Internal Knowledge Representation for Conversational AI

Zachary Brown & William Myers
Brigham Young University
amazon alexa prize
What makes a good conversation?
Aspects of Natural Language

1. How do we process what the user says?

2. How can we create a response in a naturally-worded way?

3. Given several different responses, how do we pick the most relevant one?
Aspects of Natural Language

1. How do we process what the user says? - Understanding

2. How can we create a response in a naturally-worded way?

3. Given several different responses, how do we pick the most relevant one?
Aspects of Natural Language

1. How do we process what the user says? - **Understanding**

2. How can we create a response in a naturally-worded way? - **Generating**

3. Given several different responses, how do we pick the most relevant one?
Aspects of Natural Language

1. How do we process what the user says? - **Understanding**

2. How can we create a response in a naturally-worded way? - **Generating**

3. Given several different responses, how do we pick the most relevant one? - **Selecting**
understanding  generating  selecting
System Architecture

Illustration credit to Pixar Animation Studio and their film "WALL-E"
System Architecture - Understanding

Illustration credit to Pixar Animation Studio and their film "WALL-E"
System Architecture - Generating

Illustration credit to Pixar Animation Studio and their film "WALL-E"
System Architecture - Selecting

Illustration credit to Pixar Animation Studio and their film "WALL-E"
Knowledge Graphs:
Using Graphs to Understand and Generate Text

- General Knowledge
- High-Level and Low-Level Topic Identification
- User and Self Modelling
- Knowledge Disambiguation and Conflict Resolution
J.R.R. Tolkien Wrote The Hobbit
The Hobbit

Fiction

J.R.R. Tolkien

Genre

Is a

Belongs to the Genre

Wrote in the Genre

Written By

Wrote
User said “I like J.R.R. Tolkien.”:

1) Look for a node named “J.R.R. Tolkien” in the knowledge graph

2) Grab all edges and nodes connecting to the J.R.R. Tolkien node

3) Now you know:
   a) J.R.R. Tolkien is a person
   b) J.R.R. Tolkien wrote The Hobbit

4) Repeat steps 1-3 for the nodes connected to the J.R.R. Tolkien node for more information:
   a) The Hobbit belongs to the genre called Fiction

General Knowledge: Understanding Text
User said “I like J.R.R. Tolkien.”:

1) You now know that:
   a) J.R.R. Tolkien is a person
   b) J.R.R. Tolkien wrote The Hobbit
   c) The Hobbit belongs to the genre called Fiction

2) Given these facts and that the user ‘likes’ J.R.R. Tolkien, it’s straightforward to create templates for generating text:
   a) “Since you like node1 and node1 edge node2, do you also like node2?”

3) In this case, we get:
   a) “Since you like J.R.R. Tolkien, and J.R.R. Tolkien Writes in the Genre Fiction, do you also like Fiction?”

Note: The node and edge names determine the output of the template. This doesn’t sound natural!
You said: “Since you like J.R.R. Tolkien, and J.R.R. Tolkien Writes in the Genre Fiction, do you also like Fiction?”

How do you know the topic of the conversation?
- Topics are hierarchical:
  - The Fiction topic has subtopics, one of which is The Hobbit
- Topics are sometimes never said:
  - J.R.R. Tolkien is a topic within the Person topic

Solution:
- Count number of edges leading to nodes
- Layer topics
- Consider having ‘meta-topics’ determined by combinations of edges present in the conversation

Topic Identification: High-Level and Low-Level

Which nodes and edges should we focus on?

Our models of the user and the EVE system tell us to focus on the The Hobbit node, and not the J.K. Rowling node.

Good models lead to interesting, engaging, and informative dialogue.

Text can be generated by trying to connect the EVE node to the User node. If the User likes something that connects to a node that EVE doesn’t like, we could use the following:

Template:
- “But node1 edge2 node2 and node2 edge3, so why would you edge1 node1?”

Map:

To obtain:
- “But J.R.R. Tolkien wrote Fiction and Fiction is boring, so why would you like J.R.R. Tolkien?”

EVE said: “But J.R.R. Tolkien wrote Fiction and Fiction is boring, so why would you like J.R.R. Tolkien?”

User said: “Good point, I typically don’t like fiction.”

We appear to have a conflict in our knowledge graph, so we ask for clarification; if the user likes something directly connected to something they dislike:

Template:
- “How can you edge1 node1 if you edge2 node2?”

Map:

EVE said: “How can you like J.R.R. Tolkien if you dislike Fiction?”

User said: “Because J.R.R. Tolkien was my grandfather.”
Summary: Knowledge Graphs

Pros:

● Easy to Template
● Enables scalable internal representation of vast amounts of data
  ○ ~ 50 million items
● Straight forward to map out and add new functionality

Cons:

● Naming Conventions
● Time intensive maintenance
● Must try to envision every possible template for user interaction
● Not scalable for broad tasks such as human conversation
understanding

generating

selecting
understanding  generating  selecting
Text Embeddings:

Using Vector Representations of Text to Understand and Select

- What makes a sentence interesting?
- Predicting what the user would like to hear
- Semantic meaning
Text Embeddings

- Map complex data like English text to a numerical representation more suitable for computers.
- Position (distance and direction) captures semantic meaning in these spaces
  - Semantically similar ideas are close
  - Dissimilar ideas are far apart
- Uses:
  - As input for neural networks (e.g. for classification)
  - Directly for NLP tasks
Uses of Text Embeddings for Understanding
Analogical Reasoning

Verb Tense

swam -> walked
swimming -> walking

Arbitrary Relationships

software -> building
architect -> programmer

Vector Representations of Words
https://www.tensorflow.org/tutorials/word2vec
Semantic Similarity

How old are you?
What is your age?
My phone is good.
Your cellphone looks great.

The Universal Sentence Encoder
https://www.tensorflow.org/hub/modules/google/universal-sentence-encoder/1
Classification

How old are you
What is your age
My phone is good
Your cellphone looks great

...
Selection
Which is the best response?

Where are you from?

- New York.
- I grew up in California.
- Have you ever been to California?
- I am from California.
- Yes.
- Babies are usually born in a hospital.
- Manhattan.
- NY.
- Cali.
- I love Star Wars.
- California.
- I grew up in Santa Fe, but my family just moved to Salt Lake City.
We will use:

- Google’s Universal Sentence Encoder
- Our Chit Chat dataset
Embed user’s input

Where are you from?

Universal Sentence Encoder

[0.124, 1.34, … .9]
Find most similar utterances to user’s input

Find closest $n$ points in Chit Chat dataset.
Get their responses

Get the response vectors from Chit Chat dataset for each point.

I am from south carolina.
If money were no object for me...

I am from south carolina.

Where did you grow up?

Where are you from?

Where are you from?

Where were you born?

Are any of my messages sending NOOOOOOO

Arizona:)
Construct analogy vectors

Subtract the closest $n$ points from their responses.
Construct analogy vectors

Average the vectors and extend it by the average norm

\[
\text{analogy\_vec} \ast= \text{avg\_norm} / \text{np.linalg.norm(analogy\_vec)}
\]
Estimate ideal response vector

Add re-lengthed average analogy vector to user’s input.

Where are you from?
Rank candidate outputs by distance to ideal vector

<table>
<thead>
<tr>
<th>Question</th>
<th>Score</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Where are you from?</td>
<td>0.789</td>
<td>I am from California.</td>
</tr>
<tr>
<td></td>
<td>0.843</td>
<td>California.</td>
</tr>
<tr>
<td></td>
<td>0.845</td>
<td>Cali.</td>
</tr>
<tr>
<td></td>
<td>0.855</td>
<td>I grew up in Santa Fe but my family just moved to Salt Lake City.</td>
</tr>
<tr>
<td></td>
<td>0.864</td>
<td>I grew up in California.</td>
</tr>
<tr>
<td></td>
<td>0.899</td>
<td>NY.</td>
</tr>
<tr>
<td></td>
<td>0.925</td>
<td>New York.</td>
</tr>
<tr>
<td></td>
<td>0.998</td>
<td>Manhattan.</td>
</tr>
<tr>
<td></td>
<td>1.058</td>
<td>Yes.</td>
</tr>
<tr>
<td></td>
<td>1.082</td>
<td>Have you ever been to California?</td>
</tr>
<tr>
<td></td>
<td>1.112</td>
<td>I love Star Wars.</td>
</tr>
<tr>
<td></td>
<td>1.203</td>
<td>Babies are usually born in a hospital.</td>
</tr>
</tbody>
</table>
Rank candidate outputs by distance to ideal vector

Where were you born?

1.065 California.
1.106 Cali.
1.131 NY.
1.132 New York.
1.158 Manhattan.
1.18 I am from California.
1.21 I grew up in California.
1.231 I grew up in Santa Fe but my family just moved to Salt Lake City.
1.277 Have you ever been to California?
1.375 Babies are usually born in a hospital.
1.394 Yes.
1.487 I love Star Wars.
Summary: Text Embeddings

Pros:
- Captures semantics
- Semantic similarity is very useful
- Building block to build many different ML models on

Cons:
- Doesn’t readily generate text:
  - Currently can’t decode from the vector space - nearest neighbor searches are the best we have
understanding, generating, selecting
Deep Methods:

Using Machine Learning for End to End Text Understanding, Generation, and Selection

- Novel text generation
- End to End
- Robust
Recurrent Neural Network (RNN)

Sequential text generation

Pros:
- Simple loss function
- Mimic good data

Cons:
- Difficult to train

Pros and cons are not shown on the diagram.

http://colah.github.io/posts/2015-08-Understanding-LSTMs/
Latent Variable Hierarchical Recurrent Encoder-Decoder (VHRED)

Sequential text generation

Pros:
- Contextual integrity improved
- Mimic good data

Cons:
- Difficult to train

A Hierarchical Latent Variable Encoder-Decoder Model for Generating Dialogues (2016)
Convolutional Sequence to Sequence (ConvS2S)

Non-sequential text generation

Pros:
- Parallelizable
- Faster Training
- Mimic good data

Cons:
- More complex loss function than sequential generators

Convolutional Sequence to Sequence Learning (2017)
## Summary: Deep Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Type</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
</table>
| RNN    | Sequential text generation | ● Simple loss function  
● Mimic good data | ● Difficult to train |
| VHRED  | Sequential text generation | ● Contextual integrity improved  
● Mimic good data | |
| ConvS2S| Non-sequential text generation | | ● More complex loss function than sequential generators |

**CON:**
- Lack of good data
- Can’t ‘seed’ for a targeted result

**PRO:**
- Potential for self-play
Unanswered:
How do we produce targeted text with ML?
Unanswered: How do you score a conversation?
What makes a good conversation?
Thank you.

pcc.cs.byu.edu
Questions?