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?

https://cdn.mantelligence.com/wp-content/uploads/2017/11/weird-conversation-starters.png

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?

https://cdn.mantelligence.com/wp-content/uploads/2017/11/weird-conversation-starters.png

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?

https://cdn.mantelligence.com/wp-content/uploads/2017/11/weird-conversation-starters.png

Aspects of Natural Language

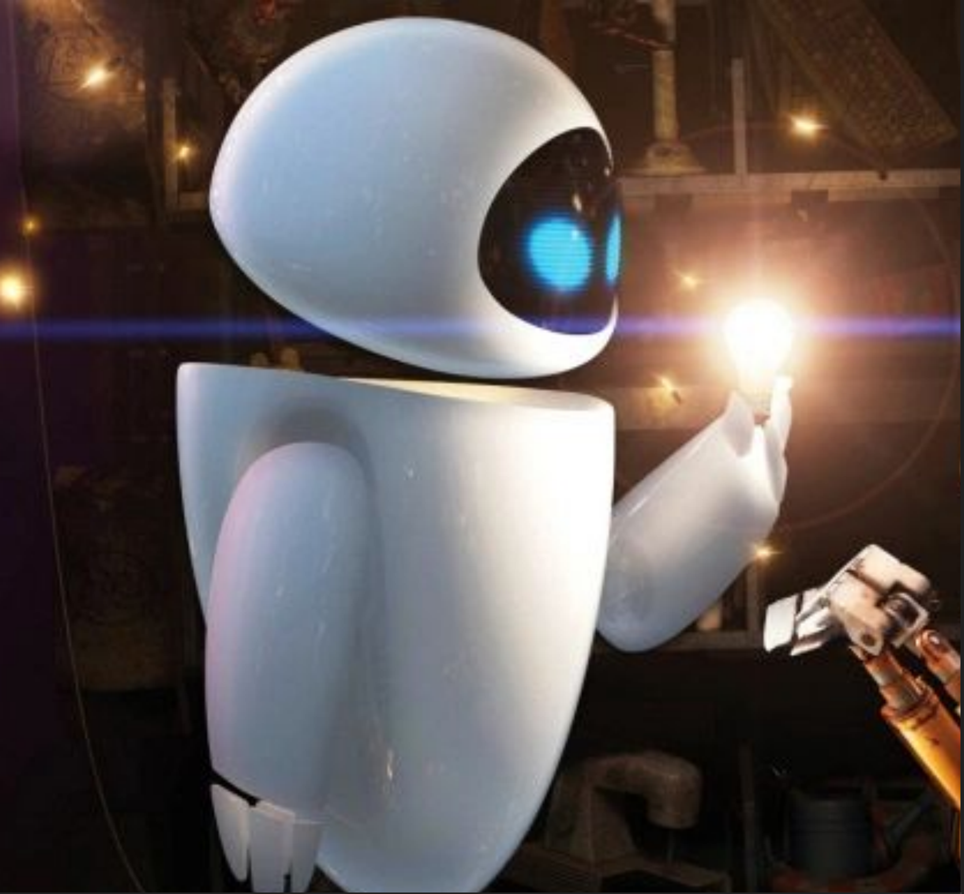
1. How do we process what the user says? - Understanding
2. How can we create a response in a naturally-worded way? - Generating
3. Given several different responses, how do we pick the most relevant one? - Selecting

https://cdn.mantelligence.com/wp-content/uploads/2017/11/weird-conversation-starters.png

understanding

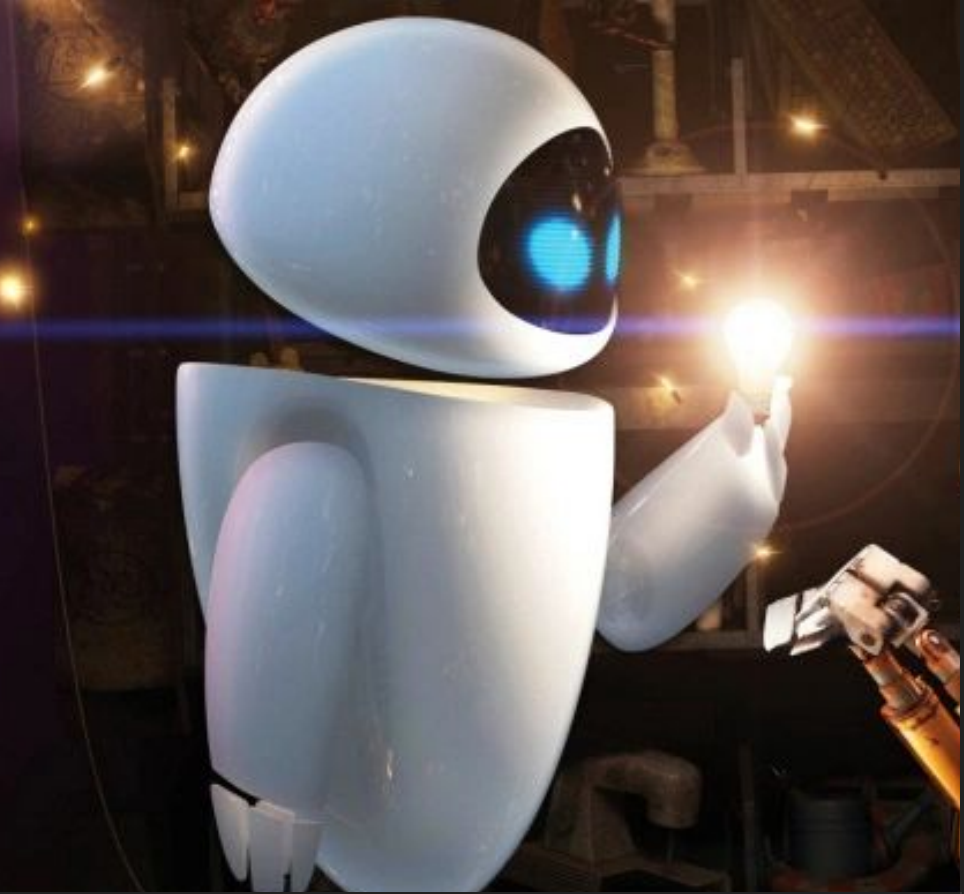
generating

selecting



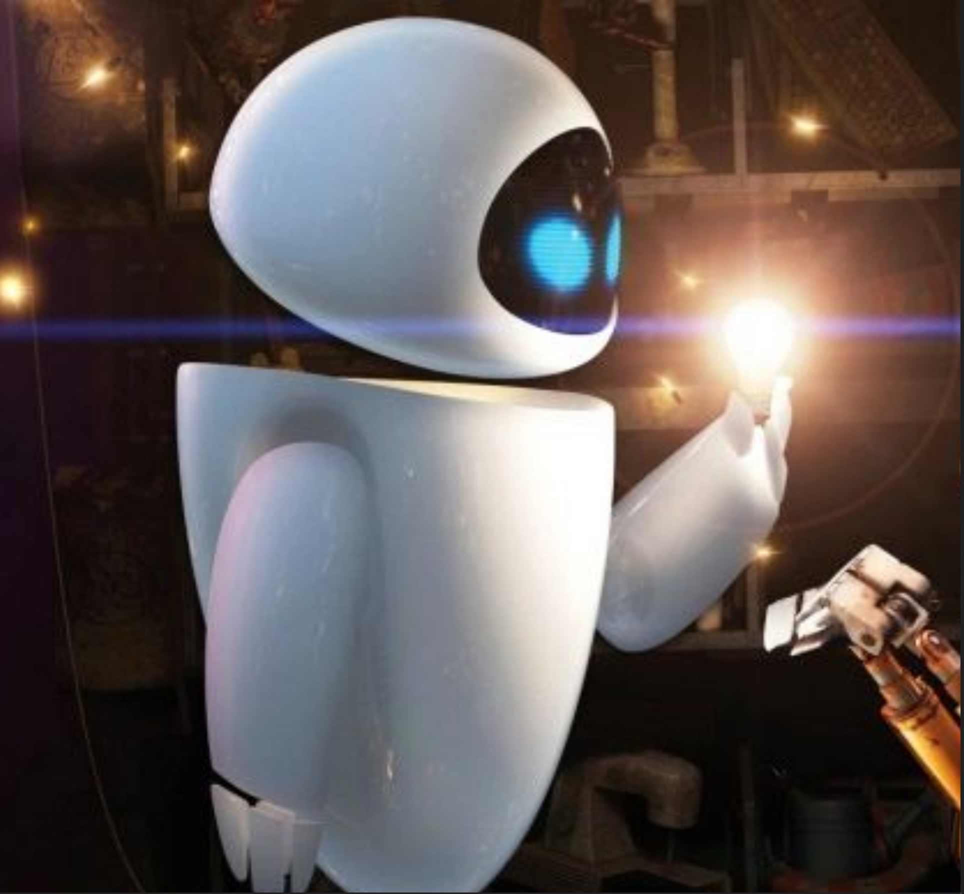
System Architecture

Illustration credit to Pixar Animation Studio and their film “WALL-E”



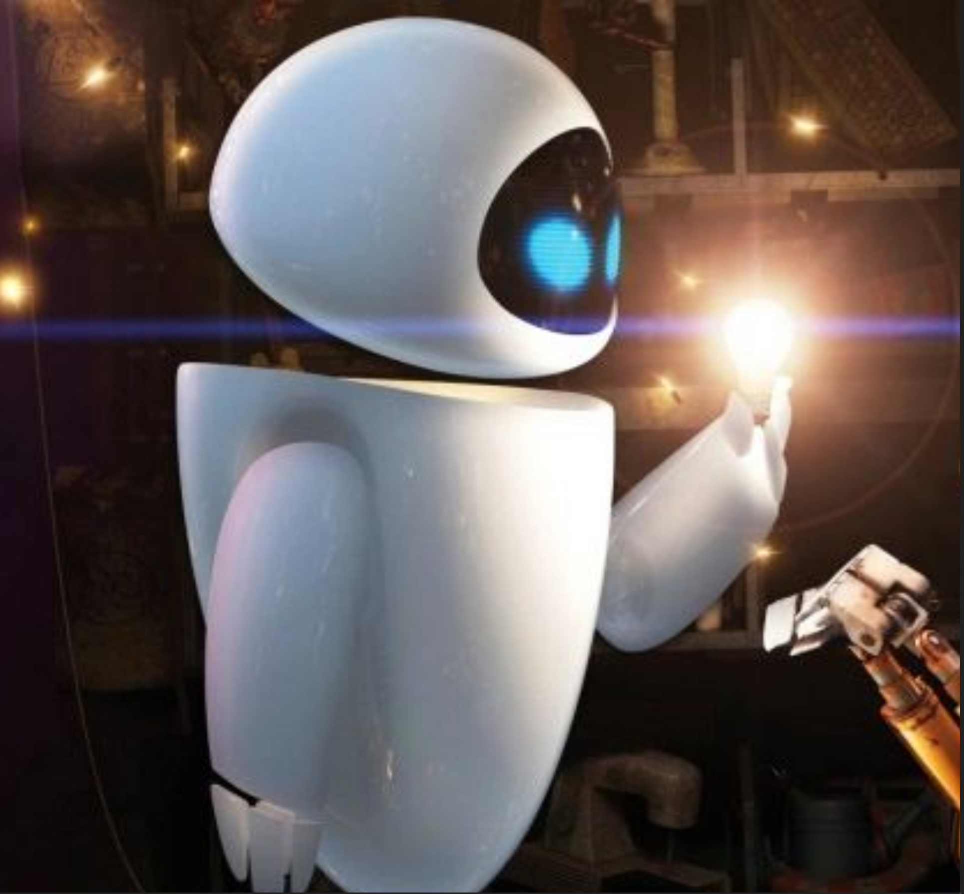
System Architecture - Understanding

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”

understanding

generating

selecting

understanding

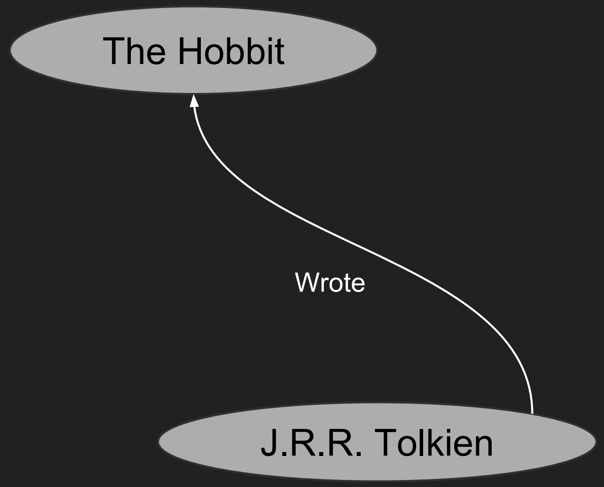
generating

selecting

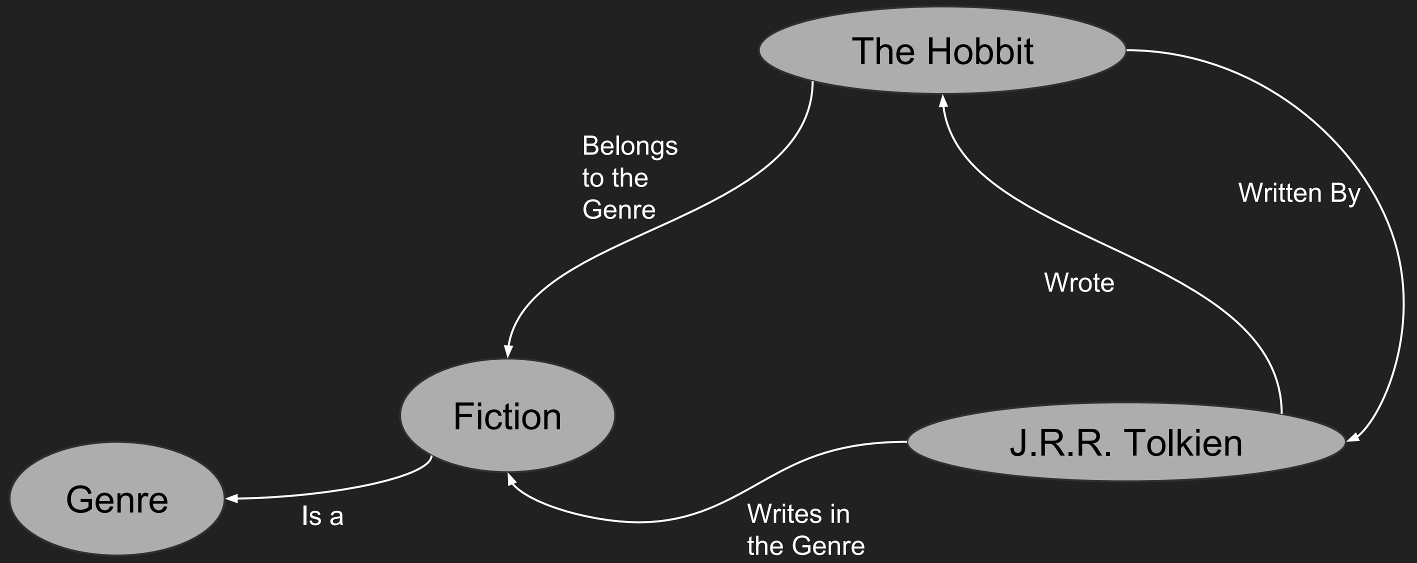
Knowledge Graphs:

Using Graphs to Understand and Generate Text

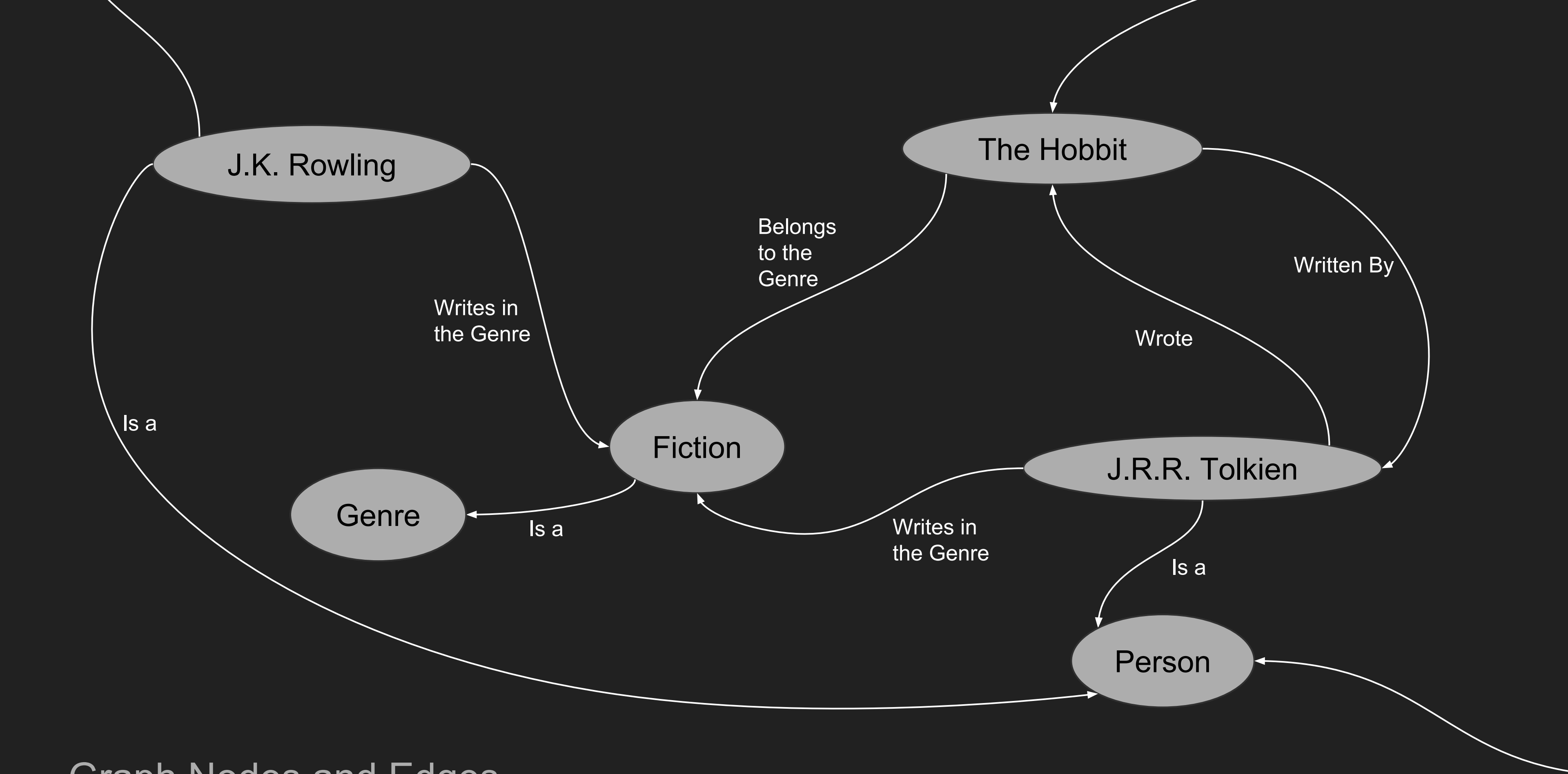
- General Knowledge
- High-Level and Low-Level Topic Identification
- User and Self Modelling
- Knowledge Disambiguation and Conflict Resolution



Graph Nodes and Edges



Graph Nodes and Edges



Graph Nodes and Edges

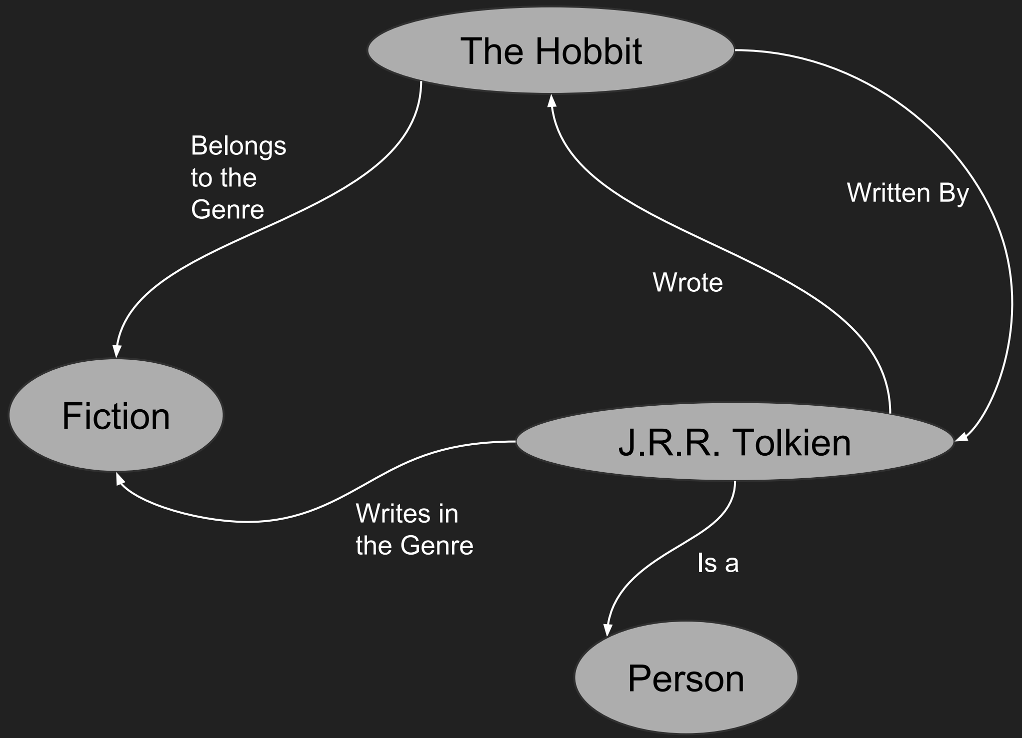
User said “I like J.R.R. Tolkien.”:

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



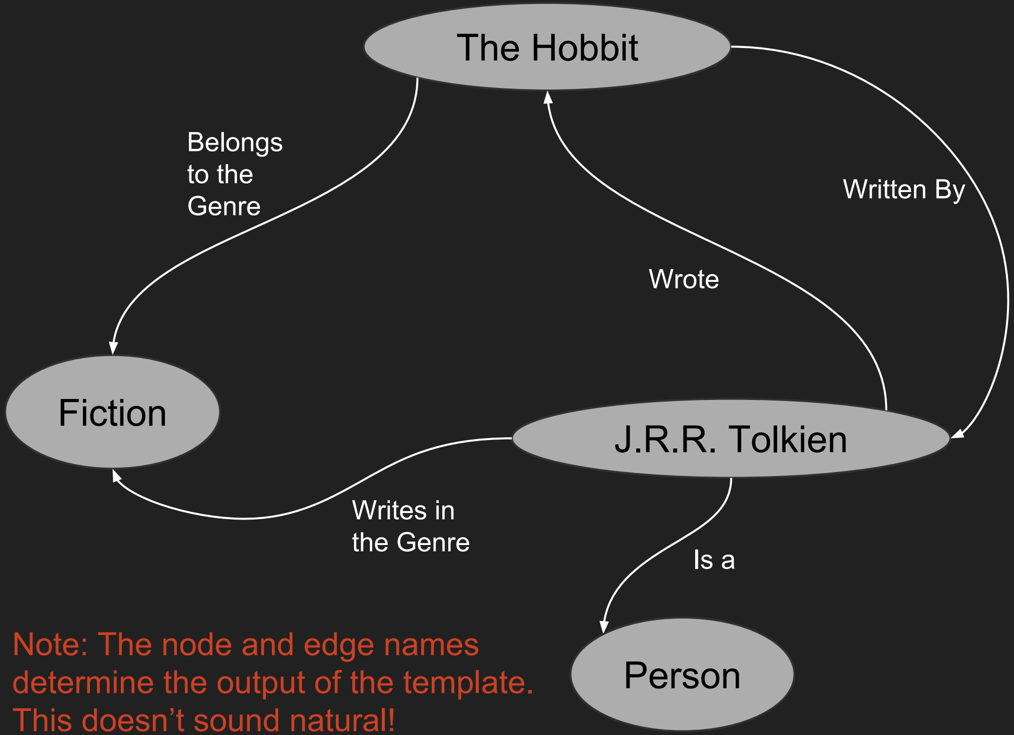
General Knowledge: Understanding Text

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!

General Knowledge: Generating Text

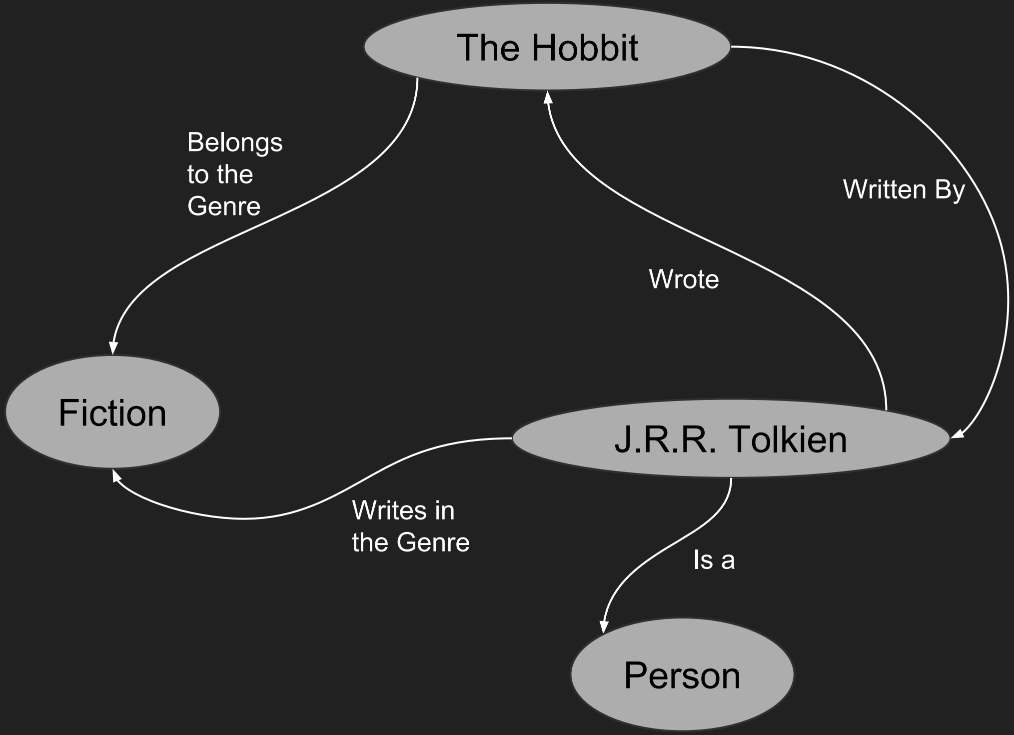
User said: "I like J.R.R. Tolkien."
You said: "Since you like J.R.R. Tolkien, and J.R.R. Tolkien Writes in the Genre Fiction, do you also like Fiction?"

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

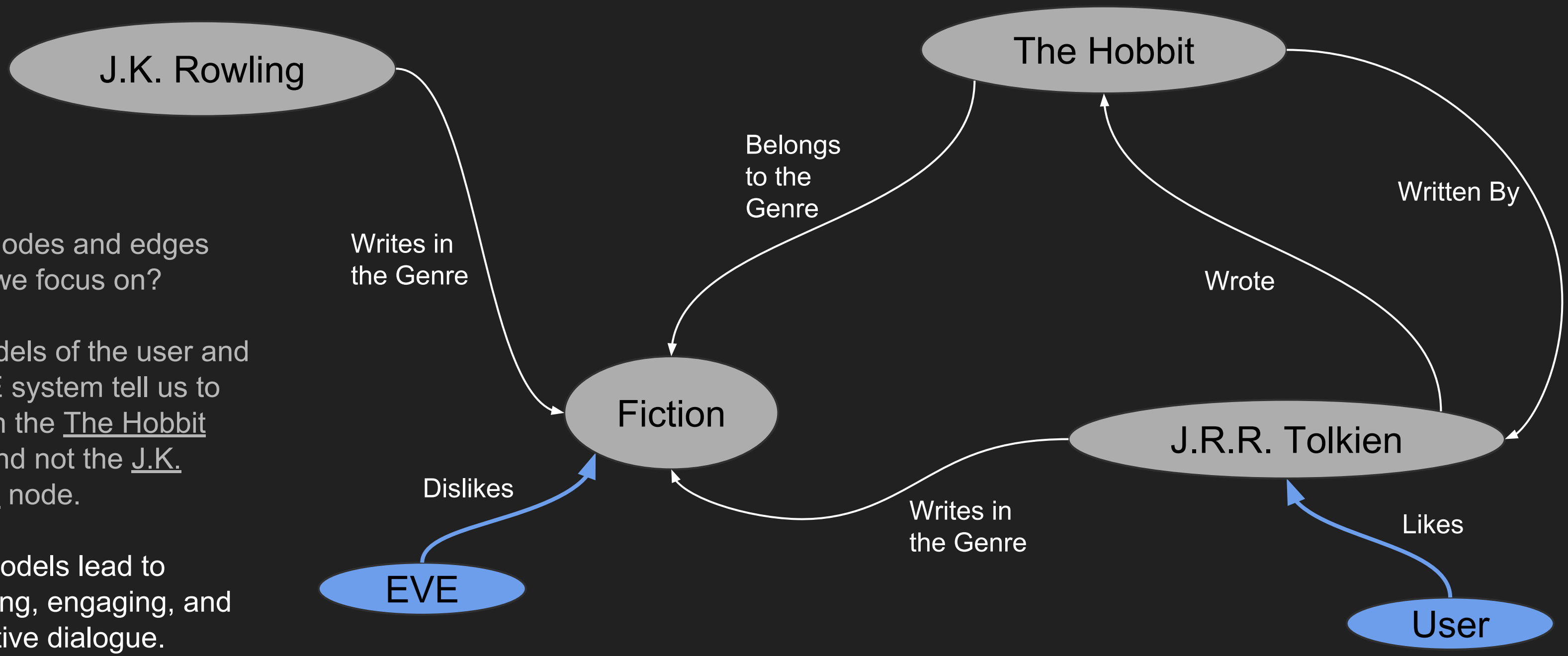
Solution:

- 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

User said: “I like J.R.R. Tolkien.”



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.

User and Self Modelling: Understanding Dialogue

User said: “I like J.R.R. Tolkien.”

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:

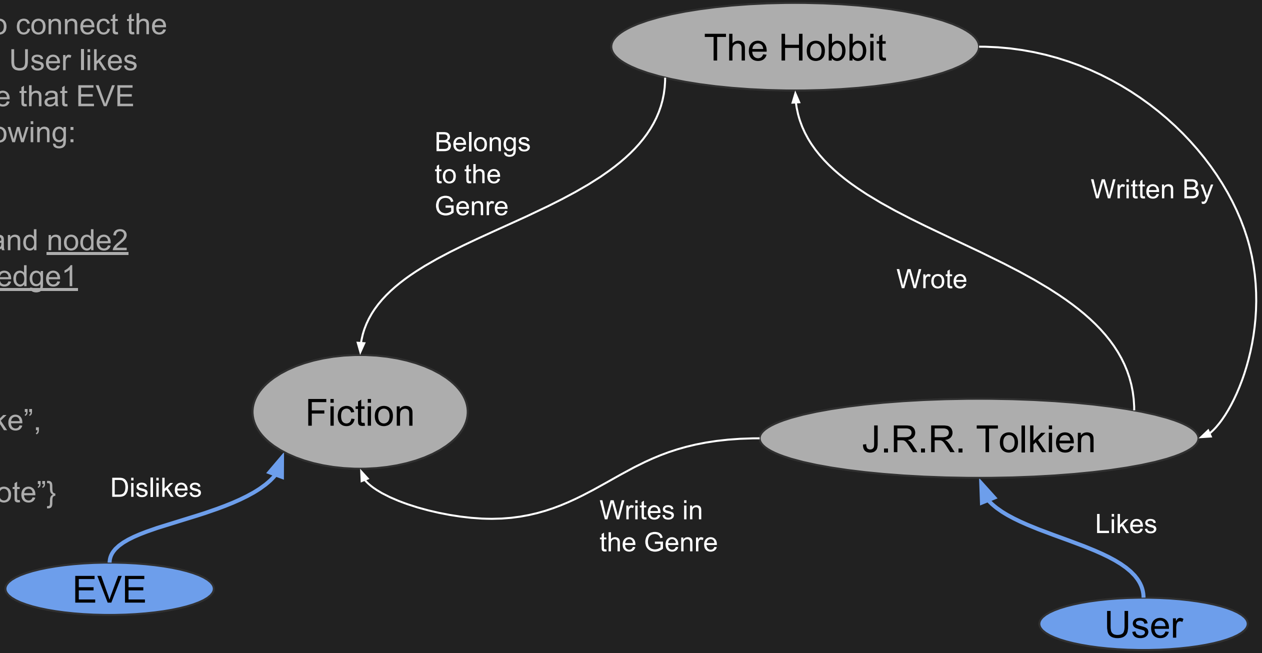
- “But node1 edge2 node2 and node2 edge3, so why would you edge1 node1?

Map:

- {“User” : “you”, “Likes” : “like”, “Dislikes” : “is boring”, “Writes in the Genre” : “wrote”}

To obtain:

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



User and Self Modelling: Generating Text

User said: “I 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.”

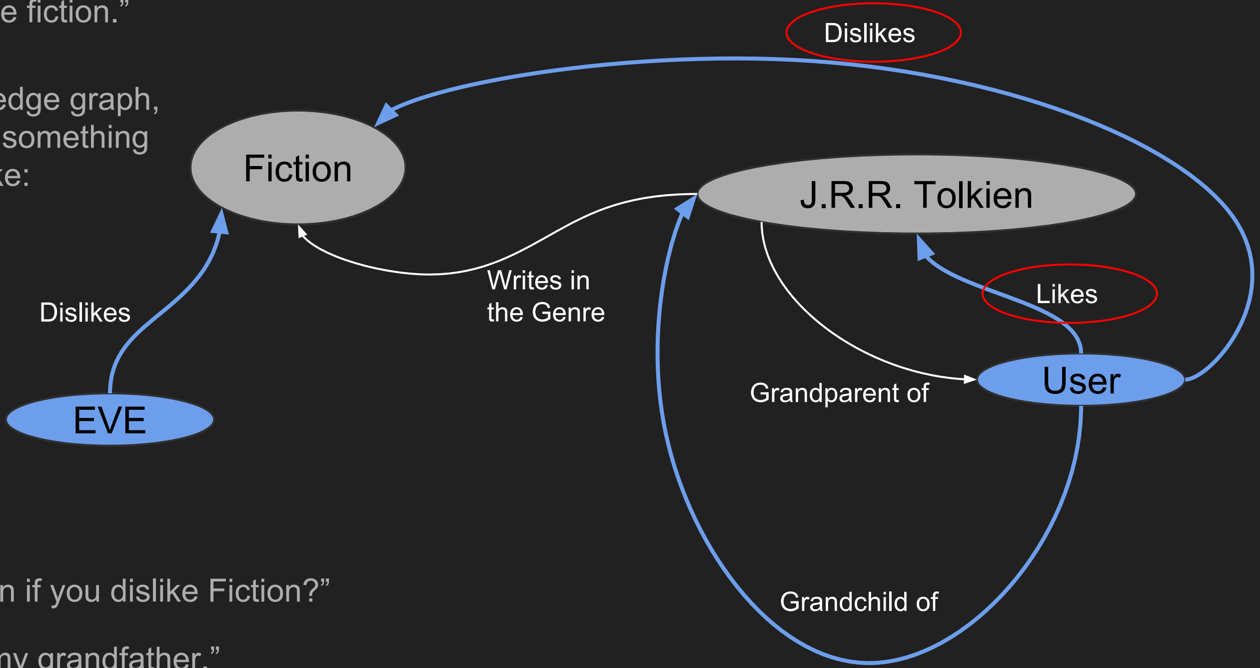
We appear to have a conflict in our knowledge graph, so we ask for clarification; if the user likes something directly connected to something they dislike:

Template:

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

Map:

- {“User” : “you”, “Likes” : “like”, “Writes in the Genre” : “wrote”, “Dislikes” : “dislike”}

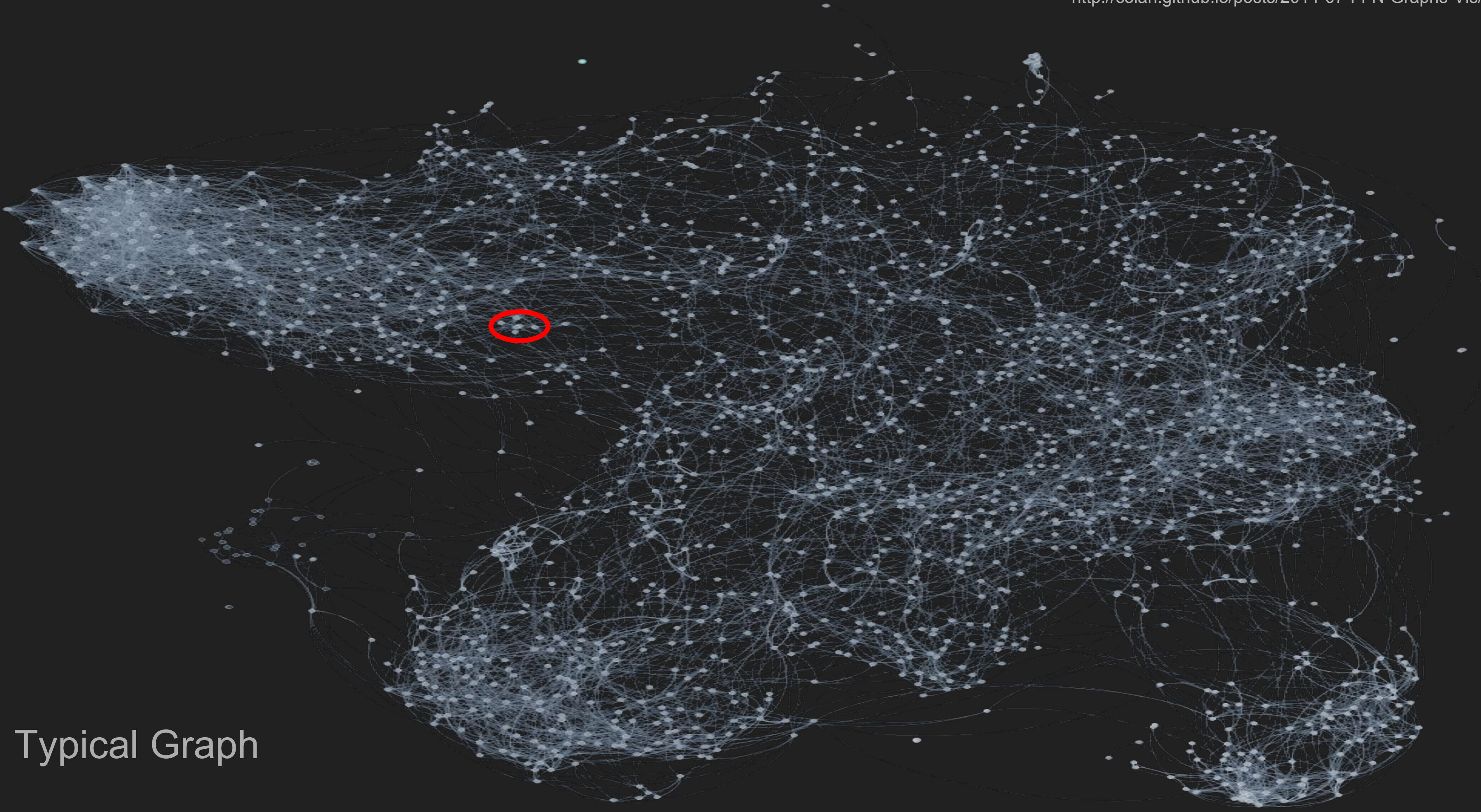


EVE said: “How can you like J.R.R. Tolkien if you dislike Fiction?”

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

Knowledge Disambiguation and Conflict Resolution

http://colah.github.io/posts/2014-07-FFN-Graphs-Vis/



Typical Graph

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

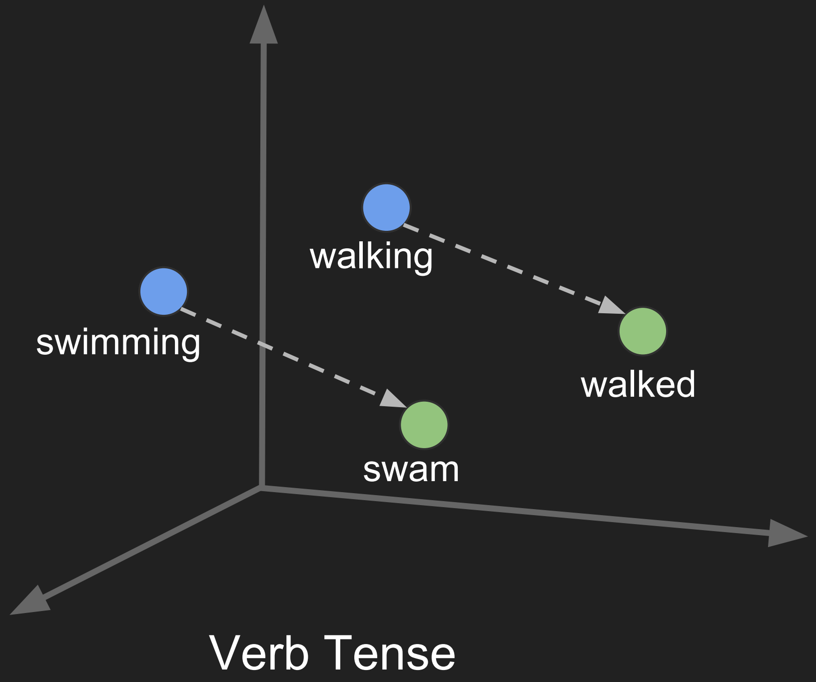
- What makes a sentence interesting?
- Predicting what the user would like to hear
- Semantic meaning

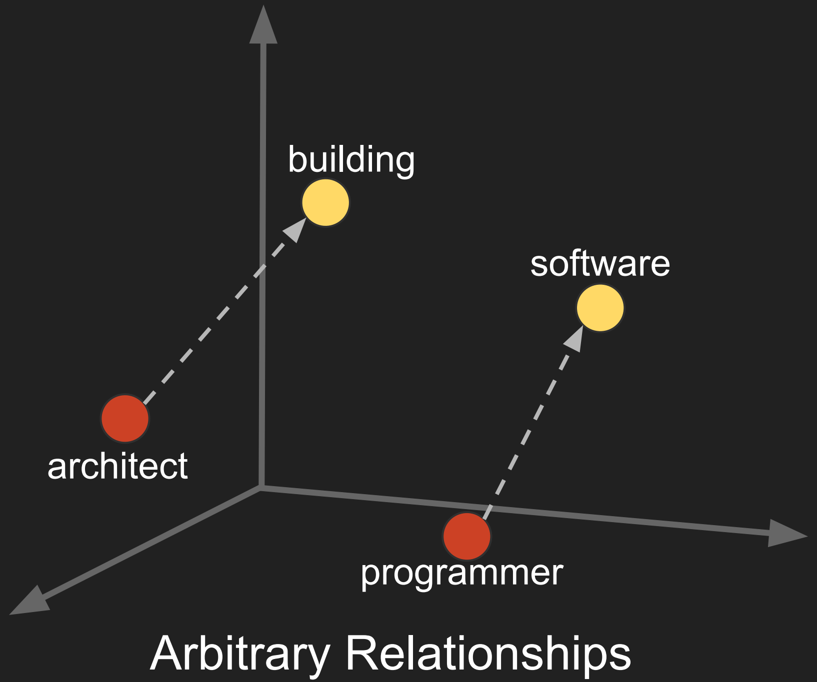
Text Embeddings

- 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





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.

...

Embed

[0.1, 0.6, …, 0.7]

[0.3, 0.8, …, 0.5]

[0.9, 0.2, …, 0.1]

[0.4, 0.7, …, 0.2]

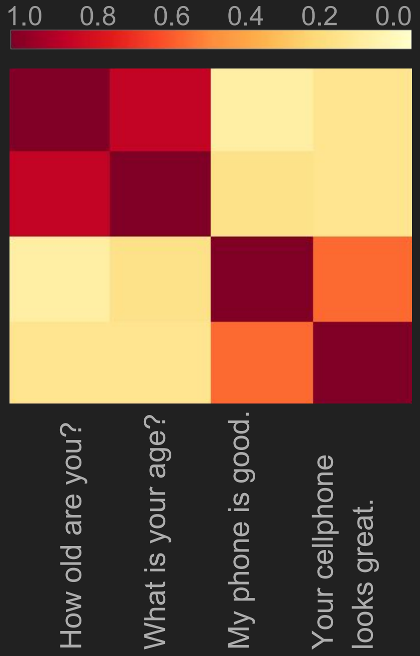
…

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

Input		Embedding		Score	Question
How old are you	Embed	[0.1, 0.6, …, 0.7]	Question Classifier	**96%**	How old are you
What is your age		[0.3, 0.8, …, 0.5]		**98%**	What is your age
My phone is good		[0.9, 0.2, …, 0.1]		**7%**	My phone is good
Your cellphone looks great		[0.4, 0.7, …, 0.2]		**8%**	Your cellphone looks great
...		…		...	

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

Selection

Which is the best response?

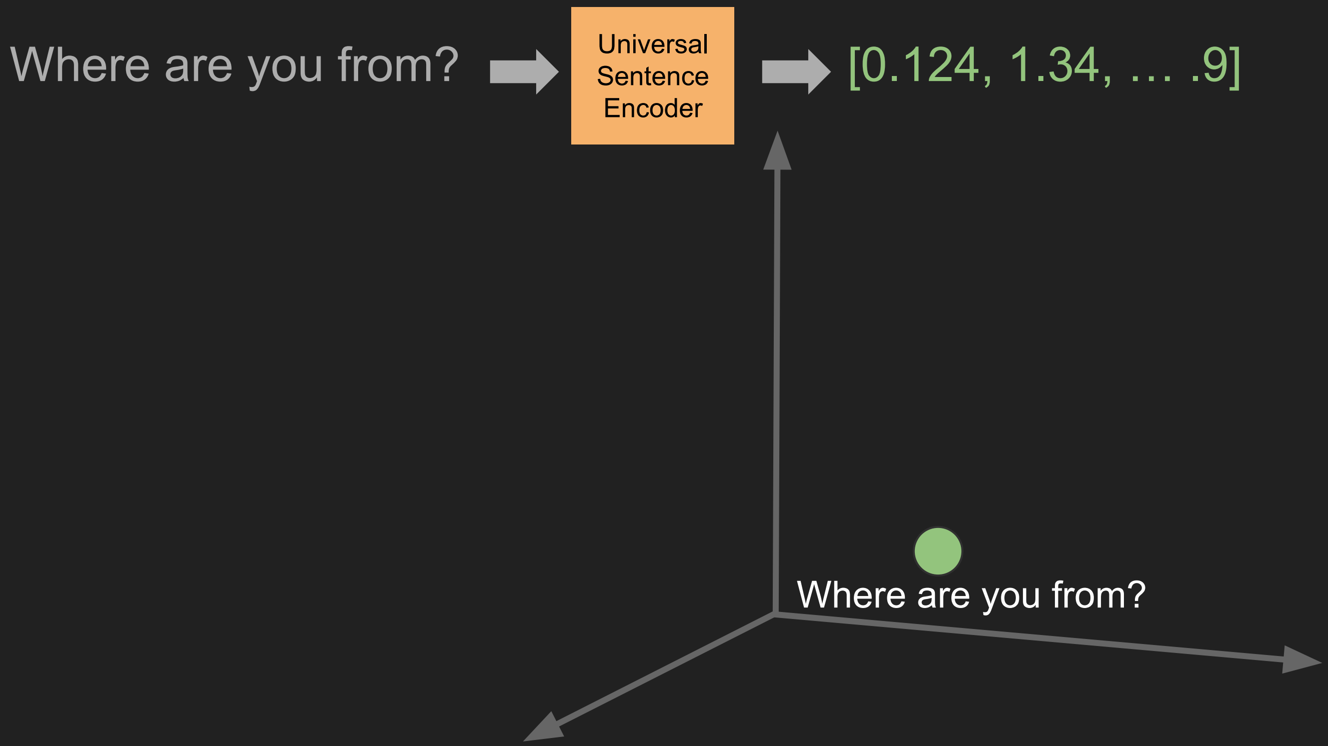
Where are you from?

- 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:

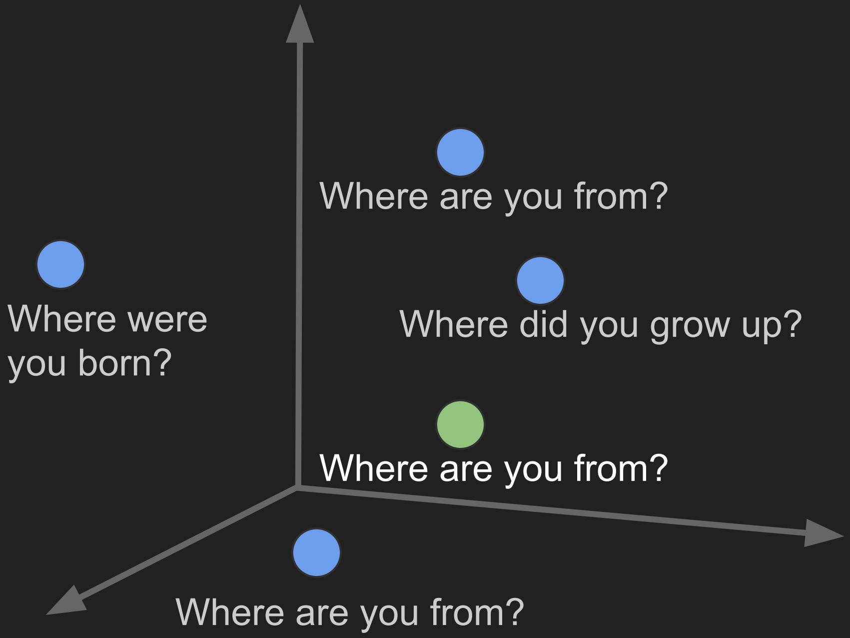
- Google’s Universal Sentence Encoder
- Our Chit Chat dataset

Embed user’s input



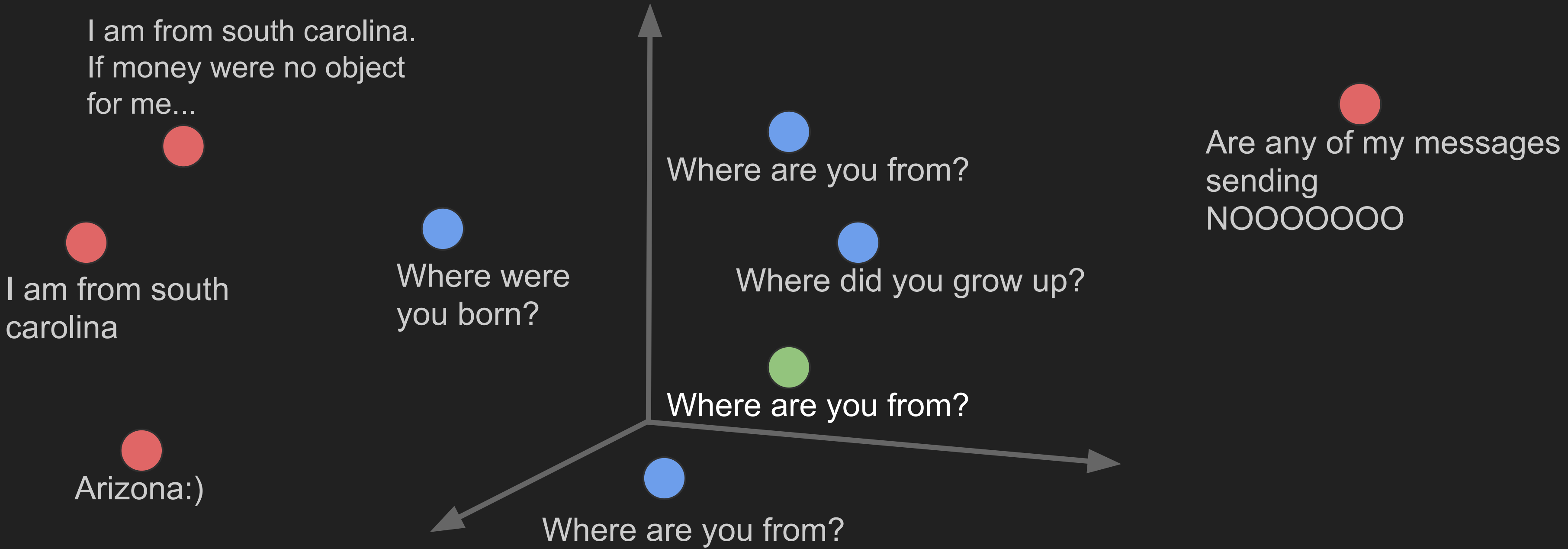
Find most similar utterances to user’s input

Find closest n points in Chit Chat dataset.



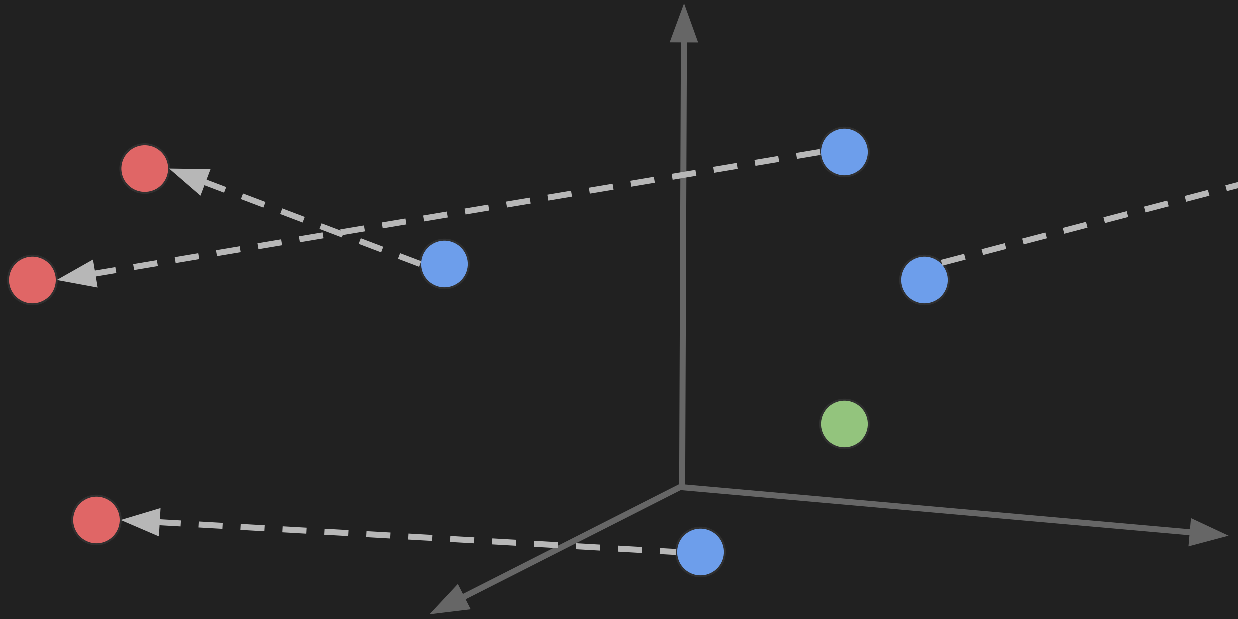
Get their responses

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



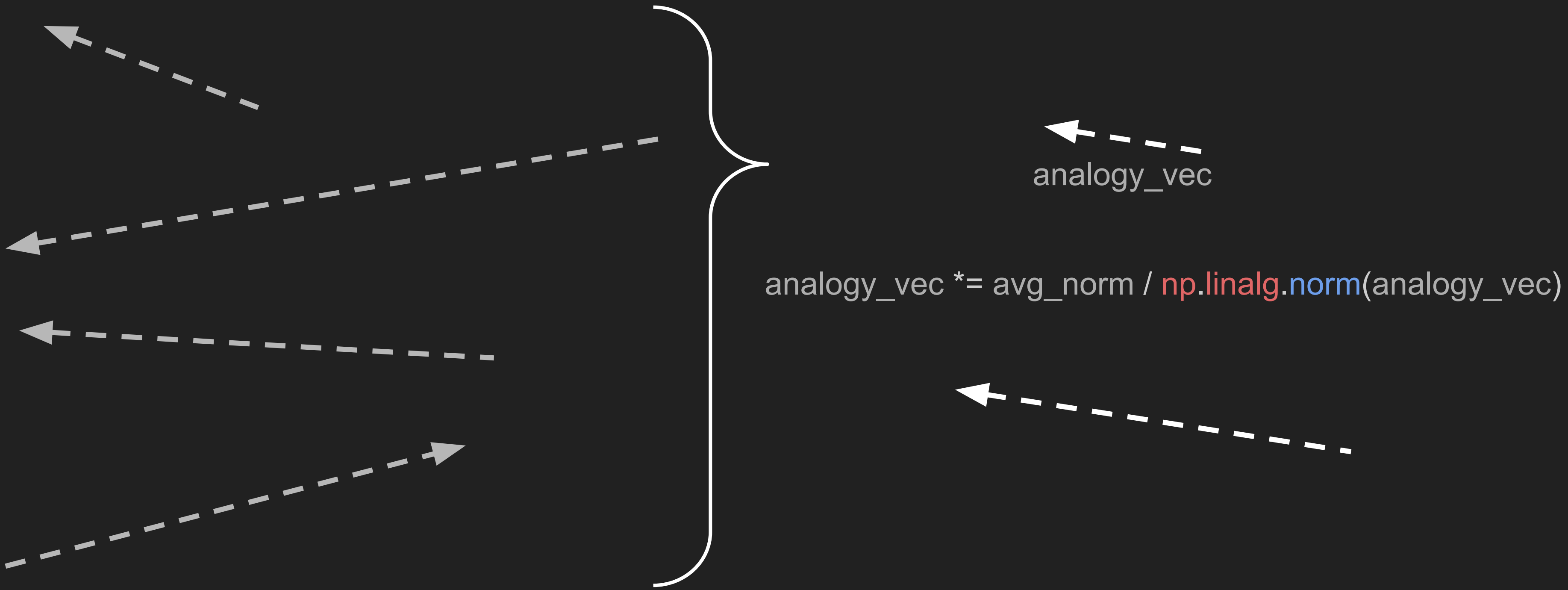
Construct analogy vectors

Subtract the closest n points from their responses.



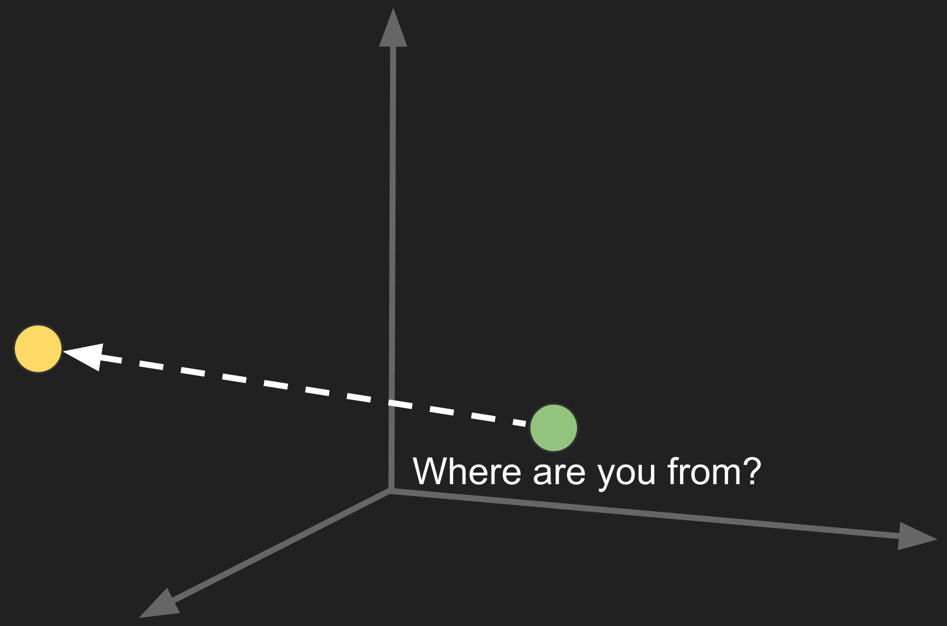
Construct analogy vectors

Average the vectors and extend it by the average norm



Estimate ideal response vector

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



Rank candidate outputs by distance to ideal vector

Where are you from?

0.789	I am from California.
0.843	California.
0.845	Cali.
0.855	I grew up in Santa Fe but my family just moved to Salt Lake City.
0.864	I grew up in California.
0.899	NY.
0.925	New York.
0.998	Manhattan.
1.058	Yes.
1.082	Have you ever been to California?
1.112	I love Star Wars.
1.203	Babies are usually born in a hospital.

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

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)

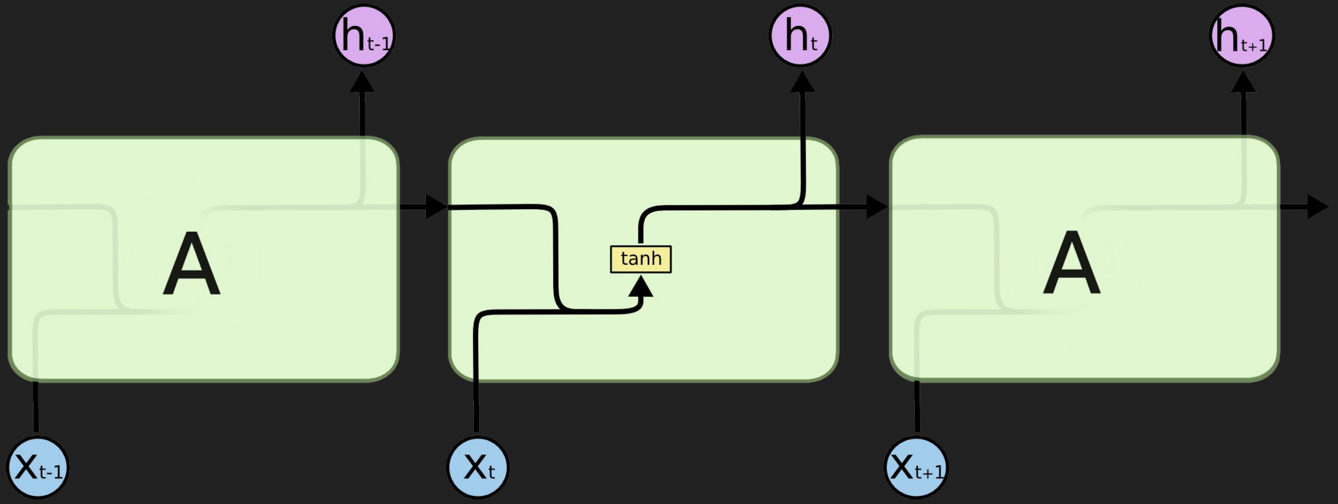
Sequential text generation

Pros:

- Simple loss function
- Mimic good data

Cons:

- Difficult to train



http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Latent Variable Hierarchical Recurrent Encoder - Decoder (VHRED)

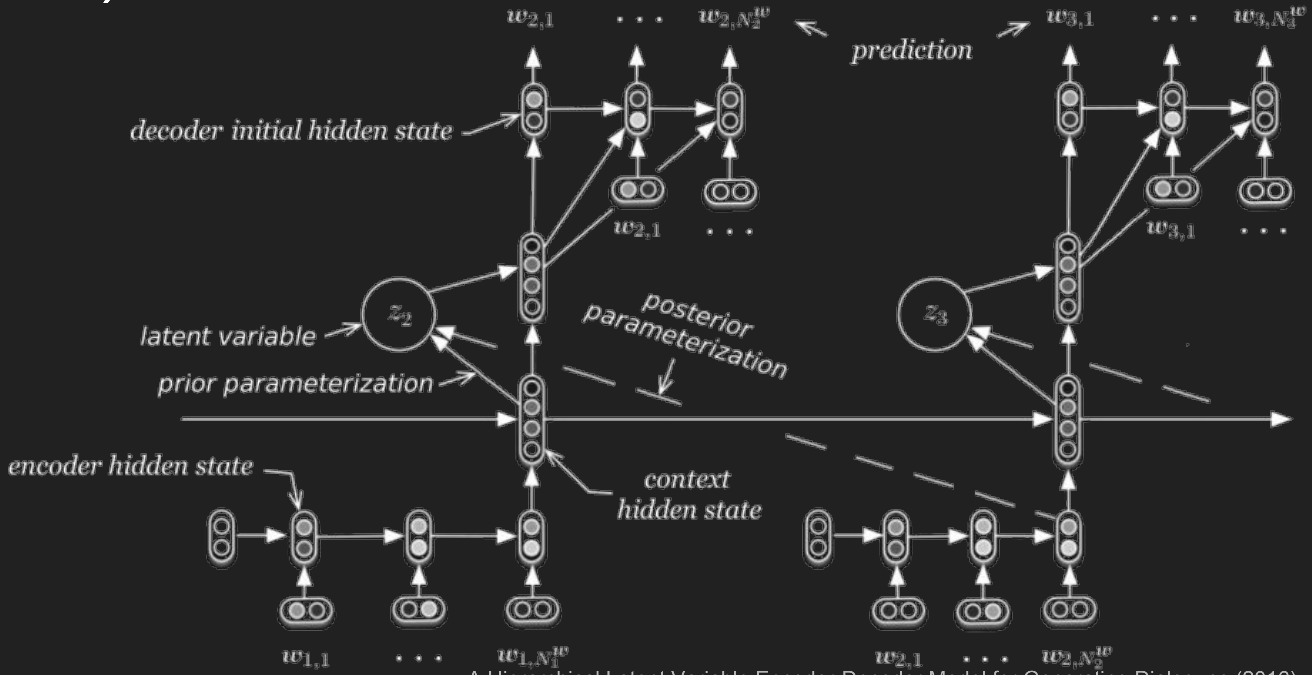
Sequential text generation

Pros:

- Contextual integrity improved
- Mimic good data

Cons:

- Difficult to train



A Hierarchical Latent Variable Encoder-Decoder Model for Generating Dialogues (2016)
https://arxiv.org/pdf/1605.06069.pdf

Convolutional Sequence to Sequence (ConvS2S)

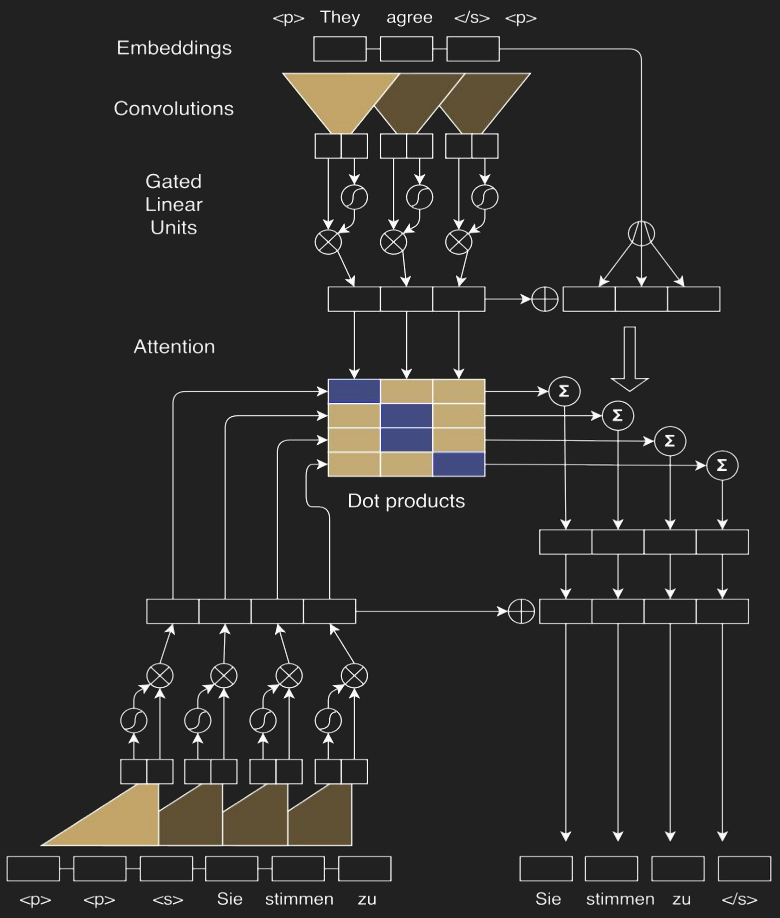
Non-sequential text generation

Pros:

- Parallelizable
- Faster Training
- Mimic good data

Cons:

- More complex loss function than sequential generators



Convolutional Sequence to Sequence Learning (2017)
https://arxiv.org/pdf/1705.03122.pdf

Summary: Deep Methods

RNN	VHRED	ConvS2S
Sequential text generation	Sequential text generation	Non-sequential text generation
Pros: ● Simple loss function ● Mimic good data	Pros: ● Contextual integrity improved ● Mimic good data	Pros: ● Parallelizable ● Faster Training ● Mimic good data
Cons: ● Difficult to train	Cons: ● Difficult to train	Cons: ● 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?