

NAIST-IS-DD1361205

Doctoral Dissertation

**Statistical Approaches to Robust
Chat-Oriented Dialog Systems**

Lasguido Nio

8 August, 2016

Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Lasguido Nio

Thesis Committee:

Professor Satoshi Nakamura	(Supervisor)
Professor Yuji Matsumoto	(Nara Institute of Science and Technology)
Dr. Ryuichiro Higashinaka	(NTT Media Intelligence Laboratories)
Assistant Professor Sakriani Sakti	(Co-supervisor)
Assistant Professor Graham Neubig	(Co-supervisor)
Assistant Professor Koichiro Yoshino	(Co-supervisor)

Statistical Approaches to Robust Chat-Oriented Dialog Systems*

Lasguido Nio

Abstract

This thesis describes the design and evaluation of new statistical models for building robust dialog systems. In the majority of previous works, conventional dialog systems required a specific hand-crafted rule which necessitates a lot of human work, especially when the dialog tries to accommodate various topics. In addition, because chat-oriented dialog systems deal with a vast variety of human languages, there are many cases in which the system ends up giving an uncorrelated response because it can't find a good match in the conversation database. Moreover, relying on the unfiltered conversation databases can result in unnatural responses.

Given these challenges in this field of work, my team and I tried to scale up statistical models for chat-oriented dialog systems. In dealing with the unnatural responses, we built our statistical models from real human-to-human conversation examples lifted from movie scripts and Twitter conversations. We propose a unit of conversation called a tri-turn, as well as extraction and semantic similarity analysis techniques to help ensure that the content extracted from raw movie/drama script files forms appropriate dialog-pair (query-response) examples. Our goal was to build a conversational agent that can interact with users in as natural a fashion as possible, while reducing the time requirement for database design and collection.

Next, we also dealt with the case where a close match for the user query is not available in the database (out of example; OOE). This problem becomes

*Doctoral Dissertation, Graduate School of Information Science,
Nara Institute of Science and Technology, NAIST-IS-DD1361205, 8 August, 2016.

important since many dialog system architectures rely heavily on, and assume that the user query is well portrayed in the conversation database. In our study however, this is not always the case. Previous approaches focused on handling this problem with a canned response or using a response template, which reduces the natural feel of the dialog systems.

Here we approached this problem with new statistical models for building robust dialog systems using neural networks to either retrieve or generate dialog responses based on existing data sources. In the retrieval task, we proposed an approach that uses paraphrase identification during the retrieval process. This is done by employing recursive autoencoders and dynamic pooling to determine whether two sentences with arbitrary length have the same meaning. For both the generation and retrieval tasks, we propose a model using long short term memory (LSTM) neural networks that work by first using an LSTM encoder to input a user's utterance and converting it into a continuous vector-space representation, then using an LSTM decoder to generate the most probable word sequence.

This system's performance was evaluated based on objective and subjective metrics. It showed that the new proposed approaches have the ability to deal with user inputs that are not well covered in the database compared to standard example-based dialog baselines. Our experimental results also show that the proposed filtering approach for our conversation database effectively improves the performance.

Keywords:

Example-based dialog system, chat-oriented dialog system, dialog corpora, response retrieval, response generation, paraphrase-based response retrieval, recursive auto-encoder, long short-term memory, neural network

Acknowledgements

First and foremost, I would like to thank my God, Jesus Christ for His grace and infinite love. Thankful for everything that He allow to cross my path.

I wish to express my gratitude to Professor Satoshi Nakamura for his supervision lasted for 3 years since I began my doctorate course. He provided me space to have an internship and recommended me for MEXT scholarship program, thus make me continue my study here in Japan. His tireless supervision and leadership are big inspiration for me to become a good researcher, in the future.

I would like to thank Assistant Professor Sakriani Sakti from whom I learned invaluable experiences of research, English writing, public speaking skills, creative thinking, and the art of presentation. She shaped my skills in so many ways that it's hard to describe. She never gives up, always push me up, and motivate me through all the hardships during my time at NAIST.

I also want to thank you to Assistant Professor Graham Neubig for the invaluable discussion that we had. His ideas and helps also shaped my research here in NAIST, especially when Assistant Professor Sakriani Sakti is not around for a while. I learned so much from him and inspired by his knowledge.

Thanks to Assistant Professor Koichiro Yoshino who lead the current SD team in AHC-Laboratory for the discussion and ideas. His knowledge in dialog research topics inspired me. I also would like to thank Professor Tomoki Toda for his the discussion that we had.

Thank to all the thesis committee, Professor Yuji Matsumoto and Dr. Ryuichiro Higashinaka from NTT for giving me insightful comments and carefully checking my thesis.

I would like to thank Dr. Wu Youzheng from SONY China Research Laboratory. The experiments and discussion during the internship program gave me many inspiration of statistical dialog management. I also would like to thanks

Zhiwei Zhao, Xiao Ran, and Yu Zhijiang for becoming good friends and help me during my time in Beijing. It was a very invaluable experience there.

I want to thank Nakamura-lab, especially SD group, the biggest and loudest group in AHC-Lab. It was fun to discuss and learn together.

Special thanks to Matsuda-san who always helped me and available 24/7 to remind me of the deadlines. Thanks for helping me in life-related problem during my stay in Japan. I also would like to thank Karube-san for her help when Matsuda-san is not around. I couldn't make it until this point without their support and help.

Parents, Prildy, and Sabriela who support, encourage, and love me endlessly. Finally, for Huang Veronica who always supports me in every decision that I made.

Contents

Acknowledgements	iii
List of Figures	viii
List of Tables	1
1. Introduction	2
1.1. General Human to Machine Conversation	2
1.2. Two Type of Dialog System: Goal-oriented and Chat-oriented . . .	3
1.3. Challenges in Developing Chat-Oriented Dialog Systems	5
1.4. Contributions of This Research	7
1.5. Thesis Outline	9
2. State-of-the-art in the Chat-Oriented Dialog System	13
2.1. Example Based Dialog Management Architecture	16
2.2. Word Vector Representation	17
2.3. Response Retrieval with Cosine Similarity	18
2.4. Response Retrieval with Syntactic-Semantic Similarity	19
2.5. BLEU: Bilingual Evaluation Understudy	19
2.6. Dialog System Evaluation	20
3. Construction of Multi-Domain Dialog Corpora	24
3.1. Related Works	26
3.2. Preprocessing	27
3.3. Filtering	27
3.3.1. Tri-Turn Extraction	27
3.3.2. Semantic Filtering	28

3.4.	Experimental Set-up	29
3.5.	Evaluation of Tri-turn and Semantic Filtering	31
4.	Combination of SMT and EBDM for Chat-Oriented Dialog Systems	34
4.1.	Technology of Statistical Machine Translation	34
4.1.1.	Phrase-Based Translation Model	34
4.1.2.	Language Model	35
4.1.3.	Learning in SMT	36
4.2.	Related Works in SMT	37
4.3.	Response Generation with SMT	38
4.4.	EBDM and SMT Hybrid Approach	38
4.5.	Experimental Set-up	39
4.6.	Evaluation of SMT Approach	39
4.6.1.	Objective Evaluation	39
4.6.2.	Subjective Evaluation	43
4.6.3.	Discussion	45
5.	Deep Neural Network for Chat-Oriented Dialog Management	48
5.1.	Technology of Deep Neural Networks	48
5.1.1.	Basic Artificial Neural Network	48
5.1.2.	Recursive Neural Network	51
5.1.3.	Recurrent Neural Network	52
5.2.	Related Works in DNN	54
5.3.	Neural Network Word Representation	55
5.4.	Recursive Auto Encoder Response Retrieval	55
5.4.1.	Recursive Auto Encoder	57
5.4.2.	Dynamic Pooling and Softmax Layer	58
5.5.	LSTM Response Retrieval and Generation	59
5.5.1.	Long Short Term Memory Neural Network	59
5.5.2.	LSTM Response Generation	61
5.5.3.	LSTM Response Retrieval	61
5.6.	Experimental Set-up	61
5.6.1.	Paraphrase-based Retrieval Setup	62

5.6.2. LSTM Network Setup	63
5.7. Evaluation of RAE and LSTM	64
5.7.1. Objective Evaluation	64
5.7.2. Subjective Evaluation	67
5.7.3. Discussion	69
6. Conclusion and Future Direction	74
6.1. Conclusion	74
6.2. Future Direction	77
References	80
Publications	89
Appendices	91
A. Conversation Database Structure in JSON Format	91
B. Dialog System Evaluator Desktop Application	93
C. Dialog System Web Demo Application	95

List of Figures

1.	Conversation between user and dialog agent in goal-oriented and chat-oriented dialog system.	4
2.	Contributions Matrix.	9
3.	Thesis overview.	10
4.	EBDM illustration.	16
5.	One-hot vector representation.	17
6.	Automatic evaluation on dialog system.	22
7.	Dialog corpora construction from movie script.	24
8.	Dialog corpus construction from movie scripts.	26
9.	Example of a tri-turn with two actors.	27
10.	Percentage of total characters involved in one movie.	29
11.	Filtering effect on the movie data.	32
12.	Filtering effect on the Twitter data.	32
13.	Word Alignment Matrix.	37
14.	Objective evaluation results on the movie data by various data-driven approaches over the cosine TF-IDF similarity (top) and syntactic-semantic similarity (bottom) metric.	40
15.	Objective evaluation results on the Twitter data by various data-driven approaches over the cosine TF-IDF similarity (top) and syntactic-semantic similarity (bottom) metric.	41
16.	Objective evaluation results of the combined system given various thresholds (axis).	42
17.	Combined retrieval approach on cross-domain using syntactic-semantic similarity as an evaluation metric.	43

18.	Subjective evaluation result on the movie and Twitter data by various data-driven approaches.	44
19.	Neural network perceptron.	49
20.	Long Short Term Memory Neural Network Cell.	53
21.	Overview of neural-network-based retrieval.	56
22.	Recursive autoencoder model.	57
23.	LSTM neural model over time.	60
24.	LSTM model perplexity.	64
25.	Objective evaluation results over the cosine TF-IDF similarity (top) and BLEU-4 (bottom) metric.	65
26.	Naturalness (top) and Relevance (bottom) for each system.	67
27.	Future work matrix.	78
28.	Conversation database structure in JSON format.	92
29.	Screen shot of the dialog system evaluator desktop application.	93
30.	Screen shot of the dialog system web demo application.	95

List of Tables

1.	Conversation corpus details.	30
2.	Comparison with retrieval based distributed representations. . . .	66
3.	Various responses for each dialogue system.	69
4.	This table shows a correlation between two sentences, user input and example database. We calculated syntactic-semantic score <i>sim</i> [1] for each utterance pair (<i>S1</i> and <i>S2</i>).	70
5.	Response preference percentage between LSTM-GEN and CSM baseline.	71
6.	Summary of various approaches to dialog response creation. . . .	76
7.	Turn class structure.	91
8.	Tri-turn class structure.	92

1. Introduction

1.1. General Human to Machine Conversation

One branch of computer science that has gained in popularity is artificial intelligence. Artificial intelligence is a branch of computer science focused on making computers behave like humans. It attempts to create smarter computers for users to interact with. It aims to create human-like computers that mimic human activity and behavior, and do so on their own.

One factor that separates humans from animals is the ability to use a language for communication. While some animals may use a simple language, humans can master and use complex spoken language to communicate. With computers, scientists try to understand speech through an automatic (computer) speech recognition system. Such systems attempt to recognize speech by building word models from sequences or phonetics segments derived from abstract linguistic representations of speech called “phonemes” [2]. A simpler yet still difficult task is making the machine able to understand the written text. In order to understand a single statement completely, one should consider the vocabulary, grammar, conversation context, etc. Natural language processing is one field of artificial intelligence that is attempting to understand this phenomenon and use it to make computers more intelligent.

Recent advancements in machine learning, especially in speech and signal processing such as speech recognition and synthesis, have allowed for this technology to be integrated into gadgets used in everyday life [3]. Given these necessary building blocks, and the recent advancements in machine learning techniques, it is easy to see how an interface for communication between humans and computers (natural language dialog system) is very promising in the future.

The presence of dialog systems has become increasingly important in many

aspects of human life. Ranging from the smart personal assistant dialog system, booking assistant dialog system, to a car navigation dialog system, we can see that dialog systems come as a cheap solution to replace expensive human services. Furthermore, with the extent of the internet, the spreading of these systems has become more effective as a service solution in remote areas.

Before we further discuss the dialog system, we must separate the dialog system topics into three main themes: First is the face-to-face dialog system. This dialog system enables humans to have two-way physical interaction, usually characterized by a facial recognition feature, avatar communication, or multimodal interaction. Next is an end-to-end spoken dialog system that supports signal/voice interaction between humans and computers. This type of dialog system incorporates automatic speech recognition, (ASR), and speech synthesis modules. Lastly is the text-to-text dialog system. This dialog system focuses on the text conversation between user and computer. An example of this dialog system is a chatter robot, (chatbot). This type of dialog system often becomes an option when the dialog scientist wants to focus their experiment on improving response robustness, dialog strategies, and/or dialog management. In this paper, we will focus on text-to-text dialog system setup. This was selected to focus on improving the response robustness of dialog system and to improve the user experience by providing a natural sentence as a dialog response.

1.2. Two Type of Dialog System: Goal-oriented and Chat-oriented

Natural language dialogue systems have so far mostly focused on two main dialogue genres: goal-oriented and chat-oriented dialog.

Goal-oriented dialog is a type of dialog system that focuses on finishing a certain task or job as soon as possible. It focuses on a certain or limited domain, and thus requires this dialog system in order to be trained with a specific knowledge of a certain topic. In this type of dialog system, the system usually has a predefined slot domain and task flow [4]. Goal-oriented systems run by performing a conversation that is based on the task flow

scenario while filling the required information slots, such as place name, price range, etc. Some examples of goal-oriented dialog system are ATIS flight reservation [5], and DARPA Communicator dialog travel planning [6].

Chat-oriented dialog on the other hand, doesn't follow a specific task flow and slot domain. Instead of a specific slot domain, chat-oriented dialog systems employ statistical methods [7] that cover a broad range of topics but do not possess the depth to perform a deep discussion. Some examples of this type of system are chatterbot systems like Eliza [8] or Alice [9]. To grasp this concept clearly, conversation snippets of the chat-oriented and goal-oriented dialog systems are provided in figure 1.

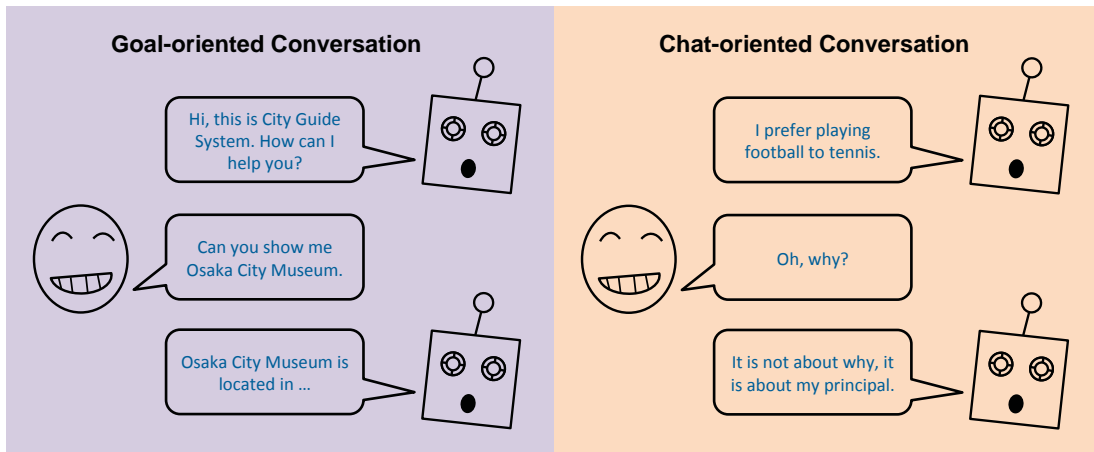


Figure 1.: Conversation between user and dialog agent in goal-oriented and chat-oriented dialog system.

Furthermore, dialog systems can also be described by the amount of human intervention used in their construction, ranging from entirely hand-made to completely data-driven. Seminal works often limit interactions to a specific scenario (e.g. a Rogerian psychotherapist [8]) or were based on complex, knowledge-rich, rule-based systems for generating responses, which require large amounts of human effort to create or add new rules [9].

1.3. Challenges in Developing Chat-Oriented Dialog Systems

The Turing Test is a type of behavioral test in the artificial intelligence field that assesses for the presence of mind, thought, or intelligence in what are generally believed to be mindless entities. The first formal instance of this test being applied is The Loebner Prize*. Established in 1990 by Hugh Loebner, this contest gives a prize for the computer whose responses are the most indistinguishable from a human's.

As an ultimate test to the “thinking” computer, one can assume that this competition produces the most human-like computer in the chat-oriented dialog task. To this date, state-of-the-art chat-oriented dialog systems are mainly dominated by rule-based systems, such as AIML[†] and ChatScript[‡]. Stuffed with rule-based dialog technology contestants, this competition becomes a match between who has the best chat rules that cover the vast topics given by the judges.

In the long run, while rule-based systems could produce a natural human-like response to the user, they do not really solve the big problem of creating an intelligent machine that can interact with humans naturally. Aiming for this overall goal, the team broke down several problems that needed to be solved, starting with the data-driven dialog systems. The team employed a data-driven dialog system, because it doesn't rely on conversation rules. As described in the previous section, given the current computer technology and machine learning advancements, one could easily see this approach as a promising direction in the future.

While there are several issues related to the chat-oriented dialog systems [10], solving these problems could enable an ideal chat-oriented dialog system:

1. The first issue in creating a good chat-oriented dialog system in a data-driven manner is **data availability**. The data used for this task should cover natural human-to-human conversations. With a natural conversation database and vast conversation topics, at least one could establish a

*<http://www.loebner.net/Prizef/loebner-prize.html>

†<http://www.alicebot.org/aiml.html>

‡<http://chatscript.sourceforge.net/>

well-constructed dialog system. In the end, collecting the ideal data for a conversation task is not a trivial job, because language develops along with human culture. Thus, a collected conversation database can easily become out of date, and strategies to update the conversation database become necessary.

2. The next problem is **selecting a good response candidate**. The perfect conversation database is meaningless if it is not used properly. In a data-driven dialog system, a good algorithm should be employed to retrieve response candidates efficiently and effectively. This task is not easy, as the computer needs to be able to understand the user sentence in order to select the appropriate response from the response pool in the conversation database.
3. However, even given the perfect conversation database and a flawless algorithm to exploit it, that does not make the dialog system perfect. Another problem that could arise in a statistical dialog system is **out of example (OOE) errors**. This problem occurs when the system handles an example that is not available in the conversation database. When this happens, the system response is incomprehensible and/or uncorrelated to the user utterance. To resolve this problem, the system should first detect when OOE occurs, and then decide what strategy should be used to handle it.
4. A good chat-oriented dialog system should be able to **comprehend the user's intentions and emotions**. In a good conversation, people should be able to grasp the situation by understanding the other person's intentions and/or emotions during the conversation. An ideal chat-oriented dialog system should be capable of this task, understanding both the situation and the context of conversation. For example, when talking about fruit, when the word orange comes, the system should be able to understand that the user means the fruit orange, not the color orange.
5. The chat-oriented dialog system should be able to **maintain a long and interesting conversation** with the user. This is not an easy task, as there are a lot of factors involved in maintaining a conversation with user. The

system needs to adapt to the user and be able to predict what kind of responses will ensure the user isn't bored with the conversation. Furthermore, the system might need to think about a good conversation strategy, maintain the response consistency, and even be able to predict the user's emotional state during the conversation.

1.4. Contributions of This Research

In this project, our aim was to create a dialog agent that can interact with the user in as natural a fashion as possible. We proposed the creation of a new kind of dialog architecture by employing basic techniques for data collection, creating an effective dialog system, and eventually overcoming the major challenges described before.

In this paper, there will be a focus on addressing dialog system problems such as data availability, selecting a good response candidate, and out of example (OOE) problems. Aligning our contribution with the dialog system problems that we presented before, our contribution can be formulated as follows: 1) To deal with the lack of data availability that encompasses natural conversation (1st problem), we constructed a conversational database based on movie scripts and Twitter conversations (1st contribution). 2) In order to select a good response candidate (2nd problem), we establish a hybrid approach that utilized both EBDM and SMT (2nd contribution). 3) Handling the OOE errors (3rd problem), we constructed a dialog retrieval with a neural-network based paraphrase-matching algorithm, and response generation and retrieval with an LSTM neural-network (3rd and 4th contributions). We analyzed and compared proposed system performances through a) objective evaluation to measure word ordering, syntactic, and semantic aspects, and b) subjective evaluation through human judgment (5th contribution).

As detailed below, we made five contributions:

1. We proposed a method to utilize human-to-human conversation examples from movies and Twitter data. The aim was to gain insight into how to build a conversational agent that can interact with users in as natural

a way as possible, while reducing the time requirement for database design and collection. Then, to help ensure that the content extracted from raw movie/drama script files consisted of appropriate dialog-pair (query-response) examples, we proposed using a unit called a tri-turn for extraction, as well as semantic similarity analysis techniques.

2. We investigated various data-driven approaches to dialog management, including two EBDM techniques (syntactic-semantic similarity retrieval and TF-IDF based cosine similarity retrieval) and using phrase-based SMT to learn about conversational mapping between user-input and system-output dialog-pairs. We also proposed a simple, yet effective method to combine system combinations of example-based and SMT-based techniques into one dialog management framework. Experimental results demonstrate that our combined system shows promise for overcoming the shortcomings of each approach.
3. We proposed a new EBDM method to retrieve dialog responses from the database by utilizing a neural-network based paraphrase-matching algorithm. In this approach, we modeled the example in our dialog-pair database and the user input query with distributed word representations, and employed recursive autoencoders and dynamic pooling to determine whether two sentences with arbitrary lengths have the same meaning.
4. We proposed a method that utilizes LSTM neural networks to perform response retrieval, in addition to generating responses directly. Given the LSTM response generation model, we calculated the perplexity of each response in the database based on the user query, and were able to obtain the best-scoring response candidate directly from the dialog database. This way we were able to reduce the chance of grammatical errors that occur when generating dialog responses, while using the ability of neural networks to perform soft matching to improve retrieval accuracy.
5. We performed an analysis and contrasting experiment to compare our proposed approach with the state-of-the-art baseline approaches in data-driven chat-oriented dialog. We evaluated the dialog system performance using

both automatic and manual evaluations, over examples that are well covered by the example base, as well as examples for which a close match does not exist.

A comprehensive list of these contributions towards the observed problems can be seen in the contributions matrix (see Figure 2) below.

Approaches and Solutions	4) Response generation and retrieval with long short term neural (LSTM) neural-network			
	3) Dialog retrieval with neural-network based paraphrase-matching algorithm			
	2) Establish a hybrid approach between EBDM response retrieval with cosine TF-IDF vector and statistical machine translation (SMT) approach			
	1) Construct a conversation database based on movie script and Twitter conversation			
		1) Data availability for natural conversation	2) Selecting a good response candidate	3) Out of example (OOE) error
		Observed Problems		

Figure 2.: Contributions Matrix.

1.5. Thesis Outline

This thesis is organized as shown in figure 3.

Chapter 2: Here is presented the basic framework for chat-oriented dialog: example based dialog management (EBDM). In this chapter a baseline retrieval algorithm using cosine similarity will be described. Next a standard

sentence representation used for the sentence similarity calculation will be introduced. At the end of this chapter, the team’s experiment assumption and evaluation of the dialog system methods will be discussed.

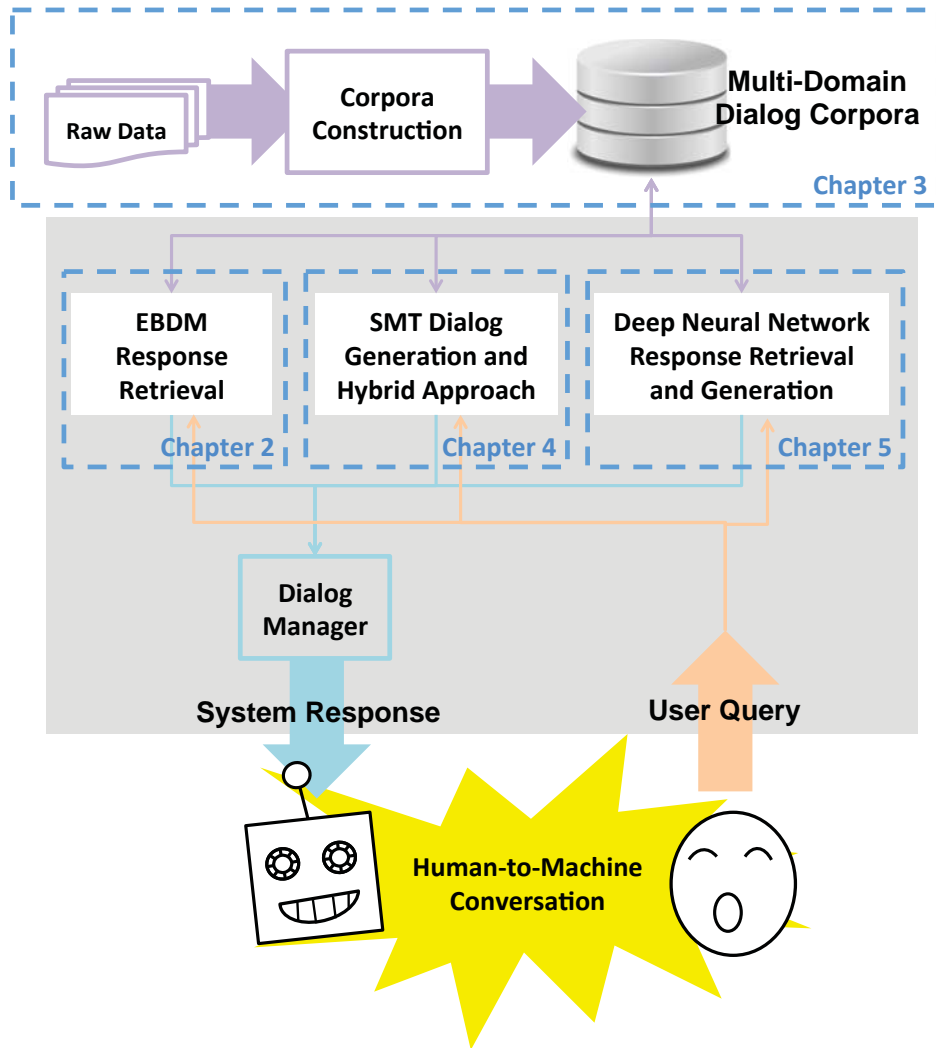


Figure 3.: Thesis overview.

Chapter 3: Here, our conversation database data collection techniques will be explained. As previously stated, raw movie scripts and Twitter conversations obtained through the internet were used as the data source. We also introduced tri-turn extraction and semantic similarity filtering to

construct a strong conversation database.

Chapter 4: In this chapter, statistical machine translation (SMT) for chat-oriented dialog management will be discussed. By treating the query-response sentence in the conversation database as a parallel corpus (as in the machine translation task), we built a response generation system using SMT. The advantages and disadvantages of this approach will be presented, leading to an explanation of a novel hybrid dialog system, which combines the baseline response retrieval and SMT approaches.

Chapter 5: Here we focus on the application of deep neural networks in dialog system response retrieval and generation. This experiment was performed in order to devise a method for handling the out of example (OOE) problems in data-driven dialog system tasks. Presented here are two approaches in neural network dialog architecture: the first is in regards to response retrieval using recursive auto-encoder paraphrase identification, and the second is the utilization of a long short-term memory neural network for dialog generation and retrieval.

Chapter 6: The conclusion of the effectiveness of our proposed systems that leverage statistical approaches for chat-oriented dialog systems, as well as proposals for future directions are presented here.

2. State-of-the-art in the Chat-Oriented Dialog System

In the past few years, the state-of-the-art chat-oriented dialog systems have been dominated by rule-based systems, such as AIML and ChatScript. However, this system relies heavily on rule-bases and there is not much that scientists can do other than evaluate the effectiveness of the chat-rules. Furthermore, some advanced hand-crafting rules suffer from the domain portability problem, which requires the dialog designer to restart the design process when developing a new application for a different domain. For example, to add a greetings module to a rule-based system, the dialog designer needs to develop a new rule to support that function.

Some early dialog systems such as ELIZA [8], SUNDIAL [11], and ARISE [12] were designed by software engineers to have domain-specific knowledge. These systems are usually restricted to a highly structured task, where a certain language set can be expected. This knowledge-based approach generally involves a finite-state automaton to its dialog strategy, thus making these systems require hand-crafted rules. Consequently, this approach requires continuous experiments with the users. This rapid prototyping of dialog system ensures strong-typed interactions with clearly-defined structures conversation [13].

A generic dialog modeling approach based on agenda and task models [14–16] was developed to overcome limitations in the typical rule-based systems. These systems were designed to avoid a domain portability problem, that is, dialog designers must perform rapid prototyping and also redesign the hand-crafted rules in the dialog. However, the design process of this dialog is still expensive and time-consuming because the knowledge sources (such as plan, hierarchical task structure) are usually designed by human experts.

Dialog scientists exploit the development of data-driven approach in automatic speech recognition (ASR) and natural language generation (NLG) technology. Although it still requires data annotation, the training process is done automatically and requires little human supervision. New dialog systems can be built by collecting and learning from the data (or specific domain data), thus spending less time and effort than the previous rule-based approach required. This has motivated the development of stochastic dialog models using reinforcement learning (RL) based on Markov decision processes (MDPs) or partially observable MDPs (POMDPs) [17, 18]. These mark a new era in data-driven dialog system development.

In real dialog application however, the deployment of RL encounters several problems, such as hand-crafted and tuned reward function, optimized policy, (which may remove control from application developers,) and such [19]. Although many dialog researchers are solving these problems [4, 20–22], this approach still needs some improvements before it can be applied to practical dialog systems [23].

Another data-driven approach for deploying dialog systems is Example-Based Dialog Modeling (EBDM). EBDM uses dialog examples that are semantically indexed to a database. Proper responses for user input are generated based on these dialog examples. In EBDM, one can easily determine a set of variables without a complexity problem, and the dialog flows can be easily controlled with the dialog examples [23].

While promising, these data-driven approaches are highly dependent on the amount of data collection. Consequently, to achieve good coverage on various types of natural conversation, recording a large data set of real human-to-human conversation is necessary, which is tedious and time consuming. Common solutions use handmade scripted dialog scenarios that may result in unnatural conversations. NPCEditor [24], a tool for building and collecting conversation corpus, is a very useful to collect stateless or semi-stateless conversation data. However, it is not appropriate to collect a conversation in which the characters have initiative and perform complex contextual or state-based reasoning. Other studies also propose constructing dialog examples from available log databases, such as conversations between human subjects and the Wizard of OZ (WOZ) system [25], or human-to-human conversation on Twitter [26]. However, covering

all possible patterns that may exist in real human-to-human conversation is still difficult. Currently, most EBDM systems rely on either canned responses, by providing error messages, [23] or templates for generation, which may result in a completely incomprehensible response [27].

My focus is in the developing of EBDM technology. EBDM has flexibility, in that it allows the extension module for variable and dialog flows to be determined and controlled. As a data-driven approach in dialog, it could employ any other domains besides the chatting purpose by a simple change in the dialog corpora. The EBDM system can also be combined with the ASR and NLG modules, which are easily trained with little-to-no human supervision.

EBDM performance in general can be improved by first improving its response naturalness. This can be done by using real human-to-human conversation data that is incorporated from a movie script and/or Twitter data. The next aim is to increase the response retrieval robustness in OOE cases, which is important in the development of a chat-oriented dialog system. In a goal-oriented system, off-topic utterances can be detected and handled easily by recognizing the unknown vocabulary and preparing hand-crafted responses. In a chat-oriented dialog system, the system is expected to understand the user query and respond naturally, although if it faces an unknown term or utterance pattern that is not available in the database. A hand-crafted response might not be an option because it will be easily detected when the OOE case occurs more in the conversation. To deal with these problems, we employ various techniques to scale up the EBDM model. These techniques include a hybrid approach between response generation and retrieval, a paraphrase identification response retrieval, and a neural network-based response retrieval that is trained end-to-end over the conversation database.

It is also important to be aware of the other development in end-to-end dialog system using neural network [28–34] that is currently being developed. By using a structure similar to memory network [35], these end-to-end systems can be used to train input-output pairs that are applicable to a task where supervision is not available, such as language modeling, question-answering, or a simple chit-chat task. This type of dialog system generates a lot of interest because they don't have limitations and their components are directly trained over dialog logs and don't have any assumption on the domain or on the structure of the dialog state.

Some of these dialog systems have even reached promising performances in chat-oriented dialog systems, predicting a next utterance in social media or forum threads [36–38].

2.1. Example Based Dialog Management Architecture

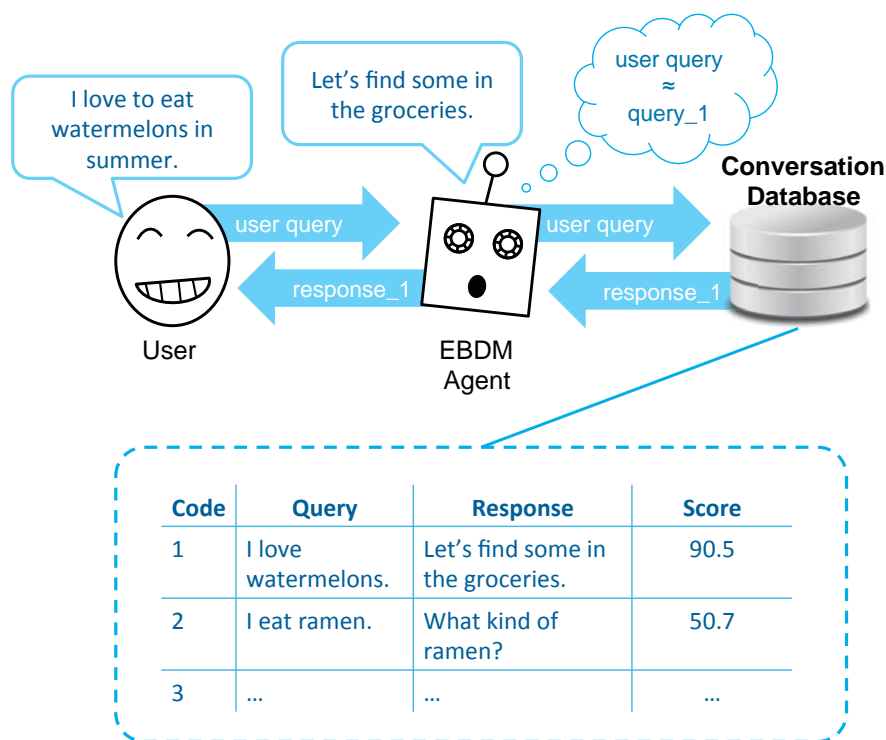


Figure 4.: EBDM illustration.

In EBDM, the system chooses a response from the examples stored in the dialog database. In order to do so, it computes a similarity measure between the user input and the query part of the query-response pairs, and returns the associated response for the query with the highest similarity. To understand this concept clearly, we can take a look at the illustration depicted in Figure 4.

There have been a number of related works published on the topic of EBDM

usage in data-driven chat [39–41]. Work by Lee et al. [23] proposes a generic dialog-modeling framework for multi-domain dialog systems to simultaneously manage goal and chat oriented dialogs for information access and entertainment. However, the chat-oriented dialog only covers small talk that is limited to 10 topics of daily conversation. If the system cannot find similar examples to determine the next system action, it simply defines a “No Example” output error and provides an in-coverage example of what the user could say at the current dialog state. Finally, Banchs et al. [42] introduce IRIS (Informal Response Interactive System), a chat oriented dialog system using movie scripts that is based on a similar cosine similarity in vector space model. However, the system did not filter any uncorrelated consecutive scripts in the movie data, and, as the authors state, this causes failures and diminishes the ability to maintain a consistent conversation.

2.2. Word Vector Representation

In natural language processing tasks, we need to convert text into a mathematical unit that is easy to compute. A simple way to do so is using a word vector representation. There are several ways we could represent our sentence as a vector. In this section we will explain one-hot bag-of-words sentence representation [43].

Vocabulary	I	you	love	hate	orange	apple	and	
Sentence 1 (S1) [I love apple]	1	0	1	0	0	1	0	→ $S1 = [1, 0, 1, 0, 0, 1, 0]$
Sentence 2 (S2) [I hate you]	1	1	0	1	0	0	0	→ $S2 = [1, 1, 0, 1, 0, 0, 0]$

Figure 5.: One-hot vector representation.

Here we use the term “vocabulary”. Vocabulary is the bag-of-words or all the unique observable words in the corpora. If we have a vocabulary with N unique words, we can imagine a vector with length N , where each word correspond to each slot in the vector. For each sentence, if the vector value slot is “1” it means that the corresponding word occurs in the sentence. On the other hand, if the vector value slot is “0” it means that the corresponding word does not occur in the sentence. Thus it is called one-hot (1 and 0) representation [44]. To clearly

understand this concept please refer to the figure 5.

By converting the sentence into a vector form, we can perform mathematical operators on the sentence. In more advance representations, the value “1” in the one-hot representation could be replaced with the TF-IDF value, thus making this vector a TF-IDF vector. In contrast, neural network models usually use a distributed word representation form. This is an n -dimension of a vector of real numbers that represent a word. More information about TF-IDF calculation can be acquired in the next section, and detailed explanation of distributed word representation can be seen in Chapter 5.

2.3. Response Retrieval with Cosine Similarity

In this work, we examine syntactic-semantic similarity and TF-IDF based cosine similarity as two similarity measures for use in EBDM. Cosine similarity with regards to the term “vector” as described in Equation 2.1 is used to retrieve a proper system response. To increase the emphasis on important words, an additional TF-IDF weighting (Equation 2.2) is performed [45].

$$\text{cos}_{sim}(S_1, S_2) = \frac{S_1 \cdot S_2}{\|S_1\| \|S_2\|}. \quad (2.1)$$

$$\text{TFIDF}(t, T) = F_{t,T} \log \left(\frac{|T|}{DF_t} \right) \quad (2.2)$$

We define $F_{t,T}$ as a term frequency ‘ t ’ in a sentence ‘ T ’, and DF_t as the total number of sentence in the query-response pairs that contain term ‘ t ’.

In this response retrieval approach, our dialog manager traverses all the query-response pairs in the conversation database. Because the system calculates the cosine similarity of the entire conversation database, we can expect the complexity of retrieving one single response in this approach as $O(N)$, where N is the amount of query-response pairs in the conversation database.

Another method that is used heavily in natural language task is kernel methods [46, 47]. There are various kernel methods that can be utilized. However, the radial basis function (RBF) kernel is the most popular and is widely used in

various kernel algorithm or support vector machine classifications [48]. In this study, the team limited itself to the cosine similarity measurement method that is widely used in response retrieval or other dialog tasks [23, 38, 42, 49, 50]

2.4. Response Retrieval with Syntactic-Semantic Similarity

We also implemented another similarity measurement that employs both semantic and syntactic relations. These two measures were combined using linear interpolation as shown in Equation (2.3). This value is calculated from the user inputted sentence (S_1) and every input example contained in the database (S_2). These values are calculated using semantic similarity in WordNet (will be discussed in Chapter 3) as a semantic factor and part-of-speech (POS) tag cosine similarity 2.1 as a syntactic factor.

$$sim(S_1, S_2) = \alpha[sem_{sim}(S_1, S_2)] + (1 - \alpha)[cos_{sim}(S_1, S_2)]. \quad (2.3)$$

In this experiment, we assumed that the semantic factor was more important than the syntactic factor, so we set the interpolation coefficient α to be 0.7.

As with the cosine similarity response retrieval, our syntactic-semantic response retrieval approach also traversed all the query-response pairs in the conversation database. It calculated the syntactic-semantic similarity over the entire conversation database. In this approach, the cost of retrieving one single response is $O(N)$, where N is the amount of query-response pairs in the conversation database.

2.5. BLEU: Bilingual Evaluation Understudy

In order to measure a sentence similarity based on its local word order, we use BLEU score [51]. BLEU score combines the modified precisions p_n for the various n-gram sizes with the sum of average logarithm. BLEU score can be calculated as follow:

$$BLEU = \min\left(1, \frac{0}{r}\right) \sum_{n=1}^N w_n \log p_n, \quad (2.4)$$

where o is the output response length, r is the reference response length, N is the n -gram length, and w_n is uniform weight. In our experiment, we used the most common setup which employs 4-gram and sets uniform weight $w_n = \frac{1}{N}$.

2.6. Dialog System Evaluation

How to evaluate dialog systems?

To this date, there is no standard method for evaluation of chat-oriented dialog systems, which adds another interesting challenge to the development of chat-oriented systems and dialog systems in general. A common approach to evaluating dialog systems is subjectively asking the opinion of users to gather insight on response naturalness, relevancy, and so on. [23,28,52]. Another variation is using crowd sourcing [30,36]. Recently some dialog researchers have also been using machine translation metrics like BLEU to judge the quality of the generated dialog responses [29].

As a comparison, evaluation in goal-oriented systems is done by using a wide range of well defined evaluation benchmarks that measure the ability to track user states and/or to reach user-defined goals [53–56]. Unlike a goal-oriented system, a chat-oriented dialog system doesn't maintain specific states in it. A recent development on the end-to-end chat-oriented dialog system evaluates the dialog system performance by classifying each response to the predefined skill/topic-sets [57–59]. This way, researchers can identify and improve their systems. Note that this approach was developed after our research was completed.

NTCIR STC* is a dialog system competition aimed to establish a short one-round conversation. (this competition is currently underway.) This task provides a large amount of conversation data extracted from Twitter and Weibo[†] and requires its participant to develop a conversation system mainly based on the information retrieval (IR) technologies. The evaluation of this task is done using IR metrics such as precision, mean average precision (MAP), normalized discount cumulative gain (nDCG), and others. Applying these metrics to the study might

*NII Testbed and Community for Information access and Research Short Text Conversation, <http://ntcir12.noahlab.com.hk/stc.htm>

[†]<http://overseas.weibo.com/>

result in a 0 score of MAP and nDCG. This happens because the focus is on OOE cases in chat-oriented dialog systems, where the normal response retrieval can not find the good response in the conversation database, often resulting in a random match between user query and retrieval results.

Dialog System Evaluation in This Study

While there is no standard evaluation in the dialog system task, here we evaluate the system response objectively by calculating the system output \hat{R} similarity compared to the actual expected output R .

$$\text{evaluation score} = \text{similarity}(\hat{R}, R). \quad (2.5)$$

During the automatic evaluation process, we used set response pairs in the conversation database as model responses when training the systems. To obtain the evaluation score, in this study we used similarity measurements from TF-IDF cosine similarity, syntactic semantic similarity, and BLEU. This was essential, as performing a human based evaluation would have been very costly. One can develop several dialog system prototypes and run this kind of automatic test to assess which approaches are better and what cases are not appropriate for human evaluation. In this way, bad approaches can be efficiently detected, and confidence in testing our dialog system with real users increases.

This objective evaluation proceeds as follows. First, we took a query-response test-bed pair (Q, R) from the conversation database (Q', R') , and treated it as a both a user query Q and response reference R consecutively. Next, we took the user query Q to the dialog agent, and collected the response \hat{R} from the dialog agent. We calculate the score from the similarity between the response of dialog agent \hat{R} and our response reference R . By doing this, we assumed that the expected appropriate response would be a response which was similar to the response reference. In order to understand this idea clearly, we depict this concept in figure 6.

Beside the objective evaluation, we also performed a subjective evaluation. Subjective evaluations are carried out by asking a human to judge. Each person was given a pair of user query and system responses at random. We asked the human annotators to give a score between 1-5 for each system response. This

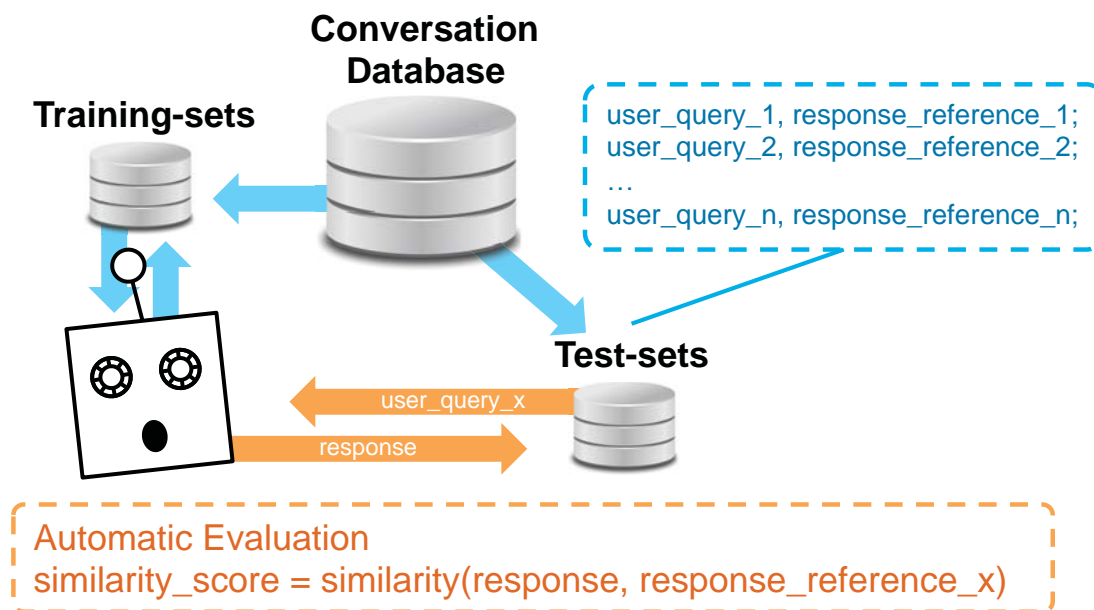


Figure 6.: Automatic evaluation on dialog system.

score reflects how natural and relevant each of the responses were. 1 represents poor performance/low quality answer and 5 represents high performance/high quality answer.

Overall, we assess our system performance with objective and subjective metrics. Through the subjective evaluation we assess system performance aspects such as response naturalness or relevance to the real user. From the objective evaluation we are able to assess the system performance by some specific features, such as: word ordering in a BLEU evaluation, syntactic and semantic cohesion in syntactic-semantic sentence similarity and TF-IDF cosine similarity. Our objective evaluation metrics are aimed towards providing a comprehensive analysis of our system performance. Later on in this paper these dialog system responses will be analyzed based on word-ordering, syntactic, and semantics aspects.

3. Construction of Multi-Domain Dialog Corpora

My team’s dialog corpora is constructed based off of movie scripts and Twitter conversations. These sources have been chosen because they resemble actual conversations between humans, and are easy to obtain from the internet. Because the movie scripts and Twitter data used in this work contain very different types of text, different processes must be used to construct them.

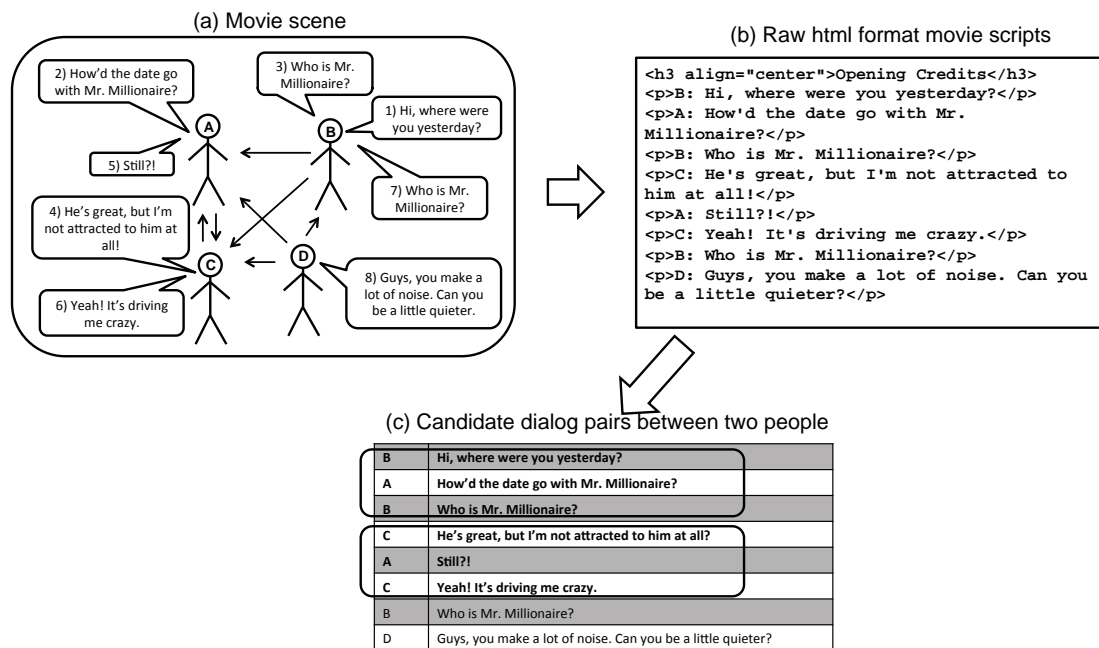


Figure 7.: Dialog corpora construction from movie script.

A movie script is a conversational manuscript that portrays the conversation between and actions of actors in a movie. Figure 7(a) illustrates an example of

one movie scene with four actors talking to each other. The corresponding raw movie scripts that are available from the web are usually written in HTML files shown in Figure 7(b). The dialog between actors is arranged in chronological order. Consequently, the conversation dialog contained in movie scenes does not have a clear indication of which utterances are responses to a particular utterance. Therefore, it is important to find a solution that is able to construct appropriate dialog-pair examples from raw movie script files. As shown in Figure 7(c), dialog tri-turn extraction is performed to find the candidate of dialog-pairs. Tri-turn is a three consecutive turn conversation in the movie script. However, it was found that tri-turn isn't always applicable in all two-way conversations. To ensure a strong relationship in these conversations, semantic similarity filtering is later applied.

In contrast to the movie script data, text from Twitter often represents real conversations between two or more people. Therefore, dialog tri-turn extraction is not necessary in order to extract the related dialog-pair sentences. Instead, the challenge with handling Twitter data is determining how to ensure the integrity of the sentences. In this case, it is necessary to filter out sentence pairs that are not likely to be useful for training the system.

Unifying both data sources into one dialog corpus, we define two basic types of information about each dialog: actor and utterance. The utterances are the actual content of each dialog turn in the movie scripts or tweets. The actor refers to the character name in the movies, or the name of the Twitter user that posted each tweet. This actor and utterance information will be utilized to construct the dialog corpus.

The details of dialog corpus construction, as illustrated in Figure 8, consists of three main steps: (1) pre-processing, which removes unnecessary information and normalizes the text, (2) dialog-pair extraction, which ensures that the conversation is between two people talking each other, and (3) semantic similarity filtering, which ensures that the each query-response pair is semantically related. Described are the details of each step in the following sections.

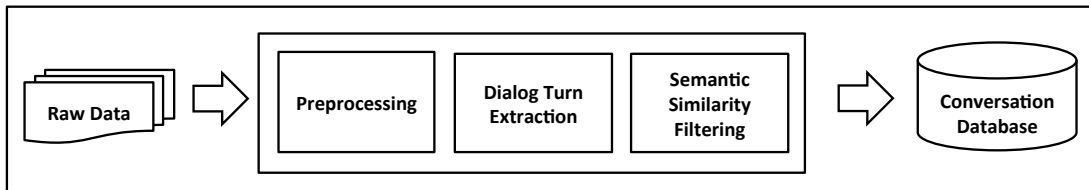


Figure 8.: Dialog corpus construction from movie scripts.

3.1. Related Works

In this work, a movie script and Twitter data obtained from internet are utilized. There are numbers of previous works utilizing these kinds of data for dialog system tasks [60]. Banchs presents a Movie DiC Corpus [42], this dialog’s data is collected from Internet Movie Script Data Collection*, and covers various genres. Ritter, et al. conducted a response generation experiment with the Twitter data collected from Twitter API [36]. There are also a number of works that utilize movie or TV Drama subtitles or scripts [61–65]. Walker, et al. [65] provides an analysis and annotation such as character sentiments and archetypes, which is useful for creating a personalized dialog system. Because a conversation obtained from a movie and from Twitter mostly covers a broad range of genres and topics, corpora created from this source, sometimes called multi-domain dialog copora.

These dialog corpora consist mostly of raw script, and might not portray an actual natural conversation between two people. We have gone a step further, by performing a filtering process over the collected dialog data. This way it can be ensured that the corpora is performed by two people that actually talk to each other, forming dialog-pair sentences. The dialog corpora constructed in this study are based on a movie script and Twitter data. This source of data is used because it portrays human-to-human dialogue [60], resembling what natural dialogue between humans.

*<http://www.imsdb.com>

3.2. Preprocessing

Preprocessing of the movie scripts is done by transforming raw HTML files into an easily readable text format. Since a variety of movie script sources were used that had differentiating formats, several parsing algorithms were implemented to fetch the information from the raw movie conversations. Unnecessary explanatory information about the movie scenes was removed.

Regarding the Twitter data, preprocessing removes information about the user’s identity, as well as removing hash tags and URLs. For both data sets, all the words in the sentences are labeled with parts of speech (POS) and named entities (NE). To ensure the integrity of the Twitter data, English language filtering[†] and non-standard word (NSW) normalization [66] is also performed.

3.3. Filtering

3.3.1. Tri-Turn Extraction

Actor	Correlated tri-turn
C	He's great, but I'm not attracted to him at all!
A	Still?!
C	Yeah! It's driving me crazy.

Actor	Un-correlated tri-turn
B	Hi, where were you yesterday?
A	How'd the date go with Mr. Millionaire?
B	Who is Mr. Millionaire?

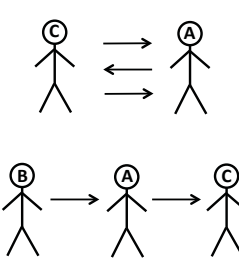


Figure 9.: Example of a tri-turn with two actors.

To certify that the dialog example database contains only query-response pairs, we proposed a simple and intuitive method for selection of the dialog data, namely, trigram turn sequences, or *tri-turn*. A tri-turn is defined as three turns in a conversation between two actors X and Y that has the pattern X-Y-X. In other words, within a tri-turn the first and last dialog turn are performed by the same

[†]search.cpan.org/~ambs/Lingua-Identify-0.51/

actor and the second dialog turn is performed by the other actor. Then, the query-response pairs are made by separating the tri-turn pattern X-Y-X into two pairs, X-Y and Y-X.

When observing this pattern, it was discovered that in the great majority of the cases, this indicated that the first and second utterances (X-Y pair), as well as the second and third utterances (Y-X pair), formed a proper input-response pair as shown in the c-a-c tri-turn in Figure 9. However, noisy cases which contain uncorrelated turns still exist (see the b-a-b tri-turn in Figure 9), this happens because the speakers are not actually speaking to each other. To address this problem, further filtering was performed using the semantic similarity measure described in the following section.

3.3.2. Semantic Filtering

Semantic similarity [67], shown in Equation (3.1), is used to make certain there is a strong semantic relationship between each dialog turn in the dialog-pair data. This is done by computing the similarity between WordNet[‡] synsets in each dialog turn. The dialog-pairs with high similarity are then extracted and included into the database. S_{syn1} and S_{syn2} respectively is a group of WordNet synsets for each word in the sentence S_1 and S_2 that are linked by a network of lexical relations. The similarity of sentence pair X-Y where $S_1 = X$ and $S_2 = Y$ can be obtained by calculating the relations between S_{syn1} and S_{syn2} . Where $|S_{syn1} \cap S_{syn2}|$ is a number of co-occurring WordNet synsets and $|S_{syn1}| + |S_{syn2}|$ is the total number of effective WordNet synsets.

$$sem_{sim}(S_1, S_2) = \frac{2 \times |S_{syn1} \cap S_{syn2}|}{|S_{syn1}| + |S_{syn2}|} \quad (3.1)$$

When dealing with the name entities (NEs), the system simply replaces the NEs with pronouns when performing the response retrieval. The NEs will be stored as variables and later can be used to replace NE slots in the retrieved/generated response.

[‡]<http://wordnet.princeton.edu/>

3.4. Experimental Set-up

We collected and constructed our conversation database from the Friends TV show[§], The Internet Movie Script Database[¶], and The Daily Script^{||}. Parsing the raw HTML data was accomplished with the Perl CPAN HTML-Parser^{**} and the filtering system was built in the Python environment using the Python NLTK tools^{††}.

From the raw data, 28.62% of the collected movie scripts are played by 11 - 20 different characters. Only 4.40% collected movie scripts are played by 1 - 10 different characters. Besides the main characters, the movie scripts usually include cameos (e.g. “*a man in the radio*,” “*man 1*,” “*radio*”). These cameo characters contribute to increasing the character variation in a single movie, which makes the filtering task is more challenging and explains the uncorrelated tri-turn during the data collection process. Figure 10 illustrates in detail the total number of different characters involved in one movie.

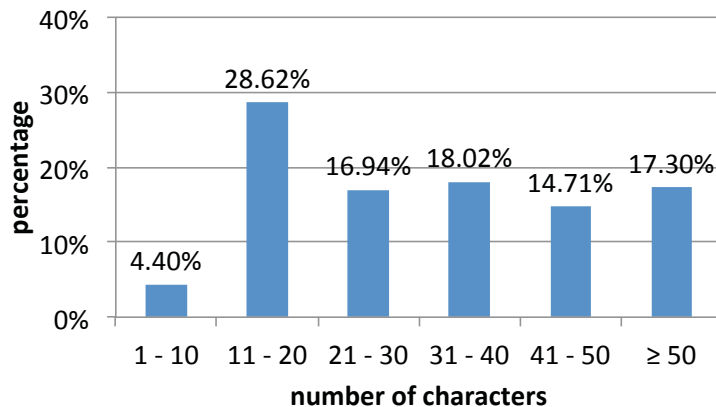


Figure 10.: Percentage of total characters involved in one movie.

The Friends TV show scripts are written in English and contain 5 seasons,

[§]<http://ufwebsite.tripod.com/scripts/scripts.htm>

[¶]<http://imsdb.com/>

^{||}<http://dailyscript.com/>

^{**}<http://search.cpan.org/dist/HTML-Parser/Parser.pm>

^{††}<http://nltk.org>

with a total of 112 episodes. Each episode contains several scenes and each scene contains several dialog turns. The total number of scenes in the corpus are 1,437. The movie script data is from The Internet Movie Script Database and The Daily Script, captured in June 2012. This resulted in a total of 1,786 conversation scripts with 1,042,288 dialog pairs. After performing dialog turn extraction and semantic similarity filtering, the total number of dialog pairs is 86,719. The summary of the conversation corpora can be seen in the Table 1.

conversation scripts	1,786
dialog pairs	1,042,288
dialog pairs after filtering	86,719

Table 1.: Conversation corpus details.

Additionally, we annotated every sentence in the dialog turn with labels such as part-of-speech tags (POS), named entities (NE), and dependency trees. POS, NE, and dependencies were tagged by using the Python NLTK Brown corpus POS Tagger, Stanford NER^{‡‡}, and the Stanford dependency parser^{§§}. We also added semantic and syntactic similarity distance between two sentences in the dialog pairs. The syntactic similarity distance obtained by calculating a syntactic similarity measure [67], given the dependency tree of a sentence as an input. Finally, we wrapped each dialog-pair with all of its annotation in JSON^{¶¶} data format.

For the Twitter data, Twitter *tweets* were collected through the Twitter API^{***}, resulting in a total of 1,076,447 dialog-pairs. After performing language filtering and semantic similarity filtering, the total number of dialog-pairs was reduced to 67,500 and 7,048 respectively.

^{‡‡}<http://nlp.stanford.edu/software/CRF-NER.shtml>

^{§§}<http://nlp.stanford.edu/software/stanford-dependencies.shtml>

^{¶¶}<http://json.org/>

^{***}<http://dev.twitter.com>

3.5. Evaluation of Tri-turn and Semantic Filtering

To demonstrate the effect of semantic similarity and tri-turn filtering in our data, we compared our system performance with and without the tri-turn filtering. In this comparison, the TF-IDF based cosine similarity (CSM) and syntactic-semantic similarity retrieval (SSSR) methods were used to retrieve responses.

Semantic similarity filtering was able to improve the performance significantly over the tri-turn filtering. However, the application of the tri-turn filtering had a role in reducing the amount of training examples while maintaining the evaluation score result. On the other hand, the difference in the amount of training examples resulting from the filtering process also effects the response retrieval time. Figure 11 and 12 depict average system evaluation score improvement and response times per input query for each applied filter. The average system evaluation score is the average performance of a certain evaluation score metric over all the test set data. The number in the horizontal axis shows the number of dialog-pairs after each filtering step.

In this experiment, we implemented our response retrieval approach without indexing optimization. All the words in the sentences were traversed to obtain the TF-IDF vector and WordNet synsets. In a practical situation dialog, the amount of query-responses in the database could be millions, and the response time becomes crucial as users may expect the response from chat-oriented dialog systems to be presented in real time. These filtering methods come to increase the conversation database effectiveness, and in this way the conversation database can be reduced and the response speed increased, while maintaining and even increasing the response retrieval performance.

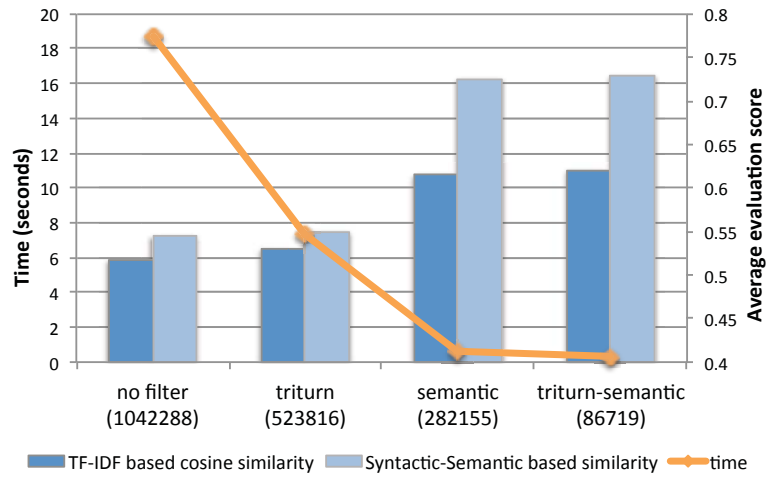


Figure 11.: Filtering effect on the movie data.

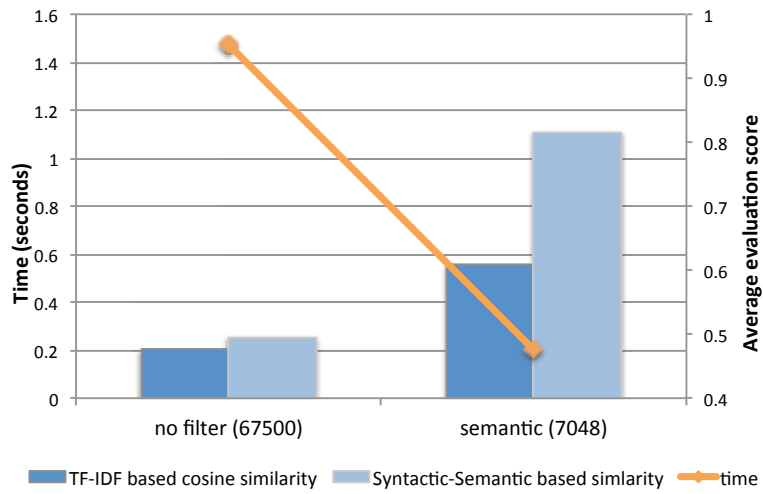


Figure 12.: Filtering effect on the Twitter data.

4. Combination of SMT and EBDM for Chat-Oriented Dialog Systems

Statistical Machine Translation (SMT) has been successfully used to address various NLP tasks [68–70]. The investigation of SMT as an approach for response generation has also been introduced by [36]. In this chapter we propose a simple but effective way to perform system combination of example-based and SMT-based techniques into one dialog management framework. Experimental results demonstrate that our combined system shows promise for overcoming the shortcomings of each approach.

4.1. Technology of Statistical Machine Translation

In this section we explain the fundamental technology of machine translation. Since our SMT is based on a phrase-based translation model, the focus of the discussion will be on the technology of phrase-based SMT. This section will be divided into three parts, a discussion of the phrase-based model in general, an explanation of the learning process in the SMT, and finally an explanation of the language model used in the phrase-based SMT.

4.1.1. Phrase-Based Translation Model

Before understanding the phrase-based translation model, word-based translation models must be discussed. Word-based translation models translate a single

sentence word by word. We can view this model as a person looking up a word in a dictionary to find the best translation for a single word in a sentence. In word-based translation models, one should consider 1) the translation table statistics by performing learning from the parallel corpus, and 2) word alignment in regards to how the word order in one language aligns itself with another language.

Besides word-based translation models, phrase-based translation models are also popular. Before the recent arrival of the neural machine translation era, this model has been the highest performing statistical machine translation system [71]. Unlike word-based translation models, phrase-based translation models translate small word sequences at a time. This model assumes phrases as a single atomic unit in the translation algorithm. The idea of a phrase-based model is based on the knowledge that a certain language word may be translated into two words in the other language.

Learning small word sequences at a time in a phrase-based model is very useful. The context learning in the phrase translation table can provide good clues as to how a language should be translated. Overall, there are several benefits to using the phrase-based translation model; due to many-to-many mapping, a word may not be the best atomic unit for translation. Translation ambiguity might be resolved by translating word groups instead of single words, and the large training corpora enables the phrase-based model to learn longer and more useful phrases, sometimes enabling the model to memorize the translation of entire sentences. Finally, the phrase-based model is more simple conceptually, since it removes the notion of insertion and deletion of the word-based model.

4.1.2. Language Model

Another essential component in statistical machine translation is the language model. In short, language models measure how likely a sequence of words would be uttered by a human speaker. This is important because it is desirable for the machine translation to produce not only the right words, but also a natural and plausible sequence of words.

The N-gram language model is one of the leading methods for language modeling. This statistical model analyzes the text in corpora and measures how likely words are to follow each other. Here we compute a probability of sentence

$S = w_1, w_2, \dots, w_n$, in such way that $p(S)$ is the probability of picking a sequence of words at random and turns out to be S .

In this task, the most common n-gram language model used is five-gram (5-gram) language model. To understand this concept clearly, from now on, we will discuss trigram (3-gram) model. The trigram model considers the usage history of two words (w_1, w_2) to predict the following third word (w_3) . To compute this 3-gram, we need to populate the statistics of three word sequences. Using maximum likelihood estimation, we can formulate 3-gram model as:

$$p(w_3 | w_1, w_2) = \frac{\text{count}(w_1, w_2, w_3) + \alpha}{\sum_w \text{count}(w_1, w_2, w)}, \quad (4.1)$$

which is a straightforward calculation for measuring the probabilities of $p(w_3 | w_1, w_2)$. We can see α as a smoothing factor. It is applied because when we can't find a term (w_1, w_2, w_3) in the dictionary, this equation will give a 0 result. An add-one smoothing uses 1 as an α value.

4.1.3. Learning in SMT

The power of phrase-based SMT lies in a good phrase translation table. A phrase table is constructed first by creating a word alignment between each sentence pair of the parallel corpus. Next, we extract phrase pairs that are consistent with this word alignment.

The translation model's learning method can be seen in the Figure 13. From the word alignment matrix (Figure 13) we can see that the phrase *assumes that* is aligned with *geht davon aus, dass*, which also serves as the actual translation between them. It can be seen that target phrases for translation could be shorter or longer than the source phrase. In the Figure 13, short phrases occur more frequently, and are more likely to be applied to translate the unseen sentences. Longer phrases help us to translate a larger chunk of text, and capture more local context.

Put it together in the equation, we can see the translation task as:

$$\hat{T} = \operatorname{argmax}_T P(S | T)P(T), \quad (4.2)$$

$$\hat{T} = \operatorname{argmax}_T P(T | S). \quad (4.3)$$

	michael	geht	davon	aus	,	dass	er	im	haus	bleibt
michael	█									
assumes		█	█	█	█	█				
that		█	█	█	█	█				
he							█			
will										█
stay										█
in								█		
the								█		
house									█	

Figure 13.: Word Alignment Matrix.

Where $P(S | T)$ is a translation model, and $P(T)$ is a language model. In order to obtain the desired translation \hat{T} , probability of output target T given the input source S should be maximized.

4.2. Related Works in SMT

Efforts in the SMT as an approach for response generation have been introduced by [36]. Other works on the topic of SMTs are mainly focused on the Question-Answering task [72, 73]. However, a QA task data structure is different than a chat-oriented dialog system task. Specifically, in that one question already has a specific answer. Chat-based dialog systems present much more of a challenge, because in a real conversation there can be more than one appropriate response given one single query. In this way, creating a word alignment matrix is challenging due to many-to-many mapping of query-response pairs in the conversation

database. Another element in query-response related task utilizes SMT as a query expansion module for question-answering retrieval tasks [74].

Here we used SMT because its earlier efforts at response generation tasks had been promising. Differing from the previous existing works, in this case the team adapted the SMT to learn from a filtered movie script conversation database. We conducted a contrastive experiment with the SMT response generation and EBDM response retrieval approach in the data-driven chat-oriented dialog system scheme. By exploiting the benefit of both approaches, we propose a simple but effective way to perform combination approaches between SMT response generation and EBDM response retrieval for a chat-based dialog system.

4.3. Response Generation with SMT

With this approach, the dialog-pair data is treated as a parallel corpus for training an SMT system. Given the trained SMT system, the user dialog is treated as an input and “translated” into the system response. The system response is chosen to be system output T of maximal probability given the user input S

$$\hat{T} = \underset{T}{\operatorname{argmax}} P(T | S). \quad (4.4)$$

4.4. EBDM and SMT Hybrid Approach

While one can say that the performance of SMT response generation is on par with standard response retrieval, upon closely inspecting each approach in this experiment, it was found that SMT response generation is more robust in the OOE case. Harnessing this advantage, we propose a hybrid approach to dialog management by combining these two core systems.

During the experiments, we observed that the EBDM approach is relatively robust in a case when an exact match is found in the conversation database. However, if the exact match does not exist (OOE case), the system performance will be down. During this case, the response generation approach in SMT performs better. Observing this phenomenon, we came up with the idea of combining

both EBDM and SMT, in the hope that SMT would be able to cover up the EBDM weakness and build a robust chat-oriented dialog system.

The system works like a switch. In a normal chat conversation, the system will find a response, just like with standard response retrieval. Through the sentence similarity equation we calculated an estimate of the confidence score from the query match in the database. Later, if the system finds a low confidence score on the fly, it will fall back to SMT response generation. This technique was found to be effective, especially when a user was provided with a natural response instead of a canned error response.

4.5. Experimental Set-up

We performed the example-based TF-IDF based cosine similarity retrieval using the Apache Lucene* tool. For the SMT approach, Moses† was used to build the translation model and perform translation for the dialog system. Here, four-gram language models built with the Kneser-Ney smoothing and the lexicalized distortion model were used.

4.6. Evaluation of SMT Approach

In this section the experimental evaluation results are presented, along with discussion about the team's approaches. We conducted two types of evaluations: objective evaluations that were performed automatically, and subjective evaluations that were accomplished by gathering opinions from human users.

4.6.1. Objective Evaluation

In objective evaluation syntactic-semantic sentence similarity and TF-IDF cosine similarity was used. In syntactic-semantic similarity the system performance was measured through its syntactic and semantic features. TF-IDF cosine similarity employs TF-IDF vector. We measured syntactic and statistical cohesion of the

*<http://lucene.apache.org/>

†<http://statmt.org/moses/>

system response. Both metrics measured similarity between system response and response reference.

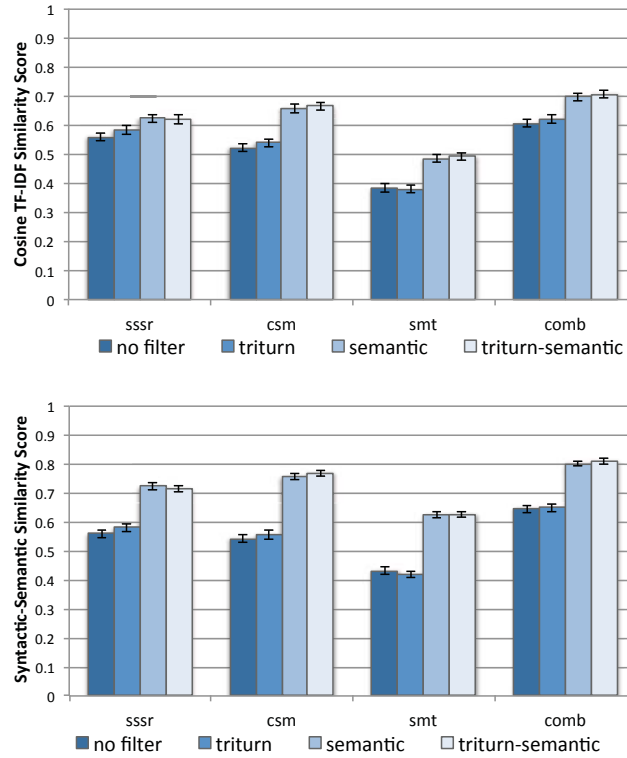


Figure 14.: Objective evaluation results on the movie data by various data-driven approaches over the cosine TF-IDF similarity (top) and syntactic-semantic similarity (bottom) metric.

Objective evaluation presented in Figures 14 and 15 is performed using TF-IDF based cosine similarity and syntactic-semantic based similarity (with calculated probability $p\text{-value} < 0.05$). The results reveal that, within EBDM approach, TF-IDF based cosine similarity retrieval (denoted as CSM) gives a better evaluation score than syntactic-semantic similarity retrieval (denoted as SSSR). This CSM approach exceeds the SSSR approach because it utilizes cosine similarity over the TF-IDF vector, compared with the SSSR approach that computes cosine similarity over the POS tag vector. Furthermore, the tri-turn and semantic similarity

filtering methods manage to increase the response score.

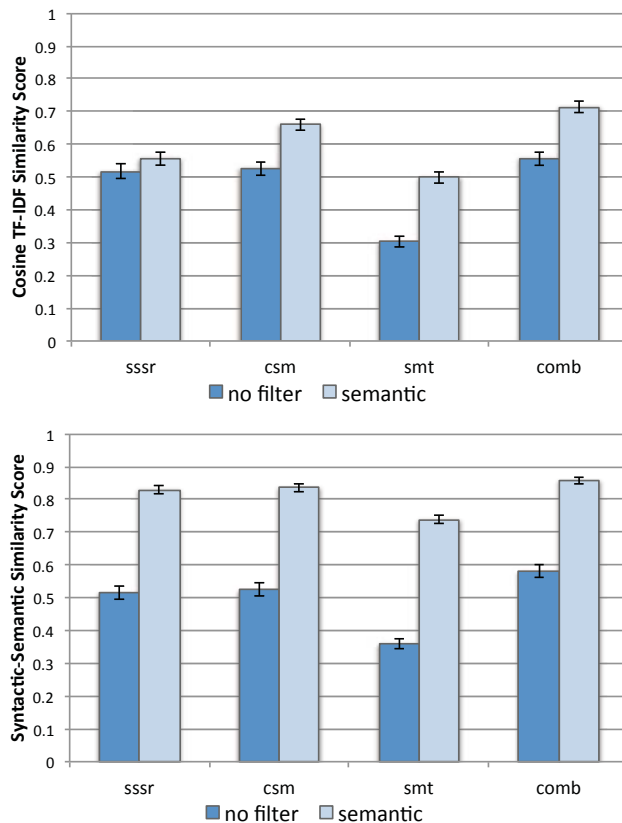


Figure 15.: Objective evaluation results on the Twitter data by various data-driven approaches over the cosine TF-IDF similarity (top) and syntactic-semantic similarity (bottom) metric.

Comparing the best EBDM approach CSM against the SMT approach SMT, CSM always give a better performance than SMT. Analyzing the data in more detail, it was found that CSM is better at handling when dialog close to Q_{test} exist in Q_{train} , while SMT can provide a better output when there is no dialog in Q_{train} similar with Q_{test} .

Combining both approaches (denoted as COMB) the system uses EBDM if the similarity between user input and dialog examples exceeds a given threshold and responding with an SMT output could overcome the shortcomings of each

approach. The objective evaluation results on system combination given various thresholds are presented in Figure 16. The axis denotes the various thresholds we experimented with. For example, threshold 0.4 means that when the CSM can not find results with a retrieval score greater than 0.4, the system will fall back and give output from the SMT approach instead. Our experiment revealed that the combined system is the best. The optimum score shown here is achieved by 0.4 and 0.6 for movie and Twitter data respectively.

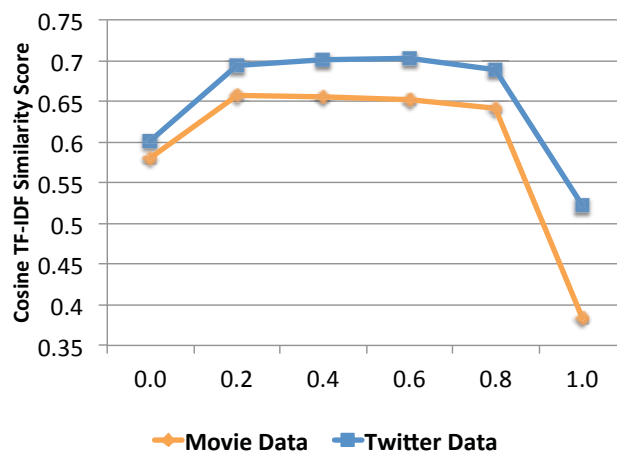


Figure 16.: Objective evaluation results of the combined system given various thresholds (axis).

A cross-domain evaluation between movie and Twitter data is also performed. In this experiment we use the COMB retrieval approach to retrieve responses within movie and Twitter filtered data. Both the Twitter and movie test data is tested in the movie, Twitter, and combined movie and Twitter database. The results of the cross-domain evaluation can be seen in the Figure 17.

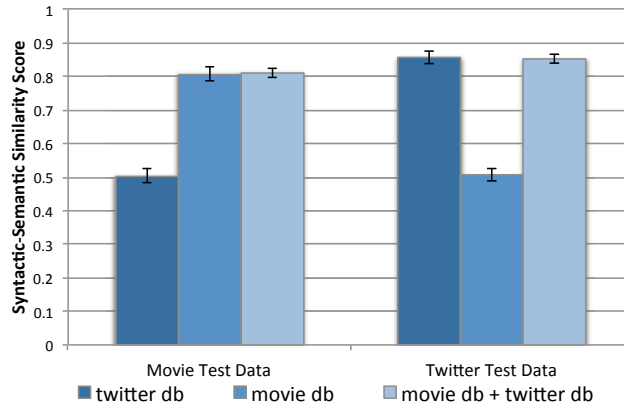


Figure 17.: Combined retrieval approach on cross-domain using syntactic-semantic similarity as an evaluation metric.

4.6.2. Subjective Evaluation

In the subjective evaluation, 40 human annotators were asked to give a naturalness score between 1-5 of the system output, with higher scores indicating that the system was producing natural and relevant system responses. Each person assessed 140 randomly selected query-response pairs that were evenly distributed over all systems. The result of this evaluation is shown in Figure 18. We also prepared a dummy system as a baseline that outputs a response by simply repeating the user input, i.e. user-input: "How are you?", then the system's output is also: "How are you?". For greeting conversations, this simple approach may work. However, for the other cases, the system may result in a completely incomprehensible response.

Along with objective evaluation, the results show that the CSM approach significantly outperforms the SMT approach. This may indicate that while the SMT responses consist of several matching phrases with the reference, they have not yet reached the naturalness of real human responses. For instance, for a query input "I'll call you back.", the SMT system will responded "I call me back.". Because this sentence is incomprehensible, many people will prefer the dummy system response "I'll call you back." instead of the SMT response. This factor seems to effect the

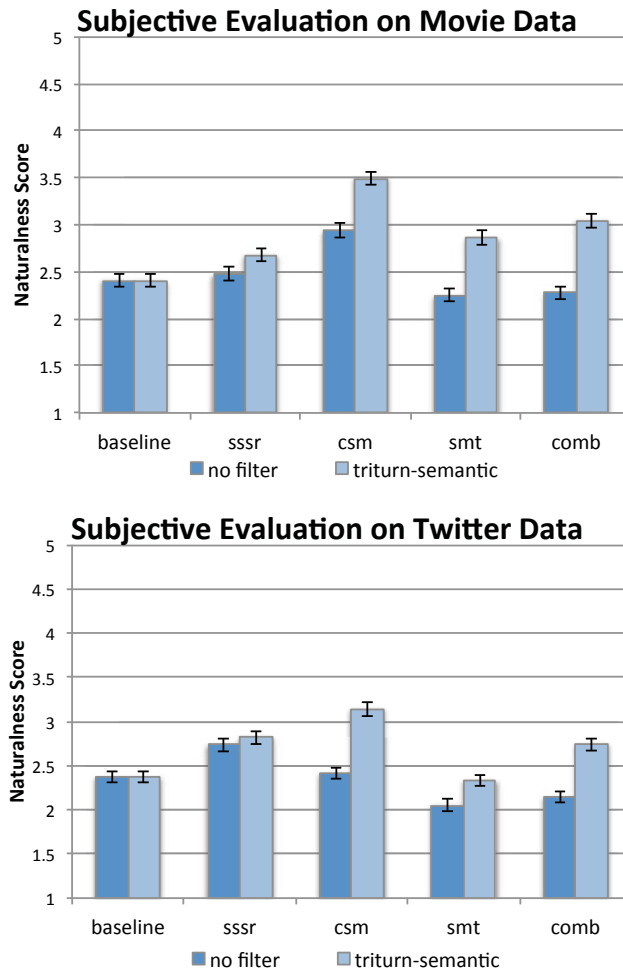


Figure 18.: Subjective evaluation result on the movie and Twitter data by various data-driven approaches.

system combination as well, where it reduced the score slightly compared to the CSM approach. Furthermore, the results of the subjective evaluation also demonstrate slightly higher scores on filtered data. This shows that the tri-turn and semantic similarity filtering methods can manage to increase the naturalness of the response.

4.6.3. Discussion

In the term of computational complexity, our CSM and SSSR approaches in EBDM have a complexity of $O(N)$, where N is the amount of traversed data in the conversation database. On the other hand, the SMT approach doesn't have to traverse the conversation database when giving a response, thus giving $O(1)$ in terms of computational complexity. This happens because the SMT approach generates the response sentence directly from the learned "translation" model.

Though CSM is inferior to SMT in terms of computational complexity, overall the CSM approach performs well, especially when it is able to find a similar sentence to the user query in the conversation database. The inverse of this case is an out of example (OOE) problem. This issue occurs when there are no sentences similar to the user query in the conversation database.

The OOE problem is serious because in a statistical chat-oriented dialog system task, the system is expected to answer various kinds of user utterances or inputs. In the task-oriented dialog system, this problem might not be as significant since conversation is usually limited to certain vocabulary and topics. In order to solve OOE, canned responses or response templates are not really good solutions as they can provide unnatural conversation responses that might interfere with the user's experience. Another way to approach OOE is to expand the conversation database to cover up a vast amount of templates and conversations, however this is not a permanent solution. Knowing that language is always developing and expanding, we can argue that there will always be conversations that such a system cannot cover.

By considering the data closely, we found that the COMB approach actually performs better for the automatic evaluation, but it doesn't perform as well in the human evaluation. This is because the SMT manages to pick an appropriate response word, but fails to create a comprehensible sentence with it. Therefore, users mostly prefer the CSM response to the SMT response, and our efforts at combining both responses (COMB approach) resulted in a decreased subjective evaluation score. This limitation made our team reconsider its approach in combining both CSM and SMT.

In the next dialog system design, we mainly focused on addressing the OOE problem. After finding the CSM approach better than the SSSR retrieval approach,

we abandoned the SSSR and adopted the CSM as our baseline EBDM approach. We moved on to the generation and neural networks strategy for a chat-oriented dialog system. The development in the neural network word representation is really promising in capturing a language phenomenon, and allows us to perform a soft matching of similar words. Furthermore, employing a language generation technique is a reasonable strategy when exact matches are not available in the conversation database.

5. Deep Neural Network for Chat-Oriented Dialog Management

5.1. Technology of Deep Neural Networks

This section will explain about the machine learning technology that is neural networks. Provided will be an introduction about neural networks and how they work, perceptrons, (which are the smallest unit in artificial neural networks,) neural network word representations, and recurrent and recursive neural networks that were used in this research.

5.1.1. Basic Artificial Neural Network

More research is still needed to explain how the brain trains itself to process information. However, some theories have arisen regarding this topic. Inside the human brain, a typical neuron collects signals from others through dendrites, which is a brain road structure that connects other neural cells to the cell body [75]. Next, neurons will forward this signal through an axon, a long, thin strand which splits into thousands of branches. At the end of each branch, there is a structure named a synapse, which converts the axon activity into electrical effects that inhibit or excite other neurons. Learning in brains occurs when synapses change effectiveness; this way the influence of a neuron to others also changes [75].

Artificial neural networks are an information processing paradigm inspired by the brain's nervous system. Applying this brain novel structure to the information processing system is the key element of the paradigm. Artificial neurons work

together as a large number of highly interconnected processing elements, such that a network is able to solve a specific problem.

Just like people, this artificial neural network learns by example. Usually artificial neural networks have a certain configuration and a specific purpose, such as data classification, pattern recognition, or pattern generation. Learning in the human brain involves adjustments to the synaptic connections that exist between the neurons. This is applied to the artificial neural networks as well. Through neural network learning algorithms such as back propagation, we adjust each neuron to fits the data and make the network function as a certain application.

Perceptrons

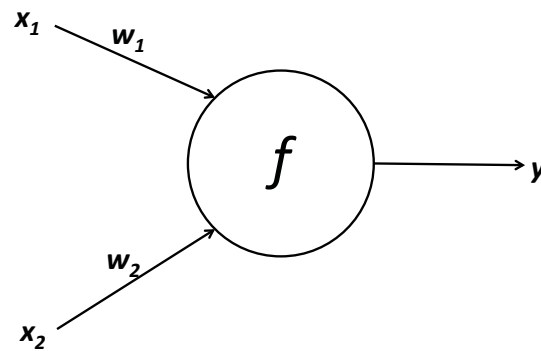


Figure 19.: Neural network perceptron.

A perceptron is a smallest learning unit in the artificial neural network. This perceptron can be viewed as a single neuron. A perceptron takes several binary inputs x_1, x_2, \dots and produces a single output y (see figure 19). Besides the inputs, perceptrons also have weights [76]. These weights w_1, w_2, \dots are real numbers representing the importance of each input. To obtain the output y , we first calculate the sum of the perceptron's input weight t :

$$t = \sum_i w_i x_i. \quad (5.1)$$

To determine the output value in the end of the perceptron, we utilize activation function f , that employs the nonlinear or linear function. To simplify this equa-

tion, the activation function is taking the sum of perceptron's input weight t as an input

$$f(t) = f\left(\sum_i w_i x_i\right). \quad (5.2)$$

The combination between many perceptrons will form a neural network (multi layer peceptron). In practice, sigmoid function is widely used to determine the output value in the end of the perceptron.

$$f(t) = \sigma(t) = \frac{1}{1 + e^{-t}}. \quad (5.3)$$

Neural Network Learning

Neural network learning is the process of adjusting the weights of each neuron in the neural network. The aim of this process is to minimize the error between the desired output and the actual output. This enables the neural network to perform specific tasks. The most widely used algorithm for neural network learning is back propagation. For this algorithm, the neural network needs to compute the error derivatives of the weights and update it. In essence it calculates the margin of errors as each weight is increased or decreased slightly.

In practice, when adjusting the weights' value, we usually introduce a variable to control how fast the learning process occurs. This value is called learning rate α , and the value is usually a real number between 0 and 1. During the neural network training, this learning rate should be adjusted. If a small value is assigned, it will take a long time for the networks to learn. On the other hand if we put a large value, the optimal networks might be hard to achieve [77]. Combining everything into the equation, we generally can update the weight of a single perceptron with

$$w_i' = w_i + \alpha \delta. \quad (5.4)$$

Where w_i' and w_i is the new weight and initial weight respectively, α is the learning rate, and δ is the error rate.

With the back propagation algorithm, we can obtain the error rate by calculating the difference between the target output (y') and network output (y) such as $\delta = y' - y$. In the neural network (which has a multiple layers) calculating

the error rate δ_j for hidden layer j might be tricky. Here we need to calculate a derivative of the layer output and propagate the error from the forward layer δ_k , thus this algorithm is called back propagation [78]. Putting it together we could calculate the error as

$$\delta_j = f'(x_j) \sum_k w_{kj} \delta_k \quad (5.5)$$

where $f'(x_j)$ is the derivative of activation function f with respect to its input x_j .

5.1.2. Recursive Neural Network

A recursive neural network (RNN) is a hierarchical network in which the same set of weights is recursively applied within a structural setting. In many cases, this hierarchical architecture is processed in a tree fashion. Given the tree structure, each node will be visited in the topological order, and will recursively apply transformations to generate further representation.

In this work, we limit our attention to RNNs over binary trees, as presented in [79]. In the binary tree structure the leaves have the initial representations, and the recursive neural networks compute the representations at each internal node η as follows:

$$x_\eta = f(w_l x_{l(\eta)} + w_r x_{r(\eta)}). \quad (5.6)$$

Where w_l and w_r are the weight parameters that connect left and right leaves to the parents, $l(\eta)$ and $r(\eta)$ are the left and right leaves.

Given this structure, an interesting interpretation can be observed. Here, the initial representation at the leaves and intermediate representation at the non-terminals lie in the same space. In the parse tree example [79], a recursive neural network combines the representations of two sub-phrases to generate a representation for the longer phrase, in the same meaning space.

We implemented a recursive neural network architecture that trains by encoding and decoding the source input. This architecture is a so-called recursive auto-encoder. More of this will be explained in the section 5.4.

5.1.3. Recurrent Neural Network

Recurrent neural network is a simple recursive neural network with a particular structure, it unfolds over time and is used for sequential inputs. This architecture is implemented when the time factor is the main differentiating factor between the elements of the sequence. In this experiment, a Long Short Term Memory (LSTM) Neural Network is employed. LSTM is one of the recurrent neural networks approaches that is widely used for natural language processing tasks.

LSTM Gate

LSTM is a particular type of recurrent neural network. The key feature of LSTM is its powerful capability to update equations in each layer structure. These equations allow LSTM to make decisions about what to store, and when to read, write, and make erasures to the memory cell. Each LSTM layer is composed of input gates i , forget gates f , output gates o , and memory cells c . Mathematically this can be viewed as:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i), \quad (5.7)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f), \quad (5.8)$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c), \quad (5.9)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o), \quad (5.10)$$

$$h_t = o_t \tanh(c_t), \quad (5.11)$$

where x_t is an input to the LSTM, in our case a single distributed word vector L of word w , σ is a logistic sigmoid function, and h is a hidden vector. The weight W and b subscript respectively represent the edge connection matrix and bias vector. For example W_{xc} indicates the input-cell (xc) weight matrix. To calculate input gates i we apply the logistic sigmoid over the sum of dot products of (1) input weight matrix W_{xi} and word input x_t , (2) hidden-input weight matrix W_{hi} and the previous hidden vector h_{t-1} , (3) cell-input weight matrix W_{ci} and previous memory cell c_{t-1} , and (4) input bias vector b_i .

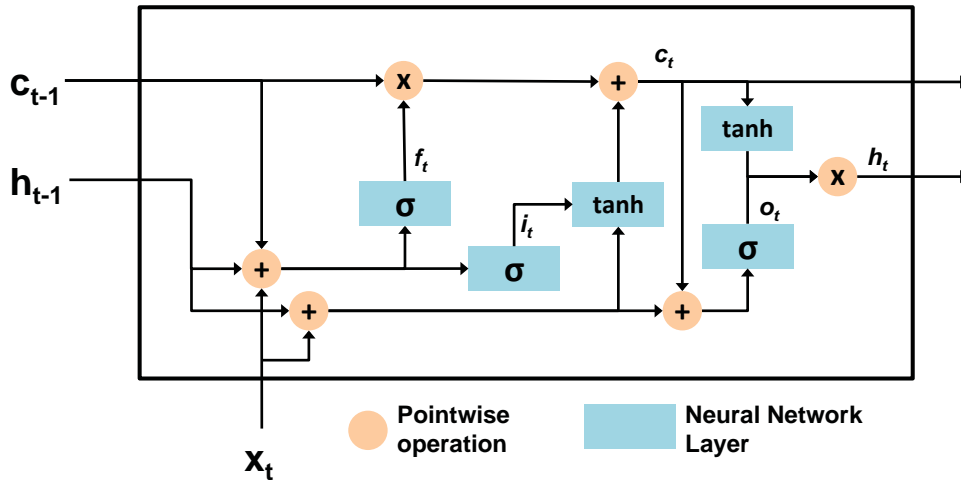


Figure 20.: Long Short Term Memory Neural Network Cell.

Back Propagation Through Time

Recurrent neural networks are often introduced as deep structure neural networks. A deep structure means that the networks have many layers that are packed together in the neural networks structure. Introduced in the early 1990s, the recurrent neural networks approach did not instantly gain popularity [77]. This happened due to the gradient vanishing problem which emerged as a major obstacle to recurrent neural network performance [77].

During the learning stage, a normal neural network will update its parameters through multiplication with the derivative product (gradient). In the mathematics principal, any quantity multiplied by less then one value (< 1) multiple times, will resulting in the vanishing value. Thus in the normal back propagation calculation, the resulting gradient value will become very small and won't add any value at all to the weight update.

LSTM is one solution to avoid the gradient vanishing problem, it is because the LSTM architecture allows limitation of the writing process by means of a gate. Learning in LSTM is done through the back propagation through time (BPTT) algorithm. BPTT works as a normal back propagation, however instead of calculating a complete back propagation through all the known layers, BPTT only performs the back propagation steps until a certain n -depth. This way, one

can maintain the gradient value and still update the weight of the networks to fit their task.

5.2. Related Works in DNN

Earlier efforts to incorporate a data-driven approach into dialog systems have relied on two main approaches. *Response retrieval* is an approach that searches for the most appropriate response in a conversation database [1, 39–42, 80, 81]. However, in the case that no response in the database could adequately respond to a given utterance, this approach will fail. *Response generation* [28, 29, 82] which has the ability to generate a new responses, is arguably more robust in handling user input compared to the other approach, however it sometimes generates unnatural responses that are incomprehensible to the user [36].

There have been a number of works on response generation for data-driven dialog systems. These works utilize a statistical machine translation system to learning the patterns between queries and responses in question-answer data [72, 73] or social media conversation data [36]. Many recent works focus on models based on recurrent neural network language models (RNNLM) [83]. Sordoni et al. [29] employs a RNN architecture to generate responses from a social media corpus, and Vinyals et al. [28] presents a long short-term memory (LSTM) neural network encoder-decoders to generate dialogue responses using movie subtitles or IT support line chats. LSTM was used because its structure is able to read the input sequence one at a time, obtain fixed-sized hidden vector representation, and utilize another LSTM to extract output sequence from that vector. The second reason is LSTM’s ability to learn data with long range temporal dependencies [83], which makes LSTM preferable for the NLP tasks, especially when it comes to learning sequences. More recently Wen et al. [82] demonstrated a more advanced LSTM that is able to control a response semantically by considering dialogue act features in the application of a goal-oriented dialogue system.

On the other hand, it was also noted that compositional distributional representations using neural networks [84] have the potential to capture a large number of linguistic phenomena, such as paraphrases. We learn these representations and use them to retrieve an appropriate response given a user utterance based on the

paraphrase detection model of Socher et al. [85].

We propose a new EBDM that retrieves dialogue responses from the database by utilizing a recursive auto-encoder paraphrase-matching algorithm. Furthermore, LSTM was employed for generation and retrieval tasks. The LSTM response generation approach, (which was developed alongside other efforts) is similar, but we were the first to train LSTM and conduct a contrastive experiment comparing the new approaches to baseline methods in a statistical, goal-oriented dialog system. Different from the previously existing works, we proposed a method to perform a response retrieval with LSTM models. The hope is that this will reduce the chance of grammatical errors occurring when generating a dialogue response.

5.3. Neural Network Word Representation

A distributed word representation is an n -dimensional vector of continuous values used to represent a word i in the vocabulary D ($i \in D$). They are often obtained by joint learning of neural network language models and distributed representation for words [86]. The reason why word representations are useful is that they allow for soft matching of similar words when exact matches are not available. This is useful especially when we are dealing with the large vocabulary. Without soft matching, the response generator has a tendency to fail and respond to the user input with another uncorrelated response based on superficial overlap of the words that do happen to have an exact match.

5.4. Recursive Auto Encoder Response Retrieval

Simple methods such as cosine similarity have problems with robustness [81]. Thus we need a more sophisticated approach to retrieve a response from the example database. In this section, we describe our proposed method to use neural network-based retrieval to retrieve more appropriate responses from the example databases. In the RAE-based retrieval (RAE) approach, given the user input the

system will find a paraphrased input sentence in the dialog pair example database. Next, it will output the corresponding response from the matched dialog pair.

In this method, a proper system response is retrieved by modeling the example database using neural word representations, and passing it to the softmax classifier that calculates a probability that the user's input Q and the query in the example database Q' are paraphrases. Thus, we can view the scoring function $S(Q, Q')$ as being the paraphrase matching probability.

Adopting the work of [85], we utilized recursive autoencoders (RAE), dynamic pooling, and a softmax classifier to decide whether the sentence was paraphrased or not. In the following sections we describe: (1) word representations, the input to the RAE, (2) recursive autoencoders, and (3) dynamic pooling and paraphrase classification. An overview of the neural-network-based retrieval method is depicted in Figure 21.

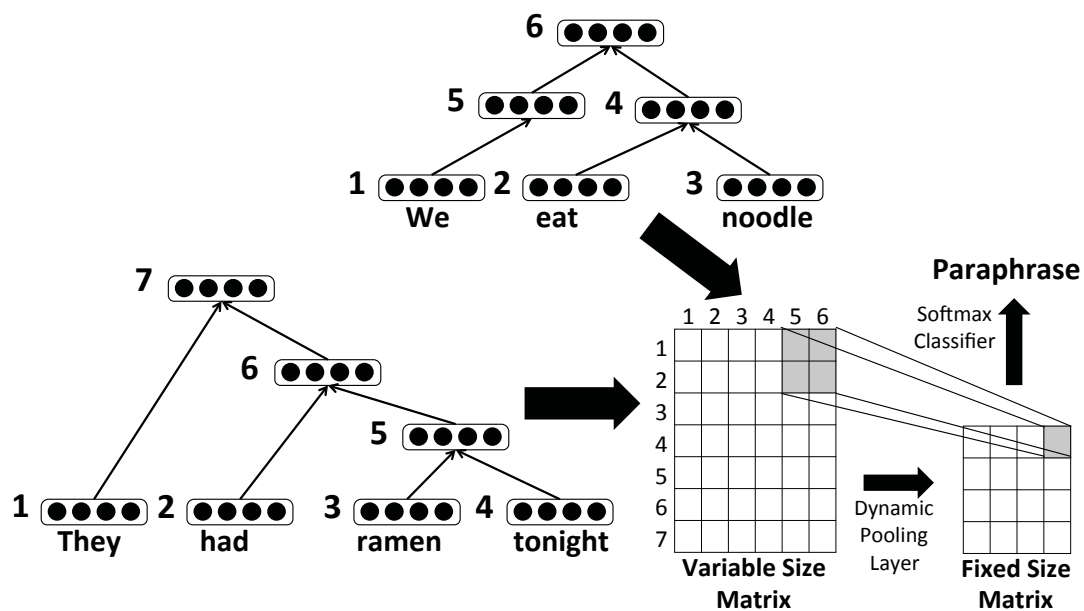


Figure 21.: Overview of neural-network-based retrieval.

5.4.1. Recursive Auto Encoder

The RAE algorithm is used to combine word representations into vector representations of longer phrases in a syntactic parse tree. The aim of using syntactic parse trees is to capture the meaning that is structurally represented by the tree. In order to construct the vector representation, this algorithm requires word representations and a binary syntactic tree as input.

When calculating the recursive autoencoders, every child and non-terminal node in the binary tree is collected as a feature representation of a sentence. The binary tree forms the parent and children triplets ($p \rightarrow c_1c_2$) where each child could be a word representation vector or a non-terminal node. A parent p is calculated through the neural network layer (Equation (5.12))

$$p = f(W_e[c_1; c_2] + b), \quad (5.12)$$

where $[c_1; c_2]$ is concatenation of the vectors of two children and f is a tanh activation function.

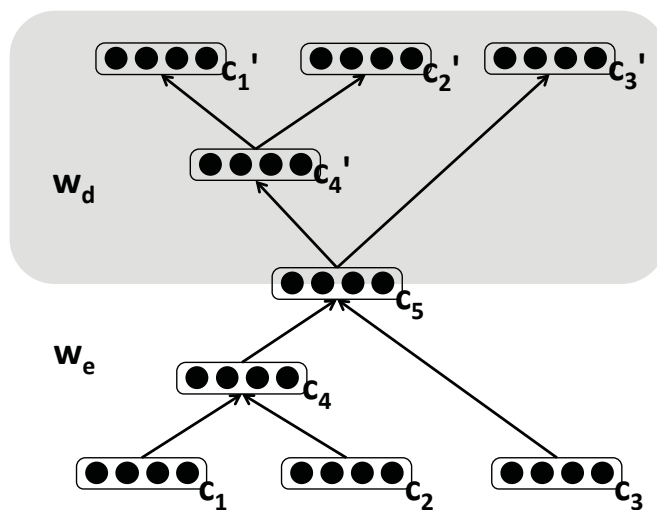


Figure 22.: Recursive autoencoder model.

The parameters W_e and b are trained using recursive autoencoders as shown in Figure 22. The RAE performance is evaluated through the Euclidean distance between original input and its estimated reconstruction node (Equation (5.13))

$$E(p) = \|[c_1; c_2] - [c'_1; c'_2]\|^2 \quad (5.13)$$

where

$$[c'_1, c'_2] = f(W_d p + b_d). \quad (5.14)$$

This process will be repeated recursively for all non-terminal nodes. In the course of RAE training, we want to minimize the total error of all inputs pairs on every non-terminal node. The total error can be determined by adding up all the calculated errors from a single parse tree T

$$E_{tree}(T) = \sum_{p \in T} E(p). \quad (5.15)$$

The benefit of recursive autoencoders is that they can capture the compositional structure of phrases, and their similarity within two given sentences. For example, the sentence “tons of stuff to throw away” and “a lot of junk to dispose of” there are relationship between words and phrases such as “tons of stuff” with “a lot” and “throw away” with “dispose of”. Using the recursive autoencoder, we can not only capture the word paraphrase similarity, but also the phrase similarity.

5.4.2. Dynamic Pooling and Softmax Layer

Given the RAE-derived representation of the sentence, the similarity of two sentences will be calculated. To deal with the arbitrary length of the sentence, RAE word representations are normalized into a fixed length vector with an algorithm called dynamic pooling. Every sentence fed into the RAE forms a binary tree representation. Given this, we can define a matrix M , where the rows and columns in the matrix represent two sentences with the different lengths i and j . Because the matrix includes all of the non-terminal nodes and leaves in the binary tree, the matrix M 's size is $2i - 1 \times 2j - 1$.

The dynamic pooling algorithm takes a matrix M as an input and turns it into matrix M' with the fixed size $n \times n$. This algorithm will divide the matrix M

into n roughly equal parts. Every minimal value in the rectangular window is selected to form a $n \times n$ grid.

Given this $n \times n$ grid, each utterance is classified as similar or not similar, using a softmax classifier layer. The softmax classifier takes the matrix M' as an input, and outputs a confidence score that decides whether a user input and dialog database is a paraphrase. This confidence score was used as the retrieval score when performing the RNN retrieval in this study.

5.5. LSTM Response Retrieval and Generation

In this section, we describe the LSTM network that we used in this study. The LSTM was used in two different ways, LSTM response generation (LSTM-GEN) and the novel proposed approach LSTM response retrieval (LSTM-RET). Both of these approaches are discussed in this section.

5.5.1. Long Short Term Memory Neural Network

We can view each query and response dialog pair as a set of a words (q_1, \dots, q_I) and $(r'_1, \dots, r'_{I'})$. By doing so, we can formulate a conditional probability of the response given the query as $P(R'|Q) = P(r'_1, \dots, r'_{I'} | q_1, \dots, q_I)$.

Before the LSTM takes a word from the input sentence, each word is transformed into a distributed word representation. A distributed word representation is an n -dimensional vector of continuous values used to represent a word in the vocabulary [86, 87]. Each word in the dictionary ($w \in W$) is embedded into n -dimensional space $L \in \mathbb{R}^{n \times |W|}$. From this representation, a word vector can be seen as a single vector in the column of matrix L .

At the end of each LSTM, we calculate the output probability by performing an affine transform on the LSTM output h_t , and calculating the probability with the softmax function:

$$P(r'_{t+1}|h) = \text{softmax}(W_{hy}h_t + b), \quad (5.16)$$

where W_{hy} is a hidden-output weight matrix, and b is a bias.

Our model consists of an LSTM encoder-decoder with two LSTMs, one for the query sequence and another for the response sequence. The details of how the

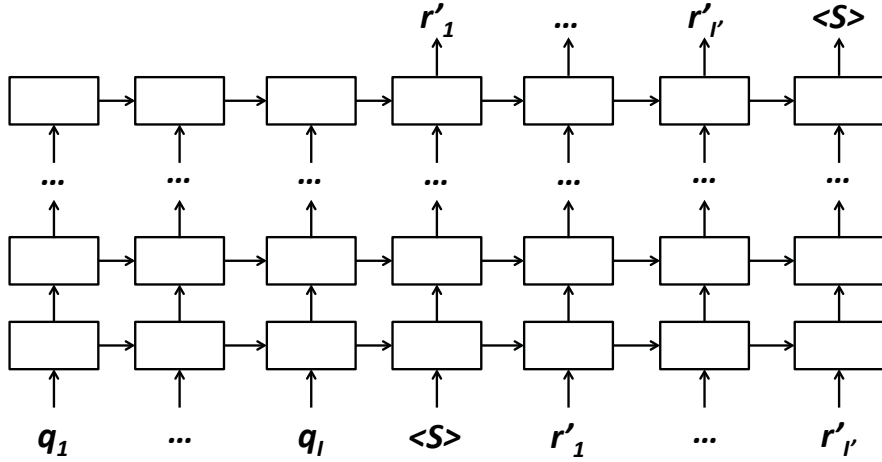


Figure 23.: LSTM neural model over time.

LSTM works can be seen in the Fig. 23. Given the query (q_1, \dots, q_I) the network will predict a response $(r'_1, \dots, r'_{I'})$ as the output. Note that the system starts to decode the output after reading the input and receiving the end of sentence symbol " $\langle S \rangle$ ".

The conditional probability of the next word in the response sentence is calculated conditioned on a hidden representation h and memory cell that encodes the input query q_1, \dots, q_I , and the previously generated words of the response sequence

$$P(r'_1, \dots, r'_{I'} | q_1, \dots, q_I) = \prod_{i=1}^{I'} P(r'_i | h). \quad (5.17)$$

We also experimented with deep LSTMs that stack memory cells one on top of another. During training we utilized back propagation through time [88] to calculate the gradient over the full sequence, minimizing the negative log likelihood $\mathcal{L}(x) = -\sum_{t=1}^T \log P(x_{t+1} | y_t)$ using stochastic gradient descent. Using the development data, we calculated the LSTM loss function, and decreased the network learning rate by half when there was no improvement over time. The learning was terminated when the learning rate was lower than a threshold.

5.5.2. LSTM Response Generation

As explained above, to encode the input sentence, we fed it word by word (q_1, \dots, q_T) to the LSTM model. By following the word probability $P(r'|h)$ from equation (5.17), we calculated the word with the highest probability as follows:

$$\hat{R} = \sum_{r' \in R} \operatorname{argmax} P(r'|h). \quad (5.18)$$

After encoding the target sentence, the LSTM response generation (LSTM-GEN) decoder is used to generate output word by word. We searched for the most likely response by using a left-to-right beam search decoder which maintains a small number h of partial hypotheses at each time step, and discards the rest [89]. The search ends when it reaches a symbol “<S>” and appends it to the highest-scoring hypothesis.

5.5.3. LSTM Response Retrieval

Differently from the LSTM response generation, LSTM response retrieval (LSTM-RET) calculates $P(R'|Q)$ for every response candidate R' in the dialog database based on its conditional probability $\log P(R'|Q)$ divided by the number of words $|R'|$.

$$\hat{R} = \operatorname{argmax}_{R' \in D} \frac{\log P(R'|Q)}{|R'|}. \quad (5.19)$$

This score shows how likely a response candidate is to be an output response, given the user utterance sentence and the LSTM model. This score is used to retrieve the highest scoring response from the dialogue database.

5.6. Experimental Set-up

Initially our experiment employed the 100-dimensional word representations computed and provided by Turian et al. [90]. Thus, for the next experiment, we chose dialogue pairs that could be transformed into a vector of word representations. When observing the conversation data, we discovered that some of the movie

corpora contains slang words. As the target aim is to provide a good, natural response to the user, we decided to filter out some tri-turns manually by removing tri-turns that contain slang words, parsing errors, and long utterances (paragraphs that consist more than three sentences.) In the end, 10,033 dialogue pairs were used as the training and test data. During the experiment, we separated the dialog pair data into 1,000 dialog pairs for tests and 9,033 dialog pairs for training randomly.

As mentioned previously, the effectiveness of example-based dialogue largely depends upon on whether a close example exists in the database. To examine how well each method works when a close example exists or doesn't exist, we further divided the test dialogue pair data into two cases [81]:

- 1 Close example found (CEF) - (587 examples): A given user query is available or there exists a close example in the dialog database. This happens when baseline CSM retrieval score is more than a threshold 0.7 [81].
- 2 Out of example (OOE) - (413 examples): The rest of the queries under the threshold.

5.6.1. Paraphrase-based Retrieval Setup

In order to train the softmax classifier, we needed to provide a good paraphrase corpora that consists of a balanced amount of sentence pairs, some paraphrase and some not. However, collecting the paraphrase data is not a trivial task and can be extremely time consuming. Here, we used an automatic approach by defining a paraphrase as a pair of sentences that have a strong syntactic-semantic relation.

To provide a balanced amount of similar and dissimilar queries during training, we cross produced all training dialog (9,033 pairs) with each other and calculated the syntactic-semantic similarity [1] $sim(S_1, S_2) = \alpha[sem_{sim}(S_1, S_2)] + (1 - \alpha)[cos_{sim}(S_1, S_2)]$. We assume that a similar query is obtained when the syntactic-semantic score is exclusively between 0.7 and 0.9, and a non-similar query is obtained when the syntactic-semantic score is exclusively between 0.2

and 0.4*. In the end, 1,421,338 pairs of training data were obtained, with a ratio of 50:50 between similar and non-similar sentences.

Regarding the recursive auto encoder (RAE) neural network, the pre-trained RAE was used, with 150,000 sentences from NYT and AP section of the Gigaword corpus provided by Socher et al. [85]. All the parse trees for the RAE algorithm were generated with the Stanford parser [91]. Lastly, the RAE and LSTM neural network in this experiment was implemented with Matlab[†] and LAMTRAM Toolkit[‡] respectively.

5.6.2. LSTM Network Setup

In order to train the LSTM network, we separated the dialog pair training data into 7,227 and 1,806 examples[§] for training and development sets. During training, we used the training set to learn the parameters, and we used the development set as a criterion to evaluate the network performance and decide whether to continue training or not.

Before evaluating the LSTM-based methods on actual dialogue performance, we first evaluated the perplexity of the model on the development set for various numbers of nodes in the hidden layers (100, 200, 300), and various numbers of hidden layers (1-7).

The perplexity results of the network training can be seen in Fig. 24. The best performance of the various settings is achieved by the 1 layer LSTM with 300 nodes in the hidden layer, with a perplexity score of 38.73. We used this network in our dialog-based evaluation in the following sections.

*Note that it would be better to manually create a corpus of similar and non-similar utterances, but this is extremely time consuming and so the more light-weight automatic approach has been taken

[†]<http://mathworks.com/products/matlab/>

[‡]<https://github.com/neubig/lamtram>

[§]This data is relatively small, but the limit of what we could collect after semantic similarity filtering. A test with a larger amount of data is reserved for future work.

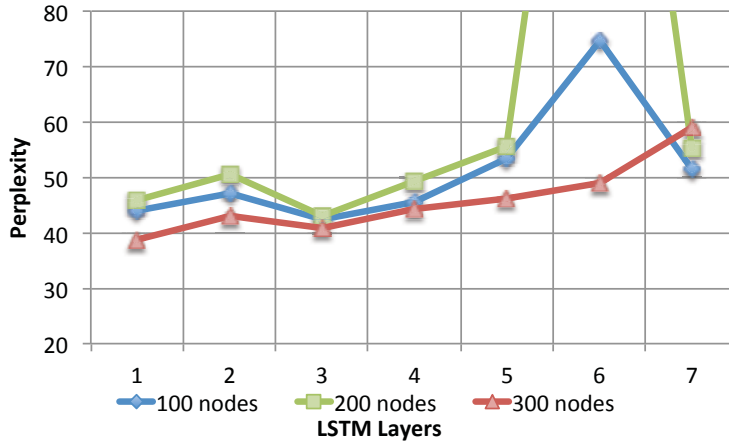


Figure 24.: LSTM model perplexity.

5.7. Evaluation of RAE and LSTM

5.7.1. Objective Evaluation

In this section we will discuss the automatic evaluation results for both RAE and LSTM systems. The LSTM models are divided into two parts: LSTM-GEN (section 5.5.2) and LSTM-RET (section 5.5.3). Both techniques utilize LSTM neural networks to learn the query-response patterns over the conversation database.

In the objective evaluation, we calculated the similarity between the system response \hat{R} and the expected output R with (1) TF-IDF cosine similarity, which focuses on content word similarity, and (2) BLEU-4, which focuses on fluency and local word order [51]. We compared the baseline retrieval systems (CSM) with the proposed paraphrase-based response retrieval (RAE), LSTM response retrieval (LSTM-RET), and baseline response generation system (SMT) with LSTM response generation approaches (LSTM-GEN).

The result of the objective evaluation over the cosine TF-IDF similarity metrics can be seen in the top section of Figure 25 (with calculated probability p -value < 0.05). Where OOE is out of example that is the case when user query is not found in database. CEF is a close example found during an inverse case of OOE.

This objective evaluation shows that both LSTM-RET and LSTM-GEN approaches

significantly outperform the baselines not only in the OOE, but also in the CEF case. In this figure, we can also see that SMT approach can pick a good selection of words when generating a response in the OOE case, however in most of the cases we observed that these responses are incomprehensible [1]. For example SMT may generate a responses like “I do exam take it” or “put you out there on a tray” which is hard to understand and grammatically wrong. This behavior resulted in the lower performance of subjective evaluation, as seen in the following section.

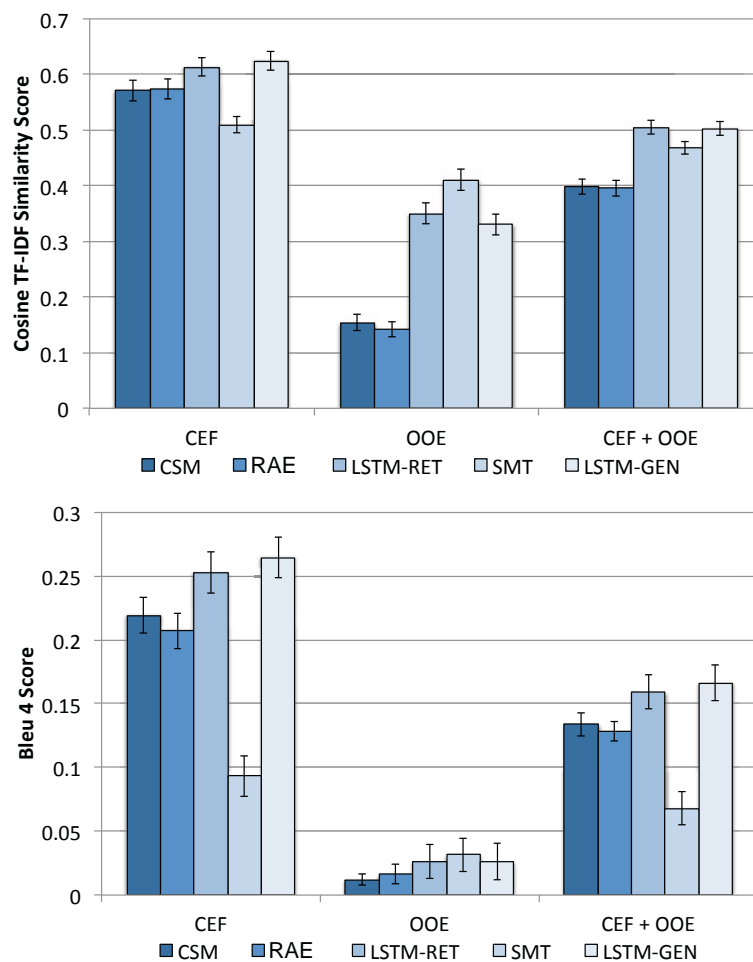


Figure 25.: Objective evaluation results over the cosine TF-IDF similarity (top) and BLEU-4 (bottom) metric.

Next, we evaluated the system’s performance with the BLEU-4 metric to calculate how well the response can capture response fluency. The results of the BLEU-4 evaluation can be seen in the bottom section of Figure 25. From this evaluation, we can see that again both LSTM approaches perform better, especially in the CEF case.

Though the RAE performance is on par with the baseline overall, it is slightly better than the baseline in the OOE case. This is because the RAE approach performs retrieval capturing paraphrase features, which is a slightly sophisticated retrieval approach compared to CSM.

One difference between the cosine TF-IDF retrieval approach (CSM) and response generation with LSTM (LSTM-GEN) is that LSTM-GEN employs distributed word embeddings while CSM employs discrete representations for words. To examine the effect of discrete vs. distributed representations, we performed a follow-up study on retrieving responses with the cosine similarity over a vector of word embeddings (CSM-EMBD). The result of our experiment can be seen in the table 2. While both the performance of LSTM-GEN and LSTM-RET is on par with the CSM-EMBD, by employing the distributed word embedding (CSM-EMBD, LSTM-RET, LSTM-GEN) these approaches could surpass the CSM approach that does not use distributed word embeddings. Furthermore, we can also see both LSTM-GEN and LSTM-RET are slightly better compared to the CSM-EMBD in the OOE case.

	BLEU-4 Score		Cos TF-IDF Score	
	CEF	OOE	CEF	OOE
CSM	0.2191	0.0121	0.5706	0.1541
CSM-EMBD	0.2625	0.0251	0.6548	0.3035
LSTM-RET	0.2526	0.0261	0.6130	0.3498
LSTM-GEN	0.2646	0.0261	0.6240	0.3302

Table 2.: Comparison with retrieval based distributed representations.

In addition, the LSTM-GEN approach has an advantage in that it is more efficient in terms of the computational complexity of creating a response. In normal response retrieval (CSM-EMBD) we need to traverse all dialog data in the database, and thus complexity is $O(n)$ where n is the amount of data in the dialog database. On the other hand, LSTM-GEN has a complexity of $O(1)$ in the

data size because we don't have to traverse all the data. Knowing this is useful, especially when we want to deliver this dialogue framework to the end user in real time.

5.7.2. Subjective Evaluation

Next, we report two varieties of subjective evaluation: naturalness and relevance. A response is categorized as natural if the sentence is comprehensible and likely

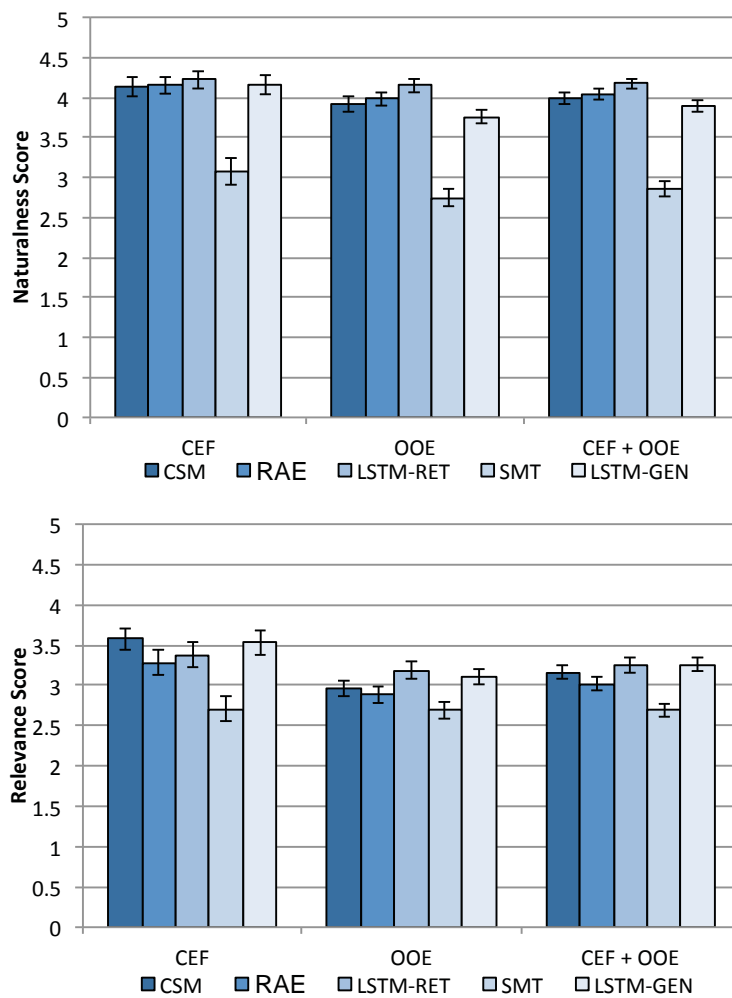


Figure 26.: Naturalness (top) and Relevance (bottom) for each system.

to be generated by a human. On the other hand, we define a relevant response as a response that is appropriately related to the user query sentence. We asked five human annotators to give a score between 1-5 to the system response. Each person was asked to annotate 255 randomly selected query-response pairs that were evenly distributed over all the systems.

The subjective evaluation on naturalness shows that our proposed methods RAE, LSTM-RET, and LSTM-GEN give responses that are on par with the baseline approaches (see Figure 26 at the top section), with LSTM-RET performing slightly better compared to the baseline approaches in both the CEF and OOE cases. By observing the LSTM-GEN generated responses sentence by sentence, we found that most of the responses are short, compared to the retrieved responses, which sometimes generate long responses. The short responses are easy to comprehend and reduce the chance of grammatical mistakes.

By looking at the subjective relevance evaluation in the bottom section of Figure 26, we can observe that the RAE relevance score is slightly under the baseline. This tells us that although the RAE approach manages to capture the paraphrase features, it still has difficulty giving a relevant answer to the user query. On the other hand, the LSTM-RET retrieval and LSTM-GEN generation approaches, compared to the other approaches, can perform significantly better in the OOE case. The performance of the LSTM-RET retrieval and LSTM-GEN generation are almost the same for the CEF case. This indicates that the LSTM response generation is relatively robust, even in cases where a close match does not exist in the database.

Finally, we display some results from each of the systems in Table 3. All of the algorithms perform relatively well in the CEF case, where we can find a good example that matches the user utterance in the dialog database. In the OOE case, both CSM and RAE may give an uncorrelated response, as they might not find a good response in the dialogue database. Both LSTM-RET and LSTM-GEN are more likely to give a short, meaningful, and correlated response, which corroborates the results of [28]. In many cases, we found that LSTM-GEN and LSTM-RET give a similar response, which happens because the same LSTM neural network model is used for both systems.

	CEF	OOE
User Utterance	What do you mean by that?	Do you know why I'm here?
CSM	I mean you're different.	I know what you mean.
RAE	I do different things on different days.	If you know what I am, you know very well I can wipe you from existence.
LSTM-GEN	I mean, what do you mean?	I know you are.
SMT	What do I mean by that.	I know what I am.
LSTM-RET	I want to talk about me.	I know. I was here.

Table 3.: Various responses for each dialogue system.

5.7.3. Discussion

First we will discuss the RAE performance. Table 4 shows the relation between user input and examples in the database. We calculate syntactic-semantic score *sim* for each utterance pair ($S1$ and $S2$). We observed that when a utterance pair has a high similarity score (a similar pair), it will generate a clear diagonal structure of dark line in the matrix representation. This matrix shows the paraphrase relations between two utterances. A clear diagonal structure of dark line in the matrix was a result from the Euclidean distance computation. During this case, the RAE-based retrieval managed to find sentences that are close/paraphrased to the input query.

In the automatic evaluation, we can see that the RAE performance is on par with the CSM baseline. This behavior is also shown in the naturalness evaluation score, where RAE performs as well as the CSM approach. However, in the term of relevance we can see that the RAE does not perform well. This shows us that the paraphrase retrieval simply doesn't perform well enough to retrieve a relevant response and it might lead to choosing irrelevant responses.

Through our experiments we found out that both the LSTM approach in response generation LSTM-GEN and retrieval LSTM-RET perform well and are preferred by users. Looking at the responses obtained by these techniques, we determined that in most cases LSTM-GEN tends to give a short answer to the users.

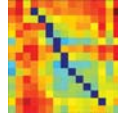
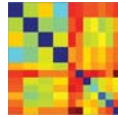

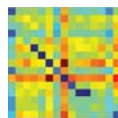
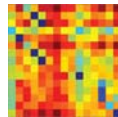
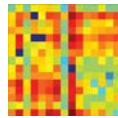
<i>sim</i>	Sentences	Matrix
0.94	<p>S_1) Captain, we can not keep going fast on these icy roads.</p> <p>S_2) We can not keep going fast on these icy roads!</p>	
0.93	<p>S_1) I'll see you there.</p> <p>S_2) I'll see you. for say to myself?</p>	
0.65	<p>S_1) So what do you have to say for yourself?</p> <p>S_2) Why should I have anything for say to myself?</p>	
0.60	<p>S_1) Hold your fire! He's got a girl.</p> <p>S_2) Looks like he's got a hostage.</p>	
0.50	<p>S_1) I've been careful, I've been waiting my chance.</p> <p>S_2) Oh, you've been under a lot of stress.</p>	
0.38	<p>S_1) Yes, I can see that too and I don't think it's so terrible.</p> <p>S_2) That's why I do all the thinking.</p>	

Table 4.: This table shows a correlation between two sentences, user input and example database. We calculated syntactic-semantic score *sim* [1] for each utterance pair (S_1 and S_2).

For example, some response that we obtained from LSTM-GEN are: “Well what do you see?”, “Don’t worry about it.”, and “I’m going to do it.”. These short responses reduce the chance of grammatical mistakes and are actually preferred

by the users since it is easy to understand and related to the user queries. On the other hand, the answers selected from conversation database in LSTM-RET are mostly similar, and only some are slightly longer compared to the LSTM-GEN results. These responses are taken from the conversation database which have a natural feel and correct grammar. This response is preferred by the user because it is grammatically correct and relevant to the user’s query.

Since the responses obtained from LSTM-GEN and LSTM-RET are similar in most of the cases, we can say that our LSTM model is fitting the data well. This may be a sign that the LSTM is overfitting the training data. Choosing between LSTM-GEN and LSTM-RET, we prefer the LSTM-GEN approach better. This is because the LSTM-GEN approach has an advantage, in that it is more efficient in terms of the computational complexity of creating a response.

In the end, we present a subjective response comparison between LSTM-GEN and CSM baseline in terms of naturalness and relevance. This percentage shows the amount of users that prefer the one response over the other. In the term of naturalness, 36.84% of subjective evaluation results prefer CSM responses over the LSTM-GEN responses. Only 27.63% of subjective evaluation results prefer LSTM-GEN responses over the CSM. In terms of relevance, 35.96% of the subjective evaluation results prefer LSTM-GEN over the CSM baseline. This shows that LSTM-GEN approach in 35.96% of the cases manage to give a more relevant response to the user query.

	LSTM-GEN	CSM
naturalness	27.63%	36.84%
relevance	35.96%	30.26%

Table 5.: Response preference percentage between LSTM-GEN and CSM baseline.

Looking closely at the data, we observe that users prefer the CSM in terms of naturalness, because sometimes the LSTM-GEN gives a short template-like question as a response. For example, when we query the system with the sentence “Oh, nothing happened, ya know, but it was great.”, LSTM-GEN response with “What was it?”, and CSM response with “Yeah, I know”. However, in terms of response relevance, users prefer LSTM-GEN because it manages to respond with the

relevant response, rather than the CSM response that is general and not related to the user's query.

Sometimes there is a case where the response is logically incorrect. For example, given a user utterance "You wanna be friends?", the LSTM-GEN responds with "I'll be my friend". We can see that although the responses' grammar is correct, the response generated from the LSTM-GEN is not correct logically. As mentioned by Higashinaka, et al. [10], semantic errors happens when the response cannot constitute any meaning. In this response we can see that a person cannot be a friend with himself/herself.

Another issue in the chat-oriented dialogue is that the system should be able to maintain a longer conversation. The system will, preferably, be able to give an open-ended type response. However, this ability doesn't manifest clearly in our system yet. We can see that the response given by our system is relevant, but is boring and does not encourage the continuation of the conversation. This is not a good response for the chat-oriented dialogue system. For example, given a user utterance "How do you know?", the system responds with "I just know". This response ends the conversation and does not encourage the user to chat more with the system.

6. Conclusion and Future Direction

6.1. Conclusion

In this thesis, we investigated several approaches to creating a robust chat-oriented dialogue system. Conventional chat-oriented dialog systems require well hand-crafted rules, which necessitate a lot of human work, especially when the dialogue tries to accommodate a variety of topics. Moreover, relying on the unfiltered conversation databases also results in unnatural conversation. We also deal with the OOE problem that occurs in the chat-oriented dialog system.

Data availability is the first problem we worked on regarding establishing chat-oriented dialogue systems. Dealing with unnatural responses, we utilized real human-to-human conversation examples from movie scripts and Twitter conversations. We proposed that tri-turn extraction and semantic similarity filtering are able to extract dialog-pair examples from multi-speaker dialogue of raw movie scripts and Twitter data. Experimental results also reveal that that tri-turn and semantic filtering improve the objective evaluation metrics score (TF-IDF based cosine similarity and syntactic-semantic similarity evaluation metrics). This approach also helps to reduce the retrieval response time by reducing dialogue examples in the training set. It relies on no explicit domain knowledge, and should therefore be applicable to other dialog applications with little-to-no modification. However, our collected corpus is not enough, and was only used in a small study. Further work in collecting more comprehensive conversation dialogue should be done.

The next problem is to retrieve a good response candidate from the conversation database. Here, we performed a contrastive experiment with various data-driven

approach in EBDM. We found that the EBDM approach is very good at handling the queries which are similar to the examples in the database, but it demonstrates poor performance in handling the queries which are different from existing examples. SMT response generation based systems had an opposite tendency. We introduced a system that combines example-based and SMT-based approaches to take advantage of the characteristics of both approaches. The drawback from this approach is that the SMT response generation system is not able to generate a good and comprehensible response.

Another problem that we focus on is the OOE problem. This occurs when the system handles an example that is not available in the conversation database. We propose a new statistical model for building robust dialogue systems using neural networks to either retrieve or generate dialogue responses based on existing data sources. Our experimental evaluation shows that these neural network retrieval and generation approaches were effective and can generate a response on par with the baseline system or even better. By focusing on addressing the case where a similar example does not exist in the training data (the OOE case), we found that our proposed approach can perform well, improving the robustness over the baseline approaches. Though promising, our approach is still far from the goal in creating a good and robust chat-oriented dialogue system. Some problems still exist, such as illogical responses, short and uninteresting responses, and close-ended responses.

In summary, there are advantages and disadvantages for every system that we proposed and/or implemented in this thesis. We put all of the dialog system approach summary data into table 6.

Approach	Abbr.	Properties	Pros/Cons
Syntactic Similarity Retrieval	SSR	POS-Tag, WordNet Synset	<ul style="list-style-type: none"> (-) O(n) Complexity, needs to traverse database (-) Does not perform well compared to CSM
Cosine Similarity TF-IDF Retrieval	CSM	TF-IDF Weighting, Word Vector	<ul style="list-style-type: none"> (+) Performs well when examples found in database (-) O(n) Complexity, needs to traverse database (-) Performs poorly in OOE case
Recursive Auto Encoder Paraphrase Based Retrieval	RAE	Word Ordering	<ul style="list-style-type: none"> (+) Phrase based matching (+) Performs slightly better in OOE case compare to CSM (-) O(n) Complexity, needs to traverse database
Long Short Term Memory Neural Network Response Retrieval	LSTM-RET	Word Ordering, Word Embedding	<ul style="list-style-type: none"> (+) Soft matching (+) Performs better in OOE case (-) O(n) Complexity, need to traverse database (-) Neural network train, took a lot of time
Statistical Machine Translation Response Generation	SMT	Word Ordering	<ul style="list-style-type: none"> (+) Able to pick up appropriate words (+) O(1) Complexity, doesn't need to traverse database (-) Sentence response is often incomprehensible
Long Short Term Memory Neural Network Response Generation	LSTM-GEN	Word Ordering, Word Embedding	<ul style="list-style-type: none"> (+) Soft matching (+) Performs better in OOE case (+) O(1) Complexity, doesn't need to traverse database (-) Neural network training takes a lot of time

Table 6.: Summary of various approaches to dialog response creation.

6.2. Future Direction

Improving the current approaches might be necessary in the future. It would also be interesting to solve the remaining problems in the chat-oriented dialogue system such as establishing a system that is able to comprehend the user’s intentions and emotions, and is able to maintain a long and interesting conversation.

Our proposed system is still not able to perform a conversation that is exactly relevant to the user utterance. Relevant, meaning that the response should be logical given the conversation context. For example, when the user gives a sentence that contains the word “orange” the system should be able to distinguish between orange the fruit and orange the color. Incorporating the system with the conversation context might be necessary.

Sometimes the system provides a response that is illogical due to a semantic error. In the next iteration, more research to handle this problem might be necessary as well. One idea could be to make a sentence classifier that is able to recognize the semantic error in the sentence and avoid this kind of response.

Another future possibility is to enable the chat-oriented dialogue system to maintain a long and interesting conversation with the user. There are many possibilities for this, starting by working on controlling the system response so that it understands the user’s intention and emotional state. A similar work to control the response generation output produced by the LSTM is done by [82]. In this work, a modified LSTM is used to control the generated response in the goal-oriented dialogue task. This modified LSTM is a so-called semantically controlled (SC) LSTM. This SC-LSTM introduces an additional cell gate to the conventional LSTM. This additional cell, controlled by the domain dialogue act, is able to manipulate the response generation results. However, performing response controlling on domain-slot in goal-oriented dialogue tasks can not be easily applied in the chat-oriented dialogue task. This is solely because the chat-oriented dialogue systems do not have a specific domain-slot. Therefore, further research is needed to investigate these topics.

In summary, we combined the existing problems with our approaches in chat-oriented dialogue system into a comprehensive matrix (see figure 27). The horizontal axis describes the existing problems in the chat-oriented dialog system. The vertical axis describes the approaches and solutions, and the blue region

represent our works in this thesis.

Approaches and Solutions	Dialog retrieval with neural-network based paraphrase-matching algorithm				
	Response generation and retrieval with long short term neural (LSTM) neural-network				
	Establish a hybrid approach between EBDM response retrieval with cosine TF-IDF vector and statistical machine translation (SMT) approach				
	Construct a conversation database based on movie script and Twitter conversation				
	Data availability for natural conversation	Selecting a good response candidate	Out of example (OOE) error	Comprehend the user intentions and emotions	Maintain a long and interesting conversation
Problems in Chat-Oriented Dialog System					

Figure 27.: Future work matrix.

References

- [1] L. Nio, S. Sakti, G. Neubig, T. Toda, and S. Nakamura, “Utilizing human-to-human conversation examples for a multi domain chat-oriented dialog system,” *IEICE Transactions on Information and Systems*, June 2014.
- [2] G. S., “Understanding speech understanding: Towards a unified theory of speech perception,” *Proc. of ESCA Workshop*, Staffordshire, England, pp.1–8, 1996.
- [3] J. Holmes and W. Holmes, *Speech Synthesis and Recognition*, 2nd ed., Taylor & Francis, Inc., Bristol, PA, USA, 2002.
- [4] Y. K., W. S., L. J., and H.J. R., “Statistical dialogue management using intention dependency graph,” *Proc. of the International Joint Conference on Natural Language Processing (IJCNLP)*, oct 2013.
- [5] E. Seneff, L. Hirschman, and V. Zue, “Interactive problem solving and dialogue in the ATIS domain,” *Proc. of the Fourth DARPA Speech and Natural Language Workshop*, pp.354–359, 1991.
- [6] M. Walker, J. Aberdeen, J. Boland, E. Bratt, J. Garofolo, L. Hirschman, A. Le, S. Lee, S. Narayanan, K. Papineni, B. Pellom, J. Polifroni, A. Potamianos, P. Prabhu, A. Rudnicky, G. Sanders, S. Seneff, D. Stallard, and S. Whittaker, “DARPA communicator dialog travel planning systems: the June 2000 data collection,” *Proc. of EUROSPEECH*, pp.1371–1374, 2000.
- [7] K. Yoshino, S. Mori, and T. Kawahara, “Spoken dialogue system based on information extraction using similarity of predicate argument structures,” *Proc. of SIGDIAL*, pp.59–66, 2011.

- [8] J. Weizenbaum, “Eliza - a computer program for the study of natural language communication between man and machine,” *Commun. ACM*, vol.9, no.1, pp.36–45, 1966.
- [9] R.Wallace, *Be Your Own Botmaster*, A.L.I.C.E A.I. Foundation, 2003.
- [10] R. Higashinaka, K. Funakoshi, M. Araki, H. Tsukahara, Y. Kobayashi, and M. Mizukami, “Towards taxonomy of errors in chat-oriented dialogue systems,” *Proc. of SIGDIAL*, Prague, Czech Republic, pp.87–95, Association for Computational Linguistics, 2015.
- [11] J. Peckham, “A new generation of spoken dialogue systems: results and lessons from the sundial project.,” *Proc. of EUROSPEECH*, 1993.
- [12] L. Lamel, S. Rosset, J. Gauvain, and S. Bannacef, “The limsi arise system for train travel information,” *Proc. of ICASSP*, Phoenix, USA, 1999.
- [13] M.F. McTear, “Modelling spoken dialogues with state transition diagrams: experiences with the CSLU toolkit,” *Proc. of ICSLP*, 1998.
- [14] S. Larsson and D.R. Traum, “Information state and dialogue management in the trindi dialogue move engine toolkit.,” *Natural Language Engineering*, vol.6, pp.323–340, 2000.
- [15] D. Bohus and A.I. Rudnicky, “The ravenclaw dialog management framework: Architecture and systems,” *Comput. Speech Lang.*, vol.23, no.3, pp.332–361, jul 2009.
- [16] C. Rich and C.L. Sidner, “COLLAGEN: A collaboration manager for software interface agents,” *User Modeling and User-Adapted Interaction*, vol.8, pp.315–350, 1998.
- [17] J.D. Williams and S. Young, “Partially observable markov decision processes for spoken dialog systems,” *Comput. Speech Lang.*, vol.21, no.2, pp.393–422, apr 2007.
- [18] L. Esther and P. Roberto, “A stochastic model of computer-human interaction for learning dialogue strategies,” *Proc. of EUROSPEECH*, pp.1883–1886, 1997.

- [19] T. Paek, “Reinforcement learning for spoken dialogue systems: Comparing strengths and weaknesses for practical deployment,” tech. rep., May 2006.
- [20] J. Williams and S. Young, “Scaling up pomdps for dialog management: The "summary pomdp" method,” Proc. of ASRU, 2005.
- [21] B. Thomson, S. Keizer, F. Mairesse, J. Schatzmann, K. Yu, and S. Young, “User study of the bayesian update of dialogue state approach to dialogue management,” Proc. of Interspeech, 2008.
- [22] S. Young, M. Gašić, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and K. Yu, “The hidden information state model: A practical framework for pomdp-based spoken dialogue management,” *Comput. Speech Lang.*, vol.24, no.2, pp.150–174, apr 2010.
- [23] C. Lee, S. Jung, S. Kim, and G.G. Lee, “Example-based dialog modeling for practical multi-domain dialog system,” *Speech Commun.*, vol.51, no.5, pp.466–484, May 2009.
- [24] A. Leuski and D. Traum, “NPCEditor: A Tool for Building Question-Answering Characters,” Proc. of LREC, 2011.
- [25] H. Murao, N. Kawaguchi., S. Matsubara, Y. Yamaguchi, and Y. Inagaki, “Example-based spoken dialogue system using WOZ system log,” Proc. of SIGDIAL, Sapporo, Japan, pp.140–148, 2003.
- [26] F. Bessho, T. Harada, and Y. Kuniyoshi, “Dialog system using real-time crowdsourcing and twitter large-scale corpus,” Proc. of SIGDIAL, Seoul, South Korea, pp.227–231, 2012.
- [27] N. Chambers and J. Allen, “Stochastic language generation in a dialogue system: Toward a domain independent generator.,” Proc. of SIGDIAL, Cambridge, Massachusetts, USA, pp.9–18, 2004.
- [28] O. Vinyals and Q. Le, “A neural conversational model,” Proc. of ICML Deep Learning Workshop, Lille, France, 2015.

- [29] A. Sordoni, M. Galley, M. Auli, C. Brockett, Y. Ji, M. Mitchell, J. Nie, J. Gao, and B. Dolan, “A neural network approach to context-sensitive generation of conversational responses,” Proc. of NAACL-HLT, 2015.
- [30] L. Shang, Z. Lu, and H. Li, “Neural responding machine for short-text conversation,” Proc. of ACL, 2015.
- [31] I.V. Serban, A. Sordoni, Y. Bengio, A.C. Courville, and J. Pineau, “Building end-to-end dialogue systems using generative hierarchical neural network models,” Proc. of AAAI, pp.3776–3784, 2016.
- [32] J. Dodge, A. Gane, X. Zhang, A. Bordes, S. Chopra, A. Miller, A. Szlam, and J. Weston, “Evaluating prerequisite qualities for learning end-to-end dialog systems,” Proc. of ICLR, 2015.
- [33] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, “End-to-end memory networks,” Proc. of NIPS, pp.2440–2448, 2015.
- [34] R. Yan, Y. Song, and H. Wu, “Learning to respond with deep neural networks for retrieval-based human-computer conversation system,” Proc. of ACM SIGIR, pp.55–64, 2016.
- [35] J. Weston, S. Chopra, and A. Bordes, “Memory networks,” Proc. of ICLR, 2015.
- [36] A. Ritter, C. Cherry, and W.B. Dolan, “Data-driven response generation in social media,” Proc. of EMNLP, Edinburgh, Scotland, UK., pp.583–593, July 2011.
- [37] H. Wang, Z. Lu, H. Li, and E. Chen, “A dataset for research on short-text conversations,” Proc. of EMNLP, 2013.
- [38] R. Lowe, N. Pow, I. Serban, and J. Pineau, “The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems,” Proc. of SIGDIAL, 2015.
- [39] S. Jung, C. Lee, and G. Lee, “Dialog studio: An example based spoken dialog system development workbench,” Proc. of the Dialogs on dialog:

Multidisciplinary Evaluation of Advanced Speech-based Interactive Systems. Interspeech2006-ICSLP satellite workshop,, Pittsburgh, USA, 2006.

- [40] C. Lee, S. Lee, S. Jung, K. Kim, D. Lee, and G. Lee, “Correlation-based query relaxation for example-based dialog modeling,” Proc. of ASRU, Merano, Italy, pp.474–478, 2009.
- [41] K. Kim, C. Lee, D. Lee, J. Choi, S. Jung, and G. Lee, “Modeling confirmations for example-based dialog management,” Proc. of SLT, Berkeley, California, USA, pp.324–329, 2010.
- [42] R.E. Banchs and H. Li, “IRIS: a chat-oriented dialogue system based on the vector space model,” Proc. of ACL (System Demonstrations), pp.37–42, 2012.
- [43] D.M. Harris and S.L. Harris, Digital design and computer architecture, Morgan Kaufmann Publishers, Corp., Amsterdam, Boston, 2007.
- [44] Z. Harris, “Distributional structure,” Word, vol.10, no.23, pp.146–162, 1954.
- [45] G. Salton, A. Wong, and C.S. Yang, “A vector space model for automatic indexing,” Commun. ACM, vol.18, no.11, pp.613–620, Nov. 1975.
- [46] D. Beck, “Bayesian kernel methods for natural language processing,” Proc. of ACL (Student Research Workshop), pp.1–9, 2014.
- [47] D. Beck, T. Cohn, C. Hardmeier, and L. Specia, “Learning structural kernels for natural language processing,” Proc. of ACL, 2015.
- [48] Y.W. Chang, C.J. Hsieh, K.W. Chang, M. Ringgaard, and C.J. Lin, “Training and testing low-degree polynomial data mappings via linear svm,” The Journal of Machine Learning Research, vol.11, pp.1471–1490, Aug. 2010.
- [49] L. Gandy and K. Hammond, “Creating conversations: An automated dialog system,” 2011.
- [50] W.Y. Wang and J. Hirschberg, “Detecting levels of interest from spoken dialog with multistream prediction feedback and similarity based hierarchical fusion learning,” Proc. of SIGDIAL, pp.152–161, 2011.

- [51] K. Papineni, S. Roukos, T. Ward, and W.J. Zhu, “BLEU: A method for automatic evaluation of machine translation,” Proc. of ACL, Philadelphia, Pennsylvania, USA, pp.311–318, 2002.
- [52] R. Higashinaka, N. Kobayashi, T. Hirano, C. Miyazaki, T. Meguro, T. Makino, and Y. Matsuo, “Syntactic filtering and content-based retrieval of twitter sentences for the generation of system utterances in dialogue systems,” Proc. of IWSDS, pp.113–123, 2014.
- [53] M.A. Walker, D.J. Litman, C.A. Kamm, and A. Abella, “Paradise: A framework for evaluating spoken dialogue agents,” Proc. of ACL, pp.271–280, 1997.
- [54] J. Williams, A. Raux, D. Ramachandran, and A. Black, “The dialog state tracking challenge,” Proc. of SIGDIAL, 2013.
- [55] D. Griol, L.F. Hurtado, E. Segarra, and E. Sanchis, “A statistical approach to spoken dialog systems design and evaluation,” Speech Communication, vol.50, no.8-9, pp.666–682, aug 2008.
- [56] T. Paek, “Empirical methods for evaluating dialog systems,” Proc. of ELDS, 2001.
- [57] R. Lowe, I.V. Serban, M. Noseworthy, L. Charlin, and J. Pineau, “On the evaluation of dialogue systems with next utterance classification,” Proc. of SIGDIAL, 2016.
- [58] J. Weston, A. Bordes, S. Chopra, and T. Mikolov, “Towards ai-complete question answering: A set of prerequisite toy tasks,” Proc. of ICLR, 2016.
- [59] J. Dodge, A. Gane, X. Zhang, A. Bordes, S. Chopra, A. Miller, A. Szlam, and J. Weston, “Evaluating prerequisite qualities for learning end-to-end dialog systems,” Proc. of ICLR, 2016.
- [60] I.V. Serban, R. Lowe, L. Charlin, and J. Pineau, “A survey of available corpora for building data-driven dialogue systems,” CoRR, 2015.
- [61] C. Danescu-Niculescu-Mizil and L. Lee, “Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs,” Proc. of CMCL, 2011.

- [62] A. Roy, C. Guinaudeau, H. Bredin, and C. Barras, “Tvd: A reproducible and multiply aligned tv series dataset,” Proc. of LREC, Reykjavik, Iceland, 2014.
- [63] J. Tiedemann, “Parallel data, tools and interfaces in opus,” Prof. of LREC, Istanbul, Turkey, 2012.
- [64] D. Ameixa and L. Coheur, “From subtitles to human interactions: Introducing the subtle corpus,” Technical report, 2013.
- [65] M. Walker, G. Lin, and J. Sawyer, “An annotated corpus of film dialogue for learning and characterizing character style,” Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC’12), Istanbul, Turkey, 2012.
- [66] R. Sproat, A.W. Black, S.F. Chen, S. Kumar, M. Ostendorf, and C. Richards, “Normalization of non-standard words.,” Computer Speech and Language, vol.15, no.3, pp.287–333, 2001.
- [67] D. Liu, Z. Liu, and Q. Dong, “A dependency grammar and wordnet based sentence similarity measure,” Journal of Computational Information Systems, vol.8, no.3, pp.1027–1035, 2012.
- [68] A. Echihabi and D. Marcu, “A noisy-channel approach to question answering,” Proc. of ACL (1), pp.16–23, 2003.
- [69] Y.W. Wong and R.J. Mooney, “Learning for semantic parsing with statistical machine translation,” Proc. of NAACL-HLT, pp.439–446, 2006.
- [70] Y.W. Wong and R.J. Mooney, “Generation by inverting a semantic parser that uses statistical machine translation,” Proc. of NAACL-HLT, pp.172–179, 2007.
- [71] P. Koehn, Statistical Machine Translation, 1st ed., Cambridge University Press, New York, USA, 2010.
- [72] C. España Bonet and P.R. Comas, “Full machine translation for factoid question answering,” Proc. of ESIRMT and HyTra, EACL 2012, Stroudsburg, PA, USA, pp.20–29, Association for Computational Linguistics, 2012.

- [73] G. Zhou, F. Liu, Y. Liu, S. He, and J. Zhao, “Statistical machine translation improves question retrieval in community question answering via matrix factorization,” Proc. of ACL, Sofia, Bulgaria, pp.852–861, Association for Computational Linguistics, 2013.
- [74] S. Riezler, A. Vasserman, I. Tsochantaridis, V. Mittal, and Y. Liu, “Statistical machine translation for query expansion in answer retrieval,” Proc. of ACL, Prague, Czech Republic, pp.464–471, Association for Computational Linguistics, 2007.
- [75] W.S. McCulloch and W. Pitts, “Neurocomputing: Foundations of research,” ch. A Logical Calculus of the Ideas Immanent in Nervous Activity, pp.15–27, MIT Press, Cambridge, MA, USA, 1988.
- [76] C. Van Der Malsburg, Frank Rosenblatt: Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms, pp.245–248, Springer Berlin Heidelberg, Berlin, Heidelberg, 1986.
- [77] B. Kröse and P.v.d. Smagt, An introduction to Neural Networks, 1996.
- [78] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, Inc., New York, NY, USA, 1995.
- [79] R. Socher, C.C. Lin, A.Y. Ng, and C.D. Manning, “Parsing natural scenes and natural language with recursive neural networks,” Proc. of ICML, 2011.
- [80] L. Nio, S. Sakti, G. Neubig, T. Toda, and S. Nakamura, “Combination of example-based and SMT-based approaches in a chat-oriented dialog system,” Proc. of ICE-ID, 2013.
- [81] L. Nio, S. Sakti, G. Neubig, T. Toda, and S. Nakamura, “Improving the robustness of example-based dialog retrieval using recursive neural network paraphrase identification,” Proc. of IEEE SLT, 2014.
- [82] T.H. Wen, M. Gasic, N. Mrksic, P.H. Su, D. Vandyke, and S. Young, “Semantically conditioned LSTM-based natural language generation for spoken dialogue systems,” Proc. of EMNLP, Lisbon, Portugal, 2015.

- [83] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, “Recurrent neural network based language model,” Proc. of INTERSPEECH, Chiba, Japan, 2010.
- [84] R. Socher, B. Huval, C.D. Manning, and A.Y. Ng, “Semantic compositionality through recursive matrix-vector spaces,” Proc. of EMNLP, Jeju, South Korea, 2012.
- [85] R. Socher, E.H. Huang, J. Pennington, A.Y. Ng, and C.D. Manning, “Dynamic pooling and unfolding recursive autoencoders for paraphrase detection,” in Advances in Neural Information Processing Systems 24, Curran Associates, Inc., 2011.
- [86] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, “A neural probabilistic language model,” J. Mach. Learn. Res., vol.3, pp.1137–1155, 2003.
- [87] R. Collobert and J. Weston, “A unified architecture for natural language processing: Deep neural networks with multitask learning,” Proc. of ICML, Helsinki, Finland, pp.160–167, 2008.
- [88] P. Werbos, “Backpropagation through time: what does it do and how to do it,” Proc. of IEEE, 1990.
- [89] I. Sutskever, O. Vinyals, and Q. Le, “Sequence to sequence learning with neural networks,” Proc. of NIPS, 2014.
- [90] J. Turian, L. Ratinov, and Y. Bengio, “Word representations: A simple and general method for semi-supervised learning,” Proc. of ACL, 2010.
- [91] D. Klein and C.D. Manning, “Accurate unlexicalized parsing,” Proc. of ACL, Stroudsburg, Pennsylvania, USA, pp.423–430, 2003.

Publications

Journal Papers

1. Lasguido Nio, Sakriani Sakti, Graham Neubig, Koichiro Yoshino, and Satoshi Nakamura, "Neural Network Approaches to Dialog Response Retrieval and Generation," *IEICE Transactions on Information and Systems*, 2016.
2. Lasguido Nio, Sakriani Sakti, Graham Neubig, Tomoki Toda, and Satoshi Nakamura, "Utilizing Human-to-Human Conversation Examples for a Multi Domain Chat-oriented Dialog System ," *IEICE Transactions on Information and Systems*, June 2014.

International Conferences

1. Lasguido Nio, Sakriani Sakti, Graham Neubig, Tomoki Toda, and Satoshi Nakamura, "Recursive Neural Network Paraphrase Identification for Example-based Dialog Retrieval," *Asia Pacific Signal and Information Processing Association (APSIPA)*. Siem Reap, Cambodia. December 2014.
2. Lasguido Nio, Sakriani Sakti, Graham Neubig, Tomoki Toda, and Satoshi Nakamura, "Improving the Robustness of Example-based Dialog Retrieval using Recursive Neural Network Paraphrase Identification," *IEEE Spoken Language Technology Workshop (SLT)*. Lake Tahoe, USA. December 2014.
3. Lasguido Nio, Sakriani Sakti, Graham Neubig, Tomoki Toda, and Satoshi Nakamura, "Conversation Dialog Corpora from Drama Television and Movie Scripts," *The 17th Oriental COCODA Conference*. Phuket, Thailand. September 2014.
4. Lasguido Nio, Sakriani Sakti, Graham Neubig, Tomoki Toda, and Satoshi Nakamura, "Combination of Example-based and SMT-based Approaches in a Chat-oriented Dialog System," *International Conference on Electronics Technology and Industrial Development (ICE-ID)*, Bali, Indonesia. October 2013.
5. Lasguido Nio, Sakriani Sakti, Graham Neubig, Tomoki Toda, Mirna Adriani, and Satoshi Nakamura, "Developing Non-Goal Dialog System based on Examples of Drama Television," *The International Workshop on Spoken Dialog Systems (IWSDS)*. Paris, France. December 2012.

Domestic Conferences

1. Lasguido Nio, Sakriani Sakti, Graham Neubig, Tomoki Toda, and Satoshi Nakamura, “Robust Example-based Dialog Retrieval using Distributed Word Representations and Recursive Autoencoders,” The 72nd Japanese Society for Artificial Intelligence Speech and Language Understanding (SIG-SLUD). Kanagawa, Japan. December 2014.
2. Lasguido Nio, Sakriani Sakti, Graham Neubig, Tomoki Toda, and Satoshi Nakamura, “A Dialog System with Human-to-Human Conversation Example,” Autumn Conference on Acoustical Society of Japan (ASJ). Sapporo, Japan. September 2014.

Related Publications

Journal Paper

1. Masahiro Mizukami, Lasguido Nio, Hideaki Kizuki, Toshio Nomura, Graham Neubig, Koichiro Yoshino, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura, “Example Based Dialogue System Based on Satisfaction Prediction,” Japanese Society for Artificial Intelligence Vol. 31(2016) No. 1. January 2016.

Appendices

A. Conversation Database Structure in JSON Format

Below is the our conversation database structure in JSON format. We prefer JSON to XML since it is more lightweight*. In general there are two class objects: Triturn Class and Turn Class. Triturn class is responsible to store a sequence of 3 consecutive turn conversation. On the other hand, Turn Class is responsible to store a detail information of the text data. The JSON structure of these class is depicted in the figure 28. The details information of these class can be seen in the table 7 and 8.

Turn Class	
attribute	description
actor	actor name who perform the dialog turn
sentence	contains the actual tokenized sentences
actual_sentence	contains the actual sentence after the filtering process (not tokenized)
postag	postag information based on the sentence
ner	NER information based on the sentence
dependency_grammar	contain normalized dependency grammar from the sentence
semantic_set	contain the semantic information from the sentence
sentence_type	the type of sentence
additional_info	contain the additional non dialog information
original_sentence	the original sentence from the script
turn_in_file	the sentence turn in the file
script_filename	the script filename

Table 7.: Turn class structure.

*<http://www.json.org/>

TriTurn Class	
attribute	description
turn_1	(Turn) first Turn TriTurn
turn_2	(Turn) second Turn in the TriTurn
turn_3	(Turn) third Turn in the TriTurn
syntax_distance_1	(double) represents the syntax distance between turn_1 and turn_2
semantic_distance_1	(double) represents the semantic distance between turn_1 and turn_2
syntax_distance_2	(double) represents the syntax distance between turn_2 and turn_3
semantic_distance_2	(double) represents the semantic distance between turn_2 and turn_3

Table 8.: Tri-turn class structure.

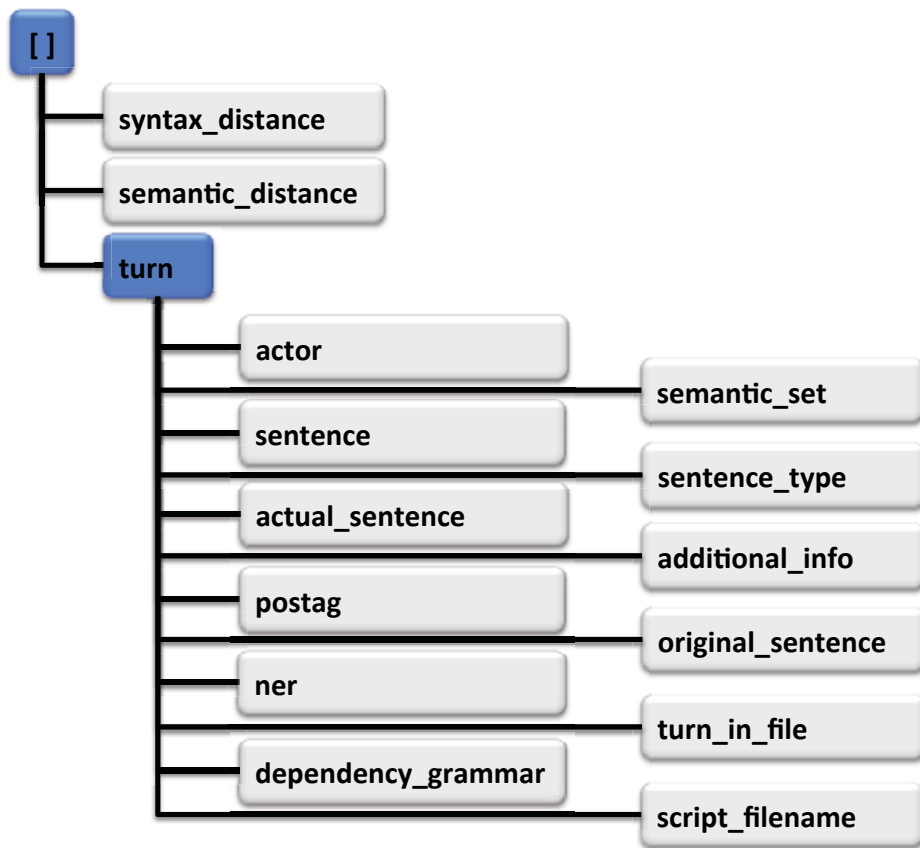


Figure 28.: Conversation database structure in JSON format.

B. Dialog System Evaluator Desktop Application

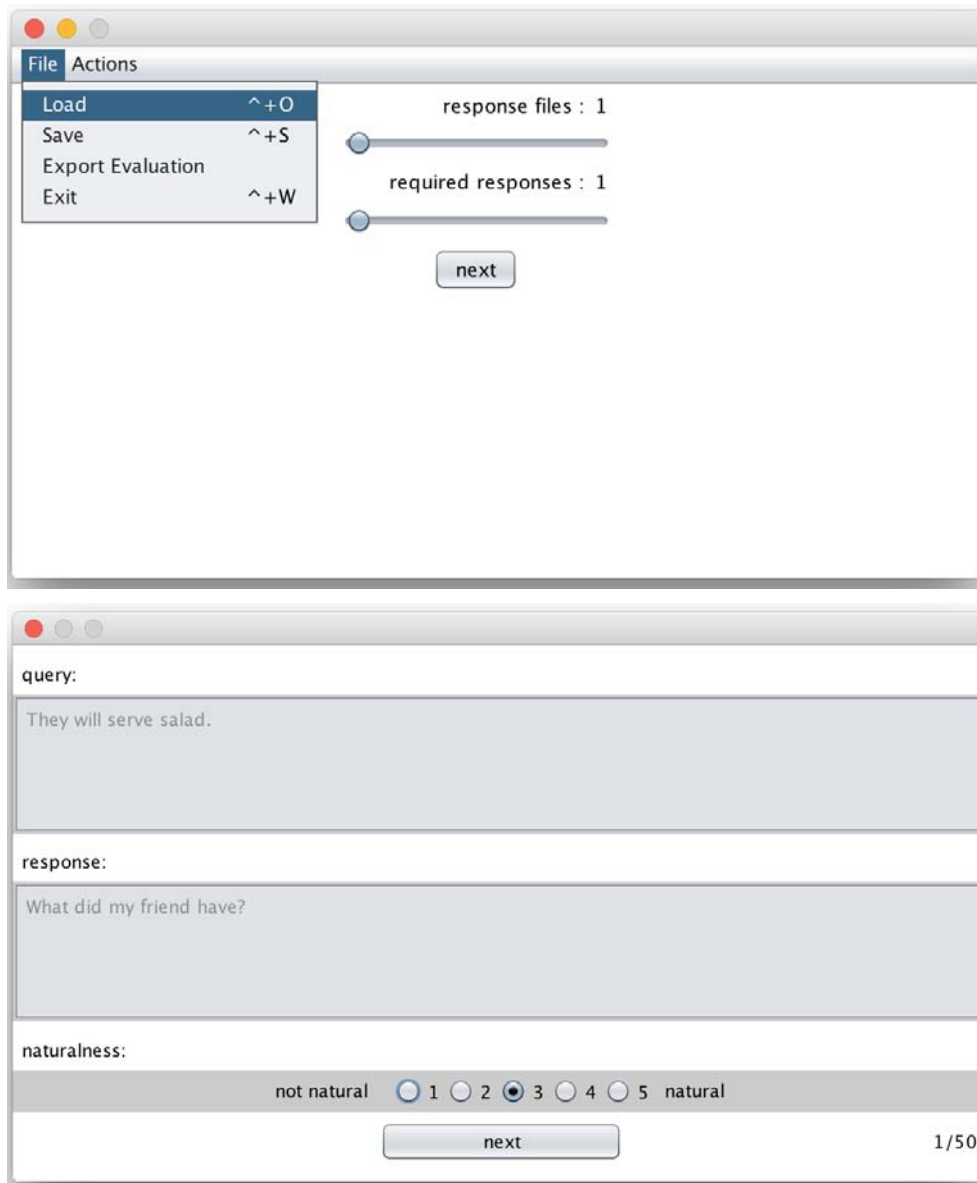


Figure 29.: Screen shot of the dialog system evaluator desktop application.

As for the subjective evaluation, we ask human and user who have a better understanding in the English language. Since we often perform this evaluation during our research, to avoid redundant and repetitive engineering steps, we build a Java[†] desktop application to help our evaluation process.

This application works as follows, first the dialog researcher decide how many response (dialog system) that he/she want to evaluate. Next input the required response number. This is depicted in the upper section of figure 29. For example, 3 response and 25 required response means that the user will need to assess query-response pair from 3 different system, and from each system user need to evaluate 25 query-response pairs. Thus make the user need to evaluate (3×25) 75 query-response pairs in overall.

When the evaluation start, user will be given pair query and response. Then user need to give an opinion score about the query-response pair, this score is ranging from 1 to 5, where 1 is not natural and 5 is natural. This evaluation page is shown in the lower section of figure 29. Since there is no clear explanation about what the “naturalness” metric is, the dialog researcher should explain it to the user manually about this metric.

[†]<https://java.com/en/>

C. Dialog System Web Demo Application

Besides the desktop application, we also build a web interface for the demo and evaluation purposes. Different from the desktop application, here we allow user to interact with the system whenever and as long as they want to. Users input their query in the bottom left part of the page, then will obtain the response from the various type of dialog systems. Later, the user could share their opinion about which response is the best, in a *drag and drop* style, in the bottom right part of the page.

There are 5 kind of dialog system (bot) presented in this evaluation, and this bot is randomly assign each time the page is loaded. In the figure 30, the 5 dialog systems are ELIZA, EBDM, NGRAM, SMT, and WIKI. ELIZA [8] is a psychotherapy dialog system adapted as a comparison to the other system. EBDM is example-based dialog system that employs TF-IDF vector. NGRAM is example-based dialog system with the N-Gram word matching. SMT is statistical dialog system, as proposed in the chapter 4. WIKI is a Wikipedia[‡] information retrieval dialog system, this system search for name entity in the dialog query from the Wikipedia using a Google search API[§].

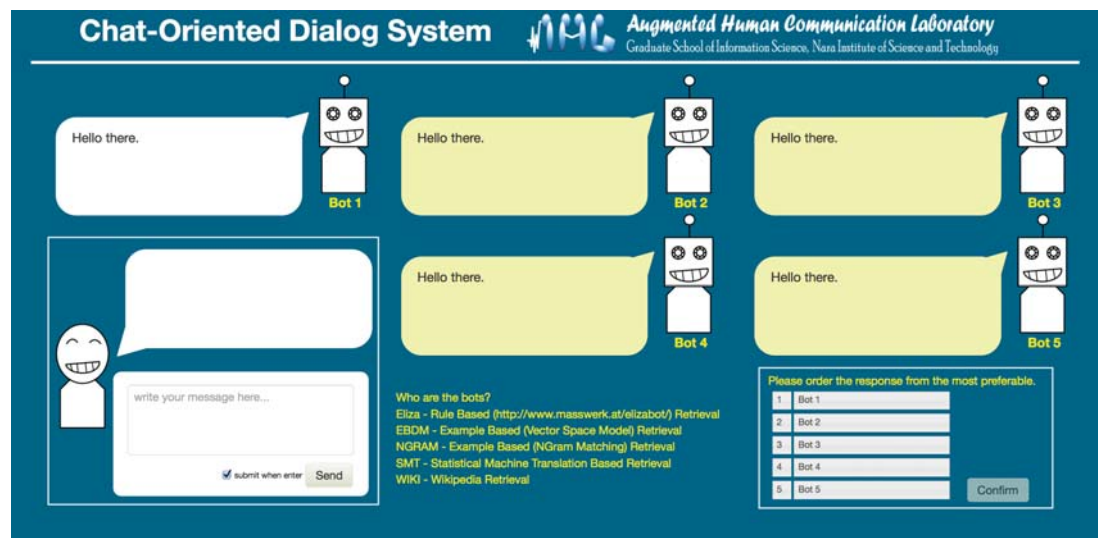


Figure 30.: Screen shot of the dialog system web demo application.

[‡]<https://www.wikipedia.org/>

[§]<https://developers.google.com/web-search/docs/>