Relational Dialogue

by

Matthew D. Huggins

SB Electrical Engineering and Computer Science
Massachusetts Institute of Technology, 2019

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

FEBRUARY 2021

Signature of Author: _____
Department of Electrical Engineering and Computer Science
January 15, 2021

Certified by: _____
Cynthia Breazeal
Professor of Media Arts and Sciences
Thesis Supervisor

Accepted by: _____
Katrina LaCurts
Chair, Master of Engineering Thesis Committee

Relational Dialogue

by

Matthew D. Huggins

Submitted to the Department of Electrical Engineering and Computer Science on January 15, 2021 in Partial Fulfillment of the Requirements for the Degree of Master of Engineering in Electrical Engineering and Computer Science

ABSTRACT

Conversational agents are increasingly common in everyday life. Dialogue with these agents is often limited to the task at hand, and is not focused on conversation as a shared social experience. Previous work has demonstrated that strengthening the user-agent relationship increases the agent's efficacy, and leads to a more enjoyable user experience. I present a relationship-driven dialogue system that aims to strengthen and expand the relationship between the agent and user. The system uses a knowledge graph to represent relevant information about the world and the agent's and user's preferences. When choosing a response, a novel probabilistic approach, called MRF-Chat, models the mutual knowledge of the agent and the user, as well as the contextual relevance of concepts in candidate responses. In human evaluations, the system was considered significantly more collaborative, engaging, and trusted by human partners in a semi-structured interaction on food preferences.

Thesis Supervisor: Cynthia Breazeal
Title: Professor of Media Arts and Sciences; Associate Director, MIT Media Lab

Table of Contents

List of Figures

Introduction

Since ELIZA, a computer psychotherapist, was created by Joseph Weizenbaum at MIT in 1966[1], there has been an incredible amount of research in creating computer programs that can emulate human conversation. After ELIZA came PARRY[2], and eventually, in 1995, perhaps the first modern chat bot, A.L.I.C.E.[3]. Since then, enormous progress has been made in both processing and generating human language. Conversational agents such as Alexa and Siri are household names. When we call the support numbers of the companies that provide the goods and services that we use in our everyday lives, we are often greeted by a friendly virtual agent instead of a human voice. Even when we converse with real humans through email, text messages, and other electronic communication media, the writing of our responses is increasingly assisted by conversational AI systems[1].

Until recently, conversational agents existed primarily on the internet or in virtual call centers. With the advent of Amazon's Alexa, Google Home, and the first social robot for the home, Jibo, conversational agents have entered mainstream life. Last year, a report from NPR and Edison Research[4] estimated that 60 million American adults have smart speakers, such as Amazon's Echo Dot, in their homes. The prevalence of conversational agents in modern life will continue to increase as virtual assistants become more common. Over time, social robots, which engage in emotionally intelligent interactions with humans, will become more common in our homes, in our workplaces, and in our hospitals. As these agents become increasingly ubiquitous,

[1] See Google Smart Compose, Smart Reply, Intercom, Drift, Point API, and many others.

there is a growing need for systems that allow the agents to engage in intelligent, empathetic, and relational conversation with humans.

The majority of research in conversational AI has focused on goal- or task-oriented dialogue systems. Generally, a goal-oriented dialogue system aims to guide a human user through a process in order to gather appropriate information, triage an existing issue, or execute a desired command. Examples include scheduling an Amtrak train ticket with a virtual agent over the phone, asking Siri to remind you about a meeting, or ordering a pizza through Alexa. In the most naive form, a goal-oriented system may be made of hand-written rules and responses that follow a defined interaction flow[5]. In more sophisticated systems, neural network language models may recognize user intent[6], information may be synthesized from large databases and communicated to the user in natural language[7], and massive transformer-based models may generate text from scratch[8]. No matter how simple or complex, these systems have well-defined end goals to accomplish.

While goal-oriented dialogue systems can provide highly engaging interactions in the contexts they are developed for, they are unable to have unstructured conversations on a wide range of topics. However, the majority of human-human conversations are not explicitly goal-oriented, and instead cover a broad range of topics in a casual setting. Small talk, chit-chat, and conversations between friends cannot be modeled with goal-oriented systems. In order to create conversational agents that can converse with humans in social contexts, we must develop ***open-domain dialogue systems*** that allow agents to talk about a wide range of topics without a structured end goal.

One may raise the question: “Why do we want to have casual conversations with computers? Why not limit interactions with machines to goal-based tasks?”. The reasons are two-fold. First, humans generally use a social model when interacting with autonomous robots[9]. By enabling robots, and conversational agents in general, to have more social dialogue with humans, they can provide a much more engaging and emotionally satisfying experience to the people that interact with them. Second, by creating a stronger social fabric between the agent and the user, the agent can more effectively deliver any desired support or intervention. Recent research has found that social robots can help college students improve their psychological wellbeing[10], help young children learn language more effectively[11], and promote social connectedness among older adults[12]. By improving an agent’s ability for social interaction, the agent can become a more effective tool for increasing the quality of human lives.

While the chat bots of the late 20th century could naively engage in broad conversation to some extent, the field of modern open-domain dialogue is very young. In order to build a system that can talk about a wide range of topics, a more complex and generalizable approach is needed than hand-crafted systems can provide, and that approach is neural language models. While the field of natural language processing (NLP) has existed for quite some time, the effective use of neural networks in NLP is recent and rapidly developing. In 2013, the long short-term memory network (LSTM) was introduced as a promising model for NLP[13], and in 2015, the advent of attention systems greatly improved the performance of LSTMs and other recurrent neural network (RNN) architectures[14]. Since 2018, transformer-based language models[15]

that are pre-trained on massive amounts of text, such as BERT[16] and GPT[17], have become the standard for high-performance on many tasks in the NLP community.

Even with many recent advancements in NLP, open-domain dialogue is still far from solved. Open-domain dialogue models can be broadly categorized into two types: ***retrieval*** models and ***generative*** models. Retrieval models take an input utterance from the user, and attempt to select a best response from a large, fixed list of candidates. Since the responses are fixed, they are generally coherent, but the system is limited by a finite set of possible responses, making it difficult or sometimes impossible to properly respond to utterances that are different than examples in the training data. On the other hand, generative models take a user utterance as input, and create a response from scratch based on probabilistic outputs from the model. Until recently, generative models struggled to produce rich and coherent responses, but extremely complex models such as GPT-2[17], and very recently, GPT-3[18], have shown more promise in generating fluent responses. However, while state-of-the-art retrieval and generative models can often produce coherent and on-topic responses, they still struggle to create responses that are logically consistent with the conversational history. In practice, dialogue systems are often complex combinations of various independent modules, including several ***skills*** or modules that are responsible for handling specific types of interactions or topics, and possibly combining one or more retrieval models with rule-based systems to cover a wide variety of use cases[20].

One of the greatest drivers of progress in open-domain dialogue has been the Alexa Prize. Started in the fall of 2016, the Alexa Prize[21] is a university competition hosted by Amazon where selected academic teams build conversational agents, referred to as "socialbots". The goal of the competition is to create engaging and coherent dialogue with humans on a variety of topics, including sports, entertainment, and politics. The grand prize objective set by the competition is a coherent 20 minute conversation between the agent and the user. As part of the competition, teams are required to publish about their work, so there is a wide range of available literature about recent work in conversational AI, especially in incorporating known information into dialogue. Many Alexa Prize dialogue systems are incredibly complex, built by teams often of nearly a dozen researchers over several months.

While recent work in open-domain dialogue has made great advances towards better human-computer conversions, there are three key issues that often occur:

1. Conversation is limited to several pre-defined topics, and if the user tries to move conversation away from one of those topics, the agent may forcefully move the conversation back to the original topic[22,64].

2. The agent is primarily focused on inserting knowledge into the conversation, not on creating a shared social experience with the user[23].

3. The quality of the conversation is defined by it's length, breadth of topics, or amount of information used, not by the user's enjoyment, rapport between the user and the agent, or the desire for the user to have more conversations in the future[21].

All three of these issues stem from a single source, a view of conversation as a set of concrete skills to acquire and metrics to achieve, not as a highly-nuanced and collaborative social experience. If an agent can talk more coherently about a wider range of topics for a longer time, mustn't it be a better conversational partner for a human to interact with? The problem with this line of reasoning is that values the technical skill of conversation over the reason why the conversation is happening in the first place: as a shared social experience between both parties. While many daily conversations between humans aim to achieve a specific end goal or to communicate certain information, many others conversations happen for purely social reasons. Small talk when meeting someone new, casual conversations between friends, and conversations about weekend plans between coworkers over coffee all serve almost exclusively social goals. When was the last time that you measured the quality of a conversation with your friend by its length and topical rigor?

I believe that conversational agents should converse with humans as partners in a shared experience that focuses on developing the relationship between the agent and the user. By focusing on the interests and knowledge of both parties, and exploring the common ground between them, conversational agents will be able to provide more enjoyable and fulfilling conversations to the people who talk with them, and will be able to more effectively deliver support and interventions to improve human lives.

In order to move towards this vision, I present a relationship-driven dialogue system that aims to strengthen and expand the relationship between the agent and user by using natural conversation to learn about each other, share experiences, and find common ground. The system uses a knowledge graph of Wikidata entities to represent

relevant information about the world, the agent's and user's preferences, and conversational history. When choosing a response, a novel probabilistic approach, called MRF-Chat, is used to model the mutual knowledge of both the agent and the user, as well as the contextual relevance of concepts included in candidate responses. Finally, I demonstrate how the system can be used to improve a semi-structured interaction for a goal-oriented task, recommending a healthy recipe for dinner. The remainder of this document is structured as follows:

Related Work

The Alexa Prize[21] is a university competition hosted by Amazon where selected university teams build conversational agents. The goal of the competition is to create engaging and coherent dialog with humans on a variety of topics, including sports, entertainment, and politics. The grand prize objective set by the competition is a coherent 20 minute conversation between the agent and the user. As part of the competition, teams are required to publish about their work, so there is a wide range of available literature about recent work in conversational AI, especially in incorporating known information into dialogue. In addition, Amazon has released the Topical Chat dataset[24], which contains nearly eleven thousand knowledge-grounded human-to-human conversations about topics used in the competition.

Perhaps the most relevant system from the Alexa Prize is BYU-EVE[25], a knowledge graph based dialogue system created at Brigham Young University. BYU-EVE uses a knowledge graph to store information about the world, as well as simple representations of the user and the agent, that consists of their “likes” and “dislikes”, as well as some other information such as nicknames. The system uses several response generators to create candidate agent utterances, including emotional mirroring and discussion about user information in the knowledge graph. In order to choose candidate responses, the system uses conversational scaffolding, which compares the user’s input and each candidate responses against a dataset of conversations. The dataset is publicly available on GitHub[26]. The creators of BYU-EVE designed dialogue with their conversational agent as both learning about the user and

sharing the agent's preferences, and their knowledge graph-based system was the first approach that explicitly modeled this task. However, their response generation did not allow for a broad range of responses, and their response selection algorithms did not model the shared conversational experience of the agent and the user when ranking candidate responses.

XiaoIce[27] is a social chatbot developed by Microsoft that is extremely popular, especially in China, with over 660 million active users. XiaoIce is designed to be an AI companion that uses emotional understanding to form a bond with the user, while serving a wide range of functionalities. XiaoIce uses a hierarchical dialogue policy to handle both chit-chat conversation and task-oriented requests, an emotional computing module, and a massive proprietary knowledge base. While XiaoIce has been extremely successful at forming long-term relationships with many users[27], the system is extremely complex, consisting of hundreds of individual skills and models built on an enormous amount of propriety data. Previously, Microsoft had also created the infamous Tay bot, which was released on Twitter in 2016, but was removed almost immediately due to inflammatory posts. Tay set a strong example towards the importance of effectively filtering how external information is used for learning, as well as carefully examining candidate responses.

Empathetic Dialogues[28] is a dataset of conversations grounded in emotional situations. Conversations were crowdsourced from 810 Amazon Mechanical Turk (MTurk) workers, who engaged in conversations about how they felt in various emotional scenarios. The resulting dataset comprises 24,850 conversations, divided into approximately 80% train, 10% validation, and 10% test partitions. The authors

trained and evaluated several models on the dataset, including a Transformer-based retrieval architecture[15], retrieval using BERT[16], and an end-to-end Transformer generative model. For retrieval models, all training set utterances are used as candidates. In order to augment the unsupervised models, the authors trained a classifier to predict an emotion label for each utterance, and prepend this label to each utterance before they are encoded, which they refer to as EmoPrepend-1. While the Empathetic Dialogues dataset allows for the creation of more empathetic retrieval-based conversational agents, additional work is required to explicitly model conversation as a shared relational experience beyond emotion-related responses.

Additionally, work led by Justine Cassell at Carnegie Melon has explored the creation of a system for human-agent rapport management[29], as well as incorporating conversational strategies employed by humans into dialogue systems[30]. At Northeastern, Ameneh Shamekhi and Timothy Bickmore have created models for predicting long-user engagement with virtual agents[31], specifically in the context of health interventions.

System Design

The relational dialogue system that I present here is composed of four main stages: Input utterance **Pre-Processing**, followed by **Knowledge Graph Development**, then **Candidate Response Generation**, and finally **Response Selection**. During Pre-Processing, the input utterance from the user is prepared for use by downstream components, and several pieces of information are extracted so that Knowledge Graph Development and Candidate Response Generation can occur. In Knowledge Graph Development, the system's knowledge graph is expanded and modified in order to model both the user and the agent preferences, as well as relevant concepts from Wikidata and their relations. During Candidate Response Generation, many potential responses for the agent to say to the user are generated and gathered, and finally, during Response Selection, a best response is chosen. Figure 1 shows a visualization of the components contained in each stage.

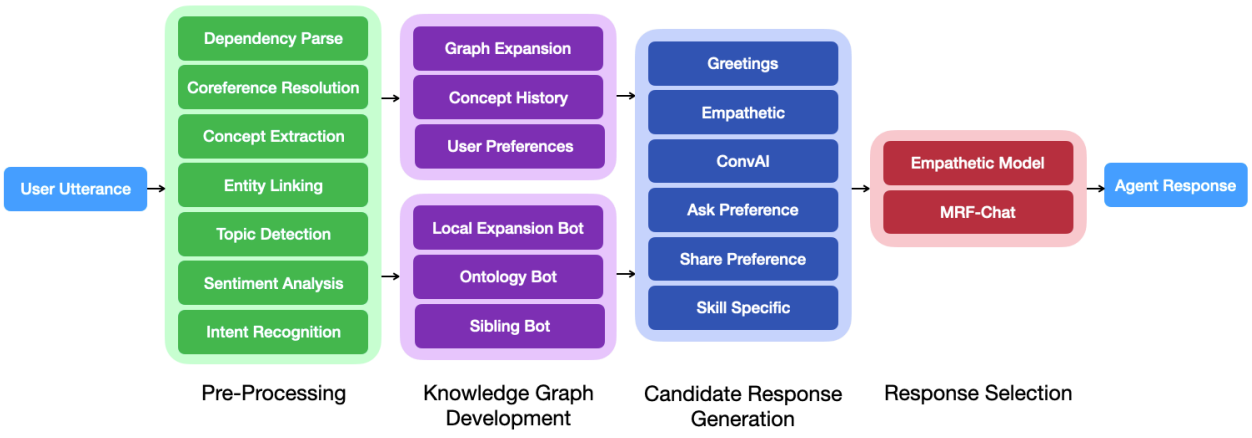


FIGURE 1: OVERVIEW OF ALL SYSTEM COMPONENTS

In this chapter, I give an overview of how each stage fits into the entire system, and how they each function independently. In the following chapter I describe in detail the system's key contribution in response selection, MRF-Chat.

Pre-Processing

Pre-processing sets the foundation for the later stages by extracting the appropriate information from the input utterance. This process occurs in three stages of increasing abstraction:

1. **Coreference Resolution & Dependency Parsing:** Coreference resolution is performed to replace pronouns with their matching nouns from earlier in the utterance, if possible. This modified input is then processed by a SpaCy[32] dependency parser, producing a syntax tree which is later used to for concept extraction and subject-verb-object triple extraction.

2. **Concept Extraction & Entity Linking:** Named entities and non-pronoun noun chunks identified in the dependency parse are compiled into a list of concepts. The matching node in the knowledge graph is found for each concept if it exists, otherwise a Wikidata search will be performed later during the Knowledge Graph Development phase to add the concept.

3. **Topic Detection & Sentiment Analysis:** A list of most relevant nodes in the knowledge graph is compiled, including the nodes representing occurring concepts, as well as other relevant nodes (e.g. the topic “pet” may be listed when “dog” is mentioned). For sentiment analysis, a sentiment score and subjectivity

score are computed for the input utterance. Highly subjective utterances with strong positive or negative sentiment will later be used to infer user preferences.

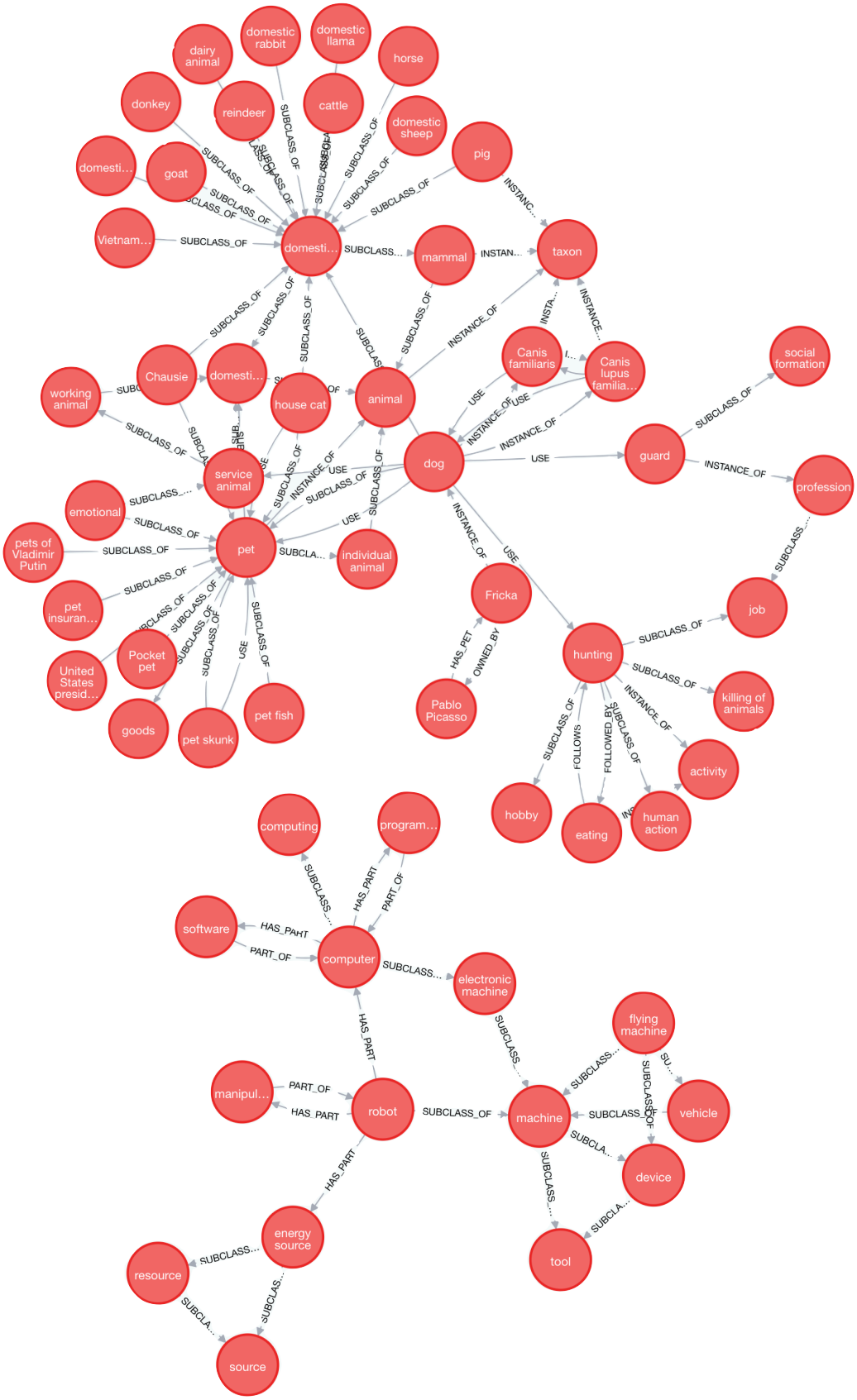
Knowledge Graph Development

In order to have conversations that seek to develop the relationship between the agent and the user, the Relational Dialogue system must be able to form a model of the user, and of the agent itself. Just like in human-human relationships, the agent's knowledge about the user will expand over time, and the agent will seek to discuss topics that are relevant to both itself and the user. Additionally, in order to converse about a wide range of topics in an open-domain dialogue setting, it is essential that the system has access to basic knowledge about the world, and the relationships of common concepts and things. In order to accomplish these goals, the system uses a knowledge graph that contains information about the world from Wikidata, and allows for user and agent models to be build on top of that platform.

Wikidata Knowledge Graph

The core of the knowledge graph is based on Wikidata[33]. Wikidata is a Wikimedia project that seeks to structure the world's knowledge, and contains information about an extremely broad variety of concepts, people, places, and things. Wikidata consists of over ninety-one million entities, which are connected to each other by various types of properties. For example, "dog" has the property "subclass of" to "pet", and "carrot" has "color" to "orange". This structure is easily adapted to a knowledge graph, in which each Wikidata entity is represented by a node, and the properties that connect entities are edges between those nodes. Examples of portions of the Wikidata-based

FIGURE 2: EXAMPLE PORTIONS OF THE WIKIDATA-BASED KNOWLEDGE GRAPH



knowledge graph can be seen in Figure 2. Neo4j[34], a popular graph database platform, is used to support the knowledge graph.

Knowledge Graph Expansion

Since the entire Wikidata knowledge graph is extremely large, only a small subgraph of relevant entities is held. During conversation, relevant information is retrieved from the Wikidata API and is added to the graph. Additionally, several "Graph Bots" run in the background during conversation, strategically expanding the knowledge graph so that relevant information can be used in conversation.

Graph Expansion During Conversation

After the user's utterance is pre-processed, all identified concepts are searched for in Wikidata, and added to the knowledge graph if a Wikidata match is found. First, each extracted concept is searched for by comparing against the names and known aliases for every node in the knowledge graph. If a concept has not already been added, a Wikidata search is done for the concept's text in the user utterance using the Wikidata API. If any matching Wikidata entities are found, the top entity returned from the search is downloaded and added to the knowledge graph as a new node, and each of its properties are added as new edges to placeholder neighboring nodes. The placeholder neighbors will later be downloaded by the Local Expansion Bot (described below).

Graph Bots

While the Relational Dialogue system is running, several Graph Bots run in the background in order to expand the knowledge graph with relevant information that may be useful in conversation:

- **Local Expansion Bot:** Downloads all neighboring entities to nodes that have been mentioned by the user. This ensures that basic information about concepts occurring in conversation is present in the knowledge graph.

- **Ontology Bot:** Many Wikidata entities have "instance of" or “subclass of” properties. For example, “dog” is a subclass of “pet”, and “Virgil” is an instance of “human”. The Ontology Bot downloads these properties up to 4 degrees from all entities mentioned by the user, in order to learn relationships between those entities.

- **Sibling Bot:** For all entities that the user is known to like or dislike (discussed next in User/Agent Modeling), find that entity’s parent entity (to which it has a “subclass of” relationship), and download all entities that also have a “subclass of” relationship to that parent node. For example, “horse” and “cattle” would be downloaded as siblings of “pig”, since they are all subclasses of “domesticated animal”.

User/Agent Modeling

The Wikidata-based knowledge graph provides a platform on which a complex models of both the user and the agent can be built. While there are many ways in which each persona can be modeled, I use a preferences-based model as an initial approach. The likes and dislikes of the user are stored in the knowledge graph by identifying the appropriate Wikidata entities that the user refers to in conversation, and can later be used for driving conversation, or for recommendations (as shown later in Evaluation Setup). In order to build a model of both the user and agent in the knowledge graph, a node is added for the user, as well as for the agent. Edges can later be added to

connect these nodes to other entities in the knowledge graph in order to build the persona model.

User Preference Extraction

During conversation, the user's preferences are extracted through a combination of a very specific rule-based system that matches statements such as "I really like tacos" or "I hate broccoli", as well as a much more generalized approach that combines sentiment and subjectivity analysis. In both systems, the concepts extracted during Pre-Processing are used as candidates for new preferences, and if a preference is found, either a "like" or "dislike" edge is added between the user's node and the appropriate entity node in the knowledge graph.

During Pre-Processing, each sentence in the user's input utterance is analyzed for both sentiment and subjectivity using the Vader Sentiment Intensity Analyzer[35] and the TextBlob Sentiment Subjectivity score[36]. If the sentence has high subjectivity and high sentiment (subjectivity ≥ 0.5, compound sentiment ≥ .05), the associated concepts are considered likes or dislikes based on the positivity/negativity of the sentiment score.

Mentioned Concept Tracking

In addition to preferences, all concepts that are mentioned by the user are recorded in the knowledge graph by adding a "mentioned" edge between the user and the appropriate entity node. This allows for a generating responses about concepts are related to past topics in the knowledge graph, and for lower-fidelity recommendations that are build upon by the user's preferences.

Pre-Loading Agent Model

In order to build a model of the agent's preferences, the same preference extraction procedures can be applied to the agent's utterances. In order to pre-build a model of the agent, existing agent utterances that are designed for its persona can be processed to form that initial model. In the case of an existing conversational agent like Jibo, that has many prompts and responses that have been hand written to use in various scenarios, each written utterance can be treated in the same way as a typical user input, undergoing Pre-Processing, and then User Preference Extraction. This process builds a model of the agent in the knowledge graph in the same way that a user model is built through conversation. In order to provide additional information to the agent model, or to create an agent model when no pre-written agent utterances are present, the agent model can be built by hand. Relationships can be directly added between the agent node and desired entities in the knowledge graph in order to build such a model. Balancing the agent's existing persona with expanding it from selected utterances is a challenging task (as is ensuring that selected responses are consistent with the persona), so in my experiments the agent's persona is not explicitly grown, however this would be valuable future work.

Candidate Response Generation

Once all pre-processing and knowledge graph development is done, many candidate responses are chosen/generated, one of which will later be chosen to use during the Response Selection phase. A base pool of candidates are chosen using a retrieval approach inspired by BYU-EVE's scattershot algorithm[25], and the pool is expanded to include responses that are generated using relevant contextual topics identified in pre-

processing, as well as information encoded in the knowledge graph. Additionally, the system includes a rule-based generation module for handling the semi-structured conversation framework used in evaluation (described later in Evaluation Setup), as well as a module specifically designed to handle greetings and initial pleasantries such as “How are you?”, which are typically very challenging for neural-only approaches to handle effectively.

Retrieval-Based Candidates

Scattershot Candidate Selection

Since scoring candidates during Response Selection is very computationally expensive, it is essential to limit the number of candidates as much as possible in order to minimize latency, while still maintaining a high quality of responses. For BYU-EVE, Fulda et al.[25] proposed the Scattershot algorithm for response prioritization (a component of candidate response selection). Given a corpus of conversations represented as a set of utterance-response pairs, and an utterance embedding function that converts a text utterance into a dense vector representing a point in a high-dimensional space, Fulda et al.’s Scattershot algorithm scores a potential response to an input utterance by first finding the top n utterances in the corpus that are most similar to the input utterance (similarity is computed as the cosine similarity of the input and reference utterance’s embeddings), and then using the embeddings of the actual responses to those top utterances as reference points. The candidate response that is closest to any of the reference embeddings is given the highest score. This approach is very lightweight as embeddings of utterances in the corpus can be computed once and cached for use during later conversation, although it is fairly naive

compared to typical neural language models that recently have shown the most success for response selection[37].

In order to leverage the original Scattershot algorithm's strength as a lightweight heuristic for response scoring, while still getting the benefits of typical neural language models, I use a hybrid approach. During the Candidate Response Generation phase, a modified version of the Scattershot algorithm is used to down-select potential candidates that are taken from the corpus, instead of scoring novel ones based on the corpus' conversations. I refer to this process as Scattershot Candidate Selection. Later, during the Response Selection phase, the retrieved candidates, alongside additional generated ones, will be scored. The Scattershot Candidate Selection algorithm uses the actual responses to the utterances that are the most similar to the input as candidates: Given a corpus of conversations D, represented as the set of utterance-response pairs $D_i = (D_{i,utterance}, D_{i,response})$, an utterance embedding function $embed()$ that converts a text utterance into a dense vector representing a point in a high-dimensional space, and an input utterance U:

1. Compute the similarity score S_i between U and $D_{i,utterance}$ for each $D_i \in D$, where similarity is computed as the cosine similarity between $embed(U)$ and $embed(D_{i,utterance})$

2. Find the list of most-similar n corpus examples $D_{j_1}, D_{j_1}, \ldots, D_{j_n}$, with the greatest similarity scores $S_{j_1}, S_{j_1}, \ldots, S_{j_n}$

3. The list of candidates is then the corresponding responses of the top corpus examples: $D_{j_1,response}, D_{j_1,response}, \ldots, D_{j_n,response}$

This approach allows for a high-quality pool of candidate responses to be rapidly found from the corpus of examples, while allowing for better response selection to happen later through a more sophisticated process. The Relational Dialogue system uses Google's Universal Sentence Encoder[38] as the embedding function, which allows for high-quality embeddings over a wide range of input utterance lengths. The embeddings of all utterances in the example corpora to be pre-computed and cached, so Scattershot Candidate Selection can be done with minimal latency (100,000 potential candidates can be searched in around 70ms on CPU).

Corpora for Retrieving Candidates

In my experiments, both the Empathetic Dialogues dataset[28] and the ConvAI dataset[39] are used for sourcing response candidates. Between the two corpora there are 96,557 utterance-response pairs (64,636 from Empathetic Dialogues, 31,921 from ConvAI). The ConvAI dataset provides a wide platform of open discussions on many topics, which the Empathetic Dialogues dataset complements with conversations about emotional experiences. The top 20 candidates are taken from each corpus using the Scattershot Candidate Selection algorithm to form the overall set of retrieval-based candidates. Using 20 candidates from each dataset was empirically found to provide an optimal balance of response quality and Response Selection latency.

Knowledge Graph Candidates

By using the existing user and agent models in the knowledge graph, and working to expand them and explore potential common ground, knowledge graph generators seek to create conversation that develops the relationship and mutual knowledge between the user and conversational agent. In the current Relational Dialogue system, there are two key knowledge graph generators: the Ask Preference Generator, and the Share Preference Generator, which seek to learn more about the user's preferences, and share the agent's preferences with the user. There are many other ways that the knowledge graph could be used to drive conversation towards these goals, I discuss several potential additions later on in the Discussion & Future Work.

Ask Preference Generator

The Ask Preference Generator uses existing knowledge about the user's likes and dislikes, along with the topic of conversation, to ask the user about their preferences around similar concepts. For example, if it is known that the user likes eating beef, a possible generated utterance would be: "Since you like beef, do you also like lamb?". Similarly, if the user likes swimming and sports are being discussed, a possible utterance would be: "I was wondering, do you like biking?". In order to identify concepts to ask about, the Ask Preference Generator uses the following algorithm, given the user's known likes/dislikes from the knowledge graph, as well as the current topic identified in Pre-Processing:

1. Identify the 3 most relevant liked/disliked concepts to the conversation, computed as the cosine similarity of their Universal Sentence Encoder encodings with that of

the current topic concept. Only include concepts that have high (greater than 0.5) cosine similarity with the topic.

2. For each of those concepts, find their children and sibling nodes (a child node has relationship “Subclass Of” to the concept’s node, and a sibling node and the concept node both have a “Subclass Of” relationship to some parent node)

3. For each concept node and up to 5 of the most relevant sibling and child nodes, generate a candidate response

Once the list of liked/disliked concepts and their related concepts to ask about has be compiled, responses are generated using a complex template that allows for a wide variety of responses to be generated, each of which asks the user about their preferences around the new concept.

Share Preference Generator

The Share Preference Generator works similarly to the Ask Preference Generator, except it instead shares the agent’s preferences of concepts that are related to the topic of conversation. Given the agent’s known likes/dislikes from the knowledge graph, as well as the current topic identified in Pre-Processing:

1. Identify most relevant liked/disliked concepts to the conversation, computed as the cosine similarity of their Universal Sentence Encoder encodings with that of the current topic concept. Only include concepts that have high (greater than 0.5) cosine similarity with the topic.

2. For each relevant liked/disliked concept generate a candidate response

Candidate utterances are generated in a similar template-based method to the Ask Preference Generator.

Response Selection

Once all of the candidate responses have been chosen, the final step is to select a best response to say to the user. This is done with a combination of a BERT language model and MRF-Chat, a novel algorithm for improving response selection by modeling the mutual knowledge of the agent and the user, as well as the contextual relevance of concepts that occur in candidate responses.

The candidate responses are first scored by a BERT-based model that was trained on the Empathetic Dialogues dataset. The model takes the user utterance and a candidate response as input, and outputs a score for the likelihood that the candidate is a good response to the user input.

The candidate responses are then also scored by MRF-Chat. During conversation, MRF-Chat builds a semantic graph of concepts occurring in conversation, as well as other relevant concepts, and models both the user's and agent's knowledge of those concepts, as well as each concept's relevance to the conversation. Similarly to the BERT model, MRF-Chat produces a likelihood score for each candidate based on conversational history. MRF-Chat's motivation, design, and independent evaluation are described in detail later in the MRF-Chat chapter.

Finally, the best response is chosen by multiplying each candidate's score from the BERT language model with its score from MRF-Chat, and then the candidate response

with the greatest final score is used. Candidates that are duplicates of utterances that previously occurred in the current conversation, as well as candidates with only three or fewer words, are never selected.

MRF-Chat

I completed the MRF-Chat project working alongside Ishaan Grover.

With advances in deep learning, the natural language understanding community has seen a recent proliferation of open-domain dialog systems[42,43], as well as several competitions such as the Amazon Alexa Prize[40] and ConvAI[41]. Most existing state-of-the-art approaches include training end-to-end models (often on very specific datasets) that account for various conversational features such as repetition, specificity[44], emotional content of response[28], knowledge graphs of related entities[45] and even the persona of the agent[37] when retrieving a response utterance from a set of candidate utterances. While deep-learning methods have shown reasonable results, the problem of open-domain dialog conversations is far from solved. In natural conversations, humans often view each other as cognitive agents. Thus, it is important for conversational agents to explicitly model the cognitive processes humans use when conversing with each other.

The cognitive theory of mutual knowledge posits that speakers and listeners maintain mental models of the knowledge and beliefs they share with each other to find common ground for communication[46,47]. It follows that in a two-person conversation, each speaker maintains both a model of their partner's knowledge, and a model of the knowledge they have communicated to their partner. These models provide information about about their mutual knowledge and help in deciding the next utterance. Further, as the conversation continues, each speaker updates their mental models as they gain

new information from and provide new information to their partner. Consider the following hypothetical conversation:

Speaker 1: Did you see the new Avenger's movie? (U1)

Speaker 2: Yes, I loved Thor's character in it. (U2)

Speaker 1: Me too! Do you watch superhero movies? (U3)

With U1, Speaker 1 offers "Avengers", "Movie" and related concepts as common ground (context) for the conversation. Speaker 2 realizes that Speaker 1 knows about Avengers, so they must also know about the related concept "Thor". Thus, Speaker 2 says U2 and offers "Thor" and related concepts as common ground. Now, Speaker 1 realizes from U1 and U2 that the concept "superhero movies" has the highest "mutual knowledge" and says U3.

Along with mutual knowledge, humans also account for contextual relevance of concepts used as the conversation flows from one topic to another. That is, even though a concept may be familiar to both interlocutors at one point during the conversation, it may not remain relevant when they discuss another topic.

While the theory of mutual knowledge forms the basis of grounding in conversation and contextual relevance plays a vital role in human conversations, to the best of our knowledge, there hasn't been an attempt to create a algorithm that explicitly models these processes. To this end, we propose a novel probabilistic approach using Markov Random Fields (MRF) to augment existing deep-learning methods for improved next

utterance prediction. In the rest of this paper, we refer to deep-learning models as base models and refer to our algorithm as MRF-Chat.

The first step towards building such an algorithm is to accurately make inferences about a person's knowledge from partial information. We draw inspiration from work by Grover et al.[48] and formulate this task as an inference problem on Markov Random Fields (MRF). For each speaker, inference on the MRF after observing known concepts gives the conditional marginal probabilities of knowing any concept in the graph. We provide further details of this work in the Background section. For next utterance predictions, we use inferences from the MRF with predictions from the base model to get the combined probability of each candidate being the next utterance given the context of the conversation.

The primary contribution of our work in this section is an algorithm (MRF-Chat) to improve open-domain conversational agents based on the cognitive theory of mutual knowledge. Our algorithm incorporates contextual relevance of all prior concepts to make predictions. It is domain agnostic and independent of the base model used. Using both, human and automatic evaluation methods, we show that MRF-Chat when used with an existing state-of-the-art model[37] significantly improves performance.

The rest of this section is organized as follows. We first give relevant related work and preliminaries essential to our problem. Next, we provide the central algorithm MRF-Chat. We then discuss the details of our experimental setup and evaluation methodology. We finally present experimental results and discussion.

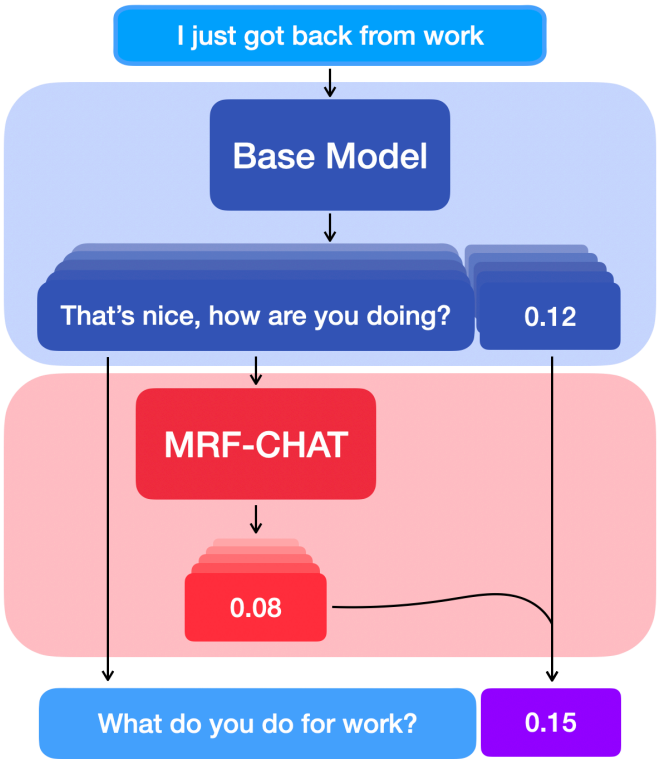


FIGURE 3: INTEGRATION OF MRF-CHAT AND A BASE MODEL. WHEN THE USER SAYS AN UTTERANCE, THE BASE MODEL PRODUCES A PROBABILITY FOR EACH CANDIDATE RESPONSE UTTERANCE. MRF-CHAT ALSO GENERATES NEW PROBABILITIES FOR EACH CANDIDATE. THESE PROBABILITIES ARE COMBINED TO CHOOSE THE FINAL RESPONSE.

Preliminaries

A Markov Random Field (MRF) is an undirected graphical model of a joint distribution.

It is specified by a graph $G = (V, E)$ and a set of random variables E where

$X = \{X_1, X_2, X_3 \ldots X_n\}$ correspond to vertices $V = \{v_1, v_2, v_3 \ldots v_n\}$. An edge e_{ij}

between nodes X_i and X_j captures interactions between nodes. These interactions are

defined by potential functions $\phi(\mathrm{x})$ which are often represented as energy functions

and then transformed into probabilities by adopting Gibbs distribution. Thus, for a

given MRF:

$$P(X_1, X_2 \ldots X_n) = \frac{1}{Z} \prod_{C} \phi_c(\mathrm{x}_c) \qquad (1)$$

$$Z = \sum_{\mathrm{x}} \prod_{C} \phi_c(\mathrm{x}_c) \qquad (2)$$

$$\phi_c(\mathrm{x}_c) = e^{-E(\mathrm{x}_c)} \qquad (3)$$

where:

- C is the set of all maximal cliques
- $\phi_c(\mathrm{x}_c)$ is the potential function for clique c
- $E(\mathrm{x}_c)$is the energy function for clique c
- $\phi_c(\mathrm{x}_c) \geq 0$
- Z is the partition function

Inference on MRF gives the marginal probabilities of each node. While exact inference on MRFs is computationally intractable, several approximation methods such as Belief Propagation and Markov Chain Monte Carlo (MCMC) are often used in practice.

Background

Recently, Grover et al.[48] experimentally validated and presented a model for predicting children's vocabulary from partial information of their existing vocabulary knowledge.

They based their model on assumptions based on the psycholinguistic theory of semantic learning which states that humans learn new words by forming semantic associations with words they already know. The fundamental assumption made by the model was that if it is observed that a child knows a given word, the child must have learned the word by forming semantic associations with words they already knew. Thus, it is likely that the child knows words that are semantically related to the given word.

The salient features and steps for model construction are as follows:

- Build a semantic network where nodes represent words and edges represent relationships between those words. The semantic network built using GloVe word vectors[49] by making pairwise comparisons between nodes and adding an edge if the cosine similarity is above a certain threshold.

- Construct an MRF factor graph corresponding to the semantic network. The nodes of the MRF represent the probability of knowing concepts and the pairwise potential functions (refer to P1: Mutual Knowledge) represent how each node influences its neighbors.

- Observe existing knowledge and perform inference to find conditional marginal probabilities of all the nodes in the graph.

While the authors used the model for predicting vocabulary knowledge of children for their particular use case, we argue that the same fundamental assumptions of learning

words/concepts apply to adults as well and the model is applicable in our problem setup.

Related Work

In recent years, great progress has been made in dialogue systems. Many traditional dialogue systems were purpose-built for specific tasks[50], and are made up of various components, such as dialogue state management, intent recognition and slot filling modules[51], as well as specific response generators to handle different scenarios. These goal-oriented systems move through specific well-defined conversational states in order to reach an end goal, e.g. retrieving information about a flight's arrival time, or booking a train ticket. With the advent of consumer voice assistants such as Amazon's Alexa, Google Home, and the social robot Jibo, these specialized dialogue approaches have been expanded into much larger systems, providing the user with many individual skills, each of which are built for a specific task or experience, each with their own dialogue state management, intent recognition, response generation, etc. While these systems are providing increasingly complex user experiences, they are generally unable to engage in chit-chat or other general, non-goal oriented, dialogue.

On the other hand, open-domain dialogue systems are designed to engage in casual conversation, without a specific end goal. Perhaps the oldest well-known example is ELIZA[1], which used a combination of pattern matching and careful substitution to emulate a psychotherapist. While today's open-domain dialogue systems are much more sophisticated, they often still leverage rule-based parsing and response templates to augment more advanced techniques[52,53]. The best examples of such

systems can be found in the Alexa Prize competition[20], a university competition hosted by Amazon where selected university teams build conversational agents, referred to as “socialbots”. The goal of the competition is to create engaging and coherent dialog with humans on a variety of topics, including sports, entertainment, and politics.

Outside of the Alexa Prize, most open-domain dialogue systems are either retrieval models that select a response based on conversational history from a large set of candidates, or generative models that produce novel responses. In this work, we focus on augmenting one key retrieval approach, the Key-Value Memory Net model trained on Persona-Chat by Zhang et al.[37]. This model is available pre-trained and open source[2].

Persona-Chat

The Persona-Chat dataset[37] is a crowd-sourced dataset of conversations where each speaker bases their responses on a given persona. The dataset consists of 162,064 utterances over 10,907 dialogues. 1000 dialogues are set aside for validation, and 968 are set aside for test. After the dataset was collected, the authors trained and evaluated several retrieval and generative models on the corpus. At the start of each conversation, the chosen model is conditioned on either the user's persona, the agent's own persona, both personas, or neither. The best performing model was a key-Value Memory Network[54] that performs attention over the training dialogue histories and personas in order to predict the best response. In our experiments, we use their

[2] https://github.com/facebookresearch/ParlAI/tree/personachat/projects/personachat

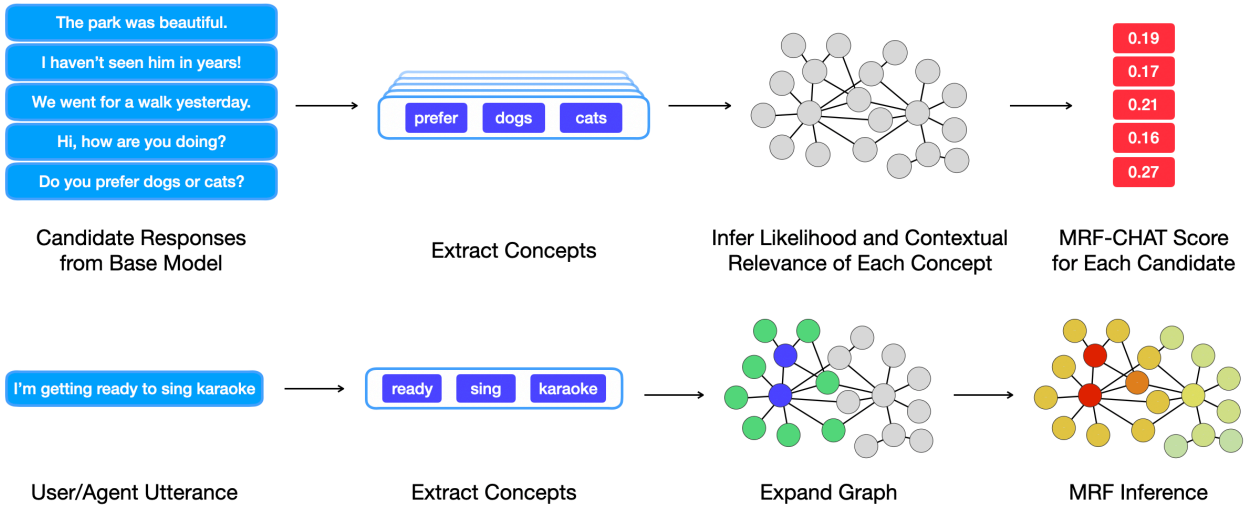


FIGURE 4: MRF-CHAT PIPELINE FOR NEXT UTTERANCE PREDICTION

pre-trained Key-Value Memory Net model trained on Persona-Chat, which we refer to as the base model.

The MRF-Chat Algorithm

We consider the setting where a user and agent take turns interacting with each other. Given a base model, we wish to incorporate the following salient features into the conversational agent.

- **P1:** The agent should account for mutual knowledge. That is, the agent should choose utterances (from a set of candidate utterances) such that both, the agent and the user have knowledge about the concepts used in those utterances (common ground).

- **P2:** The agent should account for contextual relevance of concepts being used in the conversation at any given time. That is, the agent should appropriately discount

the mutual knowledge of a concept if it is not relevant to the current conversation (even if it was relevant earlier).

More formally, for a prior utterance said by the agent U_{agent}, a prior utterance said by the user U_{user}, and a set of candidate utterances $U_{candidates}$ and a base model M, we want to select a response utterance $U_{response} \in U_{candidates}$ for the agent such that it satisfies **P1** and **P2**.

We now discuss the steps for solving **P1** and **P2** separately and then discuss a method to combine them to generate a salient response.

P1: Mutual Knowledge

Formally, we define mutual knowledge of a concept as the probability that both the agent and user know the given concept given the concepts they have used in their respective utterances. The steps for P1 (shown in Figure 4) are first broadly outlined and subsequently detailed follows.

1. Extract relevant concepts from U_{agent}, U_{user} and $U_{candidates}$.

2. Create a semantic graph containing all the extracted concepts from U_{agent}, U_{user}, $U_{candidates}$ as well as all concepts semantically related to them.

3. Construct a corresponding MRF and perform inference first using extracted concepts from U_{agent} as positive observations, then using extracted concepts from U_{user} as positive observations. This step provides the individual marginal probabilities of the agent and user knowing the concepts (Figure 5).

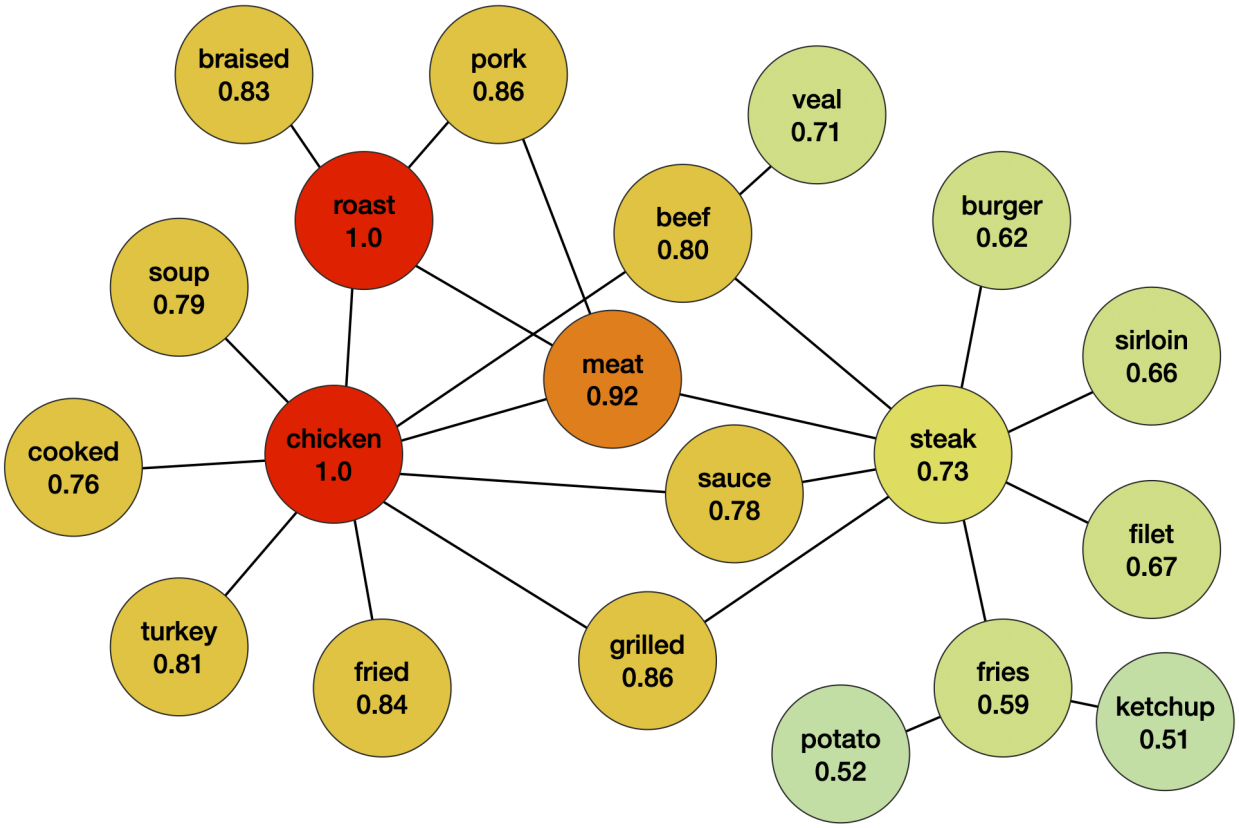


FIGURE 5: EXAMPLE OF INFERENCE ON MRF. RED NODES CORRESPOND TO POSITIVE OBSERVATIONS. PROBABILITIES OF KNOWING A NODE ARE SHOWN BELOW CONCEPT NAMES WITHIN THE NODES.

4. Compute the joint distribution of both the agent and user knowing a given concept.

Concept Extraction

The first step in processing an input utterance, U_{agent}, U_{user} or $U_{candidates}$, is to extract the relevant concepts. For example, given the utterance "I love dogs, even though I'm allergic to them", the concepts "love", "dogs", and "allergic" should be identified. In order to achieve this task, we use Yake[55], which is an open-source keyword extraction tool that provides state-of-the-art performance in a lightweight package. Given an utterance, Yake returns a list of keywords, each with a corresponding relevance score $r_{c,yake} \in [0,1]$ (since Yake scores closer to 0 indicate higher relevance, we use

$r_c = 1 - r_{c,yake}$ instead). We also experimented with RAKE[56] keyword extraction and well as spaCy[32] noun chunks, but we found Yake to be more effective for our purposes.

Thus, using the concept extraction module, we have concepts

- C_{agent} from U_{agent}
- C_{user} from U_{user}
- $C_{candidates}$ from $U_{candidates}$
- Set of all concepts $C = C_{agent} \cup C_{user} \cup C_{candidates}$

It is important to note that there exist many strategies to generate the set $U_{candidates}$ and any reasonable strategy is usable. Since our task is to improve M using MRF-Chat, in our experiments we use top k responses from M, $U_{topK} \subseteq U_{candidates}$ instead of the entire candidate set. This is done primarily for computational efficiency in running our experiments. In practice, we found that it was rarely the case where MRF-chat would find $U_{response} \notin U_{topK}$. Increasing the value of k allows for more candidates to be considered, but at the cost of latency.

Building the Semantic Network

The core component of MRF-Chat is a semantic graph, where the nodes represent individual concepts and edges represent semantic relationship between concepts. A common measure of semantic distance between two words is to take the cosine

distance between their word embeddings. Since our concepts are single words, we define the semantic distance between two concepts as the cosine distance between their pre-trained common crawl GloVe word vectors (300 dimensional)[49]. Two words with vector representations v_1 and v_2 are said to be semantically similar if $cos(v_1, v_2) \geq \epsilon$.

In order to build a semantic network, we further find a set of all semantically similar concepts $C_{similar}$ such that $c_{i,similar}$ is in the vocabulary of Common crawl word vectors. We now build a semantic graph $G_{semantics} = (V, E)$ such that $V = C \cup C_{similar}$. We perform pairwise comparisons between nodes in V and add an edge between all pairs of nodes that are semantically similar. In order for the graph to represent real-world knowledge of salient concepts, we exclude very frequent words such as "yes", "me", "what", etc. We use word frequencies from the SUBTLEX-US database[57], excluding words with a Zipf value of less than 5.75 based on empirical trials.

As the conversation continues, and the algorithm processes new utterances from the user and agent, graph G is always *augmented* with new concepts from the immediately preceding user user-agent utterance pair rather than *re-computed* for it. In this way, the algorithm expands its graph of concepts it considers as the conversation continues.

Inference on the MRF

The end goal of this step is to construct an MRF $G_{mrf} = (V_{mrf}, E_{mrf})$ and its corresponding factor graph from some semantic graph $G_r = (V_r, E_r)$ and perform

inference on it. To construct the MRF, we use the same method and potential functions (given below) as Grover et al.[48].

$$\phi(X_i, X_j) = \begin{bmatrix} e^{-(1-s(w_i,w_j))} & e^{-s(w_i,w_j)} \\ e^{-s(w_i,w_j)} & e^{-(1-s(w_i,w_j))} \end{bmatrix} \quad (4)$$

where $s(w_i, w_j)$ is the cosine distance between the word embeddings corresponding to w_i and w_j.

Thus, we know that $|V_{mrf}| = |V_r|$ and $|E_{mrf}| = |E_r|$. Inference on extremely large factor graphs can be computationally infeasible in terms of run-times for real-time conversational agents. To avoid such problems, we construct a local subgraph $G_{local} = (V_{local}, E_{local})$ of $G_{semantics}$ which only includes concepts (and related concepts) from the preceding agent-user utterance pair. That is, $V_{local} = C$. Therefore, G_{mrf} is constructed from G_{local}. Every node in G_{mrf} is mapped to a Bernoulli random variable that represents the probability of the user/agent knowing a particular concept (depending on the observed nodes).

Now, we first perform inference on G_{mrf} using concepts in C_{agent}. This gives the conditional marginal probability of the agents knowledge of each concept given the concepts used in the agents previous utterance. We also perform inference on G_{mrf} using concepts in C_{user} to get conditional marginal probability of the users knowledge of each concept given the concepts used in their previous utterance. Thus, for a given concept, let X_{agent} be a random variable that represents the probability that the agent

knows a given node and X_{user} represent the probability that the user knows a given concept. Then, inference, we have $P(X_{agent} \mid C_{agent})$ and $P(X_{user} \mid C_{user}$, and we wish to find $P(X_{agent}, X_{user} \mid C_{agent}, C_{user})$. Here, we make the simplifying assumption that the agent's and user's knowledge of a concept are conditionally independent of concepts used in each other's utterances (that is, they generate utterances based on the knowledge they had prior to the conversation) in a given agent-user utterance pair. Now, we have the joint distribution,

$$P(X_{agent}, X_{user} \mid C_{agent}, C_{user}) = P(X_{agent} \mid C_{agent})P(X_{user} \mid C_{user}) \quad (5)$$

We now define a define a Bernoulli random variable X_{mutual} for each concept representing the probability that both the user and the agent know the given concept (Mutual knowledge).

P1: Contextual Relevance

As an agent and user interact for multiple turns, the contextual relevance of each concept will vary in time. For example, if the user and agent discuss "superhero movies" initially but then go on to talk about their favorite desserts, the contextual relevance of "superhero movies" decreases with time. This relevance decays with each pair of turns as they discuss other topics. To capture this notion of relevance in time, we define contextual relevance of a concept as the mixture of distributions of all previous $P(X_{mutual})$ from the MRF where the weight for each distribution is

exponentially decayed with every pair of turns of the conversation[3]. Let mutual knowledge of concept c be $P(X_i)$ in the i^{th} turn and let R_n be a random variable representing the contextual relevance after the n^{th} turn, then:

$$P(R_n) = \frac{1}{Z} \sum_{i=1}^{n} \lambda^{n-i} P(X_i) \qquad (6)$$

where Z is the normalizing constant and $\lambda \in [0,1]$ defines the rate at which we decay the weights. A higher λ lowers the rate of decay and results in weighting prior mutual knowledge more heavily.

From Contextual Relevance to Next Utterance Probabilities

Up until now, we have found the contextual relevance of each key concept while taking into account mutual knowledge of both, the agent and the user. However, we wish to find the probability of each candidate utterance $u \in U_{candidates}$ being a salient next utterance.

In order for the agent to optimize the contextual relevance of concepts across the response utterance, it is essential to reward the presence of concepts that are believed to be shared knowledge between the agent and the user, and to also penalize the presence of concepts that are believed to not be shared. Likewise, concepts that the agent believes are neutral, i.e. neither more relevant or less relevant than all other

[3] Note that we index from the 0th turn and assume that the agent started the conversation. In this case, the first inference occurs after the first pair of turns. In case the user starts the conversation, there is no notion of mutual knowledge in the 0th turn, so we only account for the users knowledge and start accounting for the agent's knowledge from the 0th turn.

concepts, should have no effect on the response selection. In order to achieve this, for each concept c with contextual relevance $X_{c,mutual}$, we assign a score equal to $E[X_{c,mutual}]$[4]. Further, find the mean of all scores, μ of nodes in G_{local}. Now, for each candidate utterance $u \in U_{candidates}$, having concepts $c_1, c_2, c_3 \ldots c_n$ with expectations $\mu_1, \mu_2, \mu_3 \ldots \mu_n$ and yake concept relevance scores $r_{1,yake}, r_{2,yake}, r_{3,yake} \ldots r_{n,yake}$, the final score for the utterance is given by:

$$score = \frac{1}{n} \sum_{i=1}^{n} r_{i,yake}(\mu_i - \mu) \qquad (7)$$

This means that every additional concept that is believed to be mutual knowledge increases the candidate's score, each concept that is considered of average likelihood does not affect the score, and each concept that is believed to be below average likelihood reduces the score. Further each concept is weighed by its relevance in the utterance. Given final scores for each utterance, we find the final probability of each utterance being the next salient utterance according to MRF-chat by applying softmax normalization.

Combining Predictions (Augmenting with MRF-Chat)

We wish to estimate the probability of the utterance u being the next utterance, and have two separate models MRF-Chat and M (deep-learning model) that estimate this probability. Thus, we have $P(u \mid MRF - Chat)$ and $P(u \mid M)$ and want to find

[4] Since $X_{c,mutual}$ is Bernoulli, the expectation is just $P(X_{c,mutual}) = 1)$

$P(u|MRF-Chat,M)$. Assuming the two models to be conditionally independent, the bayes optimal method to combine these distributions is given by[58]:

$$P(u|MRF-chat,M) \propto P(u|MRF-chat)P(u|M) \quad (8)$$

Experimental Details

For our experimental setup, we consider two models: 1) The base model[37] and 2) The base model augmented with MRF-Chat. For each conversation in the Persona-Chat test set, we used the first 4 turns as context and processed each utterance as described in “The MRF-Chat Algorithm” section in order to update marginal probabilities and priors after each pair of turns (2 updates in total). We then produced a response to follow as the next utterance in the conversation using both MRF-Chat augmented with base model and the base model alone. For our ablation study, we repeat this response generation process for each value of the decay factor $\lambda \in \{0,0.3,0.6\}$ ($\lambda = 0$ means that we only consider the most recent user-agent utterance pair and ignore the previous turn). We exclude conversations in which MRF-Chat and the base model selected the same response from human ratings (the number of excluded conversations for each value of λ is reported in Table 1).

In our experiments, we use the top ten scoring candidates from the base model (top $K = 10$). For the semantic graph, we manually tested different values of semantic similarity on different words, we set $\epsilon = 0.6$. In order to improve latency without affecting performance, we only consider the 100,000 most common words from GloVe[49]. Further, we use GloVe vectors to build the graph instead of other publicly

available semantic graphs like ConceptNet[59], Wikidata[33], etc. due to the ability of computing semantic similarity between any given pair of words.

Evaluation Methods

When evaluating our algorithm, we primarily focus on human ratings, as automated metrics for often poor for evaluating dialogue[60]. In order to collect human ratings, we ran crowdsourcing tasks on Amazon Mechanical Turk (MTurk) as shown in Figure 6. MTurk workers then rated each conversation, comparing the MRF-Chat's (augmented with the base model) response with the base model's response. Each conversation is rated once for each unique response produced by MRF-Chat (if two values of λ produce the same response for a conversation, that response is only rated a single time to avoid redundancy).

Inspired by the Acute-Eval setup[61], we show each worker the conversational context and ask them to choose the "better" response between the MRF-Chat response and the base model's response. Specifically, we ask which response is better based on the conversation, and which response is more on topic. For both questions, we a use four-point scale, with labels "Response 1 is much better", "Response 1 is slightly better", "Response 2 is slightly better", and "Response 2 is much better". In our results, we look at the distribution of responses across the four options, as well as simplified into the binary categories of either Win/Loss for MRF-Chat against the base. We also ask the worker to briefly justify their answers, which we found increases the quality of ratings. In addition to asking the worker for justification, we require that they are located in the United States or United Kingdom (since all of our conversations are in

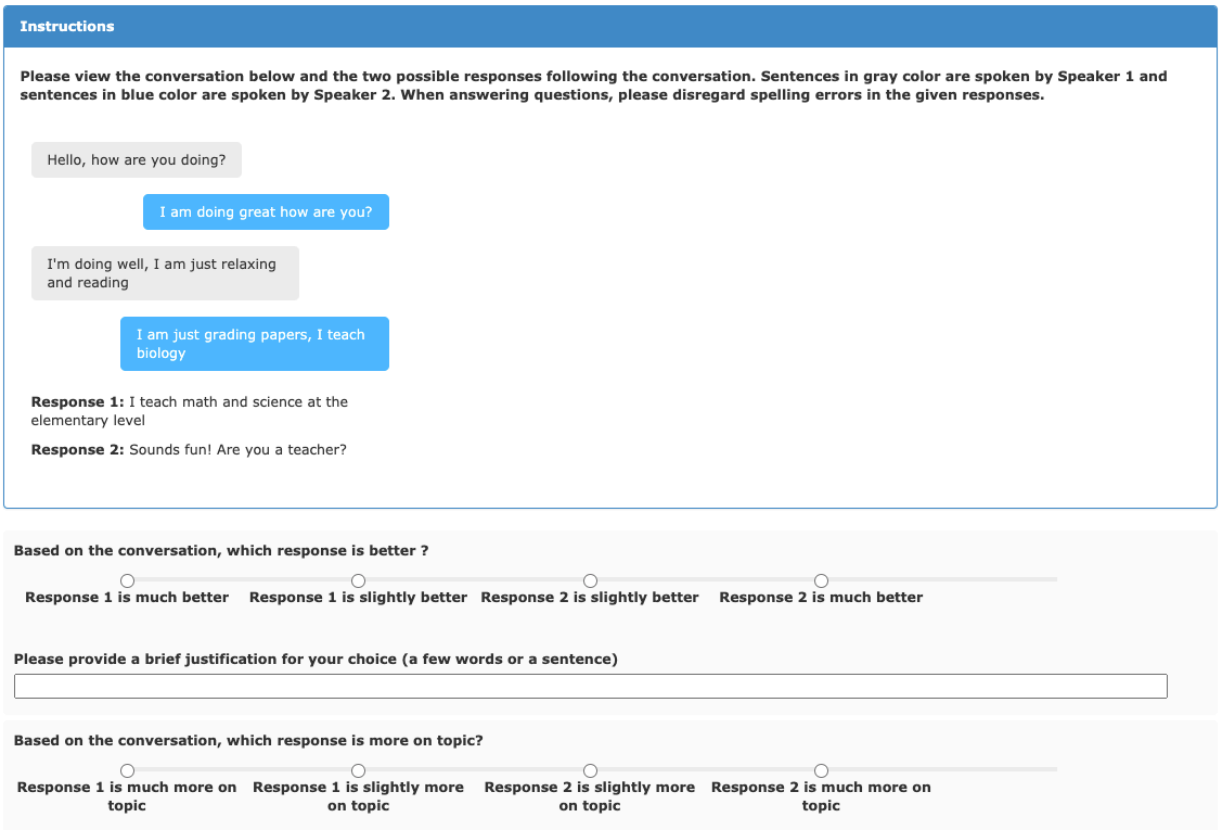


FIGURE 6: OUR MECHANICAL TURK HUMAN EVALUATION SETUP. WORKERS READ FOUR TURNS OF CONVERSATIONAL HISTORY, AND THEN ARE ASKED TO COMPARE THE RESPONSE CHOSEN BY THE BASE MODEL ALONE TO THE RESPONSE FROM MRF-CHAT.

English). These design decisions helped to ensure that we received high-quality annotations.

In addition to human ratings, we compute BLEU scores[62], comparing both the base model and MRF-Chat's response against the actual response in the test conversation, following previous work in dialogue[63]. We measure the mean length, in words, across all responses from the base model and MRF-Chat for each value of λ, as well as the number of concepts extracted by Yake[55]. Increased utterance length has been used to gain insight into the performance of dialogue systems in the past[39], but measuring the number of concepts present in utterances is a less typical approach that allows us to

not only gain insight into the length of our selected utterances, but also into the density of unique concepts within them.

Results

The results from our human evaluation can be found in Table 1. The number of wins and losses for MRF-Chat (augmented with base model) against the base KV Memory model are displayed, along with the p-values for each result. For reference, the number of conversations in which MRF-Chat produced the same response as the base model are included. MRF-Chat outperformed the base model significantly $(p < 0.001)$ on both questions, for all three values of decay factor λ. That is, human annotators believe that the response from MRF-chat (augmented with base model) were better and more on-topic.

As λ increases, we see a slight decrease in the p-value of MRF-Chat's improvement over the base model, but still outperforming it significantly. This is surprising, as a higher value of λ means that older conversation history is being weighed more heavily

Dialogue System	Q1: Better		Q1: On Topic		Equivalent Responses
	Win/Loss	p-value	Win/Loss	p-value	
MRF-Chat ($\lambda = 0$)	**335**/220	< .000001	**334**/217	< .000001	413
MRF-Chat ($\lambda = 0.3$)	**370**/279	0.000206	**368**/276	0.000168	319
MRF-Chat ($\lambda = 0.6$)	**364**/278	0.000397	**362**/276	0.000382	325

TABLE 1: HUMAN EVALUATION RESULTS: WIN/LOSS COUNT FOR MRF-CHAT AGAINST THE KV MEMORY NETWORK ALONE. MRF-CHAT CONSISTENTLY IMPROVES HUMAN RATINGS. ALL RESULTS ARE SIGNIFICANT ($p < 0.001$).

Dialogue System	Average BLEU
KV Memory	4.44
MRF-Chat ($\lambda = 0$)	4.54 (4.70)
MRF-Chat ($\lambda = 0.3$)	4.56 (4.69)
MRF-Chat ($\lambda = 0.6$)	4.56 (4.69)

TABLE 2: AVERAGE OF BLEU-1,-2,-3,-4 SCORES FOR EACH APPROACH. VALUES IN PARENTHESIS ARE COMPUTED USING ONLY THE MRF-CHAT RESPONSES THAT WERE DIFFERENT THAN THOSE FROM THE KV MEMORY MODEL.

Dialogue System	Utterance Length	Utterance Concepts
KV Memory	11.72 ± 3.42	4.46 ± 1.82
MRF-Chat ($\lambda = 0$)	11.12 ± 3.44 (10.65 ± 3.39)	4.07 ± 1.72 (3.77 ± 1.62)
MRF-Chat ($\lambda = 0.3$)	11.09 ± 3.39 (10.83 ± 3.33)	4.05 ± 1.65 (3.88 ± 1.58)
MRF-Chat ($\lambda = 0.6$)	11.08 ± 3.40 (10.85 ± 3.35)	4.05 ± 1.65 (3.90 ± 1.58)

TABLE 3: MEAN UTTERANCE LENGTH, IN WORDS, AND MEAN NUMBER OF CONCEPTS EXTRACTED BY YAKE. VALUES IN PARENTHESIS ARE COMPUTED USING ONLY THE MRF-CHAT RESPONSES THAT WERE DIFFERENT THAN THOSE FROM THE KV MEMORY MODEL.

in comparison to the last user and agent utterance, and one may expect that increasing the use of conversation history would increase performance. However, since the conversation is rather short (2 turns), we hypothesize that the concepts in the immediately preceding turn are significantly more important (compared to the turn before) when generating a response. This result is consistent when we compare the algorithm with $\lambda = 0.6$ and $\lambda = 0.3$ as a lower value of λ slightly improves the p-value. However, we believe that a larger value of λ will be essential for effective performance in longer conversations as the conversation starts to reference more and more previously mentioned concepts. Future work will examine the performance of the model on longer conversations.

Automated Metrics

Mean BLEU scores for the KV Memory base model and MRF-Chat with each value of λ, as well as BLUE scores exclusively on conversations in which MRF-Chat produced a different response than the base model, can be found in Table 2. MRF-Chat (augmented with the base model) consistently achieves better mean BLEU scores compared to the base model across all values of λ.

We find that MRF-Chat tends to produce shorter utterances than the KV Memory base alone (see Table 3). We also find that the average number of concepts extracted by Yake[55] from each of MRF-Chat's responses is less than that of the KV Memory model. Since our human evaluation indicates that MRF-Chat's responses are better and more

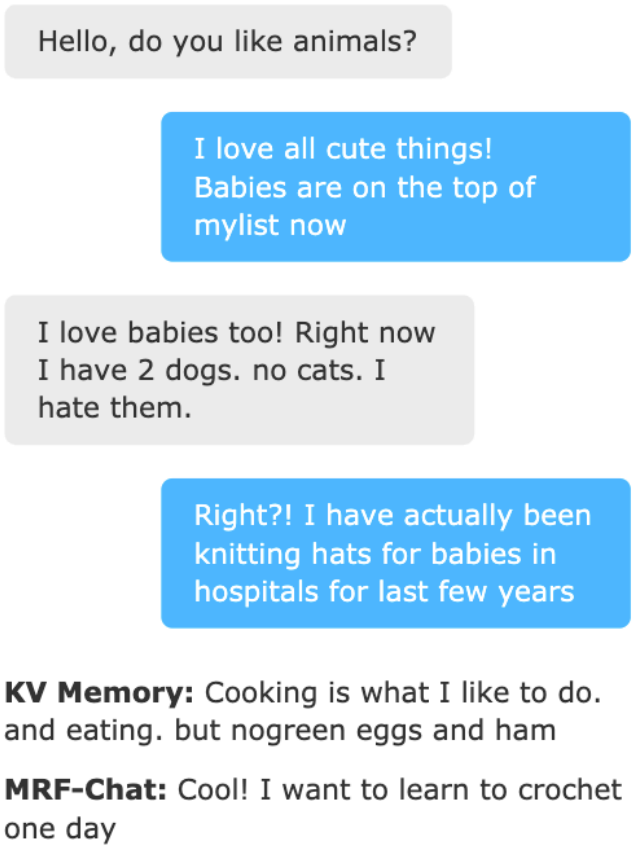


FIGURE 7: AN EXAMPLE OF HOW MRF-CHAT PRODUCES BETTER RESPONSES WITH FEWER BUT MORE RELEVANT CONCEPTS.

on topic than KV Memory alone, this suggests that MRF-Chat produces utterances with fewer, more relevant concepts. This is by design, as our algorithm rewards concepts that are believed to be known to both the agent and the user based off of the conversation, and penalizes concepts that are less relevant (Equation 7). An example conversation showing this phenomenon can be seen in Figure 7.

Conclusions and Future Work

In this chapter, we have introduced MRF-Chat, an algorithm for improving open-domain conversational agents based on the cognitive theory of mutual knowledge. The algorithm incorporates the contextual relevance of all prior concepts used in conversation in order to make predictions. Additionally, it is domain agnostic and independent of the base model used. Using human and automatic evaluation methods, we show that MRF-Chat can significantly improve the quality of responses in dialogue when combined with the KV Memory base trained on the Persona-Chat corpus. We found significant improvements over the base model in terms of the responses being on-topic as well as the overall quality of responses across various values of the prior weight λ. We further show that our algorithm produces slightly shorter utterances, with fewer but more relevant concepts, than the base alone.

While this algorithm provides a step forward in improving the quality of open-domain dialogue systems, as demonstrated by our consistent improvements over the Persona-Chat KV Memory model in human evaluation, future work will investigate and evaluate augmentation of other state-of-the-art approaches with MRF-chat. Next, our human evaluations use short conversation histories (4 turns), and we hypothesize that longer

conversations would enable our approaches to be the most effective. Additionally, it is essential that dialogue systems can effectively manage longer, more complex conversations, so we plan on evaluating MRF-Chat on longer conversation histories.

Evaluation Setup

In the MRF-Chat chapter, I showed that MRF-Chat can significantly improve the quality of responses in dialogue by augmenting the KV Memory base model. In this chapter, I describe the full end-to-end evaluation of the Relational Dialogue system. Using a semi-structured conversation framework focused on food preferences, I gathered a total of 92 conversations between 23 human participants and either the full relational dialogue system or a reduced baseline system. After each conversation, participants rated the conversational agent on typical open-domain dialogue characteristics (engaging, interesting, human sounding), as well as on social/relational characteristics (collaborative, trust, caring, understanding).

Semi-Structured Conversation Framework

In order to measure the system's performance on both general chit chat and in an application-focused environment, the interactions gathered for evaluation combined open dialogue with the tasks of deciding on a healthy meal. While the conversations are focused around the topic of food, no modifications would be necessary to discuss other common topics in a similar way (besides the knowledge graph augmentation used for the final recipe recommendation). In this sense, the system is truly an open domain dialogue system. Here, food is used as an example of a common topic that two individuals may casually discuss, as well as an example of a space where the system can be used to aid in a concrete goal, i.e. the recommendation of a healthy meal.

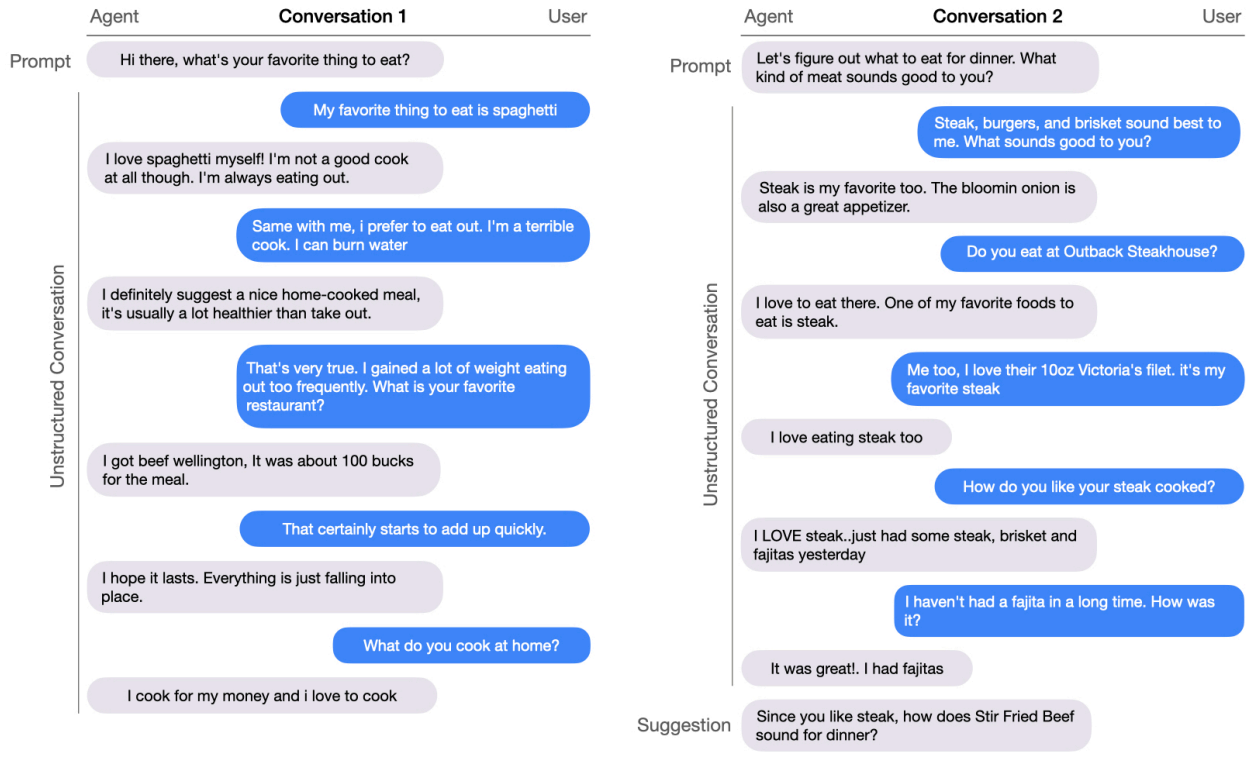


FIGURE 8: CONVERSATIONS BETWEEN PARTICIPANT AND RELATIONAL DIALOGUE SYSTEM.

The entire interaction consists of two conversations. The first conversation begins around food in general, and participants then discussed a variety of topics in the first conversations. The system starts the conversation with the prompt “Hi there, what’s your favorite thing to eat”, after which the user and system converse freely. In the second conversation, the system starts with “Let’s figure out what to eat for dinner. What kind of meat sounds good to you?”. What followed is generally a more task-focused and collaborative discussion around food preferences and what to choose for dinner. After ten turns of open dialogue (five user utterances and five system responses), the system recommends a healthy recipe for dinner chosen from a set of twenty candidates (described next in “Knowledge Graph Augmentation - Recipes”). An

example pair of conversations that a participant had with the full Relational Dialogue system can be seen in Figure 8. Both conversations are started by the fixed prompt from the system, followed by 10 turns of unstructured interaction. The second conversation is then concluded with the recipe recommendation.

Knowledge Graph Augmentation - Recipes

Recipe Scraping

For evaluation, an additional layer of healthy recipes was added to the knowledge graph. This serves as an example of how the knowledge graph can easily be expanded or adapted to serve new goals, and a similar expansion could easily be done with other topics such as sports, books, movies, etc. Twenty recipes were scraped from the USDA's ChooseMyPlate website, which provides many healthy eating recipes and recommendations. The recipes included a wide range of proteins and vegetables, as well as several vegetarian options.

Knowledge Graph Augmentation

In order to incorporate the recipes into the knowledge graph, each recipe was added as a new node type "RecipeNode". Each recipe's ingredients were searched in Wikidata and added to the knowledge graph using the same entity linking process from Pre-Processing, and an "Ingredient" edge was added between each recipe's node and its ingredients' Wikidata nodes. In this way, the recipes were incorporated seamlessly into the knowledge graph, without any modifications to the Wikidata graph structure. The knowledge graph can be easily augmented in this way for a wide variety of tasks or specific contexts.

Knowledge Graph Initialization

In order to minimize time spent downloaded Wikidata information during conversations, the knowledge graph was pre-loaded with all Wikidata entities within 2 degrees of the recipe ingredients using the Local Expansion Bot. When combined with the pre-loaded agent model created from existing Jibo utterances (as described in the Knowledge Graph Development section), the resulting graph that was used as the starting graph for each interaction contained a total of 3,972 nodes. Since the knowledge graph is always a partially-downloaded representation of all Wikidata entities and their relationships that is continuously expanded to fit the needs of conversation, these entities could be downloaded at runtime instead. However, since there is a high likelihood they will be used, it is reasonable to download them ahead of time. This practice is applicable to any time that the future topic of conversation is generally known. There is little drawback of downloading additional entities ahead of time, besides the increased storage needed to save the knowledge graph. If storage and compute are not an issue, the entirety of Wikidata could theoretically be pre-downloaded into the knowledge graph, but that would result in an enormous amount of required storage. The knowledge graph's implementation minimizes the effects of increased total graph size on runtime latency.

Recipe Recommendation

The recipe to recommend to the user leverages the knowledge graph to infer the user's preferences beyond just the ingredients or foods they have mentioned. For example, if the participant is known to like steak, it can be inferred that there's a very high probability that they like beef in general, and a possibility that they like other meats.

While this inference can be done through a wide range of techniques (e.g. MRF), for this evaluation a relatively simple approach is used to calculate the score for each recipe:

1. Find all of the paths $P = p_1, p_2, \ldots, p_i$ between the user node and the recipe node that are no longer than 5 nodes (including the user and recipe nodes). Paths may not visit the same node twice, may not include any non-Wikidata nodes outside of the user/recipe endpoints (the path may not move through another recipe, user, or agent). By definition, any node adjacent to the user node is connected to the user node by either a “likes”, “dislikes”, or “mentioned” edge.

2. Assign each path a weight inverse to its length, with a path of user-ingredient-recipe receiving a weight of 1: $w_i = \frac{1}{length(p) - 2}$

3. Assign each path a valence $v_i = -1$ if the user dislikes the adjacent node, and $v_i = 1$ otherwise.

4. Compute the recipe’s final score: $S_{recipe} = \sum_{p_i \in P} w_i v_i$

In the baseline system, each recipe’s score is simply the number of ingredients that the user is known to like or has mentioned without sentiment, minus the number of ingredients the user is known to dislike. This is equivalent to the full knowledge graph

approach with the restriction that the path found must have only three nodes (the user, the ingredient, and the recipe).

Baseline System

In order to evaluate the effects of the key components of the Relational Dialogue system (MRF-Chat, Knowledge Graph Response Generation, Use of Knowledge Graph in Recipe Recommendation), the evaluation focuses on comparing the complete Relational Dialogue system to a reduced baseline system without those key components. The baseline system uses the entire pre-processing pipeline (concept extraction, entity linking, sentiment analysis, etc.), but only uses retrieval-based candidates, and neural-only Response Selection without MRF-Chat. In order to choose a recipe to recommend, the baseline system can only use information about whether the user likes or dislikes the exact ingredients of the recipe, that is, it cannot use knowledge graph relationships to infer possible preferences. While the baseline system does not use the knowledge graph for generating responses or choosing a recipe to recommend, and does not use MRF-Chat in Response Selection, all system components are executed in the baseline system so that there is no difference in latency between the baseline system and the entire Relational Dialogue system in evaluation (for example, in the baseline system MRF-Chat is run as normal given the conversational history and candidate utterances, but the outputted scores are never used).

Experimental Details

In order to gather conversations for evaluation, twenty-three participants were recruited. All participants were current or recent undergraduate students. The majority of participants (19 of 23) had engineering, math, or science-related backgrounds, but no participant had significant experience in the design or development of conversational agents or natural language processing systems. Each participant interacted with both the full Relational Dialogue system and the baseline system, following the two-conversation format described above for each system. Half of the participants were assigned to interact with the baseline system before the full system, and half interacted the the full system before the baseline. The system was run on a 2019 MacBook Pro with 2.8 GHz Quad-Core Intel Core i7 CPU and 16 GB 2133 MHz LPDDR3 RAM. Participants were able to type in their utterances and see system responses through Zoom remote controls (unfortunately interactions could not be gathered in person due to the pandemic). Gathering in-person interactions, as opposed to through Mechanical Turk, allowed for an exceptionally high quality of data.

After each conversation, the participant answered a questionnaire, rating the system on a scale of 1-10 (1=Strongly Disagree, 10=Strongly Agree) for several metrics. In total, each participant filled out 4 questionaries (one after the first conversation and one after the second, for both the baseline and full system). The questionnaire used for the second conversation is identical to the first questionnaire except for the addition of two questions. The questions were as follows:

Typical Dialogue Metrics

My conversational partner is engaging

My conversational partner is interesting

My conversational partner sounds human

Relational Metrics

My conversational partner is collaborative

I trust my conversational partner

My conversational partner cares about me

My conversational partner understands me

Recommendation Metrics (2nd Conversation Questionnaire only)

My conversational partner knows my food preferences

My conversational partner suggested a good meal for me

Once the participant completed the interactions with both systems, they were additionally asked which conversational partner/system they preferred speaking to. This provides an absolute metric to compare the two systems overall.

Results

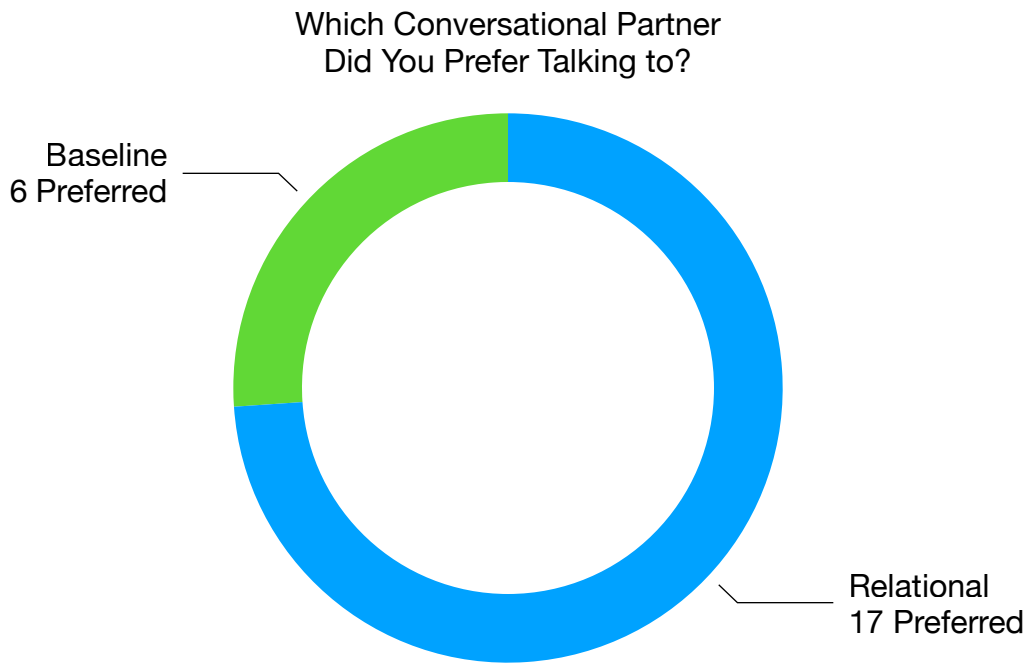


FIGURE 9: THE MAJORITY OF PARTICIPANTS PREFERRED TO TALK TO THE RELATIONAL DIALOGUE SYSTEM OVER THE BASELINE.

Overall, the Relational Dialogue system showed very strong results compared to the baseline system. When participants were asked which conversational partner they preferred talking to (between the baseline and full Relational Dialogue system), seventeen out of twenty-three said they preferred the Relational Dialogue system (participants never knew which of the two systems they interacted with was which, they only knew that they had interacted with two different systems). This is an overwhelming win showing the benefits of MRF-Chat and the knowledge graph generators, and is statistically significant even with the small sample size ($p < 0.05$).

When asked why they preferred the relational system often said that it provided “more interesting conversation” and “stayed on topic more”. Figure 10 shows an example of

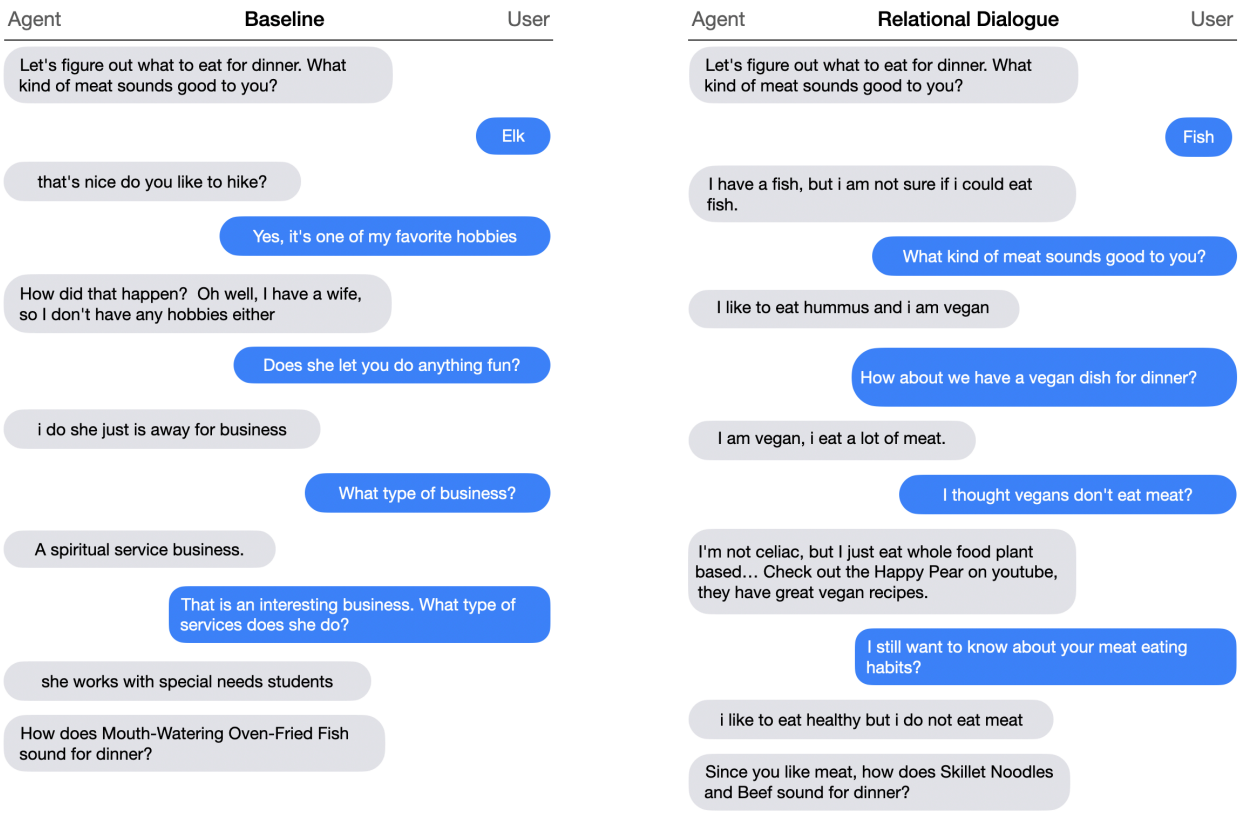


FIGURE 10: A COMPARISON OF CONVERSATION 2 BETWEEN A PARTICIPANT AND BOTH THE BASELINE AND FULL RELATIONAL DIALOGUE SYSTEM. THE RELATIONAL DIALOGUE SYSTEM PRODUCED MORE ON-TOPIC AND RELEVANT RESPONSES.

how the Relational Dialogue system provides more on-topic responses than the baseline system. One participant felt that the Relational Dialogue system had “more empathy towards [them]”, and another said that it “seemed a bit more personable, willing to talk more about themself”.

When participants preferred the baseline system, they often mentioned that the Relational Dialogue system felt more stuck on one topic when they tried to change the course of the conversation. An example of this phenomenon occurs in Figure 11. One participant felt that the baseline system’s conversations “flowed more naturally” from topic to topic, and another participant tried to change the topic away from food but the agent resisted. The combination of the these comments would suggest that while the

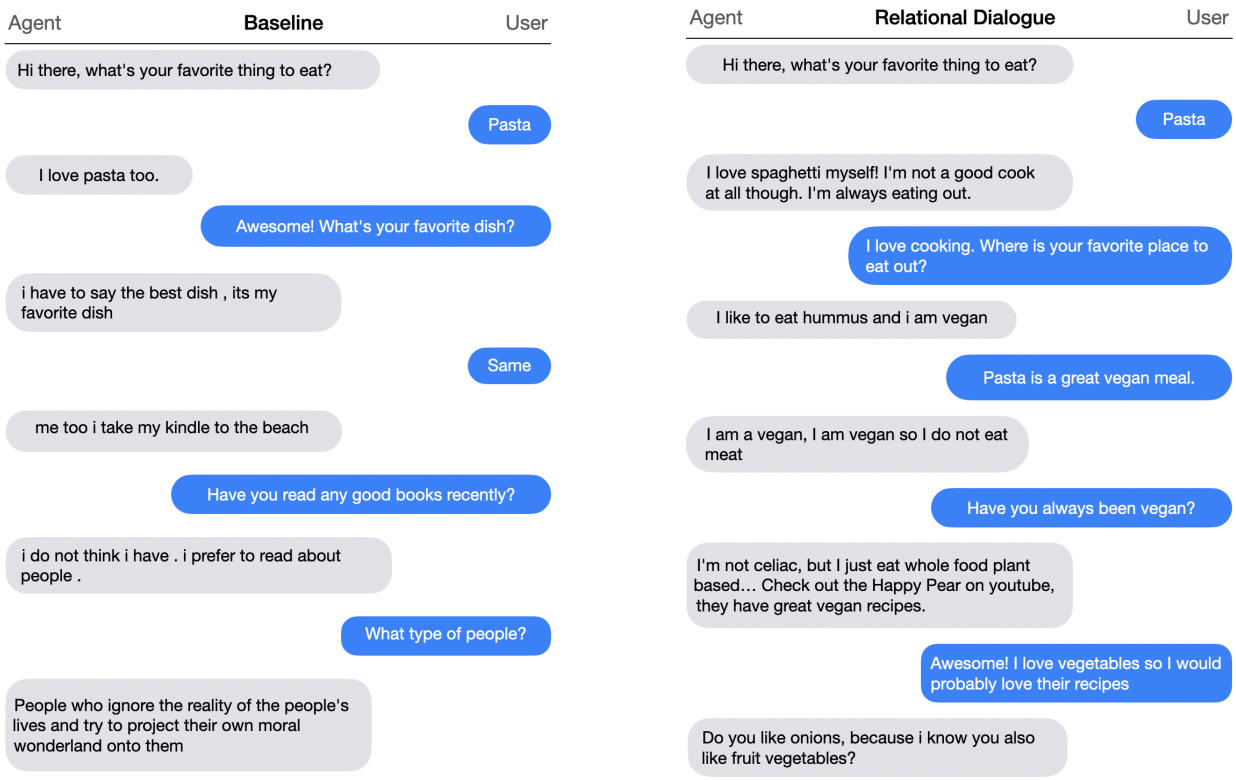


FIGURE 11: A COMPARISON OF CONVERSATION 1 BETWEEN A PARTICIPANT AND BOTH THE BASELINE AND FULL RELATIONAL DIALOGUE SYSTEM. WHILE THE RELATIONAL DIALOGUE SYSTEM'S RESPONSES ARE MORE ON-TOPIC. THEY LIMIT THE FLOW OF CONVERSATION.

Relational Dialogue system is able to have a more interesting and coherent conversation than the baseline, and at times allows for better conversational flow, additional work is needed to ensure that the system can appropriately adapt to the topic of conversation when the user desires a more significant change.

The Ask Preference Generator was used in 16 out of 46 conversations between a participant and the Relational Dialogue system (this does not include additional times that the Ask Preference Generator created candidate responses that weren't ultimately chosen). These 16 conversations happened with 12 out of the 23 participants. The Ask Preference Generator responses were typically only chosen once or twice in the entire interaction (9 out of 12 participants). Interesting, all three participants who received Ask

Characteristic	Conversation 1 Win/Tie/Loss	Conversation 2 Win/Tie/Loss	Overall Win/Tie/Loss
Engaging	9 / 4 / 10	**14 / 4 / 5**	23 / 8 / 15
Interesting	10 / 6 / 7	14 / 1 / 8	24 / 7 / 15
Sounds Human	12 / 2 / 9	12 / 2 / 9	24 / 4 / 18
Collaborative	12 / 5 / 6	**12 / 7 / 4**	**24 / 12 / 10**
Trust	8 / 11 / 4	12 / 5 / 6	**20 / 16 / 10**
Cares About Me	6 / 5 / 12	14 / 2 / 7	20 / 7 / 19
Understands Me	12 / 3 / 8	13 / 3 / 7	25 / 6 / 15
Knows My Food Preferences	N/A	13 / 4 / 6	13 / 4 / 6
Suggested a Good Meal for Me	N/A	10 / 7 / 6	10 / 7 / 6

TABLE 4: HUMAN RATING RESULTS: WIN/TIE/LOSS COUNT FOR THE RELATIONAL SYSTEM AGAINST THE BASELINE. A WIN IS WHEN THE PARTICIPANT RATED THE RELATIONAL SYSTEM HIGHER THAN THE BASELINE FOR THE RESPECTIVE CHARACTERISTIC (TIE IF SAME RATING, LOSS IF LOWER RATING). CELLS ARE GREEN IF THE RELATIONAL SYSTEM WON THE MAJORITY OF THE TIME (RED FOR MAJORITY LOSS). RESULTS IN BOLD ARE STATISTICALLY SIGNIFICANT ($p < 0.05$).

Preference Generator responses more than twice (thrice for two participants, four times for the other) preferred the baseline system. Of the other nine, the one participant that preferred the baseline received two Ask Preference Generator responses. These results would suggest that excess use of the Ask Preference Generator may result in lower conversational quality, which would not be surprising as 3 or 4 similar responses in an interaction of 13 total agent utterances may seem fairly repetitive.

Questionnaire Results

The Relational Dialogue system also showed promising results across the majority of characteristics asked about in the questionnaires. The results for participants' ratings of Relational Dialogue vs. baseline on each characteristic for each conversation

independently and combined can be see in Table 4. While results on Conversation 1 were more evenly spread, the Relational Dialogue system was given a higher rating by the majority of participants on 8 of the 9 characteristics for Conversation 2, and 6 of the 9 characteristics overall. The Relational Dialogue system was rated as significantly more engaging and collaborative on Conversation 2, and significantly more collaborative and trusted by participants overall ($p < 0.05$ for all).

The Relational Dialogue system performed much better in Conversation 2 than Conversation 1 when compared to the baseline. It is unclear whether this effect is due to the Relational Dialogue system being able to perform more effectively in the context of Conversation 2 specifically, or if conversing with the systems for longer provides more information about which the participant can make a stronger decision about the fundamental capabilities of each system.

Results show that participants felt that the the Relational Dialogue system provided better meal suggestions slightly more often than the baseline, and the majority felt that the Relational Dialogue system better understood their food preferences. While more data is needed to determine if these results are statistically significant, these initial findings are promising.

Knowledge Graph Development

During the first conversation that participants had with the full Relational Dialogue system, an average of 7.5 nodes were added to the user model in the knowledge graph (stdev=4.5, min=0, max=19), and an average of 112.1 nodes were added to the initial knowledge graph (stdev=52.3, min=23, max=255). In the second conversation, the

same average of 7.5 nodes were added to the user model (stdev=4.4, min=1, max=21), and a slightly higher average of 149.9 nodes were added to the entire knowledge graph (stdev=114.9, min=0, max=410). No correlation was found between either the number of nodes added to the user model or the number of nodes aded to the knowledge graph and user ratings. While this may seem surprising at first, this mirrors the initial MRF-Chat finding, which was that while MRF-Chat produces responses with more relevant concepts, the number of concepts occurring in conversation does not differ significantly. It is possible that a similar trend is occurring here, where the relevance and quality of the concepts occurring in conversation is more significant than the amount. Since less relevant concepts are often found in less coherent responses, it is reasonable to hypothesize that similar patterns could be found here, with the Relational Dialogue system performing at its best when producing highly relevant concepts without significant effect on the abundance of concepts, although additional work is needed to understand these effects when using the entire Relational Dialogue system.

Latency

Interactions were gathered on a 2019 MacBook Pro with a 2.8 GHz Quad-Core Intel Core i7 CPU. On average, Pre-Processing took 258ms, Candidate Response Generation took 198ms, API requests to Wikidata took 378ms, neural response scoring took 2054ms, and MRF-Chat scoring took 977ms, for a total average latency of 3865ms. This latency could be dramatically reduced (likely to below 2 seconds) by running the neural response scoring model on GPU.

Discussion & Future Work

Ethical Considerations & Data Privacy

While it is important to evaluate the ethical implications of any system, it is especially important to keep them in mind for this project. Establishing trust and rapport with a user increases how important it is that the agent's actions and uses are ethically sound and unbiased, and that the technology is not used for malice or exploitation. As transformer-based generative models such as GTP-3[17] continue to create increasingly human-sounding text, and algorithms like MRF-Chat allow for an increased perception of understanding, it is important that similar systems are not used for deception or manipulation.

It is also important to consider issues of transparency and privacy around the data gathered about each individual in the knowledge graph, as well as the ability to explain how any decisions or recommendations based on the knowledge graph are made. The knowledge graph's design makes explainability very simple, as it is a symbolic system with a relatively small number of understandable concepts. It is also very easy to visualize a user's data in the knowledge graph, and to make modifications to that information by adding or deleting nodes and edges.

Improving Agent/User Modeling

Richer User Modeling

While modeling the user's preferences allows for a valuable representation of the user that can be used to drive conversation, there are many ways in which that model could be expanded to support richer interaction. Initially, the model could be expanded to

include key information about the user, perhaps including basic knowledge about their friends and family, their age, their home town, etc. By developing the ability to effectively extract more relationships between concepts in utterances, the graph can then be expanded to include additional information that comes up. For example, if the user mentions “Mark and I went to the beach yesterday”, a “visited” relationship could be added between the user node and the beach.

Development of Agent

Through conversation, the agent could learn and its personality would grow, aligning some of its own preferences with the user’s, and expanding its own unique identity as the relationship develops. While core features of the agent’s character will be fixed, the agent can form opinions on new topics. It will be important to identify highly controversial or unwanted opinions, and prevent the agent from adopting them. This can be done initially using existing sentiment detection techniques to detect topics with negative connotations.

Improving Agent Responses

Improved Ask Preference Generator + Response Selection

In some cases, the Ask Preference generator helps improve the conversational experience by exploring the user’s interest in concepts that are relevant to conversation. However, there are other cases when the Ask Preference generator breaks down, reducing the quality of conversation. There are two main issues with the Ask Preference generator that occurred occasionally in evaluation. The first issue is asking about unusual concepts, and the second is too frequent use of the Ask Preference generator. Examples of both of these issues can be seen in Figure 12. First, the agent asks the user if they like “dishes” because they like “steaks”. While dishes (in

FIGURE 12: AN EXCERPT FROM A PARTICIPANT'S INTERACTION WHERE THE ASK PREFERENCE GENERATOR BREAKS DOWN.

this context the meaning of dish similar to a recipe, e.g. "a chicken dish") are relevant to steaks (steak is a type of food or dish in the Wikidata ontology), it would be unusual for a human to ask the question. In order to overcome this, additional work is needed to predict how likely it is that any given concept would be brought up in conversation. Since typical language models for open-domain dialogue are trained on entire utterances, an utterance that is generally a reasonable reply, but has one or two key words that are unusual, may still receive a high score.

Regarding the second issue, using the Ask Preference generator multiple times in an interaction was linked to lower system ratings (while it appeared that a single use of the

Ask Preference generator corresponded with higher ratings, additional evaluation is required to confirm this trend). While this issue could be easily fixed by limiting how often the Ask Preference generator produces candidate responses, the issue is a symptom of a deeper problem. MRF-Chat favors concepts that are relevant to the conversation, and are likely known by both the agent and the user. In most cases, this mechanism increases the quality of selected responses, however there are some cases, such as the multiple Ask Preference issue, where it results in the repetition of similar responses or topics. To address this, further work is needed to model not only what concepts are relevant, but also to model how the topic of conversation should change over time.

Additional Knowledge Graph Generation

As the knowledge graph is expanded to include more complex models of both the user and the agent, they ways in which it can be used for conversation will grow. The agent could bring up information in the knowledge graph in conversation, or use the knowledge to inquire about the user's experiences.

A significant issue with many existing dialogue systems is a lack of logical consistency in the agent's responses[19]. For example, an agent may say they like a certain food, or say they had a certain experience, and then later say the opposite. The knowledge graph could be used to prevent this issue.

Response Combination

One of the limitations of a primarily-retrieval based approach is that while the agent's utterance is chosen to be the best overall way of responding to the user, the response may not address all parts of the user's utterance. By combining various candidate responses appropriately, the system may be able to create more complex responses

that address the user's utterance and drive the conversation forward. For example, BYU-EVE[25] uses an emote/answer/offer framework, where the agent appropriate responds to the user, and then offers something new. This framework could be expanded to ensure that the user's questions are answered, to allow the agent to share things about itself at the right moments, and to increase the overall coherent-ness of responses.

Further Evaluation

While the conducted evaluation showed strong initial results for the Relational Dialogue system, additional evaluation is needed to further understand some of the trends observed in the interactions, as well as the strengths and weaknesses of various subcomponents in the system. Initially, continuing the existing evaluation with many more participants would improve the statistical significance of the results, and offer more clarity around some of the observations. The next key area to address is evaluating how the system performs on different types of conversations, and how the user's perception of the agent changes over time. Did the Relational Dialogue system outperform the baseline on the second conversation more significantly because of the nature of the second conversation's framing and content, and is that difference repeatable in other types of conversations? Does the user having more conversation with the systems result in a higher separation in ratings between the baseline and the full system over time? What kinds of conversational scenarios does the Relational Dialogue system perform will in, and where does it struggle? To address these questions, similar evaluation can be conducted with new types of conversations, different topics besides food preferences, and with longer sequences of conversations over time.

Additional work is also needed to understand the different effects of MRF-Chat vs. the knowledge graph generators on the overall quality of experience. In order to measure these effects, there are several approaches that will help give insight into each component. Initially, gathering more interactions may help reveal trends between the use of the knowledge graph generators and user ratings. Then, an ablation study could be performed, evaluating the system with knowledge graph generators and without MRF-Chat, and vise versa. Over time, developing a robust method for evaluating each knowledge graph generator independently would greatly aid in the development of new generators for new applications.

Another key area for further evaluation is how exploring different types and amounts of concepts in conversation affects the user's experience, as well as the specific effects of user model development. Results in the initial MRF-Chat evaluation found that the quality of conversational concepts was more important to human ratings than the number of them, and an analysis of similar effects is needed to determine if the same trend occurs with the entire Relational Dialogue system.

Conclusion

In this thesis, I presented the Relational Dialogue system through the following:

1. A Wikidata-based knowledge graph that contains information about the world, and provides a platform for modeling the preferences of both the user and the agent in order to generate agent utterances that seek to learn about the user and share the agent's preferences.

2. MRF-Chat, a novel probabilist approach to augmenting retrieval-based dialogue systems. MRF-Chat models the mutual knowledge of the agent and user, as well as the contextual relevance of all concepts appearing in conversation and in candidate responses. In human evaluations, MRF-Chat was found to produce significantly better and more on topic responses when compared to a state-of-the-art baseline.

3. An example application of the Relational Dialogue system, using a semi-structured conversational framework around food preferences in order to drive conversation with the user, and ultimately make a healthy meal recommendation based on their learned preferences.

In human evaluations, the majority of participants preferred conversing with the Relational Dialogue system over the baseline (17/23, $p < 0.05$). The Relational Dialogue system was rated significantly more engaging, collaborative, and trusted by the users. These results show the benefit of the knowledge graph and of MRF-Chat, and lay the groundwork for future work that approaches human-computer dialogue as a shared social experience between the agent and the user.

Acknowledgements

As this work represents a key culminating moment in my education, I'd like to recognize the individuals who this work could not have happened without, so I extend the following thanks:

My family and closest friends, thank you for your support and guidance through life. Most especially to my parents, without whom none of this would have been possible in any way, shape, or form.

For those who gave me vision and guidance through this project and my degree:
Cynthia, I cannot thank you enough for welcoming me into the group, for helping me grow and think about the future, and for building such an amazing group of people to learn and work alongside. I could never ask for a better time than I've had in the group.
Hae Won, thank you for always being there to teach and support me, for the countless things that you do to make sure that I, and everyone else, can do their best work. You've always been there to point the way when I've encountered a new challenge, and without you much of this work would have been far more difficult and much less fun.

For the educators that helped me build the path that led up to this work:
Paul and **Liam**, for showing me the wonders of human language.
Taiwo, for always walking with me in my exploration of computer science.
Blade, for teaching me to challenge the world around me.
Bruno, for helping me listen to, and fight tactfully for, my own voice.

For those who worked with me along the way and offered invaluable support:
Ishaan, for working together on MRF-Chat, and for helping on endless other things.
Pedro, **Tasia**, **Sooyeon**, **Sharifa**, **Sam**, and many others, for your support, guidance, and collaboration throughout a wonderful twenty-eight months with PRG.

References

[1] Weizenbaum, J. (1966). ELIZA—a Computer Program for the Study of Natural Language Communication between Man and MachineCommun. ACM, 9(1), 36–45.

[2] Zemčík, Tomáš. (2019). A Brief History of Chatbots. DEStech Transactions on Computer Science and Engineering. 10.12783/dtcse/aicae2019/31439.

[3] Alice (2002). A.L.I.C.E AI Foundation. http://www.Alicebot.org/.

[4] NPR (2019). NPR and Edison Research Report: 60M U.S. Adults 18 Own a Smart Speaker. https://www.npr.org/about-npr/794588984/npr-and-edison-research-report-60m-u-s-adults-18-own-a-smart-speaker.

[5] Jie Lu, Vahid Behbood, Peng Hao, Hua Zuo, Shan Xue, and Guangquan Zhang. 2015. Transfer learning using computational intelligence: A survey. Knowledge-Based Systems 80 (2015), 14–23.

[6] Yijin Liu, Fandong Meng, Jinchao Zhang, Jie Zhou, Yufeng Chen, and JinanXu. 2019. CM-Net: A Novel Collaborative Memory Network for Spoken Language Understanding. arXiv:1909.06937 [cs.CL]

[7] Jan Pichl, Petr Marek, Jakub Konrád, Petr Lorenc, Van Duy Ta, & Jan Šedivý. (2020). Alquist 3.0: Alexa Prize Bot Using Conversational Knowledge Graph.

[8] Gabriel, R., Liu, Y., Gottardi, A., Eric, M., Khatri, A., Chadha, A., ... & Hu, S. (2020). Further Advances in Open Domain Dialog Systems in the Third Alexa Prize Socialbot Grand Challenge. Proc. Alexa Prize.

[9] Breazeal, Cynthia. (2003). Toward social robots. Robotics and Autonomous Systems. 42. 167-175. 10.1016/S0921-8890(02)00373-1.

[10] Sooyeon Jeong, Sharifa Alghowinem, Laura Aymerich-Franch, Kika Arias, Agata Lapedriza, Rosalind Picard, Hae Won Park, & Cynthia Breazeal. (2020). A Robotic Positive Psychology Coach to Improve College Students' Wellbeing.

[11] Kanero, Junko & Geckin, Vasfiye & Oranç, Cansu & Mamus, Ezgi & Küntay, Aylin & Goksun, Tilbe. (2018). Social Robots for Early Language Learning: Current Evidence and Future Directions. Child Development Perspectives. 12. 146-151. 10.1111/cdep.12277.

[12] Ostrowski, A.K., DiPaola, D., Partridge E., Park, H.W., & Breazeal, C. (2019). Older Adults Living with Social Robots: Promoting Social Connectedness in Long-Term Communities. IEEE Robotics & Automation Magazine. doi: 10.1109/MRA.2019.2905234

[13] Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing. ieee Computational intelligenCe magazine, 13(3), 55-75.

[14] Hu, D. (2019, September). An introductory survey on attention mechanisms in NLP problems. In Proceedings of SAI Intelligent Systems Conference (pp. 432-448). Springer, Cham.

[15] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L., & Polosukhin, I. (2017). Attention is All you Need. In Advances in Neural Information Processing Systems (pp. 5998–6008). Curran Associates, Inc..

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, & Kristina Toutanova. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

[17] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI blog, 1(8), 9.

[18] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, & Dario Amodei. (2020). Language Models are Few-Shot Learners.

[19] Floridi, L., & Chiriatti, M. (2020). GPT-3: Its nature, scope, limits, and consequences. Minds and Machines, 1-14.

[20] Ram, A., Prasad, R., Khatri, C., Venkatesh, A., Gabriel, R., Liu, Q., ... & King, E. (2018). Conversational ai: The science behind the alexa prize. arXiv preprint arXiv:1801.03604.

[21] Khatri, C., Hedayatnia, B., Venkatesh, A., Nunn, J., Pan, Y., Liu, Q., ... & Cheng, M. (2018). Advancing the state of the art in open domain dialog systems through the alexa prize. arXiv preprint arXiv:1812.10757.

[22] Zihao Wang, Ali Ahmadvand, Jason Ingyu Choi, Payam Karisani, and Eugene Agichtein. Emersonbot: Information-focused conversational ai emory university at the alexa prize 2017 challenge. AWS, 2017.

[23] Khatri, C., Hedayatnia, B., Venkatesh, A., Nunn, J., Pan, Y., Liu, Q., ... & Cheng, M. (2018). Advancing the state of the art in open domain dialog systems through the alexa prize. arXiv preprint arXiv:1812.10757.

[24] Gopalakrishnan, Karthik, et al. "Topical-Chat: Towards Knowledge-Grounded Open-Domain Conversations." Proc. Interspeech 2019 (2019): 1891-1895.

[25] Fulda, Nancy, et al. "BYU-EVE: Mixed Initiative Dialog via Structured Knowledge Graph Traversal and Conversational Scaffolding." Proceedings of the 2018 Amazon Alexa Prize (2018).

[26] Chit-Chat Dataset. https://github.com/BYU-PCCL/chitchat-dataset

[27] Zhou, Li, et al. "The design and implementation of XiaoIce, an empathetic social chatbot." arXiv preprint arXiv:1812.08989 (2018).

[28] Rashkin, H., Smith, E. M., Li, M., & Boureau, Y. L. (2018). Towards empathetic open-domain conversation models: A new benchmark and dataset. arXiv preprint arXiv:1811.00207.

[29] Zhao, Ran, Alexandros Papangelis, and Justine Cassell. "Towards a dyadic computational model of rapport management for human-virtual agent interaction." International Conference on Intelligent Virtual Agents. Springer, Cham, 2014.

[30] Zhao, Ran, et al. "Automatic recognition of conversational strategies in the service of a socially-aware dialog system." Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue. 2016.

[31] Trinh, Ha, et al. "Predicting User Engagement in Longitudinal Interventions with Virtual Agents." IVA. 2018.

[32] Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. (2017). To appear.

[33] Wikidata. https://www.wikidata.org/wiki/Wikidata:Main_Page

[34] Neo4j. https://neo4j.com/

[35] Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.

[36] TextBlob. https://github.com/sloria/TextBlob

[37] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? arXiv preprint arXiv:1801.07243 (2018).

[38] Cer, Daniel, et al. "Universal sentence encoder." arXiv preprint arXiv:1803.11175 (2018).

[39] Dinan, E., Logacheva, V., Malykh, V., Miller, A., Shuster, K., Urbanek, J., ... & Prabhumoye, S. (2019). The second conversational intelligence challenge (convai2). arXiv preprint arXiv:1902.00098.

[40] Chandra Khatri, Anu Venkatesh, Behnam Hedayatnia, Raefer Gabriel, Ashwin Ram, and Rohit Prasad. 2018. Alexa Prize—State of the Art in Conversational AI. AI Magazine 39, 3 (2018), 40–55.

[41] Mikhail Burtsev, Varvara Logacheva, Valentin Malykh, Iulian Vlad Serban,Ryan Lowe, Shrimai Prabhumoye, Alan W Black, Alexander Rudnicky, and Yoshua Bengio. 2018. The first conversational intelligence challenge. In The NIPS'17 Competition: Building Intelligent Systems. Springer, 25–46.

[42] Ilia Kulikov, Alexander Miller, Kyunghyun Cho, and Jason Weston. 2019. Importance of search and evaluation strategies in neural dialogue modeling. In Proceedings of the 12th International Conference on Natural Language Generation. 76–87.

[43] Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M Smith, et al. 2020. Recipes for building an open-domain chatbot. arXiv preprint arXiv:2004.13637 (2020).

[44] Abigail See, Stephen Roller, Douwe Kiela, and Jason Weston. 2019. What makes a good conversation? how controllable attributes affect human judgments. arXiv preprint arXiv:1902.08654 (2019).

[45] Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 2019. Open-dialkg: Explainable conversational reasoning with attention-based walks over knowledge graphs. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 845–854.

[46] Raymond W Gibbs Jr. 1987. Mutual knowledge and the psychology of conversational inference. Journal of pragmatics 11, 5 (1987), 561–588.

[47] Gordon P Thomas. 1986. Mutual knowledge: A theoretical basis for analyzing audience. College English 48, 6 (1986), 580–594.

[48] Ishaan Grover, Hae Won Park, and Cynthia Breazeal. 2019. A Semantics-based Model for Predicting Children's Vocabulary.. In IJCAI. 1358–1365.

[49] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Doha, Qatar, 1532–1543. https://doi.org/10.3115/v1/ D14-1162

[50] Steve Young. 1999. Probabilistic Methods in Spoken Dialogue Systems. Philosophical Transactions of the Royal Society (Series A 358 (1999), 1389–1402.

[51] Deborah A. Dahl, Madeleine Bates, Michael Brown, William Fisher, Kate Hunicke-Smith, David Pallett, Christine Pao, Alexander Rudnicky, and Elizabeth Shriberg. 1994. Expanding the Scope of the ATIS Task: The ATIS-3 Corpus. In Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994. https://www.aclweb.org/anthology/H94-1010

[52] Iulian V. Serban, Chinnadhurai Sankar, Mathieu Germain, Saizheng Zhang, Zhouhan Lin, Sandeep Subramanian, Taesup Kim, Michael Pieper, Sarath Chandar, Nan Rosemary Ke, Sai Rajeshwar, Alexandre de Brebisson, Jose M. R. Sotelo, Dendi Suhubdy, Vincent Michalski, Alexandre Nguyen, Joelle Pineau, and Yoshua Bengio. 2017. A Deep Reinforcement Learning Chatbot. arXiv:1709.02349 [cs.CL]

[53] Dian Yu, Michelle Cohn, Yi Mang Yang, Chun-Yen Chen, Weiming Wen, Jiaping Zhang, Mingyang Zhou, Kevin Jesse, Austin Chau, Antara Bhowmick, Shreenath Iyer, Giritheja Sreenivasulu, Sam Davidson, Ashwin Bhandare, and Zhou Yu. 2019. Gunrock: A Social Bot for Complex and Engaging Long Conversations. arXiv:1910.03042 [cs.CL]

[54] Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-Value Memory Networks for Directly Reading Documents. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Austin, Texas, 1400–1409. https://doi.org/10.18653/v1/D16-1147

[55] Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes, and Adam Jatowt. 2020. YAKE! Keyword Extraction from Single Documents using Multiple Local Features. Information Sciences 509 (01 2020), 257–289. https://doi.org/10.1016/j.ins.2019.09.013

[56] Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic Keyword Extraction from Individual Documents. 1 – 20. https://doi.org/10.1002/9780470689646.ch1

[57] Marc Brysbaert and Boris New. 2009. Moving beyond Kucera and Francis: A Critical Evaluation of Current Word Frequency Norms and the Introduction of a New and Improved Word Frequency Measure for American English. Behavior researchmethods 41(112009),977–90. https://doi.org/10.3758/BRM.41.4.977

[58] C Bailer-Jones and Kester Smith. 2011. Combining probabilities. Data Processing and Analysis Consortium (DPAS), GAIA-C8-TN-MPIA-CBJ-053 (2011).

[59] Hugo Liu and Push Singh. 2004. ConceptNet–a practical common sense reasoning tool-kit. BT technology journal 22, 4 (2004), 211–226.

[60] Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Austin, Texas, 2122–2132. https://doi.org/10.18653/v1/D16- 1230

[61] Margaret Li, Jason Weston, and Stephen Roller. 2019. Acute-eval: Improved dialogue evaluation with optimized questions and multi-turn comparisons. arXiv preprint arXiv:1909.03087 (2019).

[62] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (Philadel- phia, Pennsylvania) (ACL '02). Association for Computational Linguistics, USA, 311–318. https://doi.org/10.3115/1073083.1073135

[63] Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Lisbon, Portugal, 1711–1721. https://doi.org/10.18653/v1/D15- 1199

[64] Geetanjali Rakshit, Kevin K Bowden, Lena Reed, Amita Misra, and Marilyn Walker. Debbie, the debate bot of the future. In Advanced Social Interaction with Agents, pages 45–52. Springer, 2017.