How many pounds are there in a stone?</td>
<td>14</td>
</tr>
</tbody>
</table>
The process of IR-based factoid QA

1. Question processing
2. Passage retrieval and ranking
3. Answer processing
Question processing

The goal is to extract a number of information pieces from the question

• **Answer type** - the kind of entity the answer consists of:
  • Person, location, time, etc

• **Query** – keywords that should be used in the IR system to search for relevant documents

• **Focus** – the sequence of words in the question that will be likely replaced by the answer

• **Question type** – what kind of question it is?
  • Definition, math, list
Question processing

Which US state capital has the largest population?

Answer type: CITY
Query: US state capital, largest, population
Focus: state capital
Answer type detection

- Detect the kind of entity the answer consists of: person, location, time, etc
Answer type taxonomy

A subset from Li and Roth (2005)
Answer type detection

• Detect the kind of entity the answer consists of: person, location, time, etc

• Rule-based system using regular expressions
  • who {is | was | are | were} PERSON --> BIOGRAPHY

• Supervised classifiers using features like:
  • Words, POS, named entities
  • WordNet synset ID
  • Question headword – head of the first NP after the question’s wh-word
What is the national flower of Estonia?
Query formulation

- The question itself: What is the national flower of Estonia
- Leave out the question word: Is the national flower of Estonia
- Use only the content of the noun phrases: National flower of Estonia
- Remove stop-words and other high-frequency words: national flower Estonia
- Query reformulation: The national flower of Estonia is
Passage retrieval

1. Segment the retrieved documents into passages
2. Answer type classification on passages
3. Filter out passages with mismatching type
4. Rank the passages using supervised machine learning:
   - The number of named entities
   - The number of question keywords
   - The longest exact sequence of question keywords occurring in the passage
   - The rank of the document
   - The proximity of the keywords in the passage
   - The ngram overlap
Answer processing

• Answer-type pattern extraction
  • Uses information about the expected answer type together with regular expressions

• N-gram tiling
  • Use n-grams from the retrieved snippets to construct the answer
Answer-type pattern extraction

• Extract potential answers using answer type info and named entity information

• Question of type HUMAN

  “Who is the prime minister of India”
  Manmohan Singh, Prime Minister of India, had told left leaders that the deal would not be renegotiated.

• Question of type DISTANCE-QUANTITY

  “How tall is Mt. Everest?”
  The official height of Mount Everest is 29029 feet
Answer-type pattern extraction

- The answers to some questions, such as DEFINITION, do not tend to be of particular named entity type.
- Hand-written or automatically learned patterns can be used to extract these answers.

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;AP&gt; such as &lt;QP&gt;</code></td>
<td>What is autism?</td>
<td>“, developmental disorders such as autism”</td>
</tr>
<tr>
<td><code>&lt;QP&gt;, a &lt;AP&gt;</code></td>
<td>What is a caldera?</td>
<td>“the Long Valley caldera, a volcanic crater 19 miles long”</td>
</tr>
</tbody>
</table>
N-gram tiling

- Uses the snippets returned by the web search engine

1. N-gram mining
   - Extract unigrams, bigrams and trigrams from the snippets

2. N-gram filtering
   - Score n-grams by how well they match the predicted answer type

3. N-gram tiling
   - Concatenate overlapping n-gram fragments into longer answers
Answer ranking

• Rank the extracted answers

• The features commonly used include features such as:
  • Answer type match
  • Pattern match
  • number of matched question keywords
  • Keyword distance
  • Novelty factor
  • Apposition features
  • Punctuation location
  • Sequences of question terms
Knowledge-based QA
Knowledge-based QA

- Use a structured database to find the answer to a question
- The question in natural language must be mapped to a query to search from that database
- The query is typically some kind of logical form
- **Semantic parsers** are used to map questions to logical forms

<table>
<thead>
<tr>
<th>Question</th>
<th>Logical form</th>
</tr>
</thead>
<tbody>
<tr>
<td>When was Ada Lovelace born?</td>
<td>birth-year (Ada Lovelace, ?x)</td>
</tr>
<tr>
<td>What states border Texas?</td>
<td>( \lambda \text{x.state(x)} \land \text{borders(x, texas)} )</td>
</tr>
<tr>
<td>What is the largest state</td>
<td>argmax(( \lambda x.\text{state}(x) ), ( \lambda x.\text{size}(x) ))</td>
</tr>
<tr>
<td>How many people survived the sinking of the Titanic</td>
<td>(count (!( \text{fb:event.disaster.survivors} \text{fb:en.sinking.of.the.titanic} )))</td>
</tr>
</tbody>
</table>
RDF database and queries

- Databases like Freebase and DBpedia contain large number of factual triples derived from Wikipedia infoboxes

```
subject       predicate       object
Ada Lovelace  birth-year     1815
```

“When was Ada Lovelace born?” → birth-year (Ada Lovelace, ?x)
“What is the capital of England?” → capital-city(?x, England)
Methods for mapping questions to queries

- Rule-based methods
- Supervised methods
- Semi-supervised methods
Rule-based mapping

- Useful for simple and frequent relations, such as *birth-year*

```
when * PERSON born?

birth-year(PERSON, ?x)
```
Supervised mapping

• Trained on a set of question-logical form pairs
  • Questions are first parsed
  • Parse trees are aligned to logical forms

• Might use a small set of seed rules
• Use these rules to bootstrap more specific alignment rules
Supervised mapping - bootstrapping

• Might use a small set of simple rules such as:

\[
\text{Who } V \text{ ENTITY } \rightarrow \text{relation}(\ ?x, \text{ entity})
\]

\[
\text{When } V \text{ ENTITY } \rightarrow \text{relation}(\ ?x, \text{ entity})
\]
Learning specific rules

• Given training examples such as:
  “When was Ada Lovelace born?”  \(\rightarrow\)  birth-year (Ada Lovelace, ?x)

• Parse the questions and form the mappings:

```
<table>
<thead>
<tr>
<th>tmmod</th>
<th>nsubj</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>When was Ada Lovelace born</td>
<td>birth-year(Ada Lovelace, ?x)</td>
</tr>
</tbody>
</table>
```

• From many examples like this, infer the specific rule:

```
<table>
<thead>
<tr>
<th>tmmod</th>
<th>nsubj</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>When was born</td>
<td>birth-year( , ?x)</td>
</tr>
</tbody>
</table>
```
Semi-supervised mapping

- It is difficult to create supervised training sets that cover enough variations of how questions can be expressed
  - One relation can be expressed using many different phrases
- This problem can be alleviated by using unlabeled web sources
- Recall Open Information Extraction:
  - Extract (subject, relation, object) triples of strings from web
- Align extracted triples with a canonical knowledge source, such as Wikipedia
Alignment with Freebase
Berant et al., 2013. Semantic Parsing on Freebase from Question-Answer Pairs

• A set of phrases that align with Freebase relation **country.capital**

<table>
<thead>
<tr>
<th>capital of</th>
<th>capital city of</th>
<th>become capital of</th>
</tr>
</thead>
<tbody>
<tr>
<td>capitol of</td>
<td>national capital of</td>
<td>official capital of</td>
</tr>
<tr>
<td>political capital of</td>
<td>administrative capital of</td>
<td>beautiful capital of</td>
</tr>
<tr>
<td>capitol city of</td>
<td>remain capital of</td>
<td>make capital of</td>
</tr>
<tr>
<td>political center of</td>
<td>bustling capital of</td>
<td>capital city in</td>
</tr>
<tr>
<td>cosmopolitan capital of</td>
<td>move its capital to</td>
<td>modern capital of</td>
</tr>
<tr>
<td>federal capital of</td>
<td>beautiful capital city of</td>
<td>administrative capital city of</td>
</tr>
</tbody>
</table>
Alignment with paraphrase databases
Fader et al., 2013. Paraphrase-Driven Learning for Open Question Answering

• PARALEX corpus (collected from wikianswers.com)

Q: What are the green blobs in plant cells?

Lemmatized synonyms from PARALEX:
what be the green blob in plant cell?
what be green part in plant cell?
what be the green part of a plant cell?
what be the green substance in plant cell?
what be the part of plant cell that give it green color?
what cell part do plant have that enable the plant to be give a green color?
what part of the plant cell turn it green?
part of the plant cell where the cell get it green color?
the green part in a plant be call?
the part of the plant cell that make the plant green be call?
Hybrid systems
DeepQA - the QA component of IBM Watson
• The main idea is to propose a large number of answer candidates extracted from different sources and rank them using various pieces of evidence

Main stages:
1. Question processing
2. Candidate answer generation
3. Candidate answer scoring
4. Confidence merging and ranking
(1) Question processing

- Run parsing, NER and relation extraction
- Extract **focus** and **answer type**
- Perform **question classification**
Poets and Poetry: **He** was a bank clerk in the Yukon before he published “Songs of a Sourdough” in 1907.

- **NER**
  - Yukon --> GEOPOLITICAL ENTITY
  - “Songs of a Sourdough” --> COMPOSITION

- **Coreference:**
  - He --> clerk

- **Relation extraction**
  - authorof(focus, “Songs of a Sourdough”)
  - publish(e1, he, “Songs of a Sourdough”)
  - in(e2, e1, 1907)
  - temporallink(publish(...), 1907)
Poets and Poetry: He was a bank clerk in the Yukon before he published “Songs of a Sourdough” in 1907.

• Extract focus with hand-written rules
Poets and Poetry: He was a bank clerk in the Yukon before he published “Songs of a Sourdough” in 1907.

• Extract focus with hand-written rules: He
Poets and Poetry: **He** was a bank clerk in the Yukon before he published “Songs of a Sourdough” in 1907.

- Extract **focus** with hand-written rules: **He**
- Extract **answer type** with hand-written rules
  - The number of different answer types is ca 5000
  - The default rule is to choose the syntactic headword of the focus
  - Extract addition answer types using for instance coreferents or even the question category
Poets and Poetry: He was a bank clerk in the Yukon before he published “Songs of a Sourdough” in 1907.

• Extract focus with hand-written rules: He
• Extract answer type with hand-written rules: He, poet, clerk
  • The number of different answer types is ca 5000
  • The default rule is to choose the syntactic headword of the focus
  • Extract addition answer types using for instance coreferents or even the question category

• Question classification:
  • Definition question
  • Multiple choice
  • Puzzle
  • Fill-in-the-blank
(2) Candidate answer generation

• Generate many candidate answers
• Knowledge bases (DBPedia, IMDB, triples from OIE)
  
  authorof(focus, "Songs of a sourdough")
  
  authorof(?x, "Songs of a sourdough")
• IR from external documents
  • Retrieve short passages and extract answer candidates (anchor texts, noun phrases)
  • Query Wikipedia pages and return the titles
(3) Candidate answer scoring

- Uses many sources of evidence
  - Lexical answer type

**Answer candidate**
Difficulty swallowing

**Lexical answer type**
Manifestation

DBPedia: Dysphagia
WordNet: Symptom
Condition

Hyponymy?
Instance-of?
Synonymy?
(3) Candidate answer scoring

- Uses many sources of evidence
  - Lexical answer type
  - Time and space relations
  - Retrieve texts with supporting evidence
(4) Answer merging and scoring

- Merge equivalent candidate answers
  - Automatically generated name dictionaries
  - Morphological parsing for common nouns
- Merge the scoring evidence of each merged answer
- Score the answers with a regularized logistic regression classifier
- The merging and ranking can be done iteratively
Evaluating factoid answers

• Mean reciprocal rank – MRR
• Assumes that the systems returns a ranked list of answers
• Each question is scored by its reciprocal of the rank of the first correct answer
• The score of a system is the mean scores of all test questions
MRR - example

<table>
<thead>
<tr>
<th>Question</th>
<th>Correct answer</th>
<th>Rank of the correct answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>How many people survived the sinking of the Titanic?</td>
<td>706</td>
<td>2</td>
</tr>
<tr>
<td>What is the average temperature in Sydney in August?</td>
<td>14°C</td>
<td>Not among the top 5</td>
</tr>
<tr>
<td>When did Mount Fuji last erupt?</td>
<td>16.12.1707</td>
<td>1</td>
</tr>
</tbody>
</table>

MRR =
# MRR - example

<table>
<thead>
<tr>
<th>Question</th>
<th>Correct answer</th>
<th>Rank of the correct answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>How many people survived the sinking of the Titanic?</td>
<td>706</td>
<td>2</td>
</tr>
<tr>
<td>What is the average temperature in Sydney in August?</td>
<td>14°</td>
<td>Not among the top 5</td>
</tr>
<tr>
<td>When did Mount Fuji last erupt?</td>
<td>16.12.1707</td>
<td>1</td>
</tr>
</tbody>
</table>

\[
\text{MRR} = \frac{1/2 + 0 + 1/1}{3} = 0.5
\]
Frame based dialogue systems
GUS system for travel planning

(Bobrow et al., 1977)

GUS: Hello. My name is GUS. I can help you plan a simple trip by air. Where do you want to go?
Client: I want to go to San Diego on May 28
GUS: What time do you want to leave?
Client: I must be in San Diego before 10 am
GUS: Would you like P.S.A. flight 102 that arrives at 9.15 am?
Client: What is the next flight?
GUS: Air California flight 310 that leaves at 8.30 am and arrives at 10.00 am
Client: I’ll take the first one
GUS: In what name should I make a reservation, please?
Client: Dan Bobrow
GUS: I have confirmed the following flight:
Client: P.S.A. flight 102 on Wednesday May 28 from San Jose to San Diego that leaves at 7.30 am and arrives at 9.15 am.
GUS: What date do you want to return on?
Client: On Friday in the evening.
GUS: Would you like the flight that leaves at 7.45 pm?
Client: That’s fine.
GUS: I have confirmed the following flight: P.S.A. flight 307 on Friday May 30 from San Diego to San Jose that leaves at 7.45 pm and arrives at 9.30 pm.
Thank you for calling. Goodbye
Frame based dialog systems

• Based on a **domain ontology** - a data structure that represents possible intentions extracted from the user’s sentences
• Ontology defines a set of frames
• Each frame consists of a set of slots
• Each frame defines a set of values that each slot can take

<table>
<thead>
<tr>
<th>Slot</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORIGIN CITY</td>
<td>city</td>
</tr>
<tr>
<td>DESTINATION CITY</td>
<td>city</td>
</tr>
<tr>
<td>DEPARTURE TIME</td>
<td>time</td>
</tr>
<tr>
<td>DEPARTURE DATE</td>
<td>date</td>
</tr>
<tr>
<td>ARRIVAL TIME</td>
<td>time</td>
</tr>
<tr>
<td>ARRIVAL DATE</td>
<td>date</td>
</tr>
</tbody>
</table>
Control structure for a frame-based dialog

- What city are you leaving from?
  - Where are you going?
  - What date do you want to leave?
    - Is it a one-way trip?
      - Yes
        - Do you want to go from <FROM> to <TO> on <DATE>?
          - No
            - Book the flight
          - Yes
            - What date do you want to return?
              - Do you want to go from <FROM> to <TO> on <DATE> returning on <RETURN>?
                - Yes
                  - Book the flight
                - No
Initiative of the dialog

• In human conversations initiative shifts forth and back
• Systems that fully control the initiative are called system-initiative
  • They typically also allow universal commands such as help and start over
• Systems that allow more flexibility are called mixed-initiative
  • The system must be able to understand switches between frames
  • The system must be able to understand which slots of which frames can be filled from the user’s utterance
Extracting information from user’s input

1. Domain classification
2. Intent determination
3. Slot filling

Show me morning flights from Boston to San Francisco on Tuesday

<table>
<thead>
<tr>
<th>DOMAIN:</th>
<th>AIR-TRAVEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTENT:</td>
<td>SHOW-FLIGHTS</td>
</tr>
<tr>
<td>ORIGIN-CITY:</td>
<td>Boston</td>
</tr>
<tr>
<td>ORIGIN-DATE:</td>
<td>Tuesday</td>
</tr>
<tr>
<td>ORIGIN-TIME:</td>
<td>morning</td>
</tr>
<tr>
<td>DEST-CITY:</td>
<td>San Francisco</td>
</tr>
</tbody>
</table>
Slot filling

• Hand-written rules

Wake me tomorrow at 6

wake me (up) | set (the|an) alarm | get me up

INTENT: SET-ALARM
Slot filling

- Hand-written rules
- Semantic grammars

```
SHOW       →  show me | i want | can i see | ...  
DEPART_TIME_RANGE →  (after | around | before) HOUR | morning | afternoon | evening  
HOUR       →  one | two | three | four... | twelve (AMPM)  
FLIGHTS    →  (a) flight | flights  
AMPM       →  am | pm  
ORIGIN     →  from CITY  
DESTINATION →  to CITY  
CITY       →  Boston | San Francisco | Denver | Washington  
```
Slot filling

- Hand-written rules
- Semantic grammars
Slot filling

• Hand-written rules
• Semantic grammars
• Supervised classifiers
  • IOB taggers
  • LSTM-IOB taggers
## Evaluation

- Slot error rate

### Slot Error Rate for a Sentence

\[
\text{Slot Error Rate for a Sentence} = \frac{\text{# of inserted/deleted/substituted slots}}{\text{# of total reference slots for sentence}}
\]

Make an appointment with Chris at 10:30 in Gates 104

<table>
<thead>
<tr>
<th>Slot</th>
<th>Filler</th>
</tr>
</thead>
<tbody>
<tr>
<td>PERSON</td>
<td>Chris</td>
</tr>
<tr>
<td>TIME</td>
<td>11:30 a.m.</td>
</tr>
<tr>
<td>ROOM</td>
<td>Gates 104</td>
</tr>
</tbody>
</table>

SER = 1/3
Evaluation

• Slot error rate
• Slot precision/recall/F-score
• Task error rate
• Human satisfaction
Chatbots
Chatbot architectures

- Rule-based
- IR-based
- Neural sequence-to-sequence models
Rule-based chatbots

- ELIZA (Weizenbaum, 1966) – designed to simulate a Rogerian psychologist

Men are all alike.
IN WHAT WAY
They’re always bugging us about something or other.
CAN YOU THINK OF A SPECIFIC EXAMPLE
Well my boyfriend made me come here.
YOUR BOYFRIEND MADE YOU COME HERE
He says I’m depressed much of the time.
I AM SORRY TO HEAR YOU ARE DEPRESSED
WHO ELSE IN YOUR FAMILY TAKES CARE OF YOU?
My father
YOUR FATHER
You are like my father in some ways.
WHAT RESEMBLANCE DO YOU SEE
You are not very aggressive but I think you don’t want me to notice that.
WHAT MAKES YOU THINK I AM NOT AGGRESSIVE
You don’t argue with me.
WHY DO YOU THINK I DON’T ARGUE WITH YOU
You are afraid of me.
DOES IT PLEASE YOU TO BELIEVE I’M AFRAID OF YOU
My father is afraid of everybody.
WHAT ELSE COMES TO MIND WHEN YOU THINK OF YOUR FATHER
Bullies.
DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR BOYFRIEND MADE YOU COME HERE
Eliza algorithm

• Pattern/transform rules

(*) you * me) --> (What makes you think I 3 you)
You hate me --> What makes you think I hate you

(l *) --> (You say you 2)
I know everybody laughed at me -->
You say you know everybody laughed at you
Eliza algorithm – keyword matching

• Keyword ranking
  • Everybody: 5
  • I: 0
• (I *) --> (You say you 2)
• (* everybody *) --> (Who in particular are you thinking of?)

• If no keyword matches
  • “Please go on”
  • “That’s very interesting”
  • “I see”
Eliza algorithm – memory trick

• When the word “my” is the highest ranked keyword:
  • select a random transform from the MEMORY list
  • Apply it to the sentence
  • Store it on the stack

(MEMORY MY
 (0 MY 0 = LETS DISCUSS FURTHER WHY YOUR 3)
 (0 MY 0 = EARLIER YOU SAID YOUR 3)
 (0 MY 0 = DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR 3)

• Later, when no keyword matches, return the top of the memory stack
PARRY (Colby et al., 1971)

- Rule-based chatbot similar to Eliza
- Was used to study schizophrenia
- Included a model of its own mental states about anger and fear
- First system to pass Turing test in 1972
ELIZA: Good Evening. Tell me your problems. Please terminate input with a period or a question mark.

PARRY: People get on my nerves sometimes.

ELIZA: I am not sure I understand you fully.

PARRY: You should pay more attention.

ELIZA: Suppose you should pay more attention.

PARRY: You're entitled to your own opinion.

ELIZA: What makes you think I am entitled to my own opinion?

PARRY: It bothers me just to be around people in general.

ELIZA: Please go on.

PARRY: I went to the track at Bay Meadows a while back.

ELIZA: What does that suggest to you?

PARRY: I went to the races.

ELIZA: Would you like to discuss your nerves?

PARRY: I guess so.

ELIZA: What does that suggest to you?

PARRY: What are you getting at?

ELIZA: What answer would please you most?

PARRY: It's not a matter or like or dislike.

ELIZA: What else would you like to discuss?

PARRY: I am getting tired of answering the same questions over and over.

ELIZA: What resemblance do you see?

PARRY: Why do you need to know that?

ELIZA: What do you think?

PARRY: I don't understand your motives.

ELIZA: Why don't you understand my motives?

IR-based chatbots

• Respond to user’s turn $X$ by repeating some appropriate turn $Y$ from a corpus of natural text

• Systems differ mainly in:
  • How they choose the corpus
  • How they choose the appropriate response from the corpus
Corpora

- Databases of human conversations
  - Twitter
- Movie dialogs
- Corpora of narrative (non-dialog) texts
- Wikipedia
- Turns that the humans have used to respond to the chatbot
Choosing the response

• Return the most similar turn

\[ r = \arg \max_{t \in C} \cos(q, t) \]

• Return the following turn to the most similar turn

\[ r = \text{response} \left( \arg \max_{t \in C} \cos(q, t) \right) \]

• q – query
• C – conversational corpus
• t – turn in C
Neural seq2seq chatbots

Encoding

How  are  you  ?

Decoding

I’m  fine  .

EOS
Problems with vanilla seq2seq

• They tend to produce dull and repetitive responses like “I’m OK” or “I don’t know”
  • Encourage beam decoder to keep more diverse responses in the beam
• They are unable to model longer prior context of the conversation
  • Hierarchical encoder over words and utterances
• They tend to generate single responses which don’t cohere over multiple turns
  • Add reinforcement or adversarial goals
<table>
<thead>
<tr>
<th>Input</th>
<th>Vanilla-SEQ2SEQ</th>
<th>Adversarial</th>
</tr>
</thead>
<tbody>
<tr>
<td>tell me ... how long have you had this falling</td>
<td>i’m not a doctor.</td>
<td>a few months, i guess.</td>
</tr>
<tr>
<td>sickness?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>so i had the doctors test sammy ’s response to</td>
<td>sammy wrote the test sammy wrote the test.</td>
<td>so he took the pills.</td>
</tr>
<tr>
<td>conditioning.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>they didn ’t have much success with this problem</td>
<td>they ’re not the only ones who have been in the</td>
<td>can we find someone else?</td>
</tr>
<tr>
<td>commander.</td>
<td>system.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Input</th>
<th>Vanilla-SEQ2SEQ</th>
<th>Adversarial</th>
</tr>
</thead>
<tbody>
<tr>
<td>tell me ... how long have you had this falling</td>
<td>i’m not a doctor.</td>
<td>a few months, i guess.</td>
</tr>
<tr>
<td>sickness?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>so i had the doctors test sammy ’s response to</td>
<td>sammy wrote the test sammy wrote the test.</td>
<td>so he took the pills.</td>
</tr>
<tr>
<td>conditioning.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>they didn ’t have much success with this problem</td>
<td>they ’re not the only ones who have been in the</td>
<td>can we find someone else?</td>
</tr>
<tr>
<td>commander.</td>
<td>system.</td>
<td></td>
</tr>
</tbody>
</table>


Evaluating chatbots

• Human evaluation

• ADEM (Lowe et al., 2017) – a classifier trained on a set of responses labelled by humans to predict the appropriateness of a dialogue turn

• Adversarial evaluation – train a “Turing-like” evaluator classifier to distinguish between human- and machine-generated responses
Thanks for attending!