Response Selection for End-to-End Retrieval-Based Dialogue Systems

Basma El Amel Boussaha

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

UNIVERSITÉ DE NANTES

THÈSE DE DOCTORAT DE

L'UNIVERSITE DE NANTES
COMUE **U**NIVERSITE **B**RETAGNE **L**OIRE

Ecole Doctorale N°601
Mathèmatique et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Informatique*

Par

Basma El Amel BOUSSAHA

Response Selection for End-to-End Retrieval-Based Dialogue Systems

Thèse présentée et soutenue à NANTES le 23 Octobre 2019

Unité de recherche : Laboratoire des Sciences du Numérique de Nantes (LS2N)

Rapporteurs avant soutenance :

Frédéric BÉCHET Professeur des universités, Aix Marseille Université, LIS
Sophie ROSSET Directrice de recherche, Université Paris-Sud, LIMSI

Composition du jury :

Président :	Yannick ESTÈVE Professeur des universités, Université d'Avignon, LIA
Examinateurs :	Frédéric BÉCHET Professeur des universités, Aix Marseille Université, LIS
	Sophie ROSSET Directrice de recherche, Université Paris-Sud, LIMSI
	Yannick ESTÈVE Professeur des universités, Université d'Avignon, LIA
Dir. de thèse :	Emmanuel MORIN Professeur des universités, Université de Nantes, LS2N
Co-enc. de thèse :	Christine JACQUIN Maître de Conférences, Université de Nantes, LS2N
	Nicolas HERNANDEZ Maître de Conférences, Université de Nantes, LS2N

“ It is said that to explain is to explain away. This maxim is nowhere so well fulfilled as in the area of computer programming, especially in what is called heuristic programming and artificial intelligence. For in those realms machines are made to behave in wondrous ways, often sufficient to dazzle even the most experienced observer. But once a particular program is unmasked, once its inner workings are explained in language sufficiently plain to induce understanding, its magic crumbles away; it stands revealed as a mere collection of procedures, each quite comprehensible. The observer says to himself "*I could have written that*". With that thought he moves the program in question from the shelf marked "intelligent", to that reserved for curios fit to be discussed only with people less enlightened than he.”

— Weizenbaum (1966)

Acknowledgment

I would like to express my gratitude and thanks to all the people who accompanied me throughout this thesis. I thank my supervisor Emmanuel MORIN and my co-supervisors Christine JACQUIN and Nicolas HERNANDEZ for their valuable and hard work during these three years to make this PhD succeed. I also thank them for their integrity, their support, their availability and their generosity to share their knowledge. Thank you to each of them for their time, effort and involvement which allowed me to finish my PhD on time.

My sincere thanks go to Frédéric BÉCHET, Sophie ROSSET and Yannick ESTÈVE for taking part of my thesis committee and for providing valuable comments. I thank Sophie again and Hoël LE CAPITAINE for taking part of my thesis supervision committee and for their valuable remarks and questions during our annual meetings.

I would also like to thank all the TALN team members who have accompanied me from near or far and shared with me this journey. Thanks to everyone I have met in the lab for the endless but always constructive debates. A special thank to Florian BOUDIN for all our interesting discussions around lunch and coffee breaks.

I especially thank my family for their endless love, support and encouragement throughout the course of this research. I am also grateful to my friends for their presence, their help and their emotional support. Finally, I thank the one who believed in me, supported me and stood by my side through the best and the worst, my dearest husband Karim.

Contents

List of Tables

List of Figures

1

Introduction

> “Computers are incredibly fast, accurate and stupid; humans are incredibly slow, inaccurate and brilliant; together they are powerful beyond imagination.”
>
> — Leo Cherne

1.1 Context

Whenever you want to change your flight departure date and, for some reason, you can not do that online. Would you prefer calling your airline company and wait for more than 15 minutes to talk or chat with a human assistant, or text a chatbot and get your ticket changed in a few minutes? According to a recent study by HubPost [1], 56% of people prefer messaging than calling customer service, 53% of people are more likely to shop with businesses they can message and 71% of people use chatbots to solve their problems fast. These numbers are pushing companies day after day to build intelligent and automatic dialogue systems to satisfy this increasing demand.

KLM Royal Dutch Airlines is one of multiple companies that are facing the increasing number of Internet users and are not able to provide the necessary assistance to its customers

1. https://www.hubspot.com/stories/chatbot-marketing-future

by relying on only humans. The company launched her chatbot called BlueBot (BB) [2] via Facebook Messenger in late 2017 to handle more than 16,000 customer interactions per week and to reinforce its 250 human assistants (Faggella, 2019). Quickly after its launching, BB sent around 2 million messages to more than 500,000 customers in the first 6 months. Today, BB is available on Google Home with voice/audio assistance and is being able to handle twice as many customer requests.

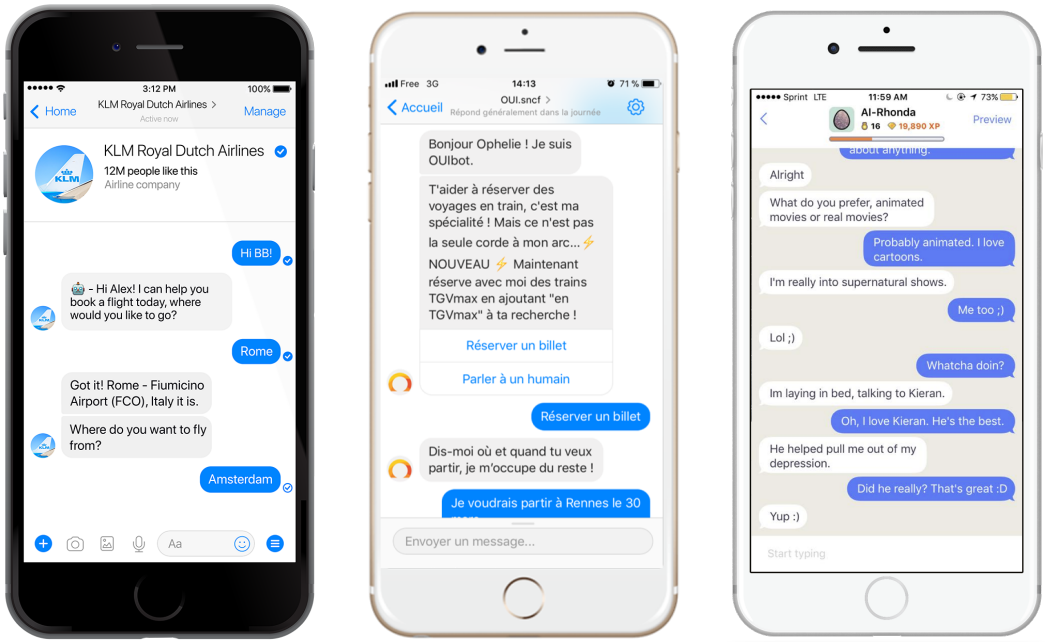


Figure 1.1 – Examples of chatbots from different companies and with different purposes. From left to right: KLM BlueBot, Oui.sncf chatbot and Replika the AI friend.

The Senior Vice President of Air France - KLM summarizes the need of such chatbot as follows (Amstelveen, 2019):

> "KLM is well known for its personal approach. On social media, we offer 24/7 service with our team of 250 human agents, handling more than 16,000 cases a week. Volumes will continue to grow. At the same time, customers require a speedy response. We have therefore been experimenting with Artificial Intelligence to support our agents to provide a personal, timely and correct answer. With BB, KLM is taking the next step in its social media strategy, offering personal service through technology, supported by human agents when needed."

A chatbot, a dialogue system or a virtual agent is a computer program that can conduct a conversation with humans via speech or text to understand their needs and provide answers to their questions. Chatbots are widely used in customer service to provide advantages to both companies and costumers in a win-win strategy. On one hand, they help companies in making their employees getting rid of routine tasks to focus on more strategic tasks and decisions for the benefit of the company. On the other hand, they help costumers processing their requests quickly and efficiently and thus gain their loyalty.

2. https://bb.klm.com/en

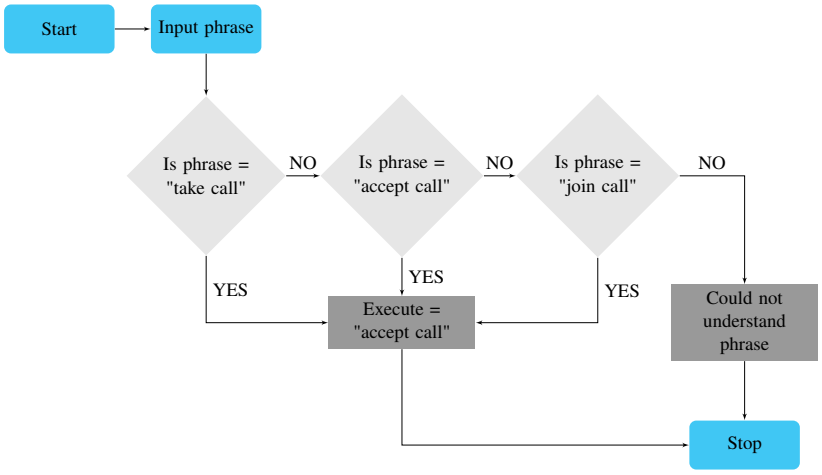


Figure 1.2 – A simple rule-based flow that describes the process of a call.

The interest in building dialogue systems dates back to the 60s with ELIZA the artificial psychotherapist which was built at the MIT Artificial Intelligence Laboratory by Joseph Weizenbaum (Weizenbaum, 1966). Later, in 1972, PARRY was created at Stanford University to simulate a person with paranoid schizophrenia (Colby, 1975). During the Turing test, professional psychiatrists were able to distinguish only 48% of responses provided by PARRY to be from a real human or a chatbot. In the 90s, two famous chatbots were born: Jabberwacky [3] the first chatbot that simulates human voice and ALICE (Wallace et al., 2003) the three-times winner of the Loebner Prize Competition in Artificial Intelligence (2000, 2001 and 2004). ALICE is the chatbot that originally inspired Spike Jonze, the American filmmaker, for his academy award-winning film Her. Afterward, the last decade witnessed the birth of a new generation of dialogue systems that have been successfully marketed and made accessible for all on their smartphones, laptops and touchpads. Siri (2010), Google Now (2012), Alexa (2015), and Cortana (2015) are examples of these dialogue systems.

Historically, the first dialogue systems were rule-based which means that the designer of the chatbot defines a set of rules that the system has to follow to produce a response. These rules might be very simple or very complex depending on the task to be accomplished. Building these systems is straightforward but defining the set of rules that cover all the possible scenarios is a time-consuming and tedious task. Therefore, the flexibility of the chatbot in answering questions whose pattern does not match the predefined rules is a drawback. Artificial Intelligence Markup Language (AIML), which formed the basis of ALICE chatbot, is an XML based programming language that allows the developers to define patterns and rules for rule-based dialogue systems. Figure 1.2 illustrates a simple rule-based flow of different patterns of accepting a call.

Lately, the 90s witnessed the birth of a new category of dialogue systems based on machine learning. These dialogue systems overcome the flexibility drawback of rule-based systems by automatically learning how to respond based on supervised approaches relying

3. http://www.jabberwacky.com/

on feature engineering. Some of them, even if they do not require rules to follow, still depend on some features defined by the developers to model the conversations. Examples of features are the length of the context, the dialogue turn, the number of utterances, the number of n-grams, etc. which, once again, constraints the flexibility of dialogue system especially when changing the domain as the features need to be re-identified.

Afterwards, more precisely, at the beginning of the last decade (2010s), many factors such as the availability of large datasets and the accessibility of large computing resources made training deep neural networks easier than before. This allowed the dialogue systems to move to another level where features are automatically extracted from data without any need of humans to define what features to be used. Deep learning changed many fields including Natural Language Processing, Computer Vision, Signal Processing, etc. and record performances were achieved. Approaches based on deep learning require much more data compared to feature-based approaches since the system automatically defines relevant features and thus the systems may take a long time to converge i.e learn the optimal representation of data. Today, a large amount of human-human conversations are available thanks to social media, emails, and community question-answering platforms (Song et al., 2018). The availability of Graphical Processing Units (GPUs) also helped to accelerate the learning and prediction processes. Therefore, researchers are now able to build automated dialogue systems that learn from human-human conversations to produce human-computer conversations with lower costs.

Even if deep learning-based systems achieved remarkable performances in several tasks and multiple domains, they receive a lot of criticism due to the difficulty of their interpretability and the large amounts of data required to train them (Doshi-Velez and Kim, 2017; Dong et al., 2019). Interpretability means the ability to explain the reasons why the model produced an output x instead of y. In the case of Deep Neural Network (DNN) composed of multiple neurons and layers, usually, we are unable to understand what do the weights of each connection mean? How can we interpret a neuron that fires for some inputs? Which is the set of weights that play the most important role in the final prediction? And other questions that we are unable to answer. For these reasons, some companies still use feature-based or even rule-based systems as they are simple, financially and computationally cheaper because they can run on only CPUs and can be quickly developed and put into production. They are also still used because the companies have small datasets of thousands of samples which is not enough to train deep learning models or because they treat a very small domain for which rules or features can be identified easily. Nevertheless, these systems are less robust and are hard to maintain.

To produce a response, self-learnable chatbots, whether they are feature-based or deep learning-based, either generate a reply word by word or retrieve the response from a set of predefined responses. Therefore, we distinguish two categories of dialogue systems. The first category regroups generative dialogue systems that emulate humans in generating responses word by word. Even if these systems are smarter enough to produce responses from scratch, yet they still have a hard constraint to satisfy: the grammar, conjugation, and spelling of the generated responses. The second category regroups retrieval-based dialogue systems which fetch the correct response from a set of candidate responses and pick the

response that matches the most the context of the conversation. As these responses are already written by humans, the structure of the sentence is not a problem to worry about. Nevertheless, both generative and retrieval-based dialogue systems have to produce adequate response on the semantics and the discourse levels.

In the literature, in the case where a dialogue system is applied for a specific task such as restaurant booking or technical help desk, retrieval-based dialogue systems were mostly used. The reason is that in such configuration, the set of responses that the chatbot has to produce can be known in advance. Hence, retrieval-based dialogue systems, if trained well, can retrieve the best response from this set of candidate responses. However, in open domains where the chatbot has to discuss different subjects ranging from weather, football to telling jokes, generative systems are more suitable. Their capacity to produce responses word by word make them more appropriate to talk on different subjects even if they suffer from the shortness and the generality of their generated response such as *"Ok"*, *"Of course"*, *"Thank you"*, etc.

In this thesis, we study task-oriented retrieval-based dialogue systems as we believe that we need more systems of this category to handle the increasing number of customers who need daily assistance. We argue that open-domain dialogue systems have a more commercial purpose which is out of our thesis scope. We orient our study towards end-to-end systems for mainly two reasons. First, we follow the trend of building end-to-end architectures as the recent state-of-the-art systems. Second, we focus on this category of architectures for the multiple advantages that it has, such as bypassing the intermediate layers of the modular architectures which are less flexible and require multiple algorithms and separate optimization of each layer. More details about these advantages are given in Chapter 2.

1.2 Problem and motivations

Making machines understand the natural language is the goal of many companies and AI labs since many years ago. Solving the ambiguities, the co-reference and determining the structure of the dialogue, etc. are the challenges that we are facing today. Many languages exist and different ways exist to express the same idea and say the same thing. Today, the recent dialogue systems are based on deep learning models and have an end-to-end architecture in which the model automatically learns implicit representation of the input and attempts to inference the output. Despite their capacity to learn deep representations, they still have issues with memorizing long input sequences and automatically determining which parts of the sequences are the most important. More specifically, in dialogues, the interlocutors follow a certain coherence and methodology when discussing. To reply to a given conversation, either by generating or retrieving the response, the dialogue system must automatically understand the implicit structure of the conversation in addition to the parts of the discussion that describe the problem and the solution. However, this implicit information are difficult to capture.

We hypothesize that correct responses are semantically similar to the context of the

conversation. For example, we suppose that in a conversation where we talk about food, responses about restaurants, recipes are possible correct responses. Thus we first build a system that matches the context with the candidate response by computing their semantic similarity. Through our study of the state-of-the-art-systems, we found that some systems based on similar hypotheses exist. We study them and show that these systems have some drawbacks that we highlight in the following.

— Some of these systems use external knowledge which is sometimes handcrafted such as a database of technical entities and verbs (Serban et al., 2017a). This makes the model very domain dependent as applying the same model to Ubuntu technical assistance while it was previously designed for restaurant booking requires defining a new knowledge base which is time-consuming and constraints the flexibility of the model.

— The complexity of the suggested architectures is quite high as the researchers tend to combine different networks without evaluating the necessity of having such complications. We argue that we need to develop simple architectures with fewer dependencies which are easier to train and evaluate and also which allow other researchers to reproduce results.

1.3 Contributions

All along this thesis, we were focusing on two main goals: (1) Building efficient retrieval-based dialogue systems that are as flexible as possible by reducing their domain dependency and (2) Making dialogue systems simpler by using the necessary components for targeting the desired information that we need to capture. To achieve these two goals, the following contributions have been made:

— We were inspired by a state-of-the-art single-turn matching system called dual encoder (Lowe et al., 2015b) for its simplicity and efficiency in matching the context and the candidate response on the sequence-level. We addressed the drawbacks that we identified in its architecture through a new system. Our proposed system is single-turn and single-level, thus, it enjoys the advantages of its simplicity and efficiency (Boussaha et al., 2018b) (Boussaha et al., 2018a). The data and the source code are made available on `https://github.com/basma-b/dual_encoder_udc`.

— We extended our single-turn and single-level system with another similarity level that we compute between the words. Our new system achieves high performances while being conceptually simple and having fewer parameters compared to the previous, substantially more complex, systems. (Boussaha et al., 2019b) (Boussaha et al., 2019c). We make publicly available the data and the source code to reproduce all our experiments on `https://github.com/basma-b/multi_level_chatbot`.

1.4 Outline

The first part of this thesis presents the current state-of-the-art of dialogue systems of both retrieval, generative and hybrid systems, their evaluation metrics as well as the available datasets used to build and evaluate dialogue systems. Chapter 2 introduces the different categories of dialogue systems and their different architectures. Moreover, it provides a succinct description of the most recent dialogue systems of each category. A detailed list of evaluation metrics used to evaluate each category is provided as well. Chapter 3 summarizes a list of the existing and most used datasets in building and evaluating dialogue systems. A classification of the datasets per category as well as a description of each dataset is provided.

The second part thoroughly describes the different contributions that we have made during the course of this thesis. Chapter 4 introduces a new retrieval-based dialogue system that was inspired by a state-of-the-art system. A deep analysis and an evaluation of the impact of different components of the system are performed as well as an evaluation of incorporating the attention mechanism. Moreover, a qualitative and quantitative analysis of the system's performance is done to understand the reasons for failure through error analysis.

Throughout this thesis, we highlight the importance of building simple and domain-independent systems and we consider the simplicity of the approach as the same level of importance as its performance as it allows researchers to easily reproduce the results of the approach on the same or different datasets. Chapter 5 introduces our second contribution as an extension of the system described in Chapter 4. This extension consists of the introduction of a new level of similarity that enriches the previous system. We evaluate our system on two different dialogue datasets of different languages and we show the efficiency of our system compared to systems of the same category and of other categories. The Dialog System Technology Challenges (DSTC7) offered a perfect evaluation platform for our system that allowed us to confront more realistic challenges. We showed that on two more datasets and with a challenging environment, our system was able to perform well. An ablation study as well as an error analysis were performed to better understand the importance of different parts of our model in addition to the origin of errors. Interesting results were obtained and new perspectives were identified.

2

Dialogue systems

> “ I believe that at the end of the century, the use of words and general educated opinion will have altered so much that one will be able to speak of machines thinking without expecting to be contradicted.”
>
> — Alan Turing

2.1 Introduction

A dialogue system also known as a dialogue agent or a chatbot is a computer program that simulates a written or a spoken conversation with one or multiple humans. It aims to emulate human behavior and intelligence in answering complex questions and handling long and coherent conversations. The interest in building dialogue systems is not recent and one of the oldest systems is ELIZA [1] the Rogerian psychotherapist bot (Weizenbaum, 1966). ELIZA was built in 1966 based on the reformulation of the interlocutor's statements to questions. Nowadays, building dialogue systems is being a hot and attractive research domain in both industry and academia and much progress has been done in the last years. Their domains of application range from help-desk assistance to AI friends.

1. An implementation of ELIZA is available on https://www.masswerk.at/elizabot/

Typically a dialogue system attempts to understand the user utterance whether it consists of a question, affirmation, greeting, etc. and to construct a relationship between this utterance and the previous ones. Once the user's intention is identified, the dialogue system proceeds to the production of a meaningful, coherent and adequate answer. Even though the impressive improvement that witnessed dialogue systems in the last decade, they are still facing multiple challenges preventing them from reaching a higher level of intelligence and robustness. Ward and DeVault (2016) in his work identifies 10 challenges among which modeling user variation, synthesizing multifunctional behaviors, low resource learning, etc. appear.

Dialogue systems can be written or spoken. In terms of application, they can be roughly divided into two categories: chitchat (chat-oriented) and task-oriented. Systems of the first category can be engaged in a conversation without necessarily being involved in a task that needs to be accomplished. Examples of chitchat systems include Replika, Cortana, Siri, XiaoIce, etc. Usually, these systems have a commercial benefit as they are developed to be artificial companions of humans. On the other hand, task-oriented dialogue systems converse with humans to fulfill a precise task such as restaurant reservations, advising students in course selection, train ticket booking, etc. Today, there is an increasing need for building task-oriented dialogue systems to help companies handling the increasing number of users. In this thesis, we focus on task-oriented dialogue systems as we believe that today we need this kind of dialogue system to help companies building smarter, smoother and more flexible dialogue systems and to help costumers by answering their requests effectively and quickly. Moreover, conversations could be between two or more users. Multi-party conversations are out of the scope of this work. In this thesis, we study written task-oriented dialogue systems with a special focus on deep end-to-end architectures for their advantages.

In this chapter, first, we describe briefly what is dialogue and dialogue systems. Then, we present the different categories of dialogue systems and the most recent works in each category with a special focus on deep end-to-end architectures in addition to the metrics used to evaluate each category. Moreover, we summarize machine learning foundations that the rest of this dissertation will build on.

2.2 Dialogue structures and dynamics

Before studying dialogue systems, it is important to understand the definition of dialogue and what are the properties and the structure of dialogue that the dialogue systems have to consider when conversing. Dialogue has been subject of multiple studies of several domains such as philosophy and logic, linguistics and psycholinguistics. A dialogue can be seen as a succession of utterances (dialogue turns) between two or multiple parties who have goals and intentions. Stent (2001) defines dialogue as *"interaction involving two participants, each of whom contributes by listening to the other and responding appropriately"*. Clark (1996) defines the dialogue as a joint activity in which language plays an especially prominent role. He says: *"Language use arises in joint activities. You call up your sister for an address, or talk to a friend about what to take on a picnic, or discuss news*

with a colleague. If you were later asked "What did you do?" you wouldn't describe your acts of speaking. You would describe the joint activities you took part in. "I got an address from my sister." "My friend and I decided what to bring on a picnic." "A colleague and I traded gossip." In each case, you take the joint activity to be primary, and the language you used along the way to be secondary, a means to an end". In this activity the interlocutors are not merely making individual moves, but actively collaborate (Schlöder and Fernández, 2015). More particularly, Clark (1996) assumes that a dialogue, as a joint activity, has the following properties:

1. A joint activity is carried out by two or more participants.
2. The participants in a joint activity assume public roles that help determine their division of labor.
3. The participants in joint activity try to establish and achieve joint public goals.
4. The participants in a joint activity may try individually to achieve private goals.
5. A joint activity ordinarily emerges as a hierarchy of joint actions or joint activities.
6. The participants in a joint activity may exploit both conventional and non-conventional procedures.
7. A successful joint activity has an entry and exit jointly engineered by the participants.
8. Joint activities may be simultaneous or intermittent and may expand, contract, or divide in their personnel.

A dialogue can not be seen merely by its linguistic structure but other concepts have to be considered. Yet the speakers, listeners, times, places, and the circumstances of utterance are important but the concept of goals (public or private), procedures (conventional or non-conventional) and actions (sequential or simultaneous) have to be considered as well (Loisel, 2008). To achieve this coordination, the speakers have to share a common ground that they accumulate in a joint activity. This common ground falls into three parts: what the participants presupposed on entering the activity, the current state of the activity, and the public events that led up to the current state. The principal task of the speakers during their joint activity is to synthesize the common ground and ensure that they share the same common ground.

Dialogue systems aim to converse with humans in natural language. For this, they have to be able to understand the context of the conversation, identify the intent and the goal of the speaker and produce a coherent response. Here we can distinguish three key concepts of dialogue. First, in the literature, the term *context* may refer to the preceding discourse, to the physical environment or to the domain of the discourse. Bunt (1999) defines the context as a set of factors relevant to the understanding of communicative behaviour. He categorizes the context into five categories: linguistic, semantic, cognitive, physical, and social. What he calls linguistic context is what is mostly considered as the dialogue history. Second, the speaker intentions and goals are usually implicit and encoded inside the utterances. Instead of telling the hearer what to do, the speaker may just state his goals, and expect a response that meets these goals at least part way (Sidner, 1983). The third concept is the dialogue cohesion. In linguistics, cohesion is what makes a text semantically meaningful

it studies how linguistic markers such as anaphoric references and pragmatic connectives structure the dialogue (Loisel, 2008). For a successful conversation, a dialogue system have to consider these concepts when conversing with humans. In the following section, we study the different categories of dialogue systems and how do they use dialogue and its structure to converse with humans.

2.3 Categories of dialogue systems

A dialogue system processes the history of the conversation called the context and attempts to respond by producing a coherent response. According to whether the response of a given context is generated, retrieved or both approaches are combined to produce the response, we distinguish three categories of dialogue systems.

2.3.1 Generative dialogue systems

Generative dialogue systems perfectly imitate humans in generating responses word by word. First, they start by understanding the input text to produce an internal representation. Then based on this representation, they produce a response. Generative models have been extensively studied over the last decade (Ritter et al., 2011; Graves, 2013; Wen et al., 2015; Serban et al., 2016; Sordoni et al., 2015b; Dušek and Jurčíček, 2016; Shao et al., 2017; Pandey et al., 2018; Zhang et al., 2018b). They are applied into different domains including machine translation, image captioning, etc. In the literature, two architectures of dialogue systems exist (Chen et al., 2017): modular and end-to-end architectures.

Modular architecture as shown in Figure 2.1, a typical modular dialogue system is composed of mainly 4 components working in pipeline (Chen et al., 2017; Liu and Lane, 2018; Gao et al., 2018a). First, given the user utterances, the Natural Language Understanding (NLU) module maps them into a semantic representation of dialogue acts. The Dialogue State Tracker (DST) processes this information and outputs the current dialogue state which is used by the Dialogue Policy (DP) to select the next system action. The DST and DP together form the Dialogue Manager (DM) which is the central controller of the dialogue system. Finally, the Natural Language Generation (NLG) module generates utterances in natural language form based on the system's action. Usually, a database such as a bus timetable or web API is required by the DST and DP modules to provide external information (Tur, 2011; Young et al., 2013; Liu and Lane, 2018).

The major limitation of this modular architecture is that the optimization of each module is often made individually and the improvement of individual modules does not necessarily mean the improvement of the whole system (Gao et al., 2018b). Thus, adapting one module to a new environment requires adapting the other modules, slots, and features which is not only expensive and time-consuming but also limits the adaptation of the whole system to other domains. Furthermore, the intermediate interfaces between the components are

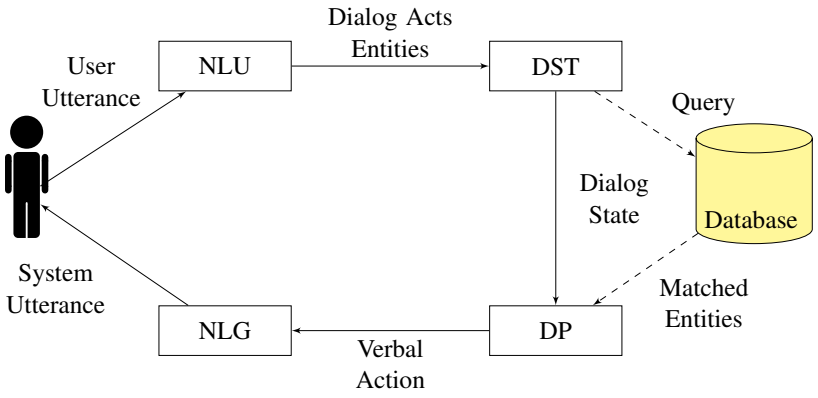


Figure 2.1 – Pipeline architecture of task-oriented dialogue systems (Liu and Lane, 2018). NLU, DST, DP and NLG denote Natural Language Understanding, Dialogue State Tracker, Dialogue Policy and Natural Language Generation respectively.

hand-crafted making the generalization of the system to multiple or new domains tedious (Tiancheng, 2019). The increasing popularity of neural networks resulted in an increasing interest in jointly optimizing multiple modules as one single module. This results in a new architecture called end-to-end.

End-to-end architecture has recently been widely adopted in multiple domains as it allows to alleviate the limitations of the modular architectures. The principle of this architecture is to regroup the 4 components into one single module trained with one objective function. Here, less manual effort is required as the intermediate interfaces between components are deleted and humans are no more needed to manually define features. End-to-end architectures enjoy the advantage of the differentiability of neural networks which can be optimized by gradient-based methods. When all the components of the dialogue system are differentiable, then the whole system becomes differentiable and can be optimized through back-propagation (Gao et al., 2018b). This represents the main advantage of end-to-end architectures compared to modular architectures which require the optimization of each module individually.

End-to-end generative dialogue systems are trained on large human-human conversations using deep neural networks to produce human-like responses (Ritter et al., 2011; Sordoni et al., 2015b; Vinyals and Le, 2015; Shang et al., 2015). The most common and powerful model for dialogue generation is the encoder-decoder model (Cho et al., 2014; Vinyals and Le, 2015). The first encoder-decoder model has been introduced by Sutskever et al. (2014) as a framework called *sequence-to-sequence* and got much interest in machine translation. Its basic idea is to use neural networks to map the input sequence to the output sequence. The architecture of an encoder-decoder model used for dialogue generation is illustrated in Figure 2.2. It is composed of two recurrent neural networks: the first one is called encoder and the second one is the decoder. The encoder learns to represent sentences in a given language by updating its hidden state each time a word embedding is fed (Cho et al., 2014). Once the end of the utterance marker is detected, the hidden state of the en-

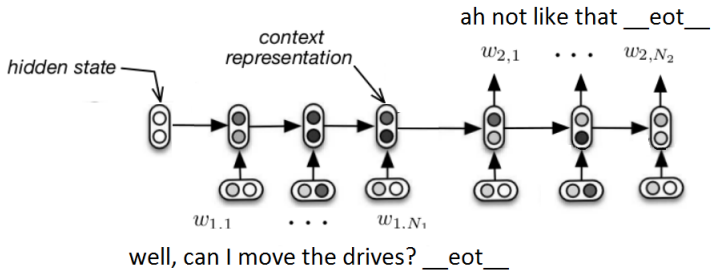


Figure 2.2 – A sequence-to-sequence model used for dialogue generation (Serban et al., 2018). The hidden state of the encoder is updated each time a token $w_{1,i}$ is fed. The decoding starts as soon as the last token w_{1,N_1} of the input is fed. *__eot__* denotes the end of turn.

coder is given as an input to the decoder. Note that this hidden state contains a representation of all the input. The decoder then will generate the response word by word.

Despite the simplicity of the encoder-decoder model, it achieved impressive results in multiple applications of Natural Language Processing, including image captioning (Vinyals et al., 2015; Rennie et al., 2016), dialogue generation (Eric et al., 2017; Pandey et al., 2018), automatic summary (Lebanoff et al., 2018; Hou et al., 2018a), machine translation (Cho et al., 2014; Zhang et al., 2018a), etc. Even if it gave good results for machine translation, its adaptation to text generation was not without flaws. The task of generating responses is considerably more difficult than translation because of the wide range of possible responses and the fact that the context and the response have different lengths unlike the original and translated sentences in machine translation (Serban et al., 2016). Moreover, with the encoder-decoder architecture, it is difficult to handle long conversations due to the memory issues of the Long Short-Term Memory Networks (LSTMs) and the Recurrent Neural Networks (RNNs) (Bengio et al., 1994). Therefore, most of the utterances generated by this model were short and general (eg. *"I am sorry"* and *"I don't know"*) with less coherence regarding the context of the conversation (Vinyals and Le, 2015; Sordoni et al., 2015b).

To address these flaws, Li et al. (2016a) suggested training a sequence-to-sequence model to maximize the mutual information instead of the log-likelihood of the response given the context. By using this objective function, the model was able to generate longer, diverse and more coherent responses since it learns not only the dependency of responses on the context but also the likelihood that the context is given for the generated response. Also, Sordoni et al. (2015a) proposed to exploit the hierarchical structure of the dialogues through its hierarchical encoder-decoder model. Simply, they encode each dialogue turn of the context separately using turn encoders and continuously update the context encoder hidden state which is then used by the decoder to generate the response. Other works focused on augmenting the input of the encoder-decoder models by extracting some information from the conversations. Li et al. (2016b) encoded background information and speaking style of the speaker into distributed embeddings that they use later to re-rank the generated responses. Xing et al. (2017) encoded topic information based on Latent Dirichlet Allo-

cation (LDA) (Blei et al., 2003) to force the system to generate topic coherent responses. Serban et al. (2017a) extended the hierarchical encoder-decoder (Sordoni et al., 2015a) with a new objective function: the joint log-likelihood. Unlike the original objective function, the log-likelihood, this new objective function allows modeling natural language generation as two parallel discrete stochastic processes: a sequence of high-level coarse tokens, and a sequence of natural language tokens.

As the encoder-decoder model became famous, many extensions and improvements have been introduced. The attention mechanism (Bahdanau et al., 2015; Luong et al., 2015) is one of the most important extensions that enables the encoder-decoder better modeling long sequences (Tiancheng, 2019). More concretely, it allows the encoder creating a dynamic size distributed representation of the context from which the decoder can retrieve the most informative parts of the input. Therefore, the model can encode-decode long input sequences while paying attention to the most important parts. Another extension of the encoder-decoder model, is the copy-mechanism (Gu et al., 2016). It allows the model to selectively replicate some segments of the input sequences in the output sequences. By this simple approach, copy-mechanism helps the encoder-decoder model handling rare and out of vocabulary (OOVs) words.

2.3.2 Retrieval-based dialogue systems

Unlike generative dialogue systems, retrieval-based dialogue systems do not generate responses word by word but have a fixed list of candidate responses from which they pick the most relevant response for the context. More particularly, they fetch the most appropriate response for the given context from a list of candidate responses. Afterward, the response with the highest score is returned as the response to the context. Hu et al. (2014) defines the process of matching textual sentences as follows: *"Matching two potentially heterogeneous language objects is central to many natural language applications (Xue et al., 2008; Bordes et al., 2014). It generalizes the conventional notion of similarity (e.g., in paraphrase identification (Socher et al., 2011)) or relevance (e.g., in information retrieval (Wu et al., 2013)), since it aims to model the correspondence between "linguistic objects" of different nature at different levels of abstractions"*. Therefore matching two textual segments is a challenging problem as the system needs to model the segments as well as their relations (Wang et al., 2015). In machine translation, the model has to determine whether the input sentence and its translation have the same meaning whereas, in dialogues, the model needs to know whether an utterance is a response to a given conversation.

Given the history of a technical conversation between two users in Figure 2.3, a response retrieval system should rank the first response before the second one. It is important that the system captures the common information (carried by terms written in bold) between the context turns and between the whole context and the candidate response. Sometimes the correct response has common words with the context which is quite simple to capture with a simple statistical model but sometimes the correct response does not share any words with the context. In this former case, the task becomes harder as implicit information has to be

	Context
A	Hi, I can not longer access the graphical **login screen** on ubuntu 12.04
B	what exactly happen?
A	I can't remember the error message, would it have auto-logged to a file or should I reboot quick?
B	you mean it won't **automaticaly start** and what happen then?
A	it just stop at a text **screen**, but I can access the command line **login** via alt F1-6, and **start** x **manually** there. I think it might me **lightdm** that's break but I'm not sure
	Candidate responses
R1	for me **lightdm** often won't start **automatically** either. It show me console tty1 instead and I have to **start lightdm manually** ✓
R2	what about sources.list ? ✗

Figure 2.3 – Example of a conversation between two participants (A and B) extracted from the Ubuntu Dialogue Corpus (Lowe et al., 2015b). R1 is the ground-truth response whereas R2 one is a wrong response. Important terms that are common between the context and the candidate responses are written in **bold**.

captured. According to Wu et al. (2017), the challenges of the task of response retrieval in dialogue systems are: (1) How to identify important information (words, phrases, and sentences) in the context and how to match this information with those in the response and (2) How to model the relationships between the context utterances.

As illustrated in Figure 2.3, the context is composed of multiple turns/utterances [2] issued from a conversation between two users. The response to this context may reply to the first utterance, to the second, .. or to the last or even to all the utterances. In the literature, retrieval-based dialogue systems are based on different hypotheses and can be roughly divided into two categories: single-turn and multi-turn matching models. As we can see in Figure 2.4a, single-turn matching systems match the response only once with the whole context or with only its last utterance. Whereas, multi-turn matching models consider the context utterances separately when matching with the response. Hence, first, they match the response with each of the context utterances U_i and then, they aggregate the matching scores to obtain the final score of the candidate response as illustrated in Figure 2.4b. Each of these categories has advantages and disadvantages that we explore in the following sections.

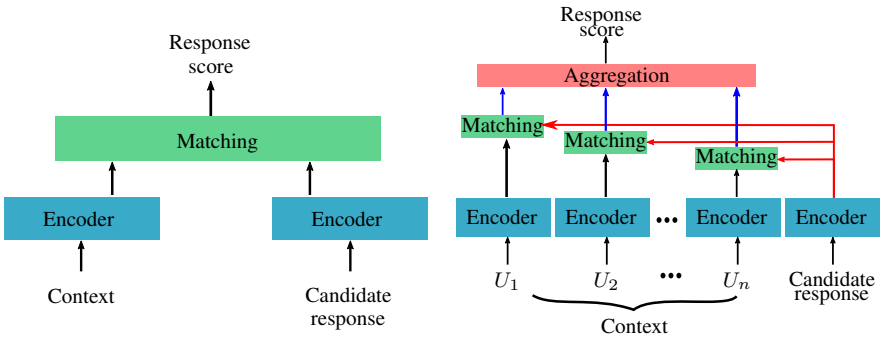
2.3.2.1 Single-turn matching models

Systems of this category match the context with the candidate response only one time (Figure 2.4a). Some systems concatenate the context utterances into one single utterance to which they match the response without explicitly distinguishing the context utterances. Others, reduce the context to its last utterance to which the response is matched. The early works on response retrieval in dialogue systems consider the problem as question answering by finding the response that replies to the last turn of the context. They do so by searching in a question answering database for questions that are semantically similar to the last question

2. We use the terms *turn* and *utterance* indifferently.

of the user based on semantic relatedness measures e.g. BM-25 (Manning et al., 2008; Graesser et al., 2004; Jeon et al., 2005; Banchs and Li, 2012).

Afterward, as neural networks became more popular, they were used in matching textual sequences. Lu and Li (2013) proposed a deep neural network model for short text matching by explicitly capturing the non-linearity and the hierarchical structure when capturing the object-object interaction between questions and answers. Hu et al. (2014) exploited the capacity of Convolutional Neural Networks (CNNs) of nicely representing the hierarchical structure of sentences with convolution and pooling and their large capacity of matching patterns at different levels. Wang et al. (2015) first represents the input sentences with their dependency trees (Filippova and Strube, 2008) which allow revealing both short-distance and long-distance grammatical relations between words. Then, dependency tree matching patterns are extracted with a mining algorithm from the direct product of the dependency trees. These patterns are incorporated into a deep neural network to determine the matching degree of the two input texts.



(a) Single-turn matching models (b) Multi-turn matching models

Figure 2.4 – General architectures of single- and multi-turn matching models. U_i denotes the i^{th} utterance of the context.

More recently, Wu et al. (2016) proposed a topic augmented neural network for matching short text conversations. Their network is composed of three layers: a sentence embedding layer which represents the question and the response into a vector space, a topic embedding layer which forms a topic vector from the embeddings of the topic words extracted by a Twitter LDA model (Zhao et al., 2011) and finally a matching layer which computes a matching score between the question and the response based on the question, response and topic vectors. Xu et al. (2017) proposed to include domain knowledge into a deep neural network. They introduce an extra memory gate to a Long Short-Term Memory (LSTM) cell to incorporate background knowledge and thus they enhance the sequence semantic modeling ability of LSTM.

Even if these single-turn matching systems achieve good performances, they still target a small category of applications. Almost all of them match short textual segments such as questions and answers. However, when considering long texts such as long conversation contexts, the performance of these systems may decrease considerably. Also, we believe that considering only the last utterance of the context and trying to match it with the pos-

sible responses may constraint the performance of the dialogue system. In a conversation, a random utterance could be as simple as *"Ok"*, *"Thanks"* or *"Yes, it is"* which can not represent the whole context of the conversation.

Other single-turn matching systems consider all the utterances of the context by concatenating them together as a single long turn to which they match the responses. A simple yet effective baseline system is the Term Frequency-Inverse Document Frequency (TF-IDF) (Salton et al., 1975) which is a statistical measure to evaluate the importance of a word in a document regarding its importance in a corpus. It hypothesizes that a textual unit is important in a given document if it appears a lot in this document and that is less frequent in the corpus. TF-IDF of a textual unit tu is given with the following formula.

$$\text{TF-IDF}(tu) = \text{TF}(tu) \times \log(\frac{N}{\text{DF}(tu)}) \tag{2.1}$$

Where TF denotes Term Frequency which represents the number of occurrence of the textual unit tu in a document, DF denotes the Document Frequency which represents the number of documents of the corpus where tu occurs and N is the total number of documents of the corpus. TF-IDF has been widely used in Information Retrieval and still represents a strong baseline for multiple domains such as keyphrase extraction (Hofmann et al., 2009) and machine translation (Eck et al., 2005). Lowe et al. (2015b) used the TF-IDF as a baseline for retrieval-based dialogue systems. The context utterances were concatenated and each word of the context is represented as a TF-IDF score. These scores are stacked into a vector that represents the context and a vector of each of the candidate responses is obtained following the same process. The hypothesis of the TF-IDF retrieval-based dialogue system is that the response which is the most similar to the context in terms of word frequency is the most probable response. Therefore, they computed cosine similarity between the context and each of the response vectors. The response with the highest similarity score is chosen as the best response.

The first fully end-to-end single-turn dialogue system was the dual encoder (Lowe et al., 2015b). First, the context and the candidate response are represented using word embeddings and are fed into an LSTM (Hochreiter and Schmidhuber, 1997) network. As its name indicates, the model is composed of two encoders which fulfill the same function of the encoder in the encoder-decoder model. They encode the context and the response into two fixed-size vectors that are multiplied with a matrix of learned parameters to produce the matching score of the response. Some variants of the dual encoder based on Convolutional Neural Networks (CNNs) (LeCun et al., 1998) and Bidirectional LSTMs were also explored by Kadlec et al. (2015).

Tan et al. (2015) built a similar framework called Attentive-LSTM for question answering. They used a Bidirectional LSTM (BiLSTM) network to encode the question and the answer combined with attention mechanism (Bahdanau et al., 2015). The attention mechanism allows the dual encoder to alleviate the problem of the LSTM when encoding long sequences which is the case of the context. The attention weights allow the model to give

more weights to certain words and thus a better matching between the question and the correct answer can be achieved. (Wan et al., 2016) proposed another semantic matching framework based on a quite similar approach called MV-LSTM. It allows matching two texts based on the positional sentence representations. A BiLSTM is used as an encoder which contains positional information of the inputs at each of its hidden states. By matching the positional information of the inputs by cosine, bilinear or tensor layer, an interaction score is obtained. Finally, and by aggregating all the interaction scores through k-Max pooling and a multi-layer perceptron, the final matching score of the context and the response is obtained.

In the same line of reasoning, Wang and Jiang (2016) proposed Match-LSTM for the natural language inference task in which the problem consists of determining whether we can derive a hypothesis H from a premise sentence P. Here, the premise and the hypothesis could be the context and the response. The difference with the dual encoder (Lowe et al., 2015b) is that, while sequentially processing the hypothesis word by word, each word is matched with an attention-weighted representation of the premise. This results in a cross-attention matching of the context and the response. More recently, Chen and Wang (2019a,b) proposed an Enhanced Sequential Inference Model (ESIM) which was originally developed for natural language inference (Chen et al., 2018). They start by encoding the context and the response with a BiLSTM encoder following the same process as Lowe et al. (2015b). Then, cross attention mechanism is applied to model the semantic relation between the context and the response. Afterwards, max and mean pooling are applied and the output is transformed into a probability that the response is the next utterance of the given context using a multi-layer perceptron classifier.

On the contrary of the previous systems which consider only the last utterance of the context and can match only short sequences, these single-turn matching systems can match longer sequences. By concatenating the context utterances as a single long utterance, they match the context with the response only one time which makes them quick and robust. Also, they enjoy the advantages of their simple architectures which are based on the dual encoder most of the time.

2.3.2.2 Multi-turn matching models

As illustrated in Figure 2.4b, systems of this category match the candidate response with every utterance of the context. Then, an aggregation mechanism is applied to combine the different matching scores and produce a response score. Yan et al. (2016) proposed a Deep Learning to Respond (DL2R) framework for open-domain dialogue systems. Their system is based on contextually query reformulation in which the last utterance of the context (called query) is concatenated with the previous utterances to formulate multiple reformulated queries. These reformulated queries, the original query, the response, and its antecedent post are encoded via a BiLSTM encoder followed by a convolution and a max-pooling layers. Then, the encoded features are matched with each other and fed into a Feed-Forward Neural Network (FFNN) to compute the final matching score of the response

and the context. Zhou et al. (2016) exploited for the first time the word level similarity between the context and the response in their Multi-view response retrieval system. The particularity of this model is that two similarity levels between the candidate response and the context are computed and the model is trained to minimize two losses. The word and sequence level similarities are computed by matching the word embeddings and the sequence vectors. The disagreement loss and the likelihood loss are computed between the prediction of the system and what the system was supposed to predict.

Later on, Wu et al. (2017) further improved the leveraging of utterances relationship and contextual information through the Sequential Matching Network (SMN). They not only match the context utterances with the response one by one but also they are matched on multiple levels of similarity. They start by encoding separately the last 10 utterances of the context in addition to the response with a shared Gated Recurrent Unit (GRU) (Chung et al., 2014) encoder. The hidden states of the GRU form a matrix that represents the sequential information of each input. Moreover, a word similarity matrix is computed as a dot product between the matrices of each utterance and the response. These two matrices are used as input channels of a Convolutional Neural Network (CNN) followed by a max-pooling that computes a two-level matching vectors between the response and each context turn. A second GRU network aggregates the obtained vectors and produces a response ranking score. This sequential matching network constitutes a solid base for later works.

More recently, Zhou et al. (2018) extended the SMN (Wu et al., 2017) through the the Deep Attention Matching Network (DAM). The DAM addresses the limitations of recurrent neural networks in capturing long-term and multi-grained semantic representations. This model is based entirely on the attention mechanism (Bahdanau et al., 2015). It is inspired by the Transformer (Vaswani et al., 2017) to rank the response using self- and cross-attention. The first GRU encoder of the SMN model is replaced by five hierarchically stacked layers of self-attention. Five matrices of multi-grained representations of the context turns and the response are obtained instead of one matrix in the case of SMN. Following the same process as the SMN, the response matrices are matched with the context turns matrices and stacked together in the form of a 3D image (matrix). This image contains self- and cross-attention information of the inputs. Later, a succession of convolution and max-pooling are applied to the image to produce the response score.

Afterward, Yang et al. (2018) proposed the Deep Matching Network (DMN) to extend the SMN [3] furthermore. The extension consists of the inclusion of external knowledge in two different ways. The first approach is based on the Pseudo-Relevance Feedback (Cao et al., 2008) named DMN-PRF and consists of extending the candidate response with relevant words extracted from the external knowledge (Question Answering (QA) data). The second approach incorporates external knowledge with QA correspondence Knowledge Distillation named DMN-KD. It adds a third input channel to the CNN of the SMN as a matrix of the Positive Pointwise Mutual Information (PPMI) between words of the response and the most relevant responses retrieved from the external knowledge. The Deep Utterance Aggregation (DUA) system (Zhang et al., 2018c) also extends the SMN with an

3. Sequential Matching Network (SMN) is called Deep Matching Network (DMN) in their paper.

explicit weighting of the context utterances. The authors hypothesize that the last utterance of the context is the most relevant and thus, they concatenate its encoded representation with all the previous utterances in addition to the candidate response. After that, a gated self-matching attention (Wang et al., 2017) is applied to remove redundant information from the obtained representation before feeding them into the CNN as in the SMN (Wu et al., 2017).

These multi-turn matching dialogue systems assume that the response replies to each of the utterances of the context. Compared to single-turn matching systems, they deploy complex mechanisms of matching and aggregation which slower the training process as more parameters need to be optimized and may constrain the adaptability of the model to multiple domains. We believe that every researcher has to ask himself/herself a question before extending an existing approach or building a new system: "What is the cost of this architecture in terms of training resources and duration? Are we able to achieve this performance while using fewer layers and fewer parameters?". Because today, we believe that researchers tend to combine different neural networks with attention and other enhancement tools without caring about the generated costs. Strubell et al. (2019) in her recent work, quantified the computational and environmental cost of training deep neural network models for NLP, and showed that the authors should report training time and computational resources required in addition to the performance of their systems. This will allow a fair comparison of different approaches as well as a preference for systems with fewer parameters and that require fewer resources for ecological reasons. Mainly, for all these reasons, we opted for single-turn matching systems as the architecture of our proposed systems. We give more details in the next chapters (4 and 5) and we show that we can do, with a simpler approach, as good as and sometimes better than complex systems.

2.3.3 Hybrid dialogue systems

Generative and retrieval-based dialogue systems have advantages and disadvantages. For instance, generative dialogue systems can produce from scratch fluent but sometimes meaningless responses. On the other hand, retrieval-based dialogue systems can produce precise and syntactically correct responses but are limited by a fixed list of candidate responses (Chen et al., 2017). Table 2.1 summarizes the characteristics of retrieval-based and generative dialogue systems. Although these two categories of dialogue systems have been studied independently in the literature, there is pioneer work on combining generative and retrieval-based dialogue systems.

Qiu et al. (2017) built a hybrid [4] system in which, for a given question (context), similar Question-Answer (QA) pairs are retrieved from a QA base using a retrieval system. Then, a ranker system computes a score for each retrieved answer A based on its related question Q. Based on these scores, responses are ranked and the response with the highest score determines whether a new response is generated. If its score is higher than a threshold, this best response is returned. Otherwise, an *attentive sequence-to-sequence* is used to generate a

4. Sometimes called ensemble systems.

Category	Pros	Cons
Retrieval	Literal human utterances; various expressions with great diversity	not tailored to queries; bottleneck is the size of the repository
Generative	tailored for queries; highly coherent	insufficient information; universal sentences

Table 2.1 – Characteristics of retrieval-based and generative dialogue systems.

new response. Serban et al. (2017b) regrouped 22 response models in their participating system to the Alexa Prize [5] including retrieval-based neural networks, generation-based neural networks, knowledge-base question answering systems and template-based systems. Liu et al. (2017) also participated in the Alexa Prize with an ensemble of rule-based, retrieval-based and generative methods.

Later on, Song et al. (2018) built a system that first retrieves candidate responses using the same process of Qiu et al. (2017). Then, the query in addition to the retrieved responses are given as input to a generative system to produce a new response. Finally, the retrieved and the generated responses are ranked and the best response is returned. Similar to the work of Qiu et al. (2017), a retrieve and refine model was proposed by Weston et al. (2018). First, it retrieves the best response and provides it, concatenated with the context, to an attentive sequence-to-sequence model to generate a new response.

The most recent work of Pandey et al. (2018) consists of an *exemplar encoder-decoder* which first constructs a list of k-exemplar context-response pairs that are the most similar to the given context and response. Then, each exemplar context and response are encoded in addition to the original context. The exemplar responses vectors are concatenated with the original context vector and are fed into the decoder to generate a response. The score of the generated response is conditioned by the similarity between the exemplar contexts and the original context. Following the same process of retrieving then generating responses, Guu et al. (2018) proposed the *prototype-then-edit* model. It generates a sentence by sampling a random example called prototype from the training set and then editing it using a randomly sampled edit vector by attending to the prototype while conditioning on the edit vector.

These hybrid systems can alleviate the limitations of generative and retrieval-based dialogue systems by combining both systems. Even if the achieved performance is interesting, we believe that the resulting hybrid systems are quite complex as at least two modules are required (even if the architecture is still end-to-end): a response retrieval and generation modules. Moreover, almost all the hybrid systems in the literature start by retrieving one or more responses from which they generate the final response. However, we believe that the inverse process i.e generating a possible response and then retrieving the most similar response in the database could be explored. It can be seen as a way of correcting the errors that may occur in the generated response.

5. https://developer.amazon.com/alexaprize

2.3.4 Synthesis

As we have seen previously, the first generative based dialogue systems were composed of multiple modules working in pipeline. These systems are quite simple to develop, are straightforward and do not require very large amounts of data and multiple resources for training. However, the different modules are optimized independently and may suffer from error propagation from one module to another (Tiancheng, 2019). Also, modular systems struggle when generalizing to new or more complex domains because of the hand-crafted features and intermediate representations as new features have to be designed and each module has to be re-optimized.

To alleviate these issues, end-to-end systems jointly optimize the modules to generate a response to the given context of conversation. Most of the end-to-end generative dialogue systems are based on the encoder-decoder model which has shown its efficiency in generating human-like responses word by word. Multiple improvements have been brought to the encoder-decoder model to solve the problem of generating general and dull responses. On the other hand, retrieval-based dialogue systems have a list of candidate responses that is matched one by one with the context and then choose the response with the best matching score. Some of the retrieval-based dialogue systems concatenate the context utterances and match them with the response. Others match the response with every utterance of the context and then aggregate the matching scores. Single-turn matching systems are quite simpler and have fewer parameters to optimize than multi-turn matching systems as less matching and aggregation is required. Overall, retrieval-based dialogue systems can produce more consistent, specific and syntactically correct responses than generative systems. We guide our research towards single-turn retrieval-based dialogue systems for their adaptability to task-oriented domains, their efficiency and to the simplicity of their architectures compared to other categories.

2.4 Incorporating external knowledge

Incorporating external knowledge can be of great benefit to dialogue systems especially when insufficient training data is available. Task-oriented dialogue systems may rely on more than training dialogues. Bus timetable and movie showtimes might be necessary for the dialogue system to provide a precise answer to the user. Also, open-domain dialogue systems may require external knowledge such as current news articles or movie reviews to be able to converse in multiple real-world events (Serban et al., 2018). External knowledge can be structured or not.

Structured external knowledge is usually in form of relational databases or structured ontologies such as bus timetables in the Let's Go! dialogue system (Raux et al., 2005) and movie showtimes from different cinemas used in the system of Nöth et al. (2004) to help users find movie information. Other structured external knowledge include general natural language processing databases and tools (Serban et al., 2018). It includes databases

of lexical relations between words such as WordNet (Miller, 1995), databases of lexical relations between verbs such as VerbNet (Schuler, 2006), word senses databases such as FrameNet (Ruppenhofer et al., 2006), semantic networks such as ConceptNet (Speer and Havasi, 2013), SenticNet (Cambria and Hussain, 2015), etc. Further structured external knowledge sources include several tools of natural language processing such as part-of-speech (POS) taggers (Brill, 1992), named entity recognition models (Nadeau and Sekine, 2007), coreference resolution models (Harabagiu et al., 2001), sentiment analysis models (Pang and Lee, 2008), word embedding models (Mikolov et al., 2013), etc.

Unstructured external knowledge can be provided by large sources of information such as online encyclopedias like Wikipedia (Fuhr et al., 2007) and the ubuntu man pages (Lowe et al., 2015a). Unstructured external knowledge enjoys the advantages of their availability with large quantities in the Web unlike structured knowledge which requires the knowledge to be structured as databases, ontologies or even separate tools. The recent research on dialogue systems has been interested more and more in including unstructured external knowledge by looking for additional information on the Web or relevant data sources to enrich the context. Lowe et al. (2015a) proposed a task-oriented retrieval-based dialogue system that selects relevant knowledge from ubuntu man pages and then use it to find the context-response pairs that match the retrieved knowledge. Following the same process, Young et al. (2018) incorporate ConceptNet (Speer and Havasi, 2013) as a commonsense knowledge into an open domain retrieval-based dialogue system. Parthasarathi and Pineau (2018) extended the encoder-decoder model with an additional knowledge vector extracted from Wikipedia summary [6] (Scheepers, 2017) or NELL KB (Carlson et al., 2010). Lian et al. (2019) extracted knowledge from the context and the response during the training and inference steps of the encoder-decoder. Similarly, Ghazvininejad et al. (2018) generalize the encoder-decoder model by conditioning responses on both conversation history and external facts extracted from Twitter (conversational data) and Foursquare (non-conversational data). Retrieval-based dialogue systems usually match the response with the context while ignoring the external knowledge beyond the dialogue utterances. Yang et al. (2018) argue that in information-seeking conversations there may be not enough signals in the current dialogue context and the candidate responses to discriminate a correct response from a wrong one due to the wide range of topics for user information needs. Thus, they propose two different approaches of incorporating external knowledge via Pseudo-Relevance Feedback (Cao et al., 2008) or via QA correspondence knowledge distillation. Long et al. (2017) also extend the encoder-decoder with external knowledge from the Web by extracting keywords from the conversation and searching for relevant Web pages using a search engine.

Incorporating external knowledge in dialogue systems is of great importance to help the system leveraging extra information that is not found in the context utterances or the response. However, we believe that extracting this knowledge from external resources such as databases or a search engine begets further effort and make the system depending on other parameters such as the performance of the search engine, the keyword extractor, etc. and may in certain cases be not compatible with the end-to-end architecture. Nevertheless, we think that extracting this external knowledge from conversational data during the trainig

6. https://thijs.ai/Wikipedia-Summary-Dataset/

or inference is more efficient as it is fully data-driven and can be easily adapted to multiple domains as no external module or data is required. In all cases, as we are interested in this thesis in building simple and domain-independent systems and for the sake of comparability of our systems to the state-of-the-art systems, we do not incorporate external knowledge into our system and we consider only non-knowledge-grounded systems to compare our results.

2.5 Machine learning foundation

In the following, we present some machine learning concepts that are required in the course of this thesis. We start by presenting the encoder-decoder model as we are interested in the encoding part in the next chapters. Then, we present the attention mechanism and its multiple forms as we use it to enhance our model in Chapter 4.

2.5.1 Encoder-decoder

As introduced previously, the encoder-decoder model generates an output sequence word by word from an input sequence. From a general point of view, the encoder embeds the input into a vector and the decoder generates the output from this vector. More specifically, the input is first embedded which means that each word of the input sequence is represented with a vector of word embeddings (Mikolov et al., 2013). The encoder is a recurrent neural network that processes the word vectors from left to right one by one. Let $\bar{w}_i$ and w_j be the i^{th} and the j^{th} embedding vectors of the source and the target sequences respectively, RNN^e and RNN^d be the encoder and decoder RNNs respectively and $\bar{h}_i$ and h_j be the i^{th} and the j^{th} hidden states of the encoder and decoder respectively. The input sequence is encoded by recursively applying:

$$\begin{aligned} \bar{h}_0 &= 0 \\ \bar{h}_i &= \text{RNN}^e(\bar{w}_i, \bar{h}_{i-1}) \end{aligned} \tag{2.2}$$

When the encoding process ends, the last hidden state of the encoder RNN^e $\bar{h}_S$ is a vector of fixed size which represents the input sequence. Then, the decoder's initial state h_0 is initialized to be $\bar{h}_S$. The words of the output are predicted as follows:

$$\begin{aligned} o_j &= \text{softmax}(W h_j + b) \\ h_j &= \text{RNN}^d(w_j, h_{j-1}) \end{aligned} \tag{2.3}$$

Where W and b are parameters and o_j denotes the decoder's (RNN^d) probability of every word of the vocabulary at each time step j. When the vocabulary size is very large

(which is usually the case), trying all the possible combinations of the vocabulary is very time consuming and may reduce the system's performance. Beam search allows the decoder to select a certain number K of candidate translations to explore (Freitag and Al-Onaizan, 2017). Large values of K generate better target sentences but decrease decoding speed (Tixier, 2018). Even if the encoder-decoder model has initially been proposed to generate texts from textual inputs and outputs, it has been successfully utilized in other domains. Convolutional Neural Networks (CNNs) (LeCun et al., 1998) have been used to generate text from images (known as image captioning) (Vinyals et al., 2015) and also used to generate images from text (known as image synthesis) (Reed et al., 2016).

2.5.2 Attention mechanism

As explained previously, the input in the encoder-decoder models is first encoded into a fixed-length vector from which the decoder infers the output sequence. As the input sequences become longer, this vector becomes a bottleneck in improving the performance of the model and may lead to information loss or wrong translation (Bahdanau et al., 2015). To tackle this problem, the attention mechanism was first developed for Neural Machine Translation (NMT) (Cho et al., 2014; Sutskever et al., 2014) by Bahdanau et al. (2015) and has been quickly adapted to multiple domains such as image captioning (Xu et al., 2015b), text summarization (Rush et al., 2015), keyphrase extraction (Meng et al., 2017), speech recognition (Chorowski et al., 2015), question answering (He and Golub, 2016; Li et al., 2017a), etc. The attention mechanism aims to help recurrent neural networks in encoding long sequences while focusing on the most relevant parts of the input sequences.

More precisely, the encoder maps the input sequence into a fixed-length vector which is the last hidden state of the encoder $\bar{h}_S$. However, since the vector is of a fixed size, information from the input is lost. The attention mechanism solves this issue by allowing the decoder to consider all the hidden state of the encoder $\bar{h}_1, \bar{h}_2, \ldots, \bar{h}_S$ which represent the information at each time step. Multiple forms of attention exist, we enumerate them in the next sections. The decoder may "*pay attention*" to all these hidden states in the case of *global attention* or to only some of them in the case of *local attention* (Luong et al., 2015). Figure 2.5 describes the global (left) and local (right) attention approaches.

2.5.2.1 Global attention

In global attention, the context vector c_t is computed as the weighted sum of **all** the hidden states called annotations $\bar{h}_i$ of the encoder as in Equation 2.4. Where α_t is the alignment score between the current target annotation h_t and each of the source annotations $\bar{h}_i$. In other words, α_t is a probability distribution over all the source annotations and indicates which words in the source sequence are more likely to help in predicting the next target word (Luong et al., 2015; Tixier, 2018). score() could be any alignment function between the source and the target annotation. Luong et al. (2015) experimented three alternatives: the dot product $\text{score}(h_t, \bar{h}_i) = h_t^\top \bar{h}_i$, a general function with a ma-

trix of parameters W_α (bi-linear) $\text{score}(h_t, \bar{h}_i) = h_t^\top W_\alpha \bar{h}_i$ and a fully connected layer $\text{score}(h_t, \bar{h}_i) = v_a^\top \tanh\left(W_a[h_t; \bar{h}_i]\right)$. They found that the dot product works better for global attention while the general function works better in the case of local attention.

$$\begin{aligned} c_t &= \sum_{i=1}^{S} \alpha_{t,i} \bar{h}_i \\ \alpha_{t,i} &= \frac{\exp\left(\text{score}(h_t, \bar{h}_i)\right)}{\sum_{i=1}^{S} \exp\left(\text{score}(h_t, \bar{h}_i)\right)} \end{aligned} \tag{2.4}$$

2.5.2.2 Local attention

Attending on all the input words in the case of global attention may be expensive and may render the mapping of long sequences such as paragraphs and documents impractical (Luong et al., 2015). Local attention addresses this drawback by allowing the decoder to focus on a small window of annotations of a fixed size $2D + 1$ where D is a parameter fixed by the user. The context vector c_t and the alignment weights α_t are computed as in Equation 2.5.

$$\begin{aligned} c_t &= \sum_{i=p_t-D}^{p_t+D} \alpha_{t,i} \bar{h}_i \\ p_t &= T \cdot \sigma(v_p^\top \tanh(W_p h_t)) \\ \alpha_{t,i} &= \frac{\exp\left(\text{score}(h_t, \bar{h}_i)\right)}{\sum_{i'=p_t-D}^{p_t+D} \exp\left(\text{score}(h_t, \bar{h}_{i'})\right)} \exp\left(-\frac{(i-p_t)^2}{2(D/2)^2}\right) \end{aligned} \tag{2.5}$$

p_t is the position where to center the window, it is set to t in the case of a *monotonic* alignment between the source and target sequences or predicted in the case of *predictive* alignment as in Equation 2.5. T is the length of the source sequence, v_p and W_p are trainable parameters and σ is the sigmoid function that allows $p_t \in [0, T]$. The alignment weights are computed similarly to the global attention (Equation 2.4), with the addition of a normal distribution term centered on p_t and with standard deviation $D/2$ to favor alignment points near p_t (Luong et al., 2015; Tixier, 2018).

2.5.2.3 Soft attention

The definition of soft attention is similar to global attention in which the weights are *"softly"* distributed over all the parts of the input sequence (Xu et al., 2015a). The model is differentiable and parameters could be learned with backpropagation. However, the training becomes expensive as the input sequences become larger.

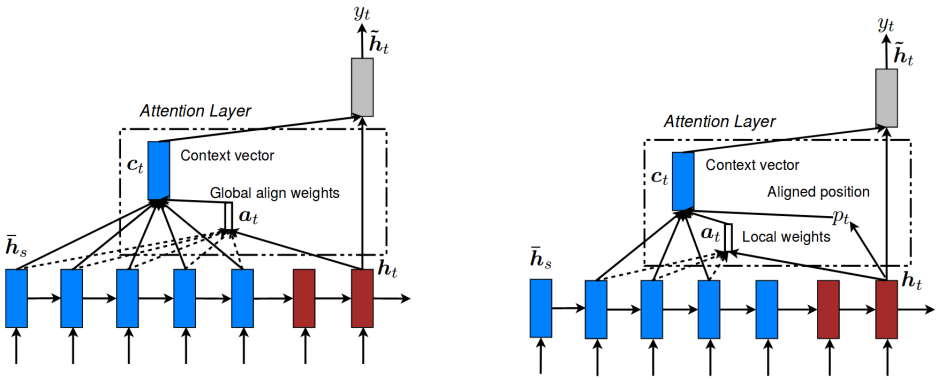


Figure 2.5 – Global (left) vs. local (right) attention (Luong et al., 2015).

2.5.2.4 Hard attention

In hard attention, each part of the input sequence is either used to compute the context vector or discarded (Xu et al., 2015a). Even if the hard attention involves less computation at the inference time, the model is non-differentiable which makes training the model end-to-end with backpropagation complicated and require some techniques such as variance reduction of reinforcement learning (Luong et al., 2015).

2.5.2.5 Self-attention

Unlike the other forms of attention, self-attention is applied to the encoder (Xu et al., 2015a). It is also known as intra-attention and it helps the encoder to look at other words in the input sequences as it encodes the current word. Considering the following input sequence that we want to translate *"The animal didn't cross the street because it was too tired"*. While processing the input word by word and reaching the word *"it"*, self-attention allows the encoder to associate it to *"animal"*. Considering that the encoder is an RNN, it maps the input sequence of length T to a sequence of annotations $(h_1, \ldots, h_T)$ where h_i is the hidden state of the RNN at the time step i. The attention vector s is computed with self-attention as a weighted sum of the annotations as in Equation 2.6.

$$\begin{aligned} u_t &= \tanh\left(W h_t\right) \\ \alpha_t &= \frac{\exp\left(\text{score}(u_t, u)\right)}{\sum_{t'=1}^{T} \exp\left(\text{score}(u_{t'}, u)\right)} \\ s &= \sum_{t=1}^{T} \alpha_t h_t \end{aligned} \tag{2.6}$$

The principle of self-attention is to first pass the annotation h_t to a dense layer, let u_t be the output. The alignment coefficient α_t is computed by comparing u_t with a trainable

context vector u randomly initialized and normalizing with a softmax. Here, the context vector can be considered as the internal representation of the relevant word, it is a trainable variable that the encoder will adjust through backpropagation. score() could be any alignment function (Tixier, 2018).

Self-attention has been the core idea of multiple works such as document classification with the hierarchical attention (Yang et al., 2016) in which self-attention is applied on the word-level and the sentence-level to encode a given document. It has been successfully applied to multiple tasks such as machine reading (Cheng et al., 2016), abstractive summarization (Paulus et al., 2018), textual entailment (Lin et al., 2017), learning task-independent sentence representations and language understanding (Shen et al., 2018). It is also the core idea of the multi-head self-attention of the Transformer (Vaswani et al., 2017) used for machine translation and one of the most relevant recent works on attention.

2.6 Evaluation metrics

The evaluation of dialogue systems is an open research problem for which the existing metrics are not adapted to the nature of the task (Lowe et al., 2017a). The evaluation metrics used today are borrowed from either information retrieval, machine translation and, automatic summary. For instance, BLEU (Papineni et al., 2002) and METEOR (Banerjee and Lavie, 2005) are machine translation metrics, ROUGE (Lin, 2004) is an automatic summary metric, Recall@k, Precision@k, Mean Reciprocal Rank and Mean Average Precision are information retrieval metrics. Depending on the nature of the evaluated system, we can group evaluation metrics of dialogue systems into two categories.

2.6.1 Generative metrics

The evaluation of generative dialogue systems is an open research domain (Liu et al., 2016). Some automatic metrics are based on word-overlap between the generated response and the response that the model is supposed to generate. These metrics are borrowed from machine translation or automatic summary. Other metrics are more flexible by tolerating semantically similar words instead of exact match. These metrics are based on word embeddings In the following, we present the existing metrics that have been used in evaluating generative dialogue systems (Ritter et al., 2011; Sordoni et al., 2015a; Serban et al., 2017a).

BLEU. BiLingual Evaluation Understudy measures the N-gram overlap between the machine translation output and the reference translation. It computes an N-gram precision for the whole dataset as follows (Liu et al., 2016).

$$P_n(r, \hat{r}) = \frac{\sum_k min(h(k, r)), h(k, \hat{r}_i)}{\sum_k h(k, r_i)} \tag{2.7}$$

Where r is the ground-truth response, $\hat{r}$ is the generated response, k indexes all the N-grams of length n and $h(k, r)$ is the number of n-grams k in r. To produce a score for the whole dataset BLEU-N, the modified precision scores for the segments are combined using the geometric mean multiplied by a brevity penalty to prevent very short candidates from receiving a too high score.

$$\text{BLEU-N} = b(r, \hat{r}) \exp(\sum_{n=1}^{N} \beta_n \log P_n(r, \hat{r})) \tag{2.8}$$

Where $b(\cdot)$ is the brevity penalty and β_n is a weighting. Usually, N is set to 4.

METEOR. Metric for Evaluation for Translation with Explicit Ordering evaluates the generated responses by computing an explicit alignment between the unigrams of the ground-truth response and the generated response. The alignment is a set of mappings between unigrams based on exact token matching, followed by WordNet synonyms, stemmed tokens and then paraphrases (Liu et al., 2016). METEOR is based on both Precision and Recall.

ROUGE. Recall-Oriented Understudy for Gisting Evaluation is a metric of automatic summary that is based on recall. Many types of ROUGE exist according to the feature used for calculating recall: ROUGE-N, ROUGE-L, ROUGE-W, and ROUGE-S based on n-grams, Longest Common Subsequence (LCS), weighted LCS, and skip-bigram co-occurrence statistics, respectively. For example, ROUGE-N counts the total number of N-grams of the reference summaries that are present in the candidate summary.

Other alternative metrics to these word-overlap metrics are based on word embeddings (Mikolov et al., 2013). They consist of representing the generated sentence and the ground-truth sentence with their vectors of embeddings. Then, the embeddings of the two sentences are matched to measure the similarity between the generated sentence and the ground-truth response. In the following, we present three word embeddings based metrics (Liu et al., 2016):

Greedy Matching. It was originally introduced for intelligent tutoring systems (Rus and Lintean, 2012). As its name indicates, it consists of greedily matching every token of the sequence r with every token of the sequence $\hat{r}$ based on the cosine similarity (cos_sim) of the word embeddings (e_w). The total greedy matching score GM is computed as the average of all the token scores. It is formulated as follows:

$$\begin{aligned} G(r, \hat{r}) &= \frac{\sum_{w \in r} \max_{\hat{w} \in \hat{r}} \text{cos\_sim}(e_w, e_{\hat{w}})}{|r|} \\ GM(r, \hat{r}) &= \frac{G(r, \hat{r}) + G(\hat{r}, r)}{2} \end{aligned} \tag{2.9}$$

Embedding Average. In contrast to greedy matching, this metric does not consider the tokens separately. First, it starts by computing an embedding of each sequence as the aver-

age of the vector embeddings of its tokens which is a popular approach in textual similarity (Wieting et al., 2016). The embedding average $\bar{e}_r$ of a sequence r is computed as mentioned in Equation 2.10. The embedding average score EA is computed as the cosine similarity between the embedding averages $\bar{e}_r$ and $\bar{e}_{\hat{r}}$.

$$\begin{aligned} \bar{e}_r &= \frac{\sum_{w \in r} e_w}{|\sum_{w' \in r} e_w|} \\ EA(r, \hat{r}) &= \text{cos\_sim}(\bar{e}_r, \bar{e}_{\hat{r}}) \end{aligned} \tag{2.10}$$

Vector Extrema. Another method of calculating the embeddings of a sequence based on its word embeddings is vector extrema (Forgues et al., 2014). More particularly, it consists of taking, at each embedding dimension d, the most extrema value amongst all word vectors in the sequence e_{wd}. This extrema value e_{rd} represents the embedding of the sequence r at the dimension d. Afterward, similarly to Embedding Average, cosine similarity is computed between the two sequence vectors to obtain the Vector Extrema score VE as in Equation 2.11. The $\min$ in the Equation refers to the selection of the largest negative value if it has a greater magnitude than the largest positive value (Liu et al., 2016).

$$\begin{aligned} e_{rd} &= \begin{cases} \max_{w \in r} e_{wd} & \text{if} e_{wd} > |\min_{w' \in r} e_{w'd}| \\ \min_{w \in r} e_{wd} & \text{otherwise} \end{cases} \\ VE(r, \hat{r}) &= \text{cos\_sim}(e_r, e_{\hat{r}}) \end{aligned} \tag{2.11}$$

Liu et al. (2016) performed a deep study of the correlation between the automatic metrics and human judgment on the responses generated by different generative dialogue systems. The results of their study showed that the correlation is very weak. In this scope, Lowe et al. (2017a) trained a classifier on human scores to automatically evaluate generative dialogue systems. Even if some effort has been done to propose automatic metrics to evaluate generative dialogue systems, human evaluation is still widely used.

2.6.2 Retrieval metrics

Precision and Recall are binary metrics used to evaluate systems with binary output such as binary classifiers. In the case of retrieval-based dialogue systems, the output of the system is a ranked list of responses which is similar to recommendation systems. Here we need to evaluate the capacity of the system to rank the ground-truth responses among the k best-ranked responses. In the following we list the most common retrieval metrics widely used in evaluating retrieval-based dialogue systems (Lowe et al., 2015b; Kadlec et al., 2015; Wu et al., 2017; Zhou et al., 2018; Yang et al., 2018).

Recall@k. It is the proportion of relevant responses that are in the top-k responses computed as follows:

$$\text{Recall@k} = \frac{\text{\# of recommended items that are relevant @k}}{\text{\# of items}} \tag{2.12}$$

Precision@k. It is the proportion of top-k responses that are relevant computed as follows:

$$\text{Precision@k} = \frac{\text{\# of recommended items that are relevant @k}}{\text{\# of recommended items @k}} \tag{2.13}$$

MAP. (Mean Average Precision) is the average of Average Precision (AP) over a set of queries (either arithmetically or geometrically). AP is the average of precision values at the ranks where relevant responses are found.

$$\text{MAP} = \frac{1}{\mid C \mid} \sum_{i=1}^{|C|} \frac{1}{m_i} \sum_{k=1}^{m_i} \text{Precision@k} \times rel(k) \tag{2.14}$$

Where C is the set of contexts, m_i is the total number of ground-truth responses of the context i and $rel(k)$ is an indicator function equal to 1 if the response at rank k is relevant, 0 otherwise.

MRR. (Mean Reciprocal Rank) is the average of the reciprocal of the ranks at which the ground-truth responses were ranked for all the contexts. Suppose that we have one ground-truth response, its reciprocal rank is 1 if it is ranked first, 0.5 if second, etc. The MRR is computed as follows:

$$\text{MRR} = \frac{1}{\mid C \mid} \sum_{i=1}^{|C|} \frac{1}{rank_i} \tag{2.15}$$

Where C is the set of contexts and $rank_i$ refers to the rank of the first retrieved ground-truth response for the i^{th} context.

These evaluation metrics used for retrieval-based dialogue systems simply look whether the ground-truth response appear in the ranked list, if yes, at which rank? However, they can not tolerate retrieving some semantically similar responses.

2.6.3 Discussion

Researchers are still working on developing evaluation metrics that are automated, repeatable, correlate to human judgements, are capable of differentiating among various dialogue strategies and explain which features of the dialogue system contribute to the quality (Deriu et al., 2019). As no clear goal is targeted by the conversational agents, it is not

clear how to evaluate the success of the conversation and the user satisfaction in contrary to task-oriented dialogue systems (Walker et al., 1997; Möller et al., 2006). Today, automatic metrics are widely used in evaluating dialogue systems based on word-overlap (BLEU, ROUGE, METEOR) or word embeddings (greedy matching, vector extrema and embedding average) in the case of generative dialogue systems or exact matching (Recall@k, Precision@k, MAP and MRR) in the case of retrieval-based dialogue systems. However, these metrics are supervised which means that they require a ground-truth response and a predicted response. Although, multiple ground-truth responses are possible for a given context. Moreover, all these metrics do not consider the context of the conversation while comparing the ground-truth response and the predicted response.

These automatic metrics have been proven to not correlate with human judgement as shown in Table 2.2. Liu et al. (2016), in their work, they measured the correlation between human judgement and the scores of generative metrics (BLEU, ROUGE, METEOR, greedy matching, vector extrema and embedding average) by computing the Pearson correlation, which estimates linear correlation, and Spearman correlation, which estimates any monotonic correlation. They showed that these metrics do not correlate with human judgement. Human evaluation remains the most used evaluation metric of open-domain dialogue systems but it is not very convenient for at least two reasons. First, it can be very expensive and time consuming as human annotators have to be recruited to evaluate the quality of the response generated by the dialogue system. Second, it does not allow a direct comparison between multiple systems as different annotators will evaluate the new systems.

Context of conversation
A: Hey! What are you doing here?
B: I'm just shopping.
A: What are you shopping for?
Generated response
Some new clothes.
Reference response
I want buy gift for my mom!

Table 2.2 – An example of zero BLEU score for an acceptable generated response in multi-turn dialogue system (Ghazarian et al., 2019).

Recently, many works were oriented towards developing evaluation models which learn automatically to evaluate open-domain dialogue systems. Lowe et al. (2017a) proposed ADEM: an Automatic Dialogue Evaluation Model which learns from a large dataset of human scores to evaluate a dialogue system by giving for each generated response a human-like score. They collected a dataset of human judgements (scores) of Twitter responses generated by four different sources: TF-IDF, dual encoder (Lowe et al., 2015b), HRED (Serban et al., 2016) and human-generated responses. The model is trained on triplets (context, ground-truth response, generated response) and learns to evaluate (give a score) the generated response regarding its context and the ground-truth response. More recently, Tao et al. (2018) proposed RUBER: a Referenced metric and Unreferenced metric Blended Evalua-

tion Routine, which evaluates a reply by taking into consideration both a ground-truth reply and a query (previous user-issued utterance). This blended metric is a combination of a referenced metric which measures the similarity between model-generated and reference responses using word-embeddings and an unreferenced metric which captures the relevance between the query and response. Ghazarian et al. (2019) attempted to improve RUBER by replacing the static word embeddings (word2vec) with pretrained BERT contextualized embeddings (Devlin et al., 2019). Another evaluation paradigm is adversarial evaluation (Bowman et al., 2016; Kannan and Vinyals, 2016; Bruni and Fernández, 2017; Li et al., 2017b; Cheng et al., 2019). The idea of adversarial evaluation is to train a "Turing-like" evaluator classifier to distinguish between responses generated by machines and responses generated by humans. The most successful generative dialogue system is the system that succeeds at fooling the evaluator.

These evaluation models correlate to human judgement better than automatic metrics and thus allow a reliable and more human-like evaluation. However, they require a large amount of labeled datasets as well as thorough engineering (Deriu et al., 2019). We believe that evaluation models are the future of dialogue evaluation metrics and the problem of the large data required by these models can be solved by collecting a large corpus of human evaluation once and for all to train a universal evaluation model which can be used in multiple domains.

2.7 Conclusion

In this chapter, we described the different categories and architectures of dialogue systems. Modular dialogue systems are composed of multiple components working in a pipeline to understand the human's utterances and generate an automatic response. This architecture is limited by its high dependency on the domain of application and the hard work of feature engineering. On the other hand, end-to-end architectures jointly optimizing multiple modules as one single module that processes the query, understands it and provides a response. This module is trained with a single objective function and automatically learns data representation with deep learning and thus reduces the effort.

In terms of application, two categories of dialogue systems exist. The first category is chit-chat (open-domain) systems which can talk with humans on different subjects without being constrained by a final objective. Even though chit-chat systems can discuss on different domains, the number of topics of conversations, the user reaction as well as the large set of possible responses decrease their performance. On the other hand, task-oriented dialogue systems focus on fulfilling a specific purpose and have a somehow countable set of choices and a straight scenario. For example, an automatic restaurant booking agent will always start by a greeting, then asks the user whether he/she wants to book or update an existing booking, etc. Such systems have more control over the conversation and can perform better than open-domain dialogue systems.

The choice of the category of dialogue system to deploy highly depends on the appli-

cation. Most of the open-domain dialogue systems are generative which perfectly imitate humans in replying by generating responses word by word. They can be modular or end-to-end. Most of the end-to-end generative systems are based on the encoder-decoder architecture which suffers from the shortness and the generality of its generated responses. In specific domains, retrieval-based dialogue systems were more used in the literature. They reply with a response that they find among a set of candidate responses. As we showed, both categories have pros and cons. This was the reason that pushed researchers to think about ensemble systems to take advantage of both architectures.

External knowledge is of great interest for dialogue systems as it helps them leveraging extra information that is not found in the context utterances or the response. Structured knowledge require the data to be stored in databases or ontologies whereas unstructured knowledge can be extracted from any source of information such as Wikipedia, the Web, etc. External knowledge can help systems enriching the representation of tokens and improve their relationships furthermore. However, we believe that extracting this knowledge from external resources such as databases or a search engine begets further effort and make the system depending on other parameters. Mainly for these reasons, we keep external knowledge out of the scope of this thesis.

The evaluation of dialogue systems is still an open research question. Today, human judgment is still largely used in evaluating generative dialogue systems. This metric is very subjective and limits the reproducibility of results and complicates the comparison between multiple systems as well as the scalability. Other generative metrics are borrowed from machine translation and automatic summary as well as information retrieval metrics in the case of retrieval-based dialogue systems.

3

Resources

> “ Things like chatbots, machine learning tools, natural language processing, or sentiment analysis are applications of artificial intelligence that may one day profoundly change how we think about and transact in travel and local experiences.”
>
> — Gillian Tans

3.1 Introduction

Data-driven dialogue systems automatically infer knowledge and strategies from data (Ritter et al., 2011; Lowe et al., 2017b; Serban et al., 2018). Most of the recent data-driven dialogue systems are based on supervised learning with deep learning models which is data-hungry and require large labeled data for training and evaluation. Moreover,, collecting these datasets is expensive and time-consuming and few datasets are available. Online conversations contain millions of samples and make an ideal source of data for the researchers (Serban et al., 2018). Even if corpus-based learning has been widely adopted for building dialogue systems, online, offline and reinforcement learning have been used as well (Cuayáhuitl and Dethlefs, 2012; El Asri et al., 2016; Dhingra et al., 2017; Carrara et al., 2017). The construction of dialogue datasets becomes expensive when human annotation is required to annotate part of or the whole dataset.

Serban et al. (2018) performed a large and complete study of the available corpora being used to build and evaluate dialogue systems. They categorize these corpora into three major categories according to the nature of the interlocutors as follows:

— Human-machine corpora contain conversations between humans and machines to obtain specific information. Examples of these datasets include DSTC1 (Williams et al., 2013) for bus information, DSTC2 (Henderson et al., 2014) for restaurant booking, and the Carnegie Mellon Communicator Corpus (Bennett and Rudnicky, 2002) which contains conversations about flight information, hotels and car rentals between humans and a travel booking system.

— Human-human spoken corpora in which conversations are more elaborated and contain fewer abbreviations and shorter phrases compared with written conversations. Some of these corpora are based on movies and TV-shows scripts such as the Movie Dic corpus (Banchs, 2012) and the Cornell Movie-Dialogue Corpus (Danescu-Niculescu-Mizil and Lee, 2011) or based on spoken conversations between tourists and tour guides over Skype for example as DSTC4 and DSTC5 (Kim et al., 2016, 2017).

— Human-human written corpora for which forums, micro-blogging, and chats are the main sources. Examples of these corpora include the Twitter Corpus (Ritter et al., 2010), the Reddit Domestic Abuse Corpus (Schrading et al., 2015) and the Ubuntu Dialogue Corpus (Lowe et al., 2015b).

As we study task-oriented retrieval-based dialogue systems, we restrict our study to human-human written corpora that have been used in building retrieval-based dialogue systems. On these human-human conversations, we want to train dialogue systems to produce human-like responses. For a complete and recent study of available datasets for building dialogue systems, we refer to the work of Serban et al. (2018). In this thesis, we focus on retrieval-based dialogue systems. Hence, in this chapter, we present a list of large datasets that have been used in training and evaluating retrieval-based dialogue systems. Some of these datasets will be used in the next chapters for building and evaluating our dialogue systems.

3.2 Datasets

Many publicly available datasets were used in evaluating most of the recent retrieval-based dialogue systems. Usually, these datasets are split into training, validation and test subsets. Each conversation of these datasets is composed of a history of a two-party conversation called context and one or multiple responses. Some of these responses are positive which means that they reply to the last turn of the context or negative which means that they do not answer. The available datasets can be roughly grouped into two categories. The first category regroups datasets where the negative response of each conversation has been randomly sampled from the whole dataset. The second category regroups datasets where humans manually annotated the incorrect responses as negative. In this section, we pro-

vide a non-exhaustive list of the available datasets split into two categories according to the approach of selecting the negative candidate responses.

3.2.1 Negative sampling based datasets

Negative sampling has been widely used in constructing dialogue datasets for retrieval-based dialogue systems (Lowe et al., 2015b; Wu et al., 2017; Yang et al., 2018; Zhang et al., 2018c; Kummerfeld et al., 2019). The popularity of this approach is mainly due to its ease, rapidity and the fact that humans are not required to annotate the candidate responses. Concretely, it consists of randomly sampling a negative response for a given context of conversation from the corpus. Therefore, for the same context, negative sampling can generate as much negative responses as we wish with less effort. However, when randomly selecting negative responses from the dataset, we might sample some responses that could be positive but we will label them as negative. We show in Chapter 5 that a large proportion of the errors made by our retrieval based dialogue systems were due to the negative sampling of the wrong responses in the dataset. In the following, we present a list of the largest, recent and most used negative sampling based datasets in building and evaluating retrieval-based dialogue systems.

3.2.1.1 Ubuntu Dialogue Corpus

Ubuntu chats interested a large research community due to the large amounts of available and open access data (Elsner and Charniak, 2008; Riou et al., 2015; Mehri and Carenini, 2017; Kummerfeld et al., 2019). In addition to the amounts of available data, the interactions in a chat-room are spontaneous and more similar to human natural conversations than micro-blogs and forums where the message lengths are constrained or the conversations are very domain-specific (Lowe et al., 2017b). In a chat-room, multiple users converse with each other on one or multiple topics around ubuntu. Today, three versions of this dataset are available. Their characteristics are summarized in Table 3.1.

The first version (V1) was constructed by Lowe et al. (2015b). They collected chat logs from the `#Ubuntu` channel of the Internet Relay Chat (IRC) [1] for the period 2004-2015. These conversations are multi-user on which they applied disentanglement heuristics to extract conversations between two users and to construct one of the largest public datasets named the Ubuntu Dialogue Corpus (UDC). This corpus targets a specific domain of technical assistance. Each dialogue in the corpus is a pair of context and response. The context is a set of utterances of two users and the response could be positive or negative. The positive response usually named the ground-truth response is the reply to the context of the conversation. However, the negative response is a randomly sampled utterance from the corpus. An example of this corpus is illustrated in Figure 3.1. As we can see, many technical words related to Ubuntu are present in the conversation as well as urls, paths, typos,

1. https://irclogs.ubuntu.com/

etc.

	UDC (V1)			UDC (V2)			UDC (V3)		
	Train	**Valid**	**Test**	**Train**	**Valid**	**Test**	**Train**	**Valid**	**Test**
# dialogues	1M	500K	500K	1M	19,560	18,920	100K	5K	1K
# cand. responses per context	2	10	10	2	10	10	100	100	100
Min # turns per context	1	2	1	2	2	2	3	3	3
Max # turns per context	19	19	19	18	18	18	75	53	43
Avg. # turns per context	10.13	10.11	10.11	3.95	3.79	3.84	5.49	5.59	3.84
Avg. # tokens per context	115.0	114.6	115.0	86.33	88.68	91.27	74.03	72.47	81.32
Avg. # tokens per response	21.86	21.89	21.94	17.24	19.09	19.26	62.92	62.82	63.06

Table 3.1 – Characteristics of different versions of the Ubuntu Dialogue Corpus.

The corpus contains about 1 million of multi-turn dialogues (dialogues with at least 3 turns) containing 7 million of utterances and 100 million words. It is divided into three subsets: train, validation, and testing with 1 million, 500 thousand and 500 thousand dialogues respectively. In the training set, each context has one positive and one negative response. However, it has one positive and 9 negative responses in the test and validation sets. Afterwards, Lowe et al. (2015b) proposed a new version of the corpus (V2) in which they update the first version and proposed some bug fixes such as the separation of the training, validation and test data by time and the addition of special tags to differentiate between the end of an utterance *(__eou__)* and the end of a turn *(__eot__)*. Recently, Kummerfeld et al. (2019) released the third version of UDC as part of the 7^{th} edition of the Dialog System Technology Challenge (DSTC7) (Yoshino et al., 2018). The particularity of this dataset is the number of candidate responses per context. As shown in Table 3.1, in each conversation, a context has 100 candidate responses among which only one is positive and 99 are negative. This dataset has been released to challenge retrieval-based dialogue systems on more realistic scenarios such as retrieving the correct response in a large pool of responses. Note also that the size of V3 is quite smaller than the previous two versions but contexts are longer.

3.2.1.2 Advising

The Advising corpus contains teacher-student English conversations collected at the University of Michigan with students playing teacher and student roles with simulated personas (Yoshino et al., 2018). It has been released recently as part of the sentence selection task of DSTC7 challenge (Gunasekara et al., 2019). The conversations in this corpus are about the courses that each student wants to take in the next semester. The teacher has a list of courses that the student has already taken and a list of suggested courses. The dataset includes additional information about student preferences and course information. A total of 815 conversations were collected and used to generate 100,000 conversations for the training and 500 for each of the validation and test sets. Similar to the Ubuntu Dialogue Corpus, the negative responses were randomly sampled from a set of 82,094 paraphrases of messages generated from the corpus.

Table 3.2 summarizes the characteristics of this dataset. The major difference between

Context

A: Could someone that is running bind9 on ubuntu explain this ... this is from a client within my network *__url__*
B: what about it are you questioning
A: if i do a nslookup through google's public dns it still resolves my internal hosts even after flushing dns
B: some routers intercept dns requests if they look like they're destined for lan hosts
A: even ubuntu running bind
B: i am using an ubuntu server box with two nics as my router which is running iptables and bind/dhcp
A: depends on its configuration i think google would reject a request for a host that's not an fqdn or something similar also if the system's running bind it's going to try using its own internal resolver first - which means that bind is going to resolve your host before it ever kicks it out to google

Candidate Responses

R1 yeah i know it isn't actually google that is resolving it ... i just wish i knew how it was doing it .. check your *__path__* conf ✓
R2 you have to be a member of a group to change a file to that group ahhhh ✗
R3 *__url__* you mean xgl ✗
R4 volume control uses the correct icon but not battery or network the program smplayer is currently not installed you can install it by typing ✗

Figure 3.1 – Example of the Ubuntu Dialogue Corpus (V1).

	Train	Valid	Test
# dialogues	100K	500	500
# cand. responses per context	100	100	100
Min # turns per context	1	1	1
Max # turns per context	41	34	36
Avg. # turns per context	9.22	9.78	9.47
Avg. # tokens per context	79.88	83.86	87.37
Avg. # tokens per response	57.83	66.13	66.60

Table 3.2 – Characteristics of the Advising corpus.

this corpus and the other corpora is the number of candidate responses per context. Here, similarly to V3 of UDC, for each context, one correct response and 99 negative responses are provided. This property makes his corpus closer to the reality in which a dialogue system has a bunch of possible responses from which it has to choose the most convenient response. Unlike other chat-based datasets such as UDC, Advising conversations are more likely to contain fewer typos, abbreviations, and emojis as we can see in Figure 3.2. Furthermore, and unlike UDC, the conversations of Advising corpus are originally two-party which reduces the chance of errors made by the disentangling heuristics in the case of multi-party conversations.

Context

Student: Hello
Advisor: Hello
Advisor: How I can help you?
Student: I'd like to talk about next semester's course scheduling
Advisor: What do you like
Student: I am most interested in software development
Advisor: You need to take EECS281
Student: Is that a hard class?
Advisor: It was a hard class.
Student: Can EECS481 and EECS 281 be taken simultaneously?

Candidate Responses

R1 First, you must take EECS281. ✓
R2 Course 370 might work well for you if you did well on course 270. ✗
R3 EECS 492 is a great class if you are interested in AI. ✗
R4 It got a ranking of 2.53 for easiness. ✗

Figure 3.2 – Example of the Advising Corpus.

3.2.1.3 MSDialog

The MSDialog dataset (Yang et al., 2018) was extracted from the Microsoft Answer Community [2] and consists of technical support conversations in English. The dataset was originally constructed for question answering and then was completed (MSDialog-ResponseRank) to allow building retrieval-based dialogue systems. In the conversations of the Microsoft Answer Community, typically a user starts a thread by asking a question and then experienced users (agents) provide answers. The users may exchange with the agents to ask for clarification or even to give feedback. Out of 35,000 dialogues, only the dialogues with a number of turns in the range of [3,99] were kept and distributed on the training, validation and test sets. The user utterances are considered as the context and the true response [3] by the agent is considered as the correct response. To generate negative responses, the following process was used:

— The ground-truth response is used as a query to retrieve 1,000 responses from the agent responses using BM25 (Robertson and Jones, 1976).
— 9 responses are randomly sampled from these selected responses and considered as negative.

In addition to the dialogues, the dataset offers metadata including question popularity, answer vote, dialogue title and time in addition to the user affiliation. This metadata is

2. https://answers.microsoft.com
3. Each utterance has a field *is_answer* that indicates whether the utterance is selected as the best answer by the community.

rich in information and can help the dialogue system to better match the context with the correct responses. Table 3.3 summarizes characteristics of the dataset[4]. Each context in the MSDialog dataset is composed of between 2 and 11 turns and as in UDC and has 10 candidate responses (1 positive and 9 negative). Similarly to the Advising Corpus, the conversations of this dataset are more likely to contain less typos, abbreviations and emojis (Example 3.3).

	Train	Valid	Test
# dialogues	173k	37k	35k
# cand. responses per context	10	10	10
Min # turns per context	2	2	2
Max # turns per context	11	11	11
Avg. # turns per context	5.0	4.9	4.4
Avg. # tokens per context	271	263	227
Avg. # tokens per response	66.7	67.6	66.8

Table 3.3 – Characteristics of the MSDialog corpus.

3.2.1.4 E-commerce Dialogue Corpus

The **E-commerce Dialogue Corpus** (EDC) (Zhang et al., 2018c) is a public dataset that contains Chinese conversations between customers and customer service staff extracted from Taobao[5] the largest e-commerce platform in China. It contains over 5 types of conversations:. commodity consultation, logistics express, recommendation, negotiation, and chitchat based on over 20 commodities. Table 3.4 summarizes the characteristics of the EDC dataset. It contains 1 million dialogues in the training set and 10 thousand in each of the validation and test sets. Conversations in this dataset have at least one dialogue turn. The example in Figure 3.4 is extracted from the EDC corpus and translated from Chinese to English.

	Train	Valid	Test
# dialogues	1M	10k	10k
# cand. responses per context	2	10	10
Min # turns per context	1	1	1
Max # turns per context	119	111	49
Avg. # turns per context	5.51	5.48	5.64
Avg. # tokens per context	38.7	38.31	40.1
Avg. # tokens per response	7.87	7.81	10.67

Table 3.4 – Characteristics of the E-commerce Dialogue Corpus.

Negative sampling based datasets are many and this is mainly due to the ease of constructing them compared to human-labeled datasets that require manual effort. They usually have one correct response per context and multiple negative responses that are randomly

4. We refer to the website https://ciir.cs.umass.edu/downloads/msdialog/ on which the dataset is published for more details.

5. https://www.taobao.com/

Context

User: I have paid for the microsoft office multiple times this week and it is your fault it won't upload. It says it uploads and then cant be used. [Original Title: Why is microsoft such an awful company?]
Agent: Hi PERSON_PLACEHOLDER, We'd like to know more information so we can provide you the best solution. Please answer these questions: We'll keep an eye out for your response.
User: I am trying to upload Office 365 Personal. I don't get an error code. It says upload complete but when I go to use the microsoft word it says there is a problem with my account, The only thing it will allow me to do on that page is select OK and when I do it sends me over to buy microsoft office which I have done.

Candidate Responses

R1 «<AGENT»>: If you are getting a message that you need to purchase Microsoft Office again which you have already done, we recommend contacting the Accounts & billing department. Select Office or Office 365 on this link to contact the said department via chat or call. Let us know if you need anything else. We're here to help. ✓
R2 «<AGENT»>: Hi Alpal, Thank you for contacting Microsoft Community. We regret the inconvenience caused and would recommend you to contact OneDrive's billing & payment support as they are the support group who could best handle this issue: https://commerce.microsoft.com/PaymentHub/Help Let us know if you've further query and we'll be happy to assist. ✗
R3 «<AGENT»>: Outlook Mail Preview IS Outlook.com. PERSON_PLACEHOLDER don't move anything to Office 365. Your mailbox may move to the same platform that Office 365 uses, but you don't need to do anything. It's automatic. PERSON_PLACEHOLDER cannot log into the Office 365 service using an Outlook.com address. Office 365 and Outlook.com are separate services even though they can run on the same platform. ✗
R4 «<AGENT»>: Hi, thank you for posting your query in Microsoft Office Community. Before we proceed, I need more information to help you better. 1. From where did you purchase Office suite? 2. How are you trying to activate it? 3. What is the edition of Office you purchased? I look forward to your reply to assist you further. Thank you. ✗

Figure 3.3 – Example of the MSDialog Corpus.

sampled from the dataset. Some systems perform an oversampling of positive or negative samples during the training to increase the size of the training dataset (Lowe et al., 2015b; Wu et al., 2017).

3.2.2 Human-labeled datasets

Unlike the datasets of the first category, to construct the following datasets, humans are required to judge whether each candidate response is a positive or negative response. In this configuration, every context may have more than one correct response. The construction of such datasets is expensive and time-consuming as many annotators are required and the agreement inter-annotator has to be managed. For these reasons, very few datasets of this category are available in the literature. In the following, we present the available ones.

Context

A: Hello there
B: Hello there
A: I added the shopping cart a few days ago and it seems to be invalid today.
B: Honey, you can add it one more time.
A: I want 100% cotton wet-dry friendly baby wipes.
B: Good, Honey, the size of this paper is 1520.
A: Right, honey, the 1020.
B: It's smaller.
A: The 2020 pre-sale version of this one.
B: There are several models of the 2020 100% cotton
A: There is also a thicker set of 3 packs of pre-sale version. Compared to the version of May 20th which is more cost-effective?
B: It is actually the 520 pre-sale version, honey. You can pay the deposit first and pay the rest on the 20th.
A: Are there any promo?
B: Honey, you can receive coupons and that's all.
A: Is this pure cotton or non-woven fabric ?
B: Non-woven honey. The 2020 is recently put on the market, it is not the thickened, the new one.
A: Is the non-thickened more thinner ? What was the price?

Candidate Responses

R1 The non-thickened is in normal thickness, it is not thin. ✓
R2 If you want a wet wipe, this one sells better. ✗
R3 Honey the size of the 1020 one is relatively small, the previous one was 1520. ✗

Figure 3.4 – Example of the E-Commerce Dialogue Corpus.

3.2.2.1 Douban

The **Douban Conversation Corpus** (Wu et al., 2017) contains human-human Chinese dialogues extracted from Douban [6] a popular social network in China. It is an open domain public dataset where conversations concern movies, books, music, etc. in contrast to the previously described datasets which are domain-specific. Wu et al. (2017) crawled 1.1 million of two-user conversations from Douban group [7] to create the training, validation and test sets. In the training and validation sets, each context has one positive response which is the last turn of the dialogue and one negative response which was randomly selected from the corpus. In the test set, a different approach of selecting the negative response was used. We describe it in the following:

— Sina Weibo [8] the largest micro-blogging service in China was used to collect 15 million post-reply pairs.

6. https://www.douban.com/
7. https://www.douban.com/group
8. http://weibo.com/

— These pairs were indexed with Lucene [9].

— The last turn of the context was expended with the top 5 keywords based on their TF-IDF scores and extracted from its previous turns.

— 9 candidate responses were retrieved from the index based on the extended turn.

	Train	Valid	Test
# dialogues	1M	50K	10K
# cand. responses per context	2	2	10
Min # turns per context	3	3	3
Max # turns per context	98	91	45
Avg. # turns per context	6.69	6.75	6.47
Avg. # tokens per context	109.8	110.6	117.0
Avg. # tokens per response	13.37	13.35	16.29

Table 3.5 – Characteristics of the Douban Corpus.

Table 3.5 contains some characteristics of the dataset and Figure 3.5 contains an example extracted from Douban Corpus and translated from Chinese to English. In Douban, only the test set has been manually annotated and this explains its small size (only 10k samples). This allows the system to be trained on random negative samples but evaluates it on a more strict datasets where responses are manually annotated. This corpus has been widely used in building and evaluating retrieval-based dialogue systems (Wu et al., 2017; Yang et al., 2018; Wu et al., 2018b; Boussaha et al., 2019c).

Context

A: Guys who need some comfort sentences, please leave your name
B: I need to be comforted, because I am looking for a job, and I am tired of it. 15805162976 Thank you, I am Miaojiang.
A: I am also looking for work and tired.
B: All kinds of tiredness.

Candidate Responses

R1 Please don't say, I think the more you talk, the more tired you are. Let's talk some happy things. This will be spiritual. Ok, I hope the big wound on my face will get better soon. ✓
R2 Why are you tired? ✓
R3 Buy a small flour mixer, what is the three light? ✗
R4 But it is my professional faith. ✗

Figure 3.5 – Example of the Douban Conversation Corpus (translated from Chinese to English).

9. https://lucenenet.apache.org/

3.2.2.2 AliMe data

AliMe data (Yang et al., 2018) is a human-machine corpus extracted from chat logs between customers and AliMe: the Alibaba chatbot (Qiu et al., 2017) for the period 2017-10-01 to 2017-10-20. The dataset was created by concatenating three consecutive dialogue turns and using them as a query of the AliMe chatbot. The chatbot returns the top-15 most similar questions from its QA database. These questions are considered as candidate responses to the context composed of 3 dialogue turns. Analysts were asked to annotate the candidate responses and assign positive labels to responses that match the context and negative labels otherwise. As a result, 63,000 context-response pairs were collected and were split into training, validation and test sets as shown in Table 3.6. The dataset contains external knowledge composed of 510,000 clicked questions with answers from the click logs of users when similar questions are suggested by the chatbot. Unfortunately, the dataset is not public.

	Train	Valid	Test
# dialogues	51K	6K	6K
# cand. responses per context	15	15	15
Min # turns per context	2	2	2
Max # turns per context	3	3	3
Avg. # turns per context	2.4	2.1	2.2
Avg. # tokens per context	38	35	34
Avg. # tokens per response	4.9	4.7	4.6

Table 3.6 – Characteristics of the AliMe Corpus (Yang et al., 2018).

As we can see, today very few human-labeled datasets are available and this is mainly due to the difficulty and the high costs of recruiting human annotators and solve the inter-annotator agreement. However, we believe that even though these high costs, it is worth it as it allows having clean and less noisy data on which we train and evaluate dialogue systems.

	UDC (V1)			UDC (V2)			Douban			MSDialog			EDC			UDC (V3)			Advising		
	Train	Valid	Test	Train	Valid	Test	Train	Valid	Test	Train	Valid	Test	Train	Valid	Test	Train	Valid	Test	Train	Valid	Test
Language	English			English			Chinese			English			Chinese			English			English		
# dialogues	1M	500K	500K	1M	19,560	18,920	1M	50K	10K	173k	37k	35k	1M	10k	10k	100K	5K	1K	100K	500	500
# cand. R per C	2	10	10	2	10	10	2	2	10	10	10	10	2	10	10	10	10	10	100	100	100
Min # turns per C	1	2	1	2	2	2	3	3	3	2	2	2	1	1	1	3	3	3	1	1	1
Max # turns per C	19	19	19	18	18	18	98	91	45	11	11	11	119	111	49	75	53	43	41	34	36
Avg. # turns per C	10.13	10.11	3.95	3.79	3.84	10.11	6.69	6.75	6.47	5.0	4.9	4.4	5.51	5.48	5.64	5.49	5.59	3.84	9.22	9.78	9.47
Avg. # tokens per C	115.0	114.6	115.0	86.33	88.68	91.27	109.8	110.6	117.0	271	263	227	38.7	38.31	40.1	74.03	72.47	81.32	79.88	83.86	87.37
Avg. # tokens per R	21.86	21.89	21.94	17.24	19.09	19.26	13.37	13.35	16.29	66.7	67.6	66.8	7.87	7.81	10.67	62.92	62.82	63.06	57.83	66.13	66.60

Table 3.7 – Statistics on the datasets. *C*, *R* and *cand.* denote context, response and candidate respectively.

3.3 Conclusion

We presented in this chapter, the available datasets used in building and evaluating retrieval-based dialogue systems. We restricted our study to large datasets for two reasons. First, because the latest systems are based on deep learning which is known to be a data hungry approach and second because, recently, almost all the state-of-the-art systems are built, evaluated and compared based on these datasets. The datasets that we studied are in two languages, English and Chinese and vary from open domain to domain specific. We recall in Table 3.7 their characteristics. The Ubuntu Dialogue Corpus is the largest available dataset, it is closed domain but since it is issued from chat channels, the utterances are short, noisy and contain multiple typos as shown in the example of Figure 3.1. Negative responses in UDC were automatically selected following the negative sampling approach. Advising and EDC were constructed following the same process. However, in MSDialog, the most similar responses to the ground-truth response were first filtered and then the negative responses were randomly selected from these responses. On the other hand, Douban is the largest human-labeled and public dataset. Humans were recruited to annotate the negative responses. However, only the test set was annotated.

The cost and the human efforts required to manually annotate the datasets are the reasons why few human-labeled datasets are available today. Even, if some of them exist, they are either small or private. However, we believe that randomly selecting responses from the corpus and annotating them as negative may introduce bias in the dataset because some of the randomly selected responses can reply to the question. A recent work studied the impact of negative sampling on the quality of retrieval-based dialogue systems (Nugmanova et al., 2019). The authors studied the performance of three variants of the dual encoder (Lowe et al., 2015b) and showed that by simply selecting the negative responses from a uniform distribution instead of randomly sampling them, the studied systems achieved better performance. The models were evaluated on a Russian dataset of human-human conversation extracted from chat logs. This result encourages researchers to investigate more elaborated techniques of negative response sampling in dialogue datasets.

Recently, some research was oriented towards developing new mechanisms to enhance the quality of response in dialogue datasets. Shang et al. (2018) argue that the conversations crawled from the Internet contain many noises such as the fact that some responses are completely irrelevant to the context of the conversations but they are marked as positive responses. Another problem which is more related to generative dialogue systems is the existence of many general responses such as *"I don't know"* and *"Ok"* which are universal i.e suitable for multiple contexts but are non-informative and uninteresting. Having this kind of responses (noisy data) in the training datasets will alter the learning process and affect the performance of the trained model. They proposed a calibration mechanism that they incorporate into generative dialogue systems to allow high quality data to have more influences on the generative dialogue system and to reduce the effect of noisy data.

In the same line of research and to enhance the quality of the dialogue datasets, Zhufeng et al. (2019) address the problem of incomplete utterances which do not take the form of

a sentence. Their observation is based on the study of Carbonell (1983) in which they showed that users of dialogue systems tend to use succinct language which often omits entities or concepts made in previous utterances. This problem is also known as *non-sentential utterances* (Fernández et al., 2007). They address the problem by constructing a large open-domain dataset called *Restoration-200K* extracted from Douban. The conversations of this dataset are manually annotated with the explicit relation between an utterance and its context. They also propose a "pick-and-combine" model to restore the incomplete utterance from its context and show how training the same dialogue system on the dataset before and after utterance restoration results in different performances. The evaluated dialogue systems produce better responses after the restoration process.

These two recent works show that enhancing the quality of the dataset enables the dialogue system to produce responses of better quality. We believe that there is much work to be done in this direction. Also, as we can notice from the available datasets, only English and Chinese are the available languages which is very limited for the community working with very low resources. Overall, in the next chapters, we consider the Ubuntu Dialogue Corpus (V1 and V3), Advising as well as the Douban Conversation Corpus for building and evaluating our retrieval-based dialogue systems. This choice is motivated by the variety of their languages (English and Chinese), the diversity of their sources (open-domain and closed-domains) and the availability of the evaluation results of multiple state-of-the-art systems on these datasets.

4

Single-level context response matching system

> “ It seems probable that once the machine thinking method has started, it would not take long to outstrip our feeble powers... They would be able to converse with each other to sharpen their wits. At some stage, therefore, we should have to expect the machines to take control.”
>
> — Alan Turing

4.1 Introduction

In the literature, response retrieval approaches match the context with the response only one time or multiple times. For instance the dual encoder (Lowe et al., 2015b), the Attentive-LSTM (Tan et al., 2015), the Match-LSTM (Wang and Jiang, 2016) and the Enhanced Sequential Inference Model (Chen and Wang, 2019a) are single-turn matching systems i.e. the whole context and the candidate response are represented as vectors and are matched together. On the other hand, the Sequential Matching Network (Wu et al., 2017), the Deep Attention Matching Network (Zhou et al., 2018), the Deep Matching Network (Yang et al., 2018) and the Deep Utterance Aggregation System (Zhang et al., 2018c) are multi-turn matching systems which match the candidate response with each utterance of the context.

Usually, single-turn matching systems are quite simpler in terms of architecture and have a smaller number of parameters compared to multi-turn matching systems.

The dual encoder (Lowe et al., 2015b) has been a basis of multiple single-turn matching systems (Kadlec et al., 2015; Tan et al., 2015; Wan et al., 2016; Wang and Jiang, 2016) for its simplicity and its performance. We studied the dual encoder as a simple and efficient baseline for response retrieval and identified some limitations that we addressed in a new and improved architecture. In this chapter, we analyze the architecture of the existing dual encoder and present our first contribution. We compare our system to systems of the same category (single-turn matching systems) and show that while keeping a simple architecture, our system can efficiently match the context with the correct response and outperforms the baseline systems with a good margin. We perform an in-depth comparison between our system and the basic dual encoder to identify whether we efficiently addressed the drawbacks that we identified. Moreover, we present an extension of our systems with the attention mechanism (Bahdanau et al., 2015).

4.2 Approach

The dual encoder of Lowe et al. (2015b) defines the task of retrieving the correct response as a classification problem. For a given context and a candidate response, it learns to predict a binary label (1 for correct response, 0 otherwise). It consists of two recurrent neural networks called encoders which encode the context and the candidate response into a fixed-size vector and learns a matrix of parameters. The overall system learns to match the context with the response by multiplying the obtained vectors and the parameters matrix to predict the matching score of the response. However, despite the simplicity and the efficiency of this approach, we believe that it contains two major limitations that we summarize in the following:

— Their system is composed of two distinct encoders, one for the context and the second for the response. However, these two encoders may project the context and the response in two different spaces. Hence, matching two vectors from different spaces may not lead to the best performance.
— In addition to the encoder parameters, their dual encoder has an extra matrix of parameters that is multiplied with the context and the response vectors. However, we believe that this parameter matrix is unnecessary and we can discard it if we share the parameters of the two encoders.

By addressing these limitations, we propose an improved architecture of the LSTM dual encoder trained in end-to-end fashion. The idea consists of representing the context C and the response R using word embeddings. Then these word embeddings $e_1, e_2, .., e_j$ are given in chronological order to a recurrent neural network with LSTM cells called encoder. The hidden layer of this recurrent network is updated each time a word embedding is fed. The encoder aims to provide a fixed-length vector for each input text which has a variable size.

This process is described in Figure 4.1. In the end, we get the hidden layer of the encoder C' and R' which represents in this case the whole context and the response respectively.

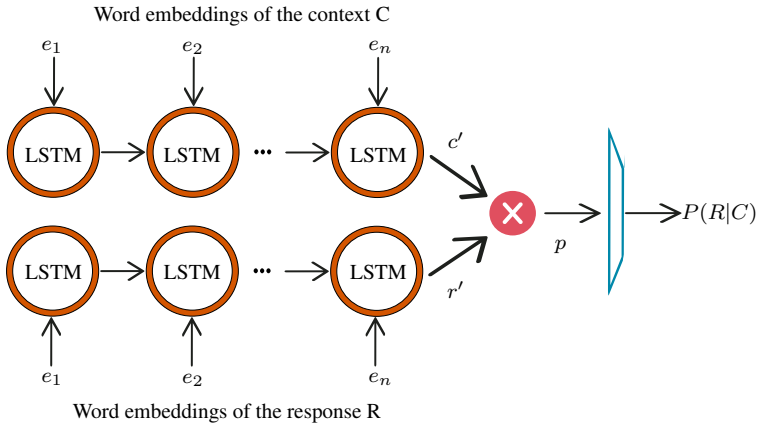


Figure 4.1 – Architecture of our single-turn single-level retrieval-based dialogue system.

In our approach, the matching score of the context and the response is computed as a cross-product p between c' and c' which reflects the similarity between the context and the candidate response on the sequence level. The similarity vector p is fed into a fully connected layer and transformed into a probability score using a sigmoid function. This architecture is motivated by the fact that the context and the response share common concepts (common words, semantics, etc). These concepts are first captured with word embeddings and then using the encoders and the similarity, we capture semantic and syntactic similarities. In the following, we elaborate on the functions of our system.

4.2.1 Sequence encoding

The first layer of our system maps each word of the input into a distributed representation $\mathbb{R}^d$ by looking up a shared embedding matrix $E \in \mathbb{R}^{|V| \times d}$ where V is the vocabulary and d is the dimension of word embeddings. We initialize the embedding matrix E using pretrained vectors (more details are given in 5.3.2). E is a parameter of our model to be learned by back-propagation. This layer produces two matrices $C = [e_{c1}, e_{c2}, ..., e_{cn}]$ and $R = [e_{r1}, e_{r2}, ..., e_{rn}]$ where $e_{ci}, e_{ri} \in \mathbb{R}^d$ are the embeddings of the i-th word of the context and the response respectively and n is a fixed sequence length. The context and response matrices $C, R \in \mathbb{R}^{d \times n}$ are then fed into a shared LSTM network word by word in order to get encoded.

Let c' and r' be the encoded vectors of C and R. They are the last hidden vectors of the encoder such as $c' = h_{c,n}$ and $r' = h_{r,n}$ where $h_{c,i}, h_{r,i} \in \mathbb{R}^m$ and m is the dimension of the hidden layer of the LSTM recurrent network. $h_{c,i}$ is obtained by Equation 4.1. $h_{r,i}$ is obtained similarly by replacing e_{ci} by e_{ri}.

$$
\begin{aligned}
z_i &= \sigma(W_z \cdot [h_{c,i-1}, e_{ci}]) \\
r_i &= \sigma(W_r \cdot [h_{c,i-1}, e_{ci}]) \\
\widetilde{h}_{c,i} &= \tanh(W \cdot [r_i * h_{c,i-1}, e_{ci}]) \\
h_{c,i} &= (1 - z_i) * h_{i-1} + z_i * \widetilde{h}_{c,i}
\end{aligned}
\tag{4.1}
$$

W_z, W_r and W are parameters, z_i and r_i are an update gate and $h_{c,0} = 0$.

4.2.2 Sequence-level similarity

We hypothesize that positive responses are semantically similar to the context. Thus, the aim of a response retrieval system is to rank the responses that share the most common semantics with the context on top of the other candidate responses. Once the input vectors are encoded, we compute a cross product p between c' and r' as follows:

$$
p = c' \wedge r' \equiv p = h_{c,n} \wedge h_{r,n} \tag{4.2}
$$

Where $\wedge$ denotes the cross product. As a result, $p \in \mathbb{R}^m$ models the similarity between C and R on the sequence-level.

4.2.3 Response score

Based on the sequence similarity vector p between the context C and the response R, we compute the score of the response. We transform the vector p into a probability using a one-layer fully-connected feed-forward neural network with a sigmoid activation (Equation 4.3). The last layer predicts the probability $P(R|C)$ of the response R being the next utterance of the context C as:

$$
P(R|C) = sigmoid(W' \cdot p + b) \tag{4.3}
$$

Where W' and b are parameters and $\oplus$ denotes concatenation. We train our model to minimize the binary cross-entropy loss.

The advantages of our system compared to the state of the art ones are: (1) we do not require any external module to provide extra information such as context and response topics or related knowledge unlike (Xu et al., 2017) and (Wu et al., 2018a); (2) the architecture is trained in end-to-end where the classification error is back-propagated through the network to improve the training process from the embedding layer to the probability prediction; (3) we designed an utterance ranking system that is domain independent. It means we can adapt this same architecture from one assistance domain to another, for example from Ubuntu to Visa and immigration assistance by simply changing the dataset.

4.3 Experimental setup

To show the efficiency of our approach, we evaluate our system and state-of-the-art single-turn systems on two datasets: the Ubuntu Dialogue Corpus (V1) and the Douban Conversation Corpus. We followed (Lowe et al., 2015b; Wu et al., 2017) in using Recall@k, Precision@1, MAP and MRR as evaluation metrics. In the following, we present the baseline systems to which we compared our system as well as the parameters of our system.

4.3.1 Baseline systems

As our approach is based on a single-turn matching of the context and the candidate response, we consider the following baseline systems of the same category described in detail in Section 2.3.2.1. These systems were considered as baselines for multiple works on retrieval-based dialogue systems (Lowe et al., 2015b; Wu et al., 2017; Zhou et al., 2018; Zhang et al., 2018c). We briefly describe them in the following:

TF-IDF We report results of the Term Frequency-Inverse Document Frequency (TF-IDF) model (Lowe et al., 2015b). The context and each of the candidate responses are represented as vectors of TF-IDF of their words. Then, a cosine similarity is computed between the context and the response vectors and used as a ranking score of the response.

LSTM dual encoder The model was introduced in the work of Lowe et al. (2015b). The context and the response were presented using their word embeddings and then they were fed word by word into an LSTM network to encode them into fixed-size vectors. Then, a response ranking score is computed using a bilinear model (Tenenbaum and Freeman, 2000).

BiLSTM dual encoder The system of Kadlec et al. (2015), very similar to the dual encoder (Lowe et al., 2015b) where the LSTM cells were replaced by Bidirectional LSTM cells in the encoder.

Attentive-LSTM Introduced by Tan et al. (2015) as an extension of the dual encoder (Lowe et al., 2015b) based on a BiLSTM encoder and the attention mechanism (Bahdanau et al., 2015).

MV-LSTM Proposed by Wan et al. (2016) based on the positional sentence representation. Each hidden state of the context encoder is matched with the corresponding hidden state of the decoder by three different methods: cosine, bilinear or tensor layer and then aggregated to produce the final score of the response.

Match-LSTM A similar work to the dual encoder which was originally proposed for natural language inference (Wang and Jiang, 2016). It is based on cross-attention i.e while sequentially processing the context word by word, each word is matched with an attention-weighted representation of the response.

System	Ubuntu Dialogue Corpus V1				Douban Conversation Corpus					
	$R_2@1$	$R_{10}@1$	$R_{10}@2$	$R_{10}@5$	$R_{10}@1$	$R_{10}@2$	$R_{10}@5$	P@1	MAP	MRR
TF-IDF (Lowe et al., 2017b)	0.659	0.410	0.545	0.708	0.096	0.172	0.405	0.180	0.331	0.359
LSTM (Lowe et al., 2017b)	0.901	0.638	0.784	0.949	0.187	0.343	0.720	0.320	0.485	0.527
BiLSTM (Kadlec et al., 2015)	0.895	0.630	0.780	0.944	0.184	0.330	0.716	0.313	0.479	0.514
Attentive-LSTM (Tan et al., 2015)	0.903	0.633	0.789	0.942	0.192	0.328	0.718	0.331	0.495	0.523
MV-LSTM (Wan et al., 2016)	0.906	0.653	0.804	0.946	0.202	0.351	0.710	0.348	0.498	0.538
Match-LSTM (Wang and Jiang, 2016)	0.904	0.653	0.799	0.944	0.202	0.348	0.720	0.345	0.500	0.537
Our system	0.917	0.685	0.825	0.957	0.209	0.357	0.702	0.358	0.500	0.543

Table 4.1 – Evaluation results on the UDC V1 and Douban Corpus using retrieval metrics.

4.3.2 System parameters

Word embeddings were initialized with Glove (Pennington et al., 2014) pre-trained on Common Crawl Corpus [1] then fine-tuned during training [2]. We used the copy of UDC (V1) shared by Xu et al. (2017) in which numbers, urls and paths were replaced by specific placeholders [3]. The System parameters were updated using Stochastic Gradient Decent (SGD) with Adam algorithm (Kingma and Ba, 2015). The model was trained on a single Titan X GPU.

Initial learning rate was set to 0.001 and Adam parameters β_1 et β_2 were set to 0.9 and 0.999 respectively. As regularization strategy we used *early-stopping* and to train the model we used mini-batch of size 256. The size of the word embeddings and the size of the hidden layer of LSTM and BiLSTM were set to 300. We limited the size of both the context and the response to 160 words. We implemented our system with Keras (Chollet et al., 2015) with Tensorflow (Abadi et al., 2015) in backend. These hyperparameters were obtained with a grid search on the development set. The code and the preprocessing scripts are available on `https://github.com/basma-b/dual_encoder_udc`.

4.4 Results analysis

We evaluated our system on two benchmark datasets, the Ubuntu Dialogue Corpus (Lowe et al., 2015b) and the Douban Conversation Corpus (Wu et al., 2017). We report in Table 4.1 the scores of the baseline systems that we obtained from the work of Wu et al. (2017). As we can see in the same Table, our system outperforms all the baseline systems of the same category (single-turn matching systems) on all the metrics and all the datasets. Compared to the original dual encoder (Lowe et al., 2015b), our improved approach of matching the context and the candidate response achieve better performance. Moreover, our system outperforms other single-turn matching systems based on the dual encoder and which utilize more complex functions such as the attention mechanism as in the Attentive-LSTM (Tan et al., 2015) and Match-LSTM (Wang and Jiang, 2016). Our system improves

1. `http://commoncrawl.org/the-data/`
2. Note that we trained word embeddings on the training set but no improvement was observed.
3. `https://www.dropbox.com/s/2fdn26rj6h9bpvl/ubuntu_data.zip`

the matching process of the context and the response by simply multiplying the context and the response vectors and transforming the result into a probability. Overall, we observe that the performance of all the systems on UDC is higher than the performance on the Douban corpus. We believe that this is mainly due to the nature of the two datasets as the Ubuntu Dialogue Corpus is closed-domain whereas Douban is an open domain dataset where the candidate responses are of multiple subjects and retrieving the most appropriate response is harder.

In addition to the numerical metrics that we reported in Table 4.1, it is important to look further into the predictions and analyze them. As Douban corpus contains conversations written in Chinese, we performed our analysis on an English dataset (UDC). In the following, we describe two kinds of analysis. In the first one, we studied the responses retrieved by our system and tried to understand the potential reasons for failure so that they can be addressed later. In the second analysis, we crossed the predictions of the basic dual encoder and our system to understand the significance of the numerical improvement that we can see in Table 4.1.

4.4.1 Prediction analysis

We analyzed the predictions made by our system on the test set to check whether we effectively match the context and the correct response and in the cases where our system fails, we would like to investigate the possible reasons. Table 4.2 contains an example that has been successfully classified according to Recall@1. As we can see, the context is composed of two utterances u1 and u2, the first user is asking a question about how to remove the chat option and the second user suggested him to log out. The correct response (written in bold) has been successfully ranked on top of the other candidate responses. Even though the candidate responses do not share common words with the context, our model was able to recognize the best response and assigned it the highest score.

	Context		Candidate responses
u1	how do i remove the chat option from the envelope icon at the top of the screen i already delete empathy	**0.98**	**i tried that but it's still there**
		0.25	thank the internet wasnt working because of this
u2	then you probably just need to log out to restart indicator-messages	0.22	thank so much
		0.14	sorry not mean for you

Table 4.2 – An extracted example from the test set where our system successfully retrieved the ground truth response (written in **bold**).

We are interested in error analysis and understanding as possible the reasons for wrong predictions made by our system. We randomly chose a test sample on which our system was not able to retrieve the correct response as shown in table 4.3. In this case, the expected response is "thank you", whereas our system predicted "it's only annoying when the cursor drag really slowly" as the correct response. We can see also that the other candidate responses were ranked on top of the ground-truth response.

	Context		Candidate responses
u1	http://www.howtogeek.com/114027/how-to-add-screensavers-to-ubuntu-12.04/ see also http://askubuntu.com/questions/64086/how-can-i-change-or-install-screensavers	0.99	it's only annoying when the cursor drag really slowly
		0.87	apt-get install hwinfo
u2	ok it won't become an issue on system upgrade	0.85	ok what is that ok just figure it out you just help me out haha
		0.27	**thank you**

Table 4.3 – An extracted example from the test set on which our system failed in retrieving the ground truth response (written in **bold**).

We believe that this happens because the problem is not clearly stated in the context, thus we can not understand what the users are talking about. For a human, it is difficult to choose between the candidate responses and for this reason, we think that our system choose the most probable response among the candidate responses in which we see the tokens "cursor" and "drag" which the system may associate to screensavers that figures in the context. We assume the second response obtained the second highest score because of the token "apt-get" and "install" that the system associated to "system upgrade" that appears in the context. This might be a particularity of the UDC as the original conversations were multi-party and some heuristics were applied to extract two-party conversations. This is why we have some contexts starting from the middle of the conversation and this may not help the system in finding the correct response.

4.4.2 Qualitative and quantitative analysis

We believe that the Ubuntu Dialogue Corpus contains some bias which makes building retrieval-based dialogue systems harder for at least three reasons. First, negative responses are randomly sampled from the whole corpus without any human judgment. Some negative responses could be potential responses for the given context such as *"Thank you", "yes !", "will try it", etc*. Second, the conversations of this corpus were originally between more than two persons and then, some heuristics were used to reduce the multi-user conversation to two users. Thus the coherence of the conversation when removing some important information risks to be lost. Third, the nature of these conversations is chat, which is very noisy compared to email, FAQ and forum datasets which contain fewer abbreviations, typos, etc.

Despite all these disadvantages, we believe that it is important to build dialogue models on this kind of datasets since chat is a large and available source of conversations as well as emails and forums. As an alternative solution, annotators could be recruited to label the candidate responses as positive or negative and in this case, we tolerate the possibility to have multiple good utterances for the same context. In this case, the Recall@k would not be the best metric to evaluate retrieval-based dialogue systems. Precision@1, Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR) will be more appropriate in the case of the presence of multiple good utterances.

In addition to the evaluation of our system with the retrieval metrics, we performed a qualitative and quantitative analysis of the obtained results. For this aim, we used the source code [4] of Lowe et al. (2015b) to train their dual encoder using the parameters that they reported on their paper. After that, we compared their predictions with the predictions of our improved dual encoder on the same test set.

Context	- Hello .. Is it possible to disable GPG check for a specific APT repository ? - Why would you ever need to do that - It's for a custom repository in enterprise environment. But that's unimportant isn't it. - Was that a statement that it's not possible ?		
Lowe et al. (2015b)	Boussaha et al. (2018b)	Label	Response
0,06	**0,87**	1	3rd party repo ? PPA ? what is it ?
0,29	0,25	0	Find it sticky edge
0,17	0,40	0	That response doesn't help me in the slightest

Context	- How can I remount a drive as read/write ? - mount -o rw /dev/whatever /whever I belive theres a remount option i think - Thanks - I'd say check the mount man page also. I forgot the syntax for the remount option		
Lowe et al. (2015b)	Boussaha et al. (2018b)	Label	Response
0,96	0,49	1	Okay
0,62	**0,88**	0	Thats sound like a good idea find out I'm missing authz_hos somehow
0,14	0,87	0	Thanks I will read that

Context	- Is there a length limitation on the hostname in SSH ? - 255 char for the FQDN - FQDN ?		
Lowe et al. (2015b)	Boussaha et al. (2018b)	Label	Response
0,99	**0,94**	1	Full Qualify Domain Name: mycomputer.kitchen.myhouse.com
0,01	0,27	0	Alright good luck
0,01	0,08	0	You have to do it once the bios hand off to grub

Context	- Is there a script that can generate a live cd iso of your currently run ubuntu hdd install ? - Remastersys - Have you use it ?		
Lowe et al. (2015b)	Boussaha et al. (2018b)	Label	Response
0,05	0,18	1	I hadn't have much luck with it, but that is a while ago (2 years)
0,88	**0,71**	0	It can
0,91	0,56	0	Not for me I doubt I could figure it out to be honest, but any theme installations come to mind as a guess

Table 4.4 – Examples of agreement and disagreement between our system and the baseline system Lowe et al. (2015b). Scores in **bold** are the highest scores obtained by the system.

Table 4.4 contains some examples that we extracted from the test set of the Ubuntu Dialogue Corpus. Each example is composed of a context of three or four dialogue turns, three candidate responses with their labels (1: correct, 0: wrong), and the prediction score by each system. In the first example, our system assigns the highest score to the correct response whereas the basic dual encoder fails in recognizing the correct response. Despite the difficulty of choosing a correct response, as none of the candidate responses explicitly shares common words with the context, our system successfully captured semantic relationships between *repo* et *repository*, *PPT* and *APT*.

The second example represents the case where the dual encoder of Lowe et al. (2015b) retrieved the correct response in contrast to our system which failed. We notice that even

4. Available at `https://github.com/npow/ubottu`

Lowe et al. (2015b) \ Boussaha et al. (2018b)	Success	Failure
Success	39.31%	11.33%
Failure	23.66%	25.70%

Table 4.5 – Statistics on the percentage of agreements and disagreements between the two systems.

if our system didn't attribute the highest score to the correct response, it chose the third response as the best response. This response can perfectly reply to the context while keeping the conversation coherent and logic. In the third example, both systems retrieved the correct response with very high scores. It means that it was quite easy to both systems to discard the correct response from the negative responses. The last example represents the case where both systems failed in retrieving the correct response. Here our system chose the answer "it can" because, probably, it was perfectly matched with the part of the context where "that can generate ..." appear. From this analysis, we can see the difficulty of the task as we humans we may choose a different response than the ground-truth response. Through this analysis, we realized that our system does not outperform the dual encoder (Lowe et al., 2015b) all the time. There were some cases where the dual encoder successfully retrieved the correct responses whereas our system failed.

For further comparison of the two systems, we quantify the cases of agreement and disagreement between both systems and we show the statistics in Table 4.5. As we can see, Both systems succeeded in ranking the correct response on top of the candidate responses in 39.31% of the test samples and both failed in 25% of the cases. We found out that there were 11.33% of the test samples where our system failed whereas the dual encoder succeeded. This interesting results show that there are some cases where the modifications that we did to the system did not only bring improvements but there are some cases that we miss.

4.5 Incorporation of the attention mechanism

As mentioned in Section 4.2, our main hypothesis on which we built our system, is that the context and the correct response share common semantics that we capture with an LSTM encoder on the sequence-level. However, these common semantics are either implicit or are diluted inside the sequences which make the task harder. As we explained in Section 2.5.2, the attention mechanism (Bahdanau et al., 2015) can alleviate this problem and help the systems to focus on the most important words in the context and the response. As these important words will receive higher importance, the model can use them to discard wrong responses from correct ones. The choice of which form of attention we will adopt in our architecture was defined by our architecture itself. Since soft, hard, local and global attention apply only in the case of encoder-decoder models, self-attention is the perfect match with our architecture since we have only an encoder. A summary of our model with the self-attention mechanism is provided in Figure 4.2.

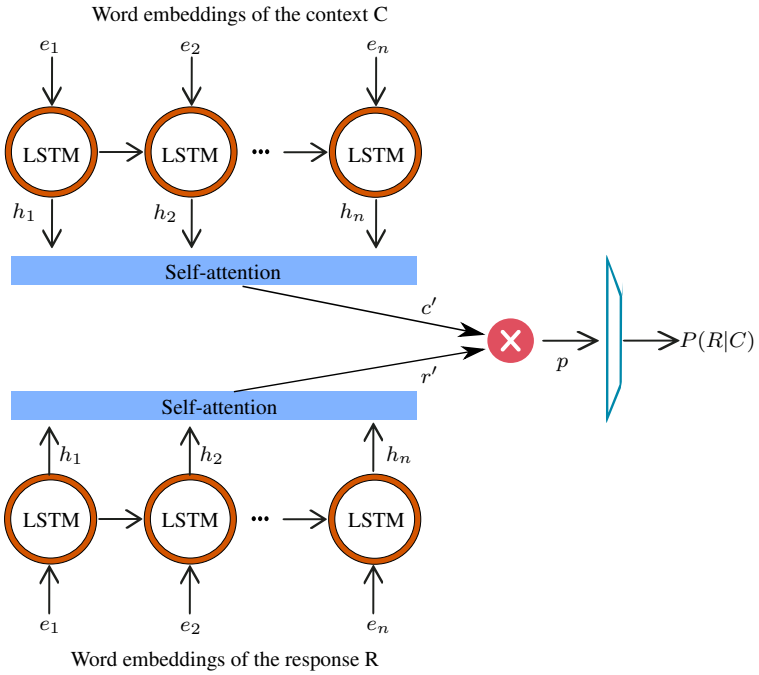


Figure 4.2 – Architecture of our system based on LSTM dual encoder. The self-attention mechanism is added on top of the LSTM encoder. Compared to our system without attention, here we return all the hidden states of the encoder in order to compute attention weights as shown in the Figure.

The shared encoder in our architecture encodes both the context and the candidate response. Each time a word embedding e_i is fed into the network, the hidden layer of the encoder is updated. Here, in contrast to the approach without attention where we consider only the last hidden state of the encoder, here we consider all the hidden states h_i of the encoder. We apply self-attention on these hidden states to obtain attention vectors on the input words of the context and the candidate response. We use the Keras implementation of self-attention [5] and we keep the same parameters as described in Section 4.3.2. Therefore, the attention vectors c' and r' are computed as follows:

$$\begin{aligned} u_t &= \tanh\left(W h_t\right) \\ \alpha_t &= \frac{\exp\left(\text{score}(u_t, u)\right)}{\sum_{t'=1}^{T} \exp\left(\text{score}(u_{t'}, u)\right)} \\ c' &= \sum_{t=1}^{T} \alpha_t h_t \end{aligned} \tag{4.4}$$

Results of the application of self-attention on our system are given in Table 4.6. As we can see, the attention mechanism brings a small improvement to our dual encoder on both

5. Available at https://pypi.org/project/keras-self-attention/

System	Ubuntu Dialogue Corpus V1				Douban Conversation Corpus					
	$R_2@1$	$R_{10}@1$	$R_{10}@2$	$R_{10}@5$	$R_{10}@1$	$R_{10}@2$	$R_{10}@5$	P@1	MAP	MRR
Our system	0.917	0.685	0.825	0.957	0.209	0.357	0.702	0.358	0.500	0.543
Our system + attention	0.918	0.696	0.830	0.957	0.204	0.363	0.742	0.345	0.507	0.544

Table 4.6 – The impact of incorporating self-attention mechanism to our system.

anyone in here have ubuntu running on a toshiba a135 eot what exactly is your issue with it eot right now i ve got a really messed up vista installed on the a135 and i m trying to put ubuntu on instead but i could n't get the livecd to boot when i select try without making changes i just get a black screen with a blinking cursor

(a) Attention weights of the context.

you have to give the darn thing some time to load the blinky cursor screen is normal for a little while and also in vista you cant shrink the active partition p the cd will let you mess with the partitions as you install it so just give it time to load up the livecd

(b) Attention weights of the correct response.

Figure 4.3 – Visualization of the attention weights on a test sample. The darker color a word gets, the higher attention weight it has. Note that "eot" denotes the end of a turn.

datasets. In Figure 4.3, we project the attention weights on the inputs of our system with attention. In Figure 4.3a, we can see that important words were effectively identified in the context by the system and different weights were attributed. For example in the first line of the context, the highest weights were attributed to *toshiba* and *a135*. When the word *a135* has been identified again in line 3, it has been given more attention since our system was able to detect that it is an important word in the conversation. As we have a shared encoder, when the response is fed into the encoder right after its associated context, the system was able to accentuate more on the common words that have been seen in the context such as *screen*, *cursor*, *livecd*, *vista*, etc. as illustrated in Figure 4.3b. What is also interesting to observe, is that our system was able to identify important words in the response and attribute them high attention scores even if they were not present in the context such as *partition*, *cd* and *shrink*. Overall, we can see that incorporating the attention mechanism helps our system accentuating certain words and slightly improve the performance.

4.6 Conclusion

As a first experience, we reduced our study to single-turn matching systems as they are simpler than multi-turn matching systems. In this category, the dual encoder (Lowe et al., 2015b) was one of the first neural approaches that was applied to the task of retrieving the next utterance of a given dialogue. It has been a strong baseline for the works that have been established after. However, these works tried to improve the performances while making their systems more complex or domain-dependent by including knowledge bases or

handcrafted features. We analyzed the basic dual encoder and identified some limitations related to the encoding part in which two separate encoders were used to produce the context and the candidate response. In addition to the separate encoders, a matrix of additional parameter was learned which make the system take longer time to converge and learn an optimal matching of the context and the correct response.

We addressed these limitations with an improved, simple and efficient model that alleviates the first issue with a shared encoder. This shared encoder, not only ensures that the context and the candidate response are projected into the same space, but also reduces the number of parameters that the system has to optimize. With a simple similarity measure between the context and the response vectors such as the cross product, the parameters matrix is no more needed and the sequence-level similarity between the context and the candidate response is computed. We conducted experiments on two public and large datasets of two different languages, the Ubuntu Dialogue Corpus and the Douban Conversation Corpus. We showed that we were able to improve the performance of the basic dual encoder with 5%, 4% and 1% on Recall@1,5,10 respectively on the UDC. Moreover, we compared our system to the state-of-the-art single-turn matching systems and showed that with a simple, domain independent and end-to-end architecture, we were able to outperform some complex and sometimes domain dependant retrieval-based dialogue systems.

We also experimented with the self-attention mechanism to help the system accentuate the most relevant words in the context and the candidate response so that they can be efficiently matched. This contribution was subject to a publication at a national conference on Traitement Automatique des Langues Naturelles (TALN 2018) (Boussaha et al., 2018b) and an international symposium Machine Learning and Data Analytics Symposium (MLDAS 2018) (Boussaha et al., 2018a).

5

Multi-level context response matching system

“ If you had all of the world’s information directly attached to your brain, or an artificial brain that was smarter than your brain, you’d be better off.”

— Sergey Brin

5.1 Introduction

We presented in the previous chapter a single-turn single-level matching system inspired by the dual encoder (Lowe et al., 2015b). Our system matches the whole context with the candidate response on the sequence-level. We showed that with a simple and end-to-end architecture we were able to outperform the original dual encoder as well as other strong single-turn matching baseline systems. On the other hand, multi-turn matching systems match the response with each of the context utterances and aggregate the matching scores to obtain the final score of the response (Wu et al., 2017; Zhou et al., 2018; Yang et al., 2018; Zhang et al., 2018c). Some of these systems match the response with each utterance on two levels: sequence-level and word-level (Wu et al., 2017; Zhang et al., 2018c). These systems, even though their efficiency, they suffer from the complexity of their architectures by combining different similarity levels, aggregation, attention mechanisms, and matching

levels.

Inspired by the multi similarity levels deployed in the Sequential Matching Network (SMN) (Wu et al., 2017) which is a multi-turn matching system, we extend our single-turn matching system presented in the previous chapter to include word-level similarity. Our new single-turn matching system matches the whole context with the candidate response only one time on the sequence-level and word-level. We evaluate our system on two large dialogue datasets of two different languages: the Ubuntu Dialogue Corpus (Lowe et al., 2015b) and the Douban Conversation Corpus (Wu et al., 2017). We show that the resulting system achieves high performances while being conceptually simpler and having fewer parameters compared to the previous, substantially more complex, systems. We also present the results of the error analysis that we performed on a test subset of the test to understand why our system fail on some test cases. Moreover, we remove some parts of our system to evaluate their impact on the whole system. Also, we participate with our system to the 7^{th} edition of the Dialog System Technology Challenge (DSTC7) which provides a new and challenging evaluation environment to retrieval-based dialogue system. We present how we adapt our system to the new challenges and the results that we obtained.

5.2 Approach

We propose an end-to-end multi-level context response matching dialogue system. First, we project the context and the candidate response into a distributed representation (word embeddings). Second, we encode the context and the candidate response into two fixed-size vectors using a shared LSTM encoder (described in Figure 5.1 with the blue frame). This process is similar to our previous system described in Chapter 4. Then, in parallel, we compute two similarity vectors: on the word-level and sequence-level. The sequence-level similarity vector is obtained by multiplying the context and the response vectors. Whereas the word-level similarity vector is obtained by multiplying word embeddings of the context and the candidate response. Both similarity vectors are concatenated and transformed into a probability of the candidate response being the next utterance of the given context. In the following, we elaborate on the functions of our system.

The first two functions of our system are sequence encoding and sequence-level similarity. These two functions are similar to the ones described in the previous chapter (Section 4.2) as we extend our single-turn matching system. First, we compute c' and r' vectors by encoding C and R with the LSTM encoder (as in Equation 4.1). Afterwards, we compute the similarity between the context and the candidate response on the sequence-level with a cross-product between c' and r' as in Equation 4.2. The resulting vector s represents the similarity on the sequence-level between the context and the candidate response. In the following, we describe our new contribution in the form of another similarity level.

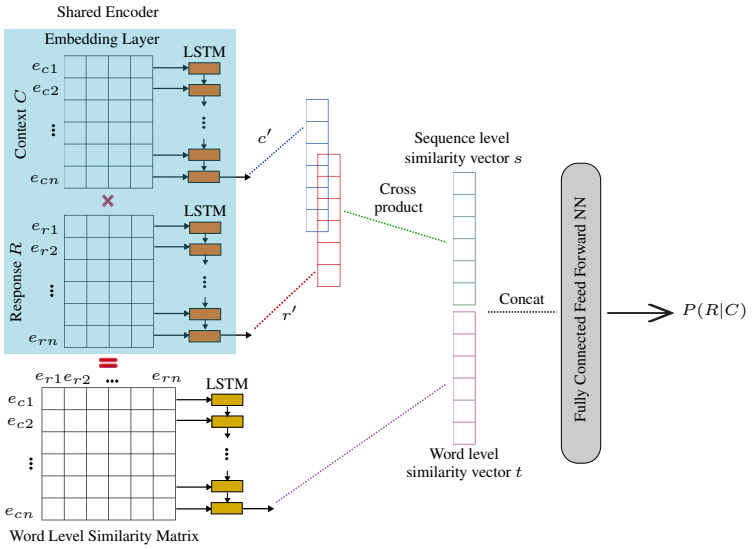


Figure 5.1 – Architecture of our multi-level context response matching dialogue system.

5.2.1 Word-level similarity

We believe that sequence-level similarity is not enough to match the context with the best response. Adding word-level similarity could help the system learning an improved relationship between C and R. This assumption was consolidated by observing the scores dropping when word-level similarity was removed from the SMN of Wu et al. (2017) (we refer to section *"Model ablation"* of their paper).

Therefore, we compute a word-level similarity matrix WLSM $\in \mathbb{R}^{n \times n}$ by multiplying every word embedding of the context e_{ci} by every word embedding of the response e_{rj} as:

$$\text{WLSM}_{i,j} = e_{ci} \cdot e_{rj} \tag{5.1}$$

This similarity matrix contains matching scores between the words of the context and the response two by two. As we compute this similarity between word embeddings using the dot product, we can capture the similarities between words like *"Paris"* and *"France"*, *"Queen"* and *"King"*, etc. thanks to the arithmetic properties of word embeddings. To transform the Word-Level Similarity Matrix into a vector, we feed every row WLSM_i into an LSTM recurrent network which learns a representation of the chronological dependency and the semantic similarity between the context and response words (see Figure 5.1). Similarly to Equation 4.1, we encode the word-level similarity matrix into a vector $t = h'_n \in \mathbb{R}^{l'}$ where l' is the dimension of the hidden layer of the LSTM network and h'_n is the last hidden vector of the network.

5.2.2 Response score

At this stage we have two vectors: s representing the similarity between C and R on the sequence-level and t representing the word-level similarity. We concatenate both vectors and transform the resulting vector into a probability using a one-layer fully-connected Feed-Forward Neural Network with a sigmoid activation (Equation 5.2). The last layer predicts the probability $P(R|C)$ of the response R being the next utterance of the context C as:

$$P(R|C) = sigmoid(W' \cdot (s \oplus t) + b) \tag{5.2}$$

Where W' and b are parameters and $\oplus$ denotes concatenation. We train our model to minimize the binary cross-entropy loss.

The advantage of our multi-level retrieval-based dialogue system compared to our previous system is the addition of an extra similarity level on which we match the context and the candidate response word by word. Thus, we explicitly consider the similarity between the words represented by their contextual vectors (word embeddings) which is implicitly computed in the sequence-level similarity. In the following sections, we compare our system to state-of-the-art systems of the same category (single-turn) and multi-turn matching systems and show that while preserving the simplicity of our architecture, we can do as good as and sometimes better than complex or domain-dependent systems.

5.3 Experimental setup

In this section, we present the experimental environment in which we evaluated our system. First, we present the baseline systems to which we compare our system.

5.3.1 Baseline systems

Our system belongs to the single-turn category in which the response is matched with the whole context without explicitly distinguishing its utterances. We compare our system to baseline systems of the same category (single-turn). We refer to their description given briefly in Section 4.3.1 and more in detail in Section 2.3.2.1. Moreover, we compare our system to other systems of a different category (multi-turn) which are more complex to show the efficiency of our approach. We briefly describe each system of this category below and we refer to Section 2.3.2.2 for the full description.

Deep Learning to Respond (DL2R) Proposed by Yan et al. (2016) based on contextually query reformulation and an aggregation of three similarity scores computed on the sequence-level. The reformulated query is matched with the response, the original query and the previous post.

Multi-View This system was designed by Zhou et al. (2016) in which two similarity levels between the candidate response and the context are computed and the model is trained to minimize two losses. The disagreement loss and the likelihood loss between the prediction of the system and what the system was supposed to predict.

Sequential Matching Network (SMN) Proposed by Wu et al. (2017). The candidate response and every dialogue turn of the context are encoded using a GRU network (Chung et al., 2014). Then, the response is matched with every turn using a succession of convolutions and max-pooling.

Deep Attention Matching Network (DAM) Introduced in the work of Zhou et al. (2018). This system is an improvement of the SMN (Wu et al., 2017) in which the Transformer (Vaswani et al., 2017) was used to produce utterance representations based on self-attention. These representations are matched together to produce self- and cross-attention scores which are stacked as a 3D matching image. Then, a ranking score is produced from this image via convolution and max pooling operations.

Deep Utterance Aggregation (DUA) (Zhang et al., 2018c) extends the SMN with an explicit weighting of the context utterances. The authors hypothesize that the last utterance of the context is the most relevant and thus concatenates its encoded representation with all the previous utterances in addition to the candidate response. After that, a gated self-matching attention (Wang et al., 2017) is applied to remove redundant information from the obtained representation before feeding them into the CNN as in the SMN (Wu et al., 2017).

5.3.2 System parameters

The initial learning rate was set to 0.001 and Adam's parameters β_1 and β_2 were set to 0.9 and 0.999 respectively. As a regularization strategy we used *early-stopping* and to train the model we used mini batch of size 256. We trained word embeddings of size 300 on UDC and 100 on Douban using FastText (Bojanowski et al., 2017). The sizes of the hidden layers of the sequence LSTM and the word LSTM were set to 300 and 200 respectively. The system parameters were updated using Stochastic Gradient Descent with Adam algorithm (Kingma and Ba, 2015). We limited the maximum length of both the context and the response sequences to 160 words. Sequences with more than 160 words were truncated and the last 160 words were kept. Smaller sized sequences were padded with the `unknown` word. All the hyper-parameters were obtained with a grid search on the validation set. We implemented our system with Keras (Chollet et al., 2015) and Theano (Theano Development Team, 2016) in the backend. We release our source code on `https://github.com/basma-b/multi_level_chatbot`.

5.4 Results and analysis

In this section, we provide a table summarizing the results of our system and the baseline systems in addition to a visualization of the WLSM matrix, error analysis, and a model

System	Ubuntu Dialogue Corpus V1				Douban Conversation Corpus					
	$R_2@1$	$R_{10}@1$	$R_{10}@2$	$R_{10}@5$	$R_{10}@1$	$R_{10}@2$	$R_{10}@5$	P@1	MAP	MRR
TF-IDF (Lowe et al., 2017b)	0.659	0.410	0.545	0.708	0.096	0.172	0.405	0.180	0.331	0.359
LSTM (Lowe et al., 2017b)	0.901	0.638	0.784	0.949	0.187	0.343	0.720	0.320	0.485	0.527
BiLSTM (Kadlec et al., 2015)	0.895	0.630	0.780	0.944	0.184	0.330	0.716	0.313	0.479	0.514
Attentive-LSTM (Tan et al., 2015)	0.903	0.633	0.789	0.942	0.192	0.328	0.718	0.331	0.495	0.523
MV-LSTM (Wan et al., 2016)	0.906	0.653	0.804	0.946	0.202	0.351	0.710	0.348	0.498	0.538
Match-LSTM (Wang and Jiang, 2016)	0.904	0.653	0.799	0.944	0.202	0.348	0.720	0.345	0.500	0.537
DL2R (Yan et al., 2016)	0.899	0.626	0.783	0.944	0.193	0.342	0.705	0.330	0.488	0.527
Multi-View (Zhou et al., 2016)	0.908	0.662	0.801	0.951	0.202	0.350	0.729	0.342	0.505	0.543
$SMN_{dynamic}$ (Wu et al., 2017)	0.926	0.726	0.847	0.961	0.233	0.396	0.724	0.397	0.529	0.569
DAM (Zhou et al., 2018)	0.938	0.767	0.874	0.969	0.254	0.410	0.757	0.427	0.550	0.601
DUA (Zhang et al., 2018c)	-	0.752	0.868	0.962	0.243	0.421	0.780	0.421	0.551	0.599
Our single-level system	0.917	0.685	0.825	0.957	0.209	0.357	0.702	0.358	0.500	0.543
Our multi-level system	0.935	0.763	0.870	0.968	0.255	0.414	0.758	0.418	0.548	0.594

Table 5.1 – Evaluation results on the UDC (V1) and Douban Corpus using retrieval metrics.

ablation study.

5.4.1 Results

Table 5.1 summarizes the evaluation results on UDC (V1) and Douban Conversation Corpus. Compared to the single-turn systems (the first six rows), our system achieves the best results on all metrics and both datasets. These systems are based on only sequence-level similarity between the context and the candidate response whereas our system incorporates word-level similarity in addition to the sequence similarity. Moreover, compared to multi-turn matching systems, our system outperforms the DL2R (Yan et al., 2016), Multi-View (Zhou et al., 2016) and the $SMN_{dynamic}$ (Wu et al., 2017) with a good margin (around 4% and 3% on Recall@1 and 2 respectively on UDC compared to $SMN_{dynamic}$). These three systems match the response with every context utterance and uses multiple convolutions and max-pooling to rank the response. However, they are outperformed by our simple single-turn system with two similarity levels without requiring complex matching and aggregation mechanisms.

The two last systems, the Deep Attention Matching (DAM) (Zhou et al., 2018) and the Deep Utterance Aggregation (DUA) (Zhang et al., 2018c) use the attention mechanism. For instance, the DAM as detailed in Section 2.3.2.2, is based on multiple layers of the self-attention (Transformer) and Convolutional Neural Networks (LeCun et al., 1998). The advantages of the Transformer are related to the performance improvement and the acceleration of the learning compared to neural networks (Vaswani et al., 2017). However, we proposed an architecture that is fully based on neural networks but that achieves almost the same results as the DAM and sometimes better. In contrast to the advantages of the Transformer, our system converges quickly. According to the authors (Zhou et al., 2018), their system was trained on one Nvidia Tesla P40 GPU, on which one epoch lasts for 8 hours on UDC and their system converges after 3 epochs. However, training our system for one epoch lasts for 50 minutes on one Nvidia Titan X Pascal GPU (Both GPUs have almost

the same characteristics [1]). Moreover, our system converges after only two epochs [2]. Having such architectures (as DAM) makes reproducibility of results harder due to hardware limitations and the time necessary to perform training and cross-validation.

The way we compute the Word-Level Similarity Matrix (WLSM) looks like the cross-attention mechanism since we match the words from the context and the response and because higher values of the matrix mean higher importance. Also, fine-tuning the cell values during training looks like fine-tuning the attention weights. For these reason we do not incorporate attention mechanism to our model as we think that the WLSM matrix is enough to help the system identifying important words. Note that on Douban, the overall performance of all the systems is lower than on UDC. We believe that this is due to the language of the corpus (Chinese) and the nature of the Douban corpus in which a context may have more than one ground-truth response and hence every retrieval system must find all the responses.

5.4.2 Error analysis

To understand the reasons for failure of our system in retrieving the correct response in some test samples, we performed a human evaluation of 200 randomly selected test samples from UDC where the ground-truth response was not retrieved by our system. By observing the test samples that were misclassified, we identified 4 error classes. Table 5.7 summarizes the distribution of the test samples over these classes and Table 5.6 contains an example of each error class.

(a) **Functionally equivalent**: This class regroups 62 cases (31%) where our system predicted a response that we believe it could replace (substitute) the ground-truth response without carrying the same meaning. For example both *"Yes I tried that also but it does not work"* and *"how can I do that ?"* are possible responses to the following context *"check if it appears in the package manager"* without being semantically equivalent.

(b) **Semantically equivalent**: In this class, we find 40 test samples (20%) where the predicted response has similar meaning as the ground-truth response. For instance, *"nice"* and *"great"* are semantically similar.

(c) **Out of context**: This last class contains 71 test samples (35.5%) where our system predicted a response that is not general (i.e that it is a technical response) which is neither functionally nor semantically related to the ground-truth response.

(d) **Very general responses**: In this class we regrouped 27 predictions (13.5%) where the ground-truth response was very specific to the context whereas our system predicted a thanking, greeting, apologizing, feedback informing, etc. responses. Examples include *"Thank you"*, *"Great"*, *"Ok"* and *"Yes"*.

We can sum up the findings of our error analysis into two important points:

1. https://technical.city/en/video/Titan-X-Pascal-vs-Tesla-P40
2. The number of trainable parameters of our system and DAM are almost the same.

Context	Candidate responses
have some problems with laptop lid action vaio ubuntu 12.04 any ideas anyone *eot* get inform about pm utils *eot*	- why don't you just me off 2gb ✓ - **i get operating system not find message** ✗

Table 5.2 – Functionally equivalent.

Context	Candidate responses
hi i'm trying to skip fsck on boot and edit the kernel line in boot add but it still auto runs fsck what else can i try thank *eot* edit your fstab file *eot*	- i can't get to it because i can't boot the server ✓ - **yeah copy paste** ✗

Table 5.4 – Out of context.

Context	Candidate responses
what is default instant message client for 12.10 *eot* empathy *eot*	- thank you ✓ - **thanks** ✗

Table 5.3 – Semantically equivalent.

Context	Candidate responses
hello i have ubuntu 10.10 what is the best way to update to 12.10 *eot* fastest and most efficient way is fresh install *eot*	- i have tried it but then i had a group rescue mode because of windows 7 ✓ - **ok thanks** ✗

Table 5.5 – Very general response.

Table 5.6 – Examples of errors raised by our system are grouped by error classes. The first response in the candidate response column is the ground-truth response whereas the second response (in bold) is the one predicted by our system. "*eot*" denotes the end of turn.

— Through this in-depth analysis we observed that around 50% of errors are due to general and completely out of context (classes c and d) responses highly ranked by our retrieval system. However, these limitations were originally observed in generative systems since they were not able to produce coherent, syntactically correct and specific responses to the context of the conversation (Li et al., 2016a). This finding encourages us to perform in-depth comparative studies between retrieval-based and generative dialogue systems.

— We highlight the importance of performing human evaluation on the candidate responses in the validation and the test sets. They should be carefully selected instead of randomly sampled from the corpus. 51% of the errors were due to the presence of responses that were functionally and semantically equivalent to the ground-truth response that were considered as negative responses in the corpus (classes a and b).

We argue that these findings of the error analysis are important and have to be considered when building dialogue corpora and dialogue systems. We believe that the evaluation environment including the dataset, the evaluation metrics, the compared systems, etc. has to be correct and adapted to the task. Because any bias may falsify the evaluation and the conclusions. Here, we can blame the way the Ubuntu Dialogue Corpus was constructed fol-

Error class	Percentage
Functionally equivalent	31%
Semantically equivalent	20%
Out of context	35.5%
Very general responses	13.5%

Table 5.7 – Error classes.

lowing negative sampling approach which allowed annotating correct responses as negative and also the evaluation metrics, as we mentioned in Section 2.6.3, which are not flexible enough to allow the system retrieving responses that are semantically similar to the ground-truth response.

5.4.3 Visualization

Furthermore, to verify the hypothesis on whether the Word-Level Similarity Matrix (WLSM) captures the relationships between the words of the context and the response, we visualized the WLSM matrix for the following test sample. The last two utterances of the context are: **A**: *hey anybody know how i can share file between xp guest and ubuntu 12.04 lts host in vmware ?* **B**: *"install ssh on ubuntu and use winscp on xp"*. The positive response is *"do i need to upload it to internet and download it again"*. In Figure 5.2, we plotted the Word-Level Similarity Matrix (WLSM) between the context (x-axis) and the response (y-axis). For formatting matters, we visualize only the last dialogue utterance (**B**) of the context.

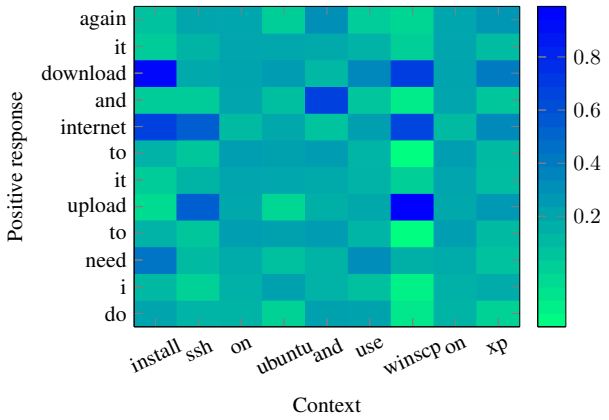


Figure 5.2 – Visualization of the Word-Level Similarity Matrix (WLSM).

As we can see, important (key) words in the context and the response were successfully recognized by our system and were given higher scores. For instance, *upload*, *internet* and *download* were matched with *install*, *ssh*, *winscp* and *xp*. This observation illustrates the importance of computing word-level similarity from word embeddings to match the

context with the best response. A quantitative evaluation of the impact of WLSM on the performance of our system is given in the next section.

5.4.4 Model ablation

We report in the two last rows of Table 5.1 the performance of our system while having only one similarity level. We notice that having only one level of similarity causes a drop in the system performance. Performances are higher when matching the context with the candidate response on the word-level compared to the sequence-level. We believe that this is because the word-level matching is more fine-grained compared to sequence-level matching. When encoding the whole sequence, some information carried by single words may vanish, but it can be captured during the word-level matching. Considering the example of Section 5.4.3, the whole context and the response are semantically similar. Having in addition to this sequence similarity, the fact that *upload*, *internet* and *download* match with *install*, *ssh* and *winscp* will help the system better recognizing the good responses. Vice versa, we can have responses that share semantically equivalent words with the context while the whole meaning of the response is not related to the whole meaning of the context.

These results demonstrate that both similarity levels are complementary and jointly incorporating them into our model results in better performances.

5.5 Dialog System Technology Challenge (DSTC7)

The existing retrieval-based dialog systems are evaluated on non-realistic scenarios. Usually, these systems select the correct response from a very small set of candidate responses of size 10 (Lowe et al., 2015b; Wu et al., 2018a; Baudiš et al., 2016; Wu et al., 2017; Zhou et al., 2018). However, when building task-oriented dialogue systems, the set of possible responses is usually very large. Moreover, the actual systems respond even if no correct response is available in the pool of candidate responses in addition to the fact that most of them hypothesize that only one response is correct. However, multiple candidate responses could be correct responses. Addressing these limitations was the goal of the 1^{st} track (sentence selection) of the 7^{th} edition of the Dialog System Technology Challenge (DSTC7) (Yoshino et al., 2018). This track aims to push the state-of-the-art task-oriented dialogue systems in more realistic evaluation scenarios (Gunasekara et al., 2019). We participated in this track with our single-turn and multi-level context response matching system with 20 other teams. Our system was ranked 7^{th} on the Ubuntu dataset and 6^{th} on Advising dataset by achieving 75.6% on Recall@10 and 57.3% on MRR outperforming the baseline system by 39.7% on Recall@10 on the Ubuntu dataset.

5.5.1 Task description

DSTC7 [3] is the 7^{th} edition of the Dialog System Technology Challenges. This edition contains three tracks: sentence selection, sentence-generation and audiovisual scene-aware dialog. The first track aims to retrieve the correct response for a given conversation's history called the context from a set of candidate responses. The goal of the sentence generation track is to generate conversational responses that go beyond chitchat, by injecting informational responses that are grounded in external knowledge. The last track aims to understand the scenes of an input video to have conversations with users about the objects and events around the video. The common point between the three tracks is: the participating systems must be data-driven and end-to-end. Our participation to this challenge, focuses on the *sentence selection* track. In the following, we describe the track and its related subtasks.

5.5.2 Sentence selection track

Until today, the recent studies evaluated retrieval-based dialogue systems in non-realistic conditions that do not reflect the reality. We can summarize the limitations of the state-of-the-art retrieval-based dialogue systems in the following three points.

— Most of the recent systems were challenged to retrieve the ground-truth response among a set of only 10 candidate responses randomly sampled from the dataset which is far from approaching the reality (Lowe et al., 2015b; Xu et al., 2017; Wu et al., 2018a, 2017). In real configuration, a dialogue system has a large base of responses usually collected from human conversations from which the system has to pick one or more responses.

— Recent works limit the number of correct responses of a given context to only one. However, in most cases, multiple correct responses are possible.

— Even when no correct response is included in the pool of candidate responses, most of the systems are not able to know what is wrong and retrieve a response anyway. However, they should be able to not provide a response in such situations and ask the help of humans for example.

The main aim of the first track of DSTC7 is to address these limitations and to push task-oriented dialogue systems to more realistic problems that every practical automated agent has to deal with (Gunasekara et al., 2019). In this track, two dialogue datasets were provided: the Ubuntu Dialogue Corpus and the Advising Corpus (V3). Five subtasks were proposed where each subtask concerns one or both datasets. In the following, we describe the subtasks and the datasets. A summary of the subtasks is given in Table 5.8.

Subtask 1 Given the context of a conversation and a set of 100 candidate responses, the task consists of selecting the correct response. On 100 candidate responses, only one is correct. This subtask is available on both datasets.

3. http://workshop.colips.org/dstc7

Subtask	Description	Ubuntu	Advising
1	Select one response from a pool of 100 candidate responses	✓	✓
2	Select one response from a pool of 120000 candidate responses	✓	✗
3	Select a response and its paraphrases from a pool of 100 candidate responses	✗	✓
4	Select one response from a pool of 100 candidate responses that may not contain the response	✓	✓
5	Same as 1 but the usage of a provided external data is mandatory	✓	✓

Table 5.8 – Subtasks of the sentence selection task of DSTC7.

Subtask 2 This subtask challenges the logical capability of the dialogue model by increasing the size of the candidate responses set. Hence, the task consists of selecting the correct response from a pool of 120,000 candidate responses which is 12,000 times larger than the usual size of the candidate set. The 120,000 candidate responses are shared across training, validation and test sets and also across samples. Only Ubuntu Dialogue Corpus (V3) is concerned with this task.

Subtask 3 In this subtask, between one and five correct responses are available in the set of candidate responses of size 100. The set of correct responses if available are paraphrases of the original correct response and the number of paraphrases has been chosen randomly. This subtask aims to evaluate the ability of the participating systems to retrieve all the correct responses (the correct response and its paraphrases) by ranking them on top of the candidate responses. This subtask is only available on Advising corpus.

Subtask 4 The candidate set contains 100 responses that may not include the correct response. Here, retrieval systems must be able to respond with a `None` response when no correct response is found. This subtask is applicable to both datasets.

Subtask 5 In this last subtask, an external knowledge base is provided and the model should incorporate it to retrieve the only correct response in a set of 100 candidate responses. The knowledge bases are Ubuntu manual pages in the case of the Ubuntu Dialogue Corpus (V3) and course descriptions in the case of the Advising Corpus.

In this challenge, we participated in subtasks 1, 3, 4 and 5.

5.5.3 System description

We used the same system described in Section 5.2 in the four subtasks to which we participated with/without extension depending on the subtask. For instance, in subtask 1, we used the system described in Figure 5.1. In subtask 3, we hypothesize that if our system can match the context with the correct response, it should be able to match its paraphrases with the same context as well. Thus, we used the same system as subtask 1. In subtask 4, our system should be able to recognize cases where no correct response is available in the set of candidate responses. Therefore, we extended the same system used in subtasks 1 and 3 with an SVM classifier (Ben-Hur et al., 2001) with RBF kernel as described in figure 5.3. For every candidate response and a context, our response retrieval system provides a ranking score. Once we have the ranking scores of all the candidate responses, we feed them to the SVM classifier. We train this classifier to predict the presence of a correct

response among the candidate responses using the labeled training data. Subtask 5 requires participants to include the external knowledge into their system while maintaining the end-to-end property of their architectures. Man pages and course descriptions were provided as external data for the Ubuntu Dialogue Corpus (V3) and the Advising Corpus respectively. We extract plain text from these external data and we train word embeddings on them using FastText (Bojanowski et al., 2017). These word embeddings are used later to initialize the embeddings layer in our system.

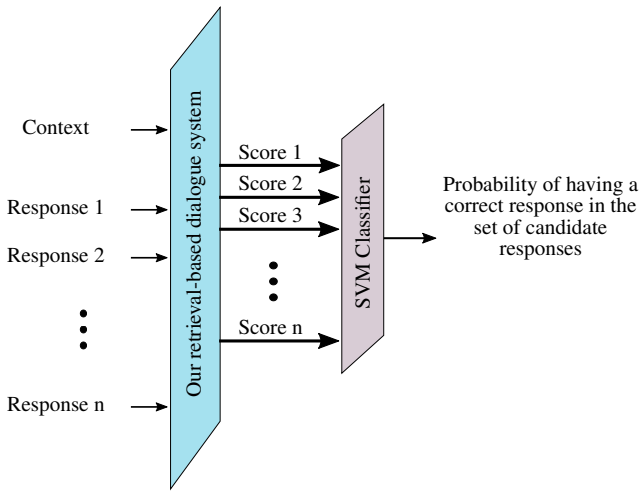


Figure 5.3 – Extension of our proposed system for subtask 4.

5.5.4 Experimental setup

In this section, we present the experimental environment of the DSTC7 challenge in which we evaluated our system. We describing the datasets, the evaluation metrics and the parameters of the system.

5.5.4.1 Datasets

DSTC7 provided two new goal-oriented dialogue datasets to build and evaluate retrieval-based dialogue systems. Each dataset is split into training, validation and testing sets. Table 5.9 summarizes statistics of both datasets for each subtask. Note that Subtask 2 concerns only the Ubuntu Dialogue Corpus (3), Subtask 3 concerns only the Advising Corpus, and the datasets of Subtask 5 are the same as Subtask 1. Also, two test sets of the Advising corpus were released. In the following, we briefly describe the two datasets (for a full description, we refer to Chapter 3).

The Ubuntu Dialogue Corpus (V3) This corpus contains two-party dialogues extracted from the Ubuntu channel on the Freenode Internet Relay Chat (IRC) (Kummerfeld et al., 2018).

	Subtask 1							Subtask 3				Subtask 4						
	Ubuntu Corpus V3			Advising Corpus				Advising Corpus				Ubuntu Corpus V3			Advising Corpus			
	Train	Dev	Test	Train	Dev	Test1	Test2	Train	Dev	Test1	Test2	Train	Dev	Test	Train	Dev	Test1	Test2
# dialogues	100K	5K	1K	100K	500	500	500	100K	500	500	500	100K	5K	1K	100K	500	500	500
# cand. R per C	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
# + responses	1	1	1	1	1	1	1	1-5	1-5	1-5	1-5	0-1	0-1	0-1	0-1	0-1	0-1	0-1
Min # turns per C	3	3	3	1	1	1	1	1	1	1	1	3	3	3	1	1	1	1
Max # turns per C	75	53	43	41	34	36	26	41	34	36	26	81	51	65	41	34	36	26
Avg. # turns per C	5.49	5.59	3.84	9.22	9.78	9.47	9.44	9.22	9.78	9.47	9.44	5.45	5.43	5.59	9.22	9.78	9.47	9.44
Avg. # tokens per C	74.03	72.47	81.32	79.88	83.86	87.37	82.22	79.88	83.86	87.37	82.22	73.24	72.90	72.73	79.88	83.86	87.37	82.22
Avg. # tokens per R	62.92	62.82	63.06	57.83	66.13	66.60	67.38	57.90	65.94	66.57	67.15	62.91	62.96	62.66	57.82	66.10	66.59	67.39

Table 5.9 – Datasets statistics. *C*, *R* and *cand.* denote context, response and candidate respectively. The dataset of Subtask 5 is the same as Subtask 1.

The corpus contains Ubuntu-related conversations. Every sample of this corpus is composed of a context which is a set of successive dialogue turns and a response which replies to the context. Moreover, a set of randomly crawled candidate responses is provided. The task consists of ranking the correct response on top of the candidate responses.

Advising Corpus The advising corpus contains conversations between teacher and student in which the teacher tries to answer the student's questions about the courses he/she will take. The teacher aims to provide information related to the duration of the course, its difficulty, whether the student's profile is adapted to the course, etc.

5.5.4.2 Evaluation metrics

For all the subtasks, DSTC7 uses Recall@1, Recall@10, Recall@50, and Mean Recall Rank (MRR) (Voorhees, 2001) as evaluation metrics. Only for subtask 3, Mean Average Precision (MAP) (Baeza-Yates and Ribeiro-Neto, 1999) is used in addition to the previous metrics. The average of Recall@10 and MRR per task and dataset are used to rank the participating systems.

5.5.4.3 System parameters

The only pre-processing performed on the dataset is tokenization using Keras Tokenizer. The system parameters were updated using Stochastic Gradient Descent with Adam algorithm (Kingma and Ba, 2015). The initial learning rate was set to 0.001 and Adam's parameters β_1 and β_2 were set to 0.9 and 0.999 respectively. As a regularization strategy we used *early-stopping* and to train the model we used mini batch of size 256. The size of word embeddings [4] and the size of the hidden layer of the encoder LSTM were set to 300. Whereas the size of the hidden layer of the second LSTM that learns the WLSM matrix was set to 200. We limited the maximum length of both the context and the response sequences to 160 words. All the hyper-parameters were obtained with a grid search on the validation set. We implemented our system with Keras (Chollet et al., 2015) with Theano (Theano Development Team, 2016) in the backend that we trained on a single Titan X GPU. We

4. We trained word embeddings on the training sets using FastText (Bojanowski et al., 2017) `-ws 5 -minCount 1 -dim 100` (Advising) `-dim 300` (Ubuntu).

used the SVM implementation provided by Scikit-learn (Pedregosa et al., 2011) with the default parameters.

5.5.5 Results and analysis

The baseline system provided by the challenge organizers is an implementation of the dual encoder of (Lowe et al., 2015b). We recall the differences between our system and the baseline system in the following. (1) Our system learns to match the context and the candidate response on the word-level and sequence-level whereas the baseline system is based on only the sequence similarity. (2) We use a shared encoder to encode the context and the candidate response while the baseline system uses different encoders. This allows the encoded context and the encoded response to be presented in the same vector space. (3) Unlike the baseline system, at each time step of the training, our system matches the context with one candidate response and thus the encoder is alternating the context and the response which is coherent to the chronological order of dialogue turns in the context and the response.

System	Subtask	Measure	Ubuntu Dialogue Corpus	Advising Corpus case 1	Advising Corpus case 2
Baseline	Subtask 1	Recall@1	0.083	0.008	0.008
		Recall@10	0.359	0.102	0.094
		Recall@50	0.794	0.542	0.498
		MRR	0.175	0.053	0.048
Our system	Subtask 1	Recall@1	**0.469**	**0.326**	**0.338**
		Recall@10	**0.756**	**0.668**	**0.646**
		Recall@50	**0.947**	**0.922**	**0.932**
		MRR	**0.573**	**0.449**	**0.440**
	Subtask 3	Recall@1		**0.212**	**0.176**
		Recall@10		**0.586**	**0.57**
		Recall@50	NA	**0.906**	**0.926**
		MRR		**0.338**	**0.297**
		MAP		**0.37**	**0.343**
	Subtask 4	Recall@1	**0.388**	**0.088**	**0.066**
		Recall@10	**0.592**	**0.31**	**0.316**
		Recall@50	**0.751**	**0.618**	**0.686**
		MRR	**0.462**	**0.163**	**0.15**
	Subtask 5	Recall@1	**0.451**	**0.282**	**0.301**
		Recall@10	**0.742**	**0.558**	**0.593**
		Recall@50	**0.926**	**0.876**	**0.902**
		MRR	**0.550**	**0.379**	**0.393**

Table 5.10 – Experimental results on the test sets of Subtasks 1, 3, 4 and 5.

5.5.5.1 Results

We used the scripts [5] provided by the organizers to evaluate the baseline system on the test set [6]. We also report the results of our system produced by the task organizers.

5. https://github.com/IBM/dstc7-noesis/tree/master/noesis-tf
6. We used the hyper-parameters defined by the organizers.

Table 5.10 summarizes these results on the four subtasks. Note that two test sets were provided for the Advising Corpus noted as `case 1` and `case 2`. As we can see, our system outperforms the provided baseline system on all the metrics with a good margin. These results confirm the effectiveness of matching the context and the response on the word-level and the sequence-level and using a shared encoder instead of different encoders for the context and the response. Also, we observe that the performance of our system in addition to the baseline system on the Advising Corpus are lower than the performance on the Ubuntu Dialogue Corpus (V3).

	Train	**Dev**	**Test**	
			Case 1	**Case 2**
Ubuntu	20%	20%	20.20%	-
Advising	20.05%	18.80%	23.40%	18.40%

Table 5.11 – Percentage of cases where no correct response is provided in the candidate set (Subtask 4).

The performance of our system on Subtask 3 are lower than Subtask 1 on all the metrics. We believe that this result is logic as the system is challenged to retrieve all the correct responses which is harder than retrieving only one correct response (as in Subtask 1). The results of subtask 4 are quite lower than expected. We analyzed the subtask datasets and found that the SVM classifier is hard to train because of the unbalanced data. As mentioned in Table 5.11, the percentages of training samples where no correct response is available in the candidate set are 20% and 20.05% in the case of Ubuntu and Advising datasets respectively. At the training step, the system will see 80% of dialogues with a correct response and thus will tend to generalize and predict a correct response most of the time. Applying some data balancing techniques may solve this problem.

After incorporation of the external knowledge as required by Subtask 5, the performance of our system did not improve. The results of Subtask 1 and 5 are comparable as they use the same datasets. As we can see in Table 5.10, the results of Subtask 5 are lower than Subtask 1 on both datasets. We believe that this is mainly due to the new word embeddings that we computed on the external data. When we used the word embeddings produced from the training data as in Subtask 1, we were able to find 89,284 and 4,534 word embedding vectors for the training data of the Ubuntu Dialogue Corpus (V3) and the Advising Corpus respectively. However, when we use the word embeddings produced from the external data, only 23,910 and 2,350 word vectors were found. This explains the drop in the system performance as more words (whose word vectors were not found) will have randomly initialized embedding vectors.

A list of scores of the 20 participating teams to the *sentence selection* track is given in Table 5.12. As we can see in Figure 5.4, the systems of teams 2, 3 and 4 ranked at the first positions are based on the ESIM framework (Chen et al., 2018) basically proposed for Natural Language Inference. Almost all the first systems are based on self and cross-attention mechanisms and use data augmentation during training to increase the number of positive samples as we discuss in Section 5.5.5.3. Systems like those of teams 17, 18

	Ubuntu, Subtask				Advising, Subtask			
Team	1	2	4	5	1	3	4	5
3	**0.819**	0.145	**0.842**	**0.822**	**0.485**	**0.592**	**0.537**	**0.485**
4	0.772	-	-	-	0.451	-	-	-
17	0.705	-	-	0.722	0.434	-	-	0.461
13	0.729	-	0.736	0.635	0.458	0.461	0.474	0.390
2	0.672	0.033	0.713	0.672	0.430	0.540	0.479	0.430
10	0.651	**0.307**	0.696	0.693	0.361	0.434	0.262	0.361
18	0.690	0.000	0.721	0.710	0.287	0.380	0.398	0.326
8	0.641	-	0.527	0.646	0.310	0.433	0.233	0.301
16	0.629	0.000	0.683	-	0.280	-	0.370	-
15	0.473	-	-	0.478	0.300	-	-	0.236
7	0.525	-	0.411	-	-	-	-	-
11	-	-	-	-	0.075	0.232	?	-
12	0.077	-	0.000	0.077	0.075	0.232	0.000	0.075
1	0.580	-	-	-	0.239	-	-	-
6	-	-	-	-	0.245	-	-	-
9	0.482	-	-	-	-	-	-	-
14	0.008	-	0.072	-	-	-	-	-
19	0.265	-	-	-	0.180	-	-	-
5	0.076	-	-	-	-	-	-	-
20	0.002	-	-	-	0.004	-	-	-

Table 5.12 – Track 1 results, ordered by the average rank of each team across the subtasks they participated in. The top result in each column is in bold. For these results the metric is the average of MRR and Recall@10 (Gunasekara et al., 2019). We participated as team number 8.

and 13 stack many neural network systems or use ensemble systems which results in more complex architectures. Compared to these systems, our system is simpler and at the time of the submission, no data augmentation technique was used. We show later, that when we augment the training set with more positive samples to balance the ratio of positive and negative samples, our system would be ranked 5^{th} on the Ubuntu corpus.

5.5.5.2 System ablation

As mentioned in previous sections, we incorporated word-level similarity to a slightly improved dual encoder (Lowe et al., 2015b). To evaluate the impact of these modifications, we performed an ablation study in which we kept only sequence-level similarity. Table 5.13 summarizes the results of this study on the validation sets of Subtask 1. As we can see, the best results were achieved by having both similarity levels which validates our hypothesis that the correct responses are those that match with the context on the sequence-level and word-level. Moreover, when considering both similarities separately, we notice that matching the context and the candidate response on the word-level is better than matching them on only the sequence-level. These results mean that explicitly considering the words separately are more meaningful than considering them implicitly when encoding the sequence and provide a fine-grained representation of the context and the response. Based on these

Team	Model Type	External Data Use	Used Raw Advising	Train Data	Model Details
1	CNN	-	No	Train+Val	Combination of CNN for utterance representation and GRU for modeling the dialogue.
2	LSTM	-	Yes	Train	ESIM with an aggregation scheme that captures the dialog-specific aspects of the data + ELMo.
3	LSTM	Embeddings	Yes	Train	ESIM plus a filtering stage for subtask 2.
4	LSTM	-	No	Train	ESIM with (1) enhanced word embeddings to address OOV issues, (2) an attentive hierarchical recurrent encoder, and (3) an additional layer before the softmax.
6	Ensemble	-	No	Train	An ensemble of CNNs.
7	LSTM	-	No	Train+Val	LSTM representation of utterances followed by a convolutional layer.
8	Other	-	Yes	Train	A multi-level retrieval-based approach that aggregates similarity measures between the context and the candidate response on the sequence and word levels.
10	LSTM	TF-IDF Extraction	No	Train	ESIM with matching against similar dialogues in training, and an extra filtering step for subtask 2.
12	RNN	TF-IDF Extraction	No	Train	BoW over ELMo with context as an RNN.
13	Ensemble	Embeddings	No	Train	Ensemble approach, combining a Dynamic-Pooling LSTM, a Recurrent Transformer and a Hierarchical LSTM.
14	Ensemble	-	No	Train	An ensemble using voting, combining the baseline LSTM, a GRU variant, Doc2Vec, TF-IDF, and LSI.
15	Memory	Memory	No	Train	Memory network with an LSTM cell.
16	LSTM	-	No	Train	ESIM with utterance-level attention, plus additional features.
17	Memory	Memory & Embeddings	Yes	Train	Self-attentive memory network, with external advising data in memory and external ubuntu data for embedding training.
18	GRU	-	No	Train	Stacked Bi-GRU network with attention, aggreagting attention across the temporal dimension followed by a CNN and softmax.
19	LSTM	-	No	Train+Val	Bidirectional LSTM memory network.
20	CNN	-	No	Train+Val	CNN with attention and a pointer network, plus a novel top-k attention mechanism.

Figure 5.4 – Summary of approaches used by participants. All teams applied neural approaches, with ESIM (Chen et al., 2018) being a particularly popular basis for system development. External data refers to the man pages for Ubuntu, and course information for Advising. Raw advising refers to the variant of the training data in which the complete dialogues and paraphrase sets are provided. Three teams (5, 9 and 11) did not provide descriptions of their approaches (Gunasekara et al., 2019).

results, we can deduce two points. (1) The modification of the dual encoder with only sequence similarity results in better performance compared to the baseline (the original dual encoder). (2) Having word similarity in addition to sequence similarity can help the system to perform a better matching between the context and the correct responses. These results correlate perfectly with our previous experiments on the UDC (V1) and the Douban Conversation Corpus.

5.5.5.3 Data Augmentation

The training set of Subtask 1 as illustrated in Table 5.9 is imbalanced. For each training sample, we have one positive response and 99 negative responses. Thus 99% of the training

			Ubuntu	Advising
	Baseline	R@1	0.083	0.062
		R@10	0.359	0.296
		R@50	0.800	0.728
		MRR	-	-
Our system	**Only sequence similarity**	R@1	0.206	0.084
		R@10	0.567	0.404
		R@50	0.885	0.791
		MRR	0.350	0.186
	Only word similarity	R@1	0.41	0.104
		R@10	0.697	0.418
		R@50	0.936	0.804
		MRR	0.512	0.209
	Word + sequence similarities	R@1	**0.463**	**0.116**
		R@10	**0.753**	**0.444**
		R@50	**0.945**	**0.848**
		MRR	**0.57**	**0.219**

Table 5.13 – Ablation results on the validation data of Subtask 1.

samples are negative while only 1% are positive. As we define the problem of response retrieval in dialogue systems as a classification problem, this imbalanced data will alter the training process. More specifically, our system will "see" more negative samples than positive ones and thus, it will tend to predict label 0 for most of the input samples. In the literature, different approaches and tricks of data balancing exist (He and Ma, 2013). We adopt a data augmentation approach to solve this problem and also to increase the number of training samples.

Each of the training samples is composed of a context, a response, and a label. The context is composed of multiple turns $t_1, t_2, \ldots, t_n$. Hence, we construct new positive samples starting from the second turn by concatenating the previous turns t_i and considering them as a new context and the turn t_j as the response with a label 1. By applying this data augmentation approach to the datasets of Subtask 1, we were able to obtain 10,349,002 and 10,727,467 training samples for the Ubuntu Dialogue Corpus (V3) and the Advising Corpus respectively which represents an increase of 3.49% and 7.27% respectively of the total number of samples. Even if the training data remains unbalanced, we show in Table 5.14 that with this small increase in the number of positive samples, the performance of our system increased considerably.

	Ubuntu Dialogue Corpus (V3)				Advising Corpus (case 1)			
	R@1	R@10	R@50	MRR	R@1	R@10	R@50	MRR
Our system	0.449	0.756	0.947	0.573	0.114	0.398	0.782	0.205
Our system + data augmentation	0.526	0.786	0.959	0.619	0.108	0.578	0.908	0.254

Table 5.14 – Results of our system after application of data-augmentation on the training sets of Subtask 1.

With this simple approach of data augmentation and while keeping the same parameters

of our system, we were able to improve Recall@(1, 10, 50) and MRR by 8%, 3%, 1%, and 4% respectively on the Ubuntu Dialogue Corpus (V3). This result sheds the light on the importance of having a balanced training set which helps the system to perform a better learning.

Data augmentation has been used as a solution to the data insufficiency problem in multiple domains such as computer vision (Krizhevsky et al., 2012), speech recognition (Hannun et al., 2014), question answering (Fader et al., 2013), and text classification (Zhang et al., 2015). Few researchers applied data augmentation techniques on dialogue systems. For instance, Kurata et al. (2016) introduced an LSTM encoder-decoder with random noise to generate more training data for the slot filling task. (Hou et al., 2018b) combined sequence-to-sequence and diversity rank to generate more diverse utterances in the training data for the task of Dialogue Language Understanding (DLU). A more recent work combines Conditional Variational Autoencoder (CVAE) and Generative Adversarial Network (GAN) (Goodfellow et al., 2014) to generate more diverse query-response pairs Li et al. (2019). These techniques are more complex than the data augmentation technique that we used which does not require training deep neural networks which requires itself large amount of training data. However, given the promising results that we obtained after augmenting the training data with a simple method, we can think of more elaborated techniques to achieve better performance.

5.6 Conclusion

We extended our single-turn sing-level matching system that we presented in the previous chapter 4 by adding an extra similarity level between the context and the candidate response to obtain a simple and efficient multi-level retrieval-based dialogue system. Our system learns to match the context with the best response based on their similarity that we capture on the word-level and sequence-level. By learning a word-level and sequence-level similarities our system was able to capture deep relationships between the context and the candidate responses. The experimental results on two large datasets (the UDC V1 and Douban Conversation Corpus) demonstrate the efficiency of our approach by bringing significant improvements compared to single- and multi-turn state-of-the-art systems.

In essence, a simple model can suffice to achieve good performance, sometimes even better than complex response matching models. For further analysis, we performed an ablation study in which we removed the similarity levels one by one. We deduced that both similarity levels are complementary and important to help the system in retrieving the correct responses of a given context. This deduction was consolidated by the visualization of the Word-Level Similarity Matrix where we important words were identified and common word between the context and the response were successfully matched. Furthermore, we performed an error analysis to understand the reasons why our system failed on some test samples and we found that half of the errors were due to the method of construction of the corpus in which negative responses where randomly sampled from the dataset. Another part of the errors were due to general and out of context responses which has been reported as

weakness in generative dialogue systems.

To evaluate our system furthermore, we participated with in the sentence selection track of 7^{th} edition of the Dialog System Technology Challenge and demonstrate the efficiency of our approach on two more datasets compared to the baseline system. DSTC7 provided a new and challenging environment for retrieval-based dialogue system and pushed researchers towards dealing with more realistic constraints. It was an opportunity to evaluate our system in a more challenging environment where we faced real problems such as the possibility of not having a correct response in the pool of candidate responses and the obligation of incorporating external knowledge. The system described in this chapter has been the subject of two publications at the international conference on Computational Linguistics and Intelligent Text Processing (CICLING 2019) (Boussaha et al., 2019c), the Dialog System Technology Challenges (DSTC7) workshop collocated with AAAI 2019 (Boussaha et al., 2019b), a submission of a journal paper at the special issue on DSTC7 at the Computer Speech and Language (CS&L) journal and a short survey paper on deep retrieval-based dialogue systems (Boussaha et al., 2019a) to be submitted.

6

Conclusion and perspectives

> “The task is . . . not so much to see what no one has yet seen; but to think what nobody has yet thought, about that which everybody sees.”
>
> — Erwin Schrodinger

6.1 Conclusion

In this thesis, we presented works related to dialogue systems that lead to the construction of two different retrieval-based dialogue systems. First, by studying the existing systems of both categories retrieval-based and generative, we were able to identify some issues with the existing systems that we need to address. We focused on retrieval-based dialogue systems as they are more adequate to discuss specific subjects and provide the necessary assistance to humans. We grouped the existing retrieval-based systems into two categories: single-turn and multi-turn matching systems. The architecture of single-turn matching systems was quite simple as some of them match the whole context (considered as one turn) with the candidate response only one time. Some of the systems of this category have domain dependency as they incorporate external knowledge. On the other hand, multi-turn matching systems perform a complex matching between the response and each turn of the context in addition to an aggregation of all the matching scores. Even more, some systems perform the matching on two levels which make the architecture more complex with

multiple operations of matching and aggregation.

After a deep study of the existing approaches, we opted for single-turn matching systems. We were inspired by the dual encoder (Lowe et al., 2015b) for the simplicity of its approach and its efficiency. We analyzed its approach and identified some drawbacks that we addressed through a new architecture. We proposed a single-turn and single-level matching system that matches the context with the candidate response on the sequence level. We used a shared LSTM encoder to encode both the whole context and the candidate response and project them into the same space in contrast to what (Lowe et al., 2015b) did in their dual encoder. Once the context and the response are encoded as vectors of a similar dimension, we hypothesize that correct responses are semantically similar to the context. Therefore, we measure the similarity between the context and the candidate response as the cross product between their respective vectors. By doing so, we eliminated the additional parameters matrix originally used in the dual encoder when computing the similarity between the two vectors. We evaluated our system on two widely used datasets in two different languages: the Ubuntu Dialogue Corpus and the Douban Conversations Corpus and compared our system to systems of the same category (single-turn matching systems). The results showed that by incorporating the described changes, we were able to improve the Recall@(1,2 and 5) by 5%, 4% and 1% respectively on UDC and outperform the other single-turn matching systems to which our system was compared. Moreover, we performed an error analysis and we studied qualitatively the predictions of our system and the dual encoder and interesting results were obtained. It turns out that, despite the higher scores that we obtained, our system does not bring only improvements. There were around 23% of the test samples where the dual encoder successfully retrieved the correct response while our system failed.

The encouraging results that we obtained with the single-turn matching system pushed us to look for further possible improvements. LSTMs are known to have difficulties in modeling long input sequences because of the vanishing gradient problem. In our case, the context of conversations is quite long as it contains multiple utterances. Therefore, we believe that the LSTM encoder suffers from this constraint and thus by improving either the input representation or matching levels can help to improve the system. The attention mechanism is a widely used solution to help neural networks in paying attention to some parts of the input that are the most relevant to the task. We incorporated the self-attention mechanism into our system as it is the most adapted attention form to our architecture (absence of the decoder). We noticed a slight improvement of the system's performance. Furthermore, we visualized of the distribution of the attention weights of some test examples and we found out that the weights perfectly correlate with the importance of words.

To help our system performing a better matching between the context and the candidate responses, we introduced another level of similarity to our single-turn and single-level system. This level of similarity could be seen as a manual attention. It consists of a matrix of pointwise product between the word embeddings of the context and the responses. Each cell of the resulting matrix represents the similarity between a word embedding from the context and another from the response. Words with high similarities give cells with high value. This process looks like the cross-attention mechanism as higher values mean higher importance as well as the attention weights and the cell values are both fine-tuned during

the training process. Although, they are different in the way that the attention weights are randomly initialized. We incorporate this matrix as an additional word similarity level to the existing sequence similarity.

This extension does not slow down the system as it is computed in parallel to the sequence encoding and helps the system identifying the common words between the inputs so that the responses that have more similar words and that is similar itself as a whole to the context is chosen. We evaluated this new system in the same test configuration and on the same datasets as our previous system. Two important results were obtained. First of all, compared to our previous system with only the sequence-level similarity, we improved Recall@1 by 8% and 5% on UDC and Douban Conversation Corpus respectively. Moreover, our system outperformed all the systems of the same category (single-turn matching systems) with a good margin. Not only our system outperformed single-turn matching systems but also outperformed several multi-turn matching systems while having a simpler approach and by matching the response with the whole context without requiring complex matching and aggregation mechanisms. It obtained almost the same results as the two recently built systems (DAM Zhou et al. (2018) and DUA (Zhang et al., 2018c)) without requiring neither attention (Transformer which needs high computing resources) nor multi-turn matching.

The visualization of the Word Level Similarity Matrix (WLSM) confirms the hypothesis that we made at the beginning by observing that similar words from the context and the candidate response were successfully matched and were assigned higher scores. As for the ablation study, the importance of each similarity level was evaluated and we concluded that both similarity levels are complementary and are necessary for more efficient retrieval of the correct response. Moreover, the error analysis revealed important information about the necessity of performing a complete comparative study between retrieval-based and generative dialogue systems as we observed that our system suffers also from general responses. We found also that some responses that were labeled as negative responses in the datasets were potential correct responses. This interesting result shed the light on the weakness of the approach of negative sampling widely used in the construction of dialogue datasets as we mentioned in Chapter 3. The 7^{th} edition of the Dialog System Technology Challenges (DSTC7) provided a perfect evaluation platform to test our system in new and more realistic scenarios. We achieved good results on different subtasks of the challenge and we obtained insights on the enhancements that we can bring to our system.

6.2 Perspectives

Due to time constraints, some research tracks were not or were partially addressed during this thesis. Some of them were identified as a results of our deep analysis. We highlight them in the following and hopefully, they can be investigated to come up with interesting results.

- **Towards simple, efficient and reproducible approaches.** From a general point of

view, during the course of this thesis, we highlight the importance of building simple but efficient architectures. With the actual success of deep learning approaches and the availability of computing resources, researchers tend to build complex architectures with multiple layers and attention mechanisms more often. However, we encourage researchers to consider the cost of efficient architectures and the time necessary to produce the same results for the sake of reproducibility.

- **Explicit modeling of the dialogue coherence.** The coherence of a dialogue is an important concept as when people discuss, they tend to keep coherent conversations. What distinguishes coherent text from random sequences of text is the semantic relations that exist between the sentences that make coherent text appears to be logically and semantically consistent for the reader/hearer. Many recent works were interested in evaluating the coherence of synchronous and asynchronous conversations (Tien Nguyen and Joty, 2017; Joty et al., 2018). But unfortunately, very few works were interested in trying to consider coherence in the dialogues to force the systems to produce coherent responses (Xu et al., 2018). Coherence could be included in the existing retrieval-based dialogue systems as a form of another level of similarity for example that measures the coherence of the candidate response and the context and therefore forces the system to pick the most coherent responses.
- **Generative vs. retrieval-based dialogue systems.** To the best of our knowledge, there is no comparative study of generative and retrieval-based dialogue systems. We generally say that generative systems produce general and short responses while retrieval-based systems can produce more specific responses. However, after the error analysis that we performed in Section 5.4.2, we found that our retrieval-based system retrieved very general responses in almost 30% of the times. These results invite researchers to perform a deep comparative study between these two categories. Maybe one difficulty that may face this kind of study is the absence of common evaluation metrics between generative and retrieval-based dialogue systems. However, we believe that much work has to be done in this perspective so that the decision of what category to choose for a given problem will be more motivated.
- **Appropriate metrics for dialogue systems.** We believe that much work needs to be done on the evaluation of dialogue systems as human evaluation is still widely used today and the currently used metrics are borrowed from different domains and which are not very suitable to either generative or retrieval-based dialogue systems.
- **More memory and more intelligence.** Overall, we believe that today, the dialogue systems are far from being perfect and this is due to many factors such as the difficulty of understanding the human language, the large number of possibilities of expressing the same idea, multiple subjects can be approached during the same conversation, etc. The current dialogue systems, even if some of them can succeed in the Turing test, they still lack intelligence, robustness, and flexibility. To overcome today's limitations, we believe that we need to improve two features of dialogue systems which are memory and intelligence. We humans beings can store millions of information in our memory and we can rapidly and efficiently recover it when needed. We are smart enough to understand the hidden meaning of words and the implicit relationships between different concepts. We argue that if we can endow dialogue systems with more memory and intelligence, we will achieve better performance.

- **Building more human-labeled datasets.** From the results of the error analyses that we performed, we noticed the importance of having datasets where wrong responses are carefully selected and annotated. We believe that randomly selecting responses and labeling them as negative responses may falsify the training and the evaluation of dialogue systems. Some wrong responses may be potential responses to the context. On the other hand, building human-labeled datasets is a very time consuming and expensive task, but we believe that it is mandatory. We can think of a computer-labeled dialogue datasets where we can make the annotation process automatic as it has been done in other domains such as machine translation (Turchi and Negri, 2014). This can be seen as a method to alleviate the subjectivity of human judgments and the noise and biases that it adds to annotations.
- **Semi or fully unsupervised dialogue systems.** Most of the retrieval-based dialogue systems that we reviewed in the course of this thesis are based on supervised approaches that require a large amount of labeled data to be trained. However, collecting labeled data is very time consuming and expensive. Recently, few effort has been made to build deep generative dialogue systems based on reinforcement learning (Li et al., 2016c; Serban et al., 2017b; Cuayáhuitl et al., 2019) and interesting performance have been achieved. Nevertheless, to the best of our knowledge, there is no similar work in the category of retrieval-based dialogue systems. We encourage researchers to guide some work towards building semi-supervised or completely unsupervised retrieval-based dialogue systems to reduce the dependency of these systems to labeled data.

List of publications

Publications in international conferences

Next Utterance Ranking Based on Context Response Similarity
Basma El Amel Boussaha, Nicolas Hernandez, Christine Jacquin and Emmanuel Morin
(Boussaha et al., 2018a)

Abstract : Building dialogue systems that converse with humans in order to help them in their daily tasks is being a priority. Some systems converse by generating dialogues word byword whereas others retrieve the best utterance among a set of candidate responses. These retrieval systems rank the candidate responses by their relevance to the history of the conversation(context), the best response is then chosen. Approaches based on deep neural networks performed well on this task. In this work,we improve a state of the art approach based on an LSTM dual encoder and propose a new response retrieval dialogue system. Based on syntactic and semantic similarities between the context and the response extracted from word embeddings, our approach learns to match the context with the best response. Experimental results on the Ubuntu Dialogue Corpus show an important improvement of about 7%, 6% and 2% on Recall@(1,2 and 5) compared to the best state of the art system.

Published in the 5^{th} Machine Learning and Data Analytics Symposium (MLDAS).

Multi-level Context Response Matching in Retrieval-Based Dialog Systems
Basma El Amel Boussaha, Nicolas Hernandez, Christine Jacquin and Emmanuel Morin
(Boussaha et al., 2019b)

Abstract: We present our work on the Dialog System Technology Challenges 7 (DSTC7). We participated in Track 1 on sentence selection which evaluates response retrieving in dialog systems on more realistic test scenarios compared to the state-of-the-art evaluations.

Our proposed dialog system matches the context with the best response by computing their semantic similarity on the word and sequence levels. Evaluation results on the provided datasets show the effectiveness of our system by achieving higher performance compared to the provided baseline system. Our system enjoys the advantages of its simple and end-to-end architecture making its training and adaptation to other domains easier.

Published in the 7^{th} Dialog System Technology Challenges workshop at the 33 rd conference on Artificial Intelligence (AAAI), (DSTC7).

Towards Simple but Efficient Next Utterance Ranking
Basma El Amel Boussaha, Nicolas Hernandez, Christine Jacquin and Emmanuel Morin (Boussaha et al., 2019c)

Abstract: Retrieval-based dialogue systems converse with humans by ranking candidate responses according to their relevance to the history of the conversation (context). Recent studies either match the context with the response on only sequence level or use complex architectures to match them on the word and sequence levels. We show that both information levels are important and that a simple architecture can capture them effectively. We propose an end-to-end multi-level response retrieval dialogue system. Our model learns to match the context with the best response by computing their semantic similarity on the word and sequence levels. Empirical evaluation on two dialogue datasets shows that our model outperforms several state-of-the-art systems and performs as good as the best system while being conceptually simpler.

Published in the 20^{th} International Conference on Computational Linguistics and Intelligent Text Processing, (CICLING).

Publications in national conferences

Ordonnancement de réponses dans les systèmes de dialogue basé sur une similarité contexte/réponse
Basma El Amel Boussaha, Nicolas Hernandez, Christine Jacquin and Emmanuel Morin (Boussaha et al., 2018b)

Résumé : Construire des systèmes de dialogue qui conversent avec les humains afin de les aider dans leurs tâches quotidiennes est devenu une priorité. Certains de ces systèmes

produisent des dialogues en cherchant le meilleur énoncé (réponse) parmi un ensemble d'énoncés candidats. Le choix de la réponse est conditionné par l'historique de la conversation appelé contexte. Ces systèmes ordonnent les énoncés candidats par leur adéquation au contexte, le meilleur est ensuite choisi. Les approches existantes à base de réseaux de neurones profonds sont performantes pour cette tâche. Dans cet article,nous améliorons une approche état de l'art à base d'un dual encodeur LSTM. En se basant sur la similarité sémantique entre le contexte et la réponse, notre approche apprend à mieux distinguer les bonnes réponses des mauvaises. Les résultats expérimentaux sur un large corpus de chats d'Ubuntu montrent une amélioration significative de 7, 6 et 2 points sur le Rappel@(1, 2 et 5) respectivement par rapport au meilleur système état de l'art.

Published in the 25^{th} conférence sur le Traitement Automatique du Langage Naturel (TALN).

Publications in international journals

End-to-End Response Selection Based on Multi-Level Context Response Matching
Basma El Amel Boussaha, Nicolas Hernandez, Christine Jacquin and Emmanuel Morin

Abstract : We present our work on the Dialog System Technology Challenges 7 (DSTC7). We participated in Track 1 on sentence selection which evaluates response retrieving in dialog systems on more realistic test scenarios compared to the state-of-the-art evaluations. Our proposed dialog system matches the context with the best response by computing their semantic similarity on the word and sequence levels. Evaluation results on the provided datasets show the effectiveness of our system by achieving higher performance compared to the provided baseline system. Our system enjoys the advantages of its simple and end-to-end architecture making its training and adaptation to other domains easier.

Submitted to the special issue of the Computer and Speech Language journal on the 7^{th} Dialog System Technology Challenge (CS&L).

Preprints

Deep Retrieval-Based Dialogue Systems: A Short Review
Basma El Amel Boussaha, Nicolas Hernandez, Christine Jacquin and Emmanuel Morin
(Boussaha et al., 2019a)

Abstract : Building dialogue systems that naturally converse with humans is being an attractive and an active research domain. Multiple systems are being designed everyday and several datasets are being available. For this reason, it is being hard to keep an up-to-date state-of-the-art. In this work, we present the latest and most relevant retrieval-based dialogue systems and the available datasets used to build and evaluate them. We discuss their limitations and provide insights and guidelines for future work.

Published on Arxiv https://arxiv.org/abs/1907.12878

Additional publications

During the course of this Ph.D., the following papers have been published, but are not included in this thesis as they are not directly connected to the research topic.

— **A Multi-Domain Framework for Textual Similarity. A Case Study on Question-to-Question and Question-Answering Similarity Tasks** Amir Hazem, Basma El Amel Boussaha, Nicolas Hernandez. In proceedings of Language Resources and Evaluation Conference (LREC), 2018.

— **'MappSent': a Textual Mapping Approach for Question-to-Question Similarity** Amir Hazem, Basma El Amel Boussaha, Nicolas Hernandez. In proceedings of Recent Advances in Natural Language Processing (RANLP), 2017.

— **Thread Reconstruction in Conversational Data using Neural Coherence Models** Dat Tien Nguyen, Shafiq Joty, Basma El Amel Boussaha, Maarten de Rijke. In proceedings of SIGIR Workshop on Neural Information Retrieval (Neu-IR), 2017.

— **Exploitation des plongements de mots pour l'analyse d'opinion et du langage** Amir Hazem, Basma El Amel Boussaha, Nicolas Hernandez. In procc dings ofConeférence sur le Traitement Automatique des Langues Naturelles (TALN), 2017.

Bibliography

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org. 68

Amstelveen (2017 (accessed June 30, 2019)). *KLM welcomes BlueBot (BB) to its service family*. 14

Baeza-Yates, R. A. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA. 90

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference for Learning Representations (ICLR'15)*, San Diego, CA, USA. 27, 30, 32, 38, 64, 67, 72

Banchs, R. E. (2012). Movie-DiC: a movie dialogue corpus for research and development. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL'12)*, pages 203–207, Jeju Island, Korea. 50

Banchs, R. E. and Li, H. (2012). IRIS: a chat-oriented dialogue system based on the vector space model. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics System Demonstrations (ACL'12)*, pages 37–42, Jeju Island, Korea. 29

Banerjee, S. and Lavie, A. (2005). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, MI, USA. 41

Baudiš, P., Pichl, J., Vyskočil, T., and Šedivỳ, J. (2016). Sentence pair scoring: Towards unified framework for text comprehension. *arXiv preprint arXiv:1603.06127*. 86

Ben-Hur, A., Horn, D., Siegelmann, H. T., and Vapnik, V. (2001). Support vector clustering. *Journal of machine learning research*, 2(Dec):125–137. 88

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166. 26

Bennett, C. and Rudnicky, A. I. (2002). The carnegie mellon communicator corpus. In *Seventh International Conference on Spoken Language Processing*. 50

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022. 27

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146. 81, 89, 90

Bordes, A., Glorot, X., Weston, J., and Bengio, Y. (2014). A semantic matching energy function for learning with multi-relational data. *Machine Learning*, 94(2):233–259. 27

Boussaha, B. E. A., Hernandez, N., Jacquin, C., and Morin, E. (2018a). Next utterance ranking based on context response similarity. In *Proceedings of the Machine Learning and Data Analytics Symposium (MLDAS'18)*, Doha, Qatar. 18, 75, 105

Boussaha, B. E. A., Hernandez, N., Jacquin, C., and Morin, E. (2018b). Ordonnancement de réponses dans les systèmes de dialogue basé sur une similarité contexte/réponse. In *Proeedings of the conférence sur le Traitement Automatique de la Langue Naturelle (TALN'2018)*, pages 115–128, Rennes, France. 18, 71, 72, 75, 106

Boussaha, B. E. A., Hernandez, N., Jacquin, C., and Morin, E. (2019a). Deep retrieval-based dialogue systems: A short review. In *arXiv preprint arXiv:1907.12878*. 97, 107

Boussaha, B. E. A., Hernandez, N., Jacquin, C., and Morin, E. (2019b). Multi-level context response matching in retrieval-based dialog systems. In *Proceedings of 7th edition of DSTC workshop at AAAI'19 (DSTC7)*, Honolulu, HI, USA. 18, 97, 105

Boussaha, B. E. A., Hernandez, N., Jacquin, C., and Morin, E. (2019c). Towards simple but efficient next utterance ranking. In *Proceedings of the 20th International Conference on Computational Linguistics and Intelligent Text Processing (CICLING'2019)*, La Rochelle, France. 18, 58, 97, 106

Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A., Jozefowicz, R., and Bengio, S. (2016). Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL'16)*, pages 10–21, Berlin, Germany. 46

Brill, E. (1992). A simple rule-based part of speech tagger. In *Third Conference on Applied Natural Language Processing (ANLP'92)*, pages 152–155, Trento, Italy. 36

Bruni, E. and Fernández, R. (2017). Adversarial evaluation for open-domain dialogue generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL'17)*, pages 284–288, Saarbrücken, Germany. 46

Bunt, H. (1999). Dynamic interpretation and dialogue theory. *The structure of multimodal dialogue*, 2:1–8. 23

Cambria, E. and Hussain, A. (2015). *Sentic Computing: A Common-Sense-Based Framework for Concept-Level Sentiment Analysis*. Springer Publishing Company, Incorporated, 1st edition. 36

Cao, G., Nie, J.-Y., Gao, J., and Robertson, S. (2008). Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '08)*, pages 243–250, New York, NY, USA. 32, 36

Carbonell, J. G. (1983). Discourse pragmatics and ellipsis resolution in task-oriented natural language interfaces. In *Proceedings of the 21st Annual Meeting on Association for Computational Linguistics (ACL'83)*, pages 164–168, Cambridge, MA, USA. 62

Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka, Jr., E. R., and Mitchell, T. M. (2010). Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI'10)*, pages 1306–1313. 36

Carrara, N., Laroche, R., and Pietquin, O. (2017). Online learning and transfer for user adaptation in dialogue systems. In *SIGDIAL/SEMDIAL joint special session on negotiation dialog 2017*, Saarbrücken, Germany. 49

Chen, H., Liu, X., Yin, D., and Tang, J. (2017). A survey on dialogue systems: Recent advances and new frontiers. *SIGKDD Explor. Newsl.*, 19(2). 24, 33

Chen, Q. and Wang, W. (2019a). Sequential attention-based network for noetic end-to-end response selection. In *Proceedings of the 7 th Dialog System Technology Challenge (DSTC7)*, Honolulu, HI, USA. 31, 63

Chen, Q. and Wang, W. (2019b). Sequential matching model for end-to-end multi-turn response selection. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'19)*, pages 7350–7354, Brighton, UK. 31

Chen, Q., Zhu, X., Ling, Z.-H., Inkpen, D., and Wei, S. (2018). Neural natural language inference models enhanced with external knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL'18)*, pages 2406–2417, Melbourne, Australia. 12, 31, 92, 94

Cheng, J., Dong, L., and Lapata, M. (2016). Long short-term memory-networks for machine reading. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP'16)*, pages 551–561, Austin, TX, USA. 41

Cheng, M., Wei, W., and Hsieh, C.-J. (2019). Evaluating and enhancing the robustness of dialogue systems: A case study on a negotiation agent. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL'19)*, pages 3325–3335, Minneapolis, MN, USA. 46

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*, pages 1724–1734, Doha, Qatar. 25, 26, 38

Chollet, F. et al. (2015). Keras. `https://github.com/keras-team/keras`. 68, 81, 90

Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., and Bengio, Y. (2015). Attention-based models for speech recognition. In *Advances in neural information processing systems (NIPS'15)*, Montreal, Canada. 38

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Workshop on Deep Learning and Representation Learning at the 28th Annual conference on Advances in Neural Information Processing Systems (NIPS'14)*, Montreal, Canada. 32, 81

Clark, H. H. (1996). *Using Language*. 'Using' Linguistic Books. Cambridge University Press. 22, 23

Colby, K. M. (1975). *Artificial Paranoia: A Computer Simulation of Paranoid Processes*. Elsevier Science Inc., New York, NY, USA. 15

Cuayáhuitl, H. and Dethlefs, N. (2012). Dialogue systems using online learning: Beyond empirical methods. In *NAACL-HLT Workshop on Future directions and needs in the Spoken Dialog Community: Tools and Data (SDCTD'12)*, pages 7–8, Montréal, Canada. 49

Cuayáhuitl, H., Lee, D., Ryu, S., Cho, Y., Choi, S., Indurthi, S., Yu, S., Choi, H., Hwang, I., and Kim, J. (2019). Ensemble-based deep reinforcement learning for chatbots. *Neurocomputing*. 103

Danescu-Niculescu-Mizil, C. and Lee, L. (2011). Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the 2nd Workshop on Cognitive Modeling and Computational Linguistics*, pages 76–87, Portland, OR, USA. 50

Deriu, J., Rodrigo, A., Otegi, A., Echegoyen, G., Rosset, S., Agirre, E., and Cieliebak, M. (2019). Survey on evaluation methods for dialogue systems. *arXiv preprint arXiv:1905.04071*. 44, 46

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL'19)*, pages 4171–4186, Minneapolis, MN, USA. 46

Dhingra, B., Li, L., Li, X., Gao, J., Chen, Y.-N., Ahmed, F., and Deng, L. (2017). Towards end-to-end reinforcement learning of dialogue agents for information access. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL'17)*, pages 484–495, Vancouver, Canada. 49

Dong, Y., Su, H., Zhu, J., and Bao, F. (2019). Towards interpretable deep neural networks by leveraging adversarial examples. In *Proceedings of the AAAI-19 Workshop on Network Interpretability for Deep Learning (AAAI'19)*, Honolulu, HI, USA. 16

Doshi-Velez, F. and Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*. 16

Dušek, O. and Jurčíček, F. (2016). Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL'16)*, pages 45–51, Berlin, Germany. 24

Eck, M., Vogel, S., and Waibel, A. (2005). Low cost portability for statistical machine translation based on n-gram frequency and tf-idf. In *International Workshop on Spoken Language Translation (IWSLT'05*. 30

El Asri, L., He, J., and Suleman, K. (2016). A sequence-to-sequence model for user simulation in spoken dialogue systems. In *Proceedings of The 17th Annual Conference of the International Speech Communication Association (Interspeech'16)*, pages 1151–1155, San Francisco, CA, USA. 49

Elsner, M. and Charniak, E. (2008). You talking to me? a corpus and algorithm for conversation disentanglement. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL'08)*, pages 834–842, Columbus, OH, USA. 51

Eric, M., Krishnan, L., Charette, F., and Manning, C. D. (2017). Key-value retrieval networks for task-oriented dialogue. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL'17)*, pages 37–49, Saarbrücken, Germany. 26

Fader, A., Zettlemoyer, L., and Etzioni, O. (2013). Paraphrase-driven learning for open question answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL'13)*, pages 1608–1618, Sofia, Bulgaria. 96

Faggella, D. (2018 (accessed July 1, 2019)). *How companies are using chatbots for marketing: Use cases and inspiration*. 14

Fernández, R., Ginzburg, J., and Lappin, S. (2007). Classifying non-sentential utterances in dialogue: A machine learning approach. *Computational Linguistics*, 33(3):397–427. 62

Filippova, K. and Strube, M. (2008). Dependency tree based sentence compression. In *Proceedings of the Fifth International Natural Language Generation Conference (INLG'08)*, pages 25–32, Salt Fork, OH, USA. 29

Forgues, G., Pineau, J., Larchevêque, J.-M., and Tremblay, R. (2014). Bootstrapping dialog systems with word embeddings. In *Proceedings of the modern machine learning and natural language processing workshop at (NIPS'14)*, Montreal, Canada. 43

Freitag, M. and Al-Onaizan, Y. (2017). Beam search strategies for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation (NMT'17)*, pages 56–60, Vancouver, Canada. 38

Fuhr, N., Lalmas, M., and Trotman, A., editors (2007). *Comparative Evaluation of XML Information Retrieval Systems: 5th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2006, Dagstuhl Castle, Germany, December 17-20, 2006, Revised and Selected Papers*. Springer-Verlag, Berlin, Heidelberg. 36

Gao, J., Galley, M., and Li, L. (2018a). Neural approaches to conversational ai. In *Proceedings of ACL 2018, Tutorial Abstracts*, pages 2–7, Melbourne, Australia. Association for Computational Linguistics. 24

Gao, J., Galley, M., and Li, L. (2018b). Neural approaches to conversational ai. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '18)*, pages 1371–1374, Ann Arbor, MI, USA. 24, 25

Ghazarian, S., Wei, J., Galstyan, A., and Peng, N. (2019). Better automatic evaluation of open-domain dialogue systems with contextualized embeddings. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation at (NAACL'19)*, pages 82–89, Minneapolis, MN, USA. 9, 45, 46

Ghazvininejad, M., Brockett, C., Chang, M.-W., Dolan, B., Gao, J., Yih, W.-t., and Galley, M. (2018). A knowledge-grounded neural conversation model. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI'18)*, pages 5110–5117, New Orlean, LA, USA. 36

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. N. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, pages 2672–2680, Montreal, Canada. 96

Graesser, A. C., Lu, S., Jackson, G. T., Mitchell, H. H., Ventura, M., Olney, A., and Louwerse, M. M. (2004). Autotutor: A tutor with dialogue in natural language. *Behavior Research Methods, Instruments, & Computers*, 36(2):180–192. 29

Graves, A. (2013). Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*. 24

Gu, J., Lu, Z., Li, H., and Li, V. O. (2016). Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL'16)*, pages 1631–1640, Berlin, Germany. 27

Gunasekara, C., Kummerfeld, J., Polymenakos, L., and Lasecki, W. S. (2019). Dstc7 task 1: Noetic end-to-end response selection. In *Proceedings of the 7th Dialog System Technology Challenges DSTC7 at the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI'19)*, Honolulu, HI, USA. 10, 12, 52, 86, 87, 93, 94

Guu, K., Hashimoto, T. B., Oren, Y., and Liang, P. (2018). Generating sentences by editing prototypes. *Transactions of the Association for Computational Linguistics (TACL)*, 6:437–450. 34

Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., et al. (2014). Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*. 96

Harabagiu, S. M., Bunescu, R. C., and Maiorano, S. J. (2001). Text and knowledge mining for coreference resolution. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL'01)*, Pittsburgh, PA, USA. 36

He, H. and Ma, Y. (2013). *Imbalanced Learning: Foundations, Algorithms, and Applications*. Wiley-IEEE Press, 1st edition. 95

He, X. and Golub, D. (2016). Character-level question answering with attention. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP'16)*, pages 1598–1607, Austin, TX, USA. 38

Henderson, M., Thomson, B., and Williams, J. D. (2014). The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL'14)*, pages 263–272, Philadelphia, PA, USA. 50

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780. 30

Hofmann, K., Tsagkias, M., Meij, E., and Rijke, M. (2009). A comparative study of features for keyphrase extraction in scientific literature. In *Proceedings of the 18th ACM Conference on Information And Knowledge Management (CIKM'09)*, Hong Kong, China. 30

Hou, L., Hu, P., and Bei, C. (2018a). Abstractive document summarization via neural model with joint attention. In Huang, X., Jiang, J., Zhao, D., Feng, Y., and Hong, Y., editors, *Natural Language Processing and Chinese Computing*, pages 329–338, Cham. Springer International Publishing. 26

Hou, Y., Liu, Y., Che, W., and Liu, T. (2018b). Sequence-to-sequence data augmentation for dialogue language understanding. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, pages 1234–1245, Santa Fe, NM, USA. 96

Hu, B., Lu, Z., Li, H., and Chen, Q. (2014). Convolutional neural network architectures for matching natural language sentences. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Proceedings of 28 th international conference on Advances in Neural Information Processing Systems (NIPS'14)*, pages 2042–2050, Montreal, Canada. 27, 29

Jeon, J., Croft, W. B., and Lee, J. H. (2005). Finding similar questions in large question and answer archives. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management (CIKM '05)*, pages 84–90, Bremen, Germany. 29

Joty, S., Mohiuddin, M. T., and Tien Nguyen, D. (2018). Coherence modeling of asynchronous conversations: A neural entity grid approach. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL'18)*, pages 558–568, Melbourne, Australia. 102

Kadlec, R., Schmid, M., and Kleindienst, J. (2015). Improved deep learning baselines for ubuntu corpus dialogs. In *Workshop on Machine Learning for Spoken Language Understanding and Interaction at the 29th Annual Conference on Neural Information Processing Systems (NIPS'15)*, Montreal, Canada. 30, 43, 64, 67, 68, 82

Kannan, A. and Vinyals, O. (2016). Adversarial evaluation of dialogue models. In *NIPS 2016 Workshop on Adversarial Training*, Barcelona, Spain. 46

Kim, S., D'Haro, L. F., Banchs, R. E., Williams, J. D., Henderson, M., and Yoshino, K. (2016). The fifth dialog state tracking challenge. In *2016 IEEE Spoken Language Technology Workshop (SLT)*, pages 511–517. 50

Kim, S., D'Haro, L. F., Banchs, R. E., Williams, J. D., and Henderson, M. (2017). The fourth dialog state tracking challenge. In *Dialogues with Social Robots*, pages 435–449. Springer. 50

Kingma, D. and Ba, J. (2015). Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference for Learning Representations (ICLR'15)*, San Diego, CA, USA. 68, 81, 90

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS'12)*, pages 1097–1105, Lake Tahoe, USA. 96

Kummerfeld, J. K., Gouravajhala, S. R., Peper, J., Athreya, V., Gunasekara, C., Ganhotra, J., Patel, S. S., Polymenakos, L., and Lasecki, W. S. (2018). Analyzing assumptions in conversation disentanglement research through the lens of a new dataset and model. *arXiv preprint arXiv:1810.11118*. 89

Kummerfeld, J. K., Gouravajhala, S. R., Peper, J. J., Athreya, V., Gunasekara, C., Ganhotra, J., Patel, S. S., Polymenakos, L. C., and Lasecki, W. (2019). A large-scale corpus for conversation disentanglement. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL'19)*, pages 3846–3856, Florence, Italy. 51, 52

Kurata, G., Xiang, B., and Zhou, B. (2016). Labeled data generation with encoder-decoder lstm for semantic slot filling. In *Proceedings of The 17th Annual Conference of the International Speech Communication Association (Interspeech'16)*, pages 725–729, San Francisco, CA, USA. 96

Lebanoff, L., Song, K., and Liu, F. (2018). Adapting the neural encoder-decoder framework from single to multi-document summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP'18)*, pages 4131–4141, Brussels, Belgium. 26

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324. 30, 38, 82

Li, H., Min, M. R., Ge, Y., and Kadav, A. (2017a). A context-aware attention network for interactive question answering. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'17)*, pages 927–935, Halifax, NS, Canada. 38

Li, J., Galley, M., Brockett, C., Gao, J., and Dolan, B. (2016a). A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL'16)*, pages 110–119, San Diego, CA, USA. 26, 84

Li, J., Galley, M., Brockett, C., Spithourakis, G., Gao, J., and Dolan, B. (2016b). A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL'16)*, pages 994–1003, Berlin, Germany. 26

Li, J., Monroe, W., Ritter, A., Jurafsky, D., Galley, M., and Gao, J. (2016c). Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP'16)*, pages 1192–1202, Austin, TX, USA. 103

Li, J., Monroe, W., Shi, T., Jean, S., Ritter, A., and Jurafsky, D. (2017b). Adversarial learning for neural dialogue generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP'17)*, pages 2157–2169, Copenhagen, Denmark. 46

Li, J., Qiu, L., Tang, B., Chen, D., Zhao, D., and Yan, R. (2019). Insufficient data can also rock! learning to converse using smaller data with augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'19)*, pages 6698–6705, Honolulu, HI, USA. 96

Lian, R., Xie, M., Wang, F., Peng, J., and Wu, H. (2019). Learning to select knowledge for response generation in dialog systems. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, (IJCAI'19)*, pages 5081–5087, Macao, China. 36

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain. 41

Lin, Z., Feng, M., Santos, C. N. d., Yu, M., Xiang, B., Zhou, B., and Bengio, Y. (2017). A structured self-attentive sentence embedding. In *Proceedings of the International Conference on Learning Representations (ICLR'17)*, Toulon, France. 41

Liu, B. and Lane, I. (2018). End-to-end learning of task-oriented dialogs. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop (NAACL'18)*, pages 67–73, New Orleans, LA, USA. 11, 24, 25

Liu, C.-W., Lowe, R., Serban, I., Noseworthy, M., Charlin, L., and Pineau, J. (2016). How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP'16)*, pages 2122–2132, Austin, TX, USA. 41, 42, 43, 45

Liu, H., Lin, T., Sun, H., Lin, W., Chang, C.-W., Zhong, T., and Rudnicky, A. (2017). Rubystar: a non-task-oriented mixture model dialog system. *arXiv preprint arXiv:1711.02781*. 34

Loisel, A. (2008). *Modélisation du dialogue homme-machine pour la recherche d'informations: approche questions-réponses*. PhD dissertation, INSA de Rouen. 23, 24

Long, Y., Wang, J., Xu, Z., Wang, Z., Wang, B., and Wang, Z. (2017). A knowledge enhanced generative conversational service agent. In *Proceedings of DSTC6 Workshop*. 36

Lowe, R., Noseworthy, M., Serban, I. V., Angelard-Gontier, N., Bengio, Y., and Pineau, J. (2017a). Towards an automatic turing test: Learning to evaluate dialogue responses. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL'17)*, pages 1116–1126, Vancouver, Canada. 41, 43, 45

Lowe, R., Pow, N., Serban, I., Charlin, L., and Pineau, J. (2015a). Incorporating unstructured textual knowledge sources into neural dialogue systems. In *Proceedings of Neural Information Processing Systems workshop on Machine Learning for Spoken Language Understanding (SLUNIPS'15)*, Montreal, Canada. 36

Lowe, R., Pow, N., Serban, I., and Pineau, J. (2015b). The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL'15)*, pages 285–294, Prague, Czech Republic. 9, 11, 18, 28, 30, 31, 43, 45, 50, 51, 52, 56, 61, 63, 64, 67, 68, 71, 72, 74, 77, 78, 86, 87, 91, 93, 100

Lowe, R. T., Pow, N., Serban, I. V., Charlin, L., Liu, C.-W., and Pineau, J. (2017b). Training end-to-end dialogue systems with the ubuntu dialogue corpus. *Dialogue & Discourse*, 8(1):31–65. 49, 51, 68, 82

Lu, Z. and Li, H. (2013). A deep architecture for matching short texts. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Proceedings of the 27th international conference on Advances in Neural Information Processing Systems (NIPS'13)*, pages 1367–1375, Lake Tahoe, NV, USA. Curran Associates, Inc. 29

Luong, T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP'15)*, pages 1412–1421, Lisbon, Portugal. 11, 27, 38, 39, 40

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA. 29

Mehri, S. and Carenini, G. (2017). Chat disentanglement: Identifying semantic reply relationships with random forests and recurrent neural networks. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (IJCNLP'17)*, pages 615–623, Taipei, Taiwan. 51

Meng, R., Zhao, S., Han, S., He, D., Brusilovsky, P., and Chi, Y. (2017). Deep keyphrase generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL'17)*, pages 582–592, Vancouver, Canada. 38

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS'13)*, pages 3111–3119, Lake Tahoe, NV, USA. 36, 37, 42

Miller, G. A. (1995). Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41. 36

Möller, S., Englert, R., Engelbrecht, K., Hafner, V., Jameson, A., Oulasvirta, A., Raake, A., and Reithinger, N. (2006). Memo: towards automatic usability evaluation of spoken dialogue services by user error simulations. In *Ninth International Conference on Spoken Language Processing*. 45

Nadeau, D. and Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticæ Investigationes*, 30(1):3–26. 36

Nöth, E., Horndasch, A., Gallwitz, F., and Haas, J. (2004). Experiences with commercial telephone-based dialogue systems (erfahrungen mit kommerziellen telefonsprachdialogsystemen). *It-Information Technology*, 46(6):315–321. 35

Nugmanova, A., Smirnov, A., Lavrentyeva, G., and Chernykh, I. (2019). Strategy of the negative sampling for training retrieval-based dialogue systems. In *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom'19)*, pages 844–848, Kyoto, Japan. 61

Pandey, G., Contractor, D., Kumar, V., and Joshi, S. (2018). Exemplar encoder-decoder for neural conversation generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL'18)*, pages 1329–1338, Melbourne, Australia. 24, 26, 34

Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135. 36

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL'02)*, pages 311–318, Stroudsburg, PA, USA. 41

Parthasarathi, P. and Pineau, J. (2018). Extending neural generative conversational model using external knowledge sources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP'18)*, pages 690–695, Brussels, Belgium. 36

Paulus, R., Xiong, C., and Socher, R. (2018). A deep reinforced model for abstractive summarization. In *Proceedings of the International Conference on Learning Representations (ICLR'18)*, Vancouver, Canada. 41

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830. 91

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*, pages 1532–1543, Doha, Qatar. 68

Qiu, M., Li, F.-L., Wang, S., Gao, X., Chen, Y., Zhao, W., Chen, H., Huang, J., and Chu, W. (2017). Alime chat: A sequence to sequence and rerank based chatbot engine. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL'17)*, pages 498–503, Vancouver, Canada. 33, 34, 59

Raux, A., Langner, B., Bohus, D., Black, A. W., and Eskenazi, M. (2005). Let's go public! taking a spoken dialog system to the real world. In *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech'05)*, pages 885–889, Lisbon, Portugal. 35

Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., and Lee, H. (2016). Generative adversarial text to image synthesis. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning (ICML'16)*, pages 1060–1069, New York, NY, USA. 38

Rennie, S. J., Marcheret, E., Mroueh, Y., Ross, J., and Goel, V. (2016). Self-critical sequence training for image captioning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1179–1195. 26

Riou, M., Salim, S., and Hernandez, N. (2015). Using discursive information to disentangle French language chat. In *2nd Workshop on Natural Language Processing for Computer-Mediated Communication (NLP4CMC 2015) / Social Media at GSCL Conference 2015*, pages 23–27, Essen, Germany. 51

Ritter, A., Cherry, C., and Dolan, B. (2010). Unsupervised modeling of twitter conversations. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL'10)*, pages 172–180, Los Angeles, CA, USA. 50

Ritter, A., Cherry, C., and Dolan, W. B. (2011). Data-driven response generation in social media. In *Proceedings of the conference on empirical methods in natural language processing*, pages 583–593, Edinburgh, Scotland. 24, 25, 41, 49

Robertson, S. E. and Jones, K. S. (1976). Relevance weighting of search terms. *Journal of the American Society for Information science*, 27(3):129–146. 54

Ruppenhofer, J., Ellsworth, M., Petruck, M. R., Johnson, C. R., and Scheffczyk, J. (2006). *FrameNet II: Extended Theory and Practice*. International Computer Science Institute, Berkeley, California. 36

Rus, V. and Lintean, M. (2012). A comparison of greedy and optimal assessment of natural language student input using word-to-word similarity metrics. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 157–162, Montréal, Canada. 42

Rush, A. M., Chopra, S., and Weston, J. (2015). A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP'15)*, pages 379–389, Lisbon, Portugal. 38

Salton, G., Wong, A., and Yang, C. S. (1975). A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620. 30

Scheepers, T. (2017). Improving the compositionality of word embeddings. Master's thesis, Universiteit van Amsterdam, Science Park 904, Amsterdam, Netherlands. 36

Schlöder, J. J. and Fernández, R. (2015). Clarifying intentions in dialogue: A corpus study. In *Proceedings of the 11th International Conference on Computational Semantics (IWCS'15)*, pages 46–51, London, UK. 23

Schrading, N., Ovesdotter Alm, C., Ptucha, R., and Homan, C. (2015). An analysis of domestic abuse discourse on Reddit. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP'15)*, pages 2577–2583, Lisbon, Portugal. 50

Schuler, K. K. (2006). *VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon*. PhD dissertation, University of Pennsylvania. 36

Serban, I. V., Klinger, T., Tesauro, G., Talamadupula, K., Zhou, B., Bengio, Y., and Courville, A. (2017a). Multiresolution recurrent neural networks: An application to dialogue response generation. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI'17)*, San Francisco, CA, USA. 18, 27, 41

Serban, I. V., Lowe, R., Henderson, P., Charlin, L., and Pineau, J. (2018). A survey of available corpora for building data-driven dialogue systems: The journal version. *Dialogue & Discourse*, 9(1):1–49. 11, 26, 35, 49, 50

Serban, I. V., Sankar, C., Germain, M., Zhang, S., Lin, Z., Subramanian, S., Kim, T., Pieper, M., Chandar, S., Ke, N. R., et al. (2017b). A deep reinforcement learning chatbot. *arXiv preprint arXiv:1709.02349*. 34, 103

Serban, I. V., Sordoni, A., Bengio, Y., Courville, A., and Pineau, J. (2016). Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI'16)*, pages 3776–3783, Phoenix, AZ, USA. 24, 26, 45

Shang, L., Lu, Z., and Li, H. (2015). Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL'15)*, pages 1577–1586, Beijing, China. 25

Shang, M., Fu, Z., Peng, N., Feng, Y., Zhao, D., and Yan, R. (2018). Learning to converse with noisy data: Generation with calibration. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18)*, pages 4338–4344. 61

Shao, Y., Gouws, S., Britz, D., Goldie, A., Strope, B., and Kurzweil, R. (2017). Generating high-quality and informative conversation responses with sequence-to-sequence models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP'17)*, pages 2210–2219, Copenhagen, Denmark. 24

Shen, T., Zhou, T., Long, G., Jiang, J., Pan, S., and Zhang, C. (2018). Disan: Directional self-attention network for rnn/cnn-free language understanding. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI'18)*, pages 5446–5455, New Orlean, LA, USA. 41

Sidner, C. L. (1983). What the speaker means: the recognition of speakers' plans in discourse. *Computers & Mathematics with Applications*, 9(1):71 – 82. 23

Socher, R., Huang, E. H., Pennington, J., Ng, A. Y., and Manning, C. D. (2011). Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Proceedings of the 24th International Conference on Neural Information Processing Systems (NIPS'11)*, pages 801–809, Granada, Spain. 27

Song, Y., Li, C.-T., Nie, J.-Y., Zhang, M., Zhao, D., and Yan, R. (2018). An ensemble of retrieval-based and generation-based human-computer conversation systems. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI'18)*, pages 4382–4388, Stockholm, Sweden. 16, 34

Sordoni, A., Bengio, Y., Vahabi, H., Lioma, C., Grue Simonsen, J., and Nie, J.-Y. (2015a). A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management (CIKM'15)*, pages 553–562, Melbourne, Australia. 26, 27, 41

Sordoni, A., Galley, M., Auli, M., Brockett, C., Ji, Y., Mitchell, M., Nie, J.-Y., Gao, J., and Dolan, B. (2015b). A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL'15)*, pages 196–205, Denver, CO, USA. 24, 25, 26

Speer, R. and Havasi, C. (2013). *ConceptNet 5: A Large Semantic Network for Relational Knowledge*, pages 161–176. Springer Berlin Heidelberg, Berlin, Heidelberg. 36

Stent, A. J. (2001). *Dialogue systems as conversational partners: applying conversation acts theory to natural language generation for task-oriented mixed-initiative spoken dialogue*. PhD dissertation, University of Rochester. Dept. of Computer Science. 22

Strubell, E., Ganesh, A., and McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL'19)*, pages 3645–3650, Florence, Italy. 33

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Proceedings of the 2014 conference on Advances in Neural Information Processing Systems (NIPS'14)*, pages 3104–3112, Montreal, Canada. 25, 38

Tan, M., Santos, C. d., Xiang, B., and Zhou, B. (2015). Lstm-based deep learning models for non-factoid answer selection. In *Proceedings of the International Conference on Learning Representation (ICLR'15)*, San Diego, CA, USA. 30, 63, 64, 67, 68, 82

Tao, C., Mou, L., Zhao, D., and Yan, R. (2018). Ruber: An unsupervised method for automatic evaluation of open-domain dialog systems. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI'18)*, pages 722–729, New Orlean, LA, USA. 45

Tenenbaum, J. B. and Freeman, W. T. (2000). Separating style and content with bilinear models. *Neural computation*, 12(6):1247–1283. 67

Theano Development Team (2016). Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688. 81, 90

Tiancheng, Z. (2019). *Learning to converse with latent actions*. PhD dissertation, Carnegie Mellon University. 25, 27, 35

Tien Nguyen, D. and Joty, S. (2017). A neural local coherence model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL'17)*, pages 1320–1330, Vancouver, Canada. 102

Tixier, A. J.-P. (2018). Notes on deep learning for nlp. *arXiv preprint arXiv:1808.09772*. 38, 39, 41

Tur, G. (2011). *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*. John Wiley and Sons. 24

Turchi, M. and Negri, M. (2014). Automatic annotation of machine translation datasets with binary quality judgements. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1788–1792, Reykjavik, Iceland. 103

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS'17)*, pages 5998–6008, Long Beach, CA, USA. 32, 41, 81, 82

Vinyals, O. and Le, Q. (2015). A neural conversational model. In *Workshop on Deep Learning at the 31 st International Conference on Machine Learning (ICML'15)*, Lille, France. 25, 26

Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2015). Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'15)*, pages 3156–3164, Boston, MA, USA. 26, 38

Voorhees, E. M. (2001). The trec question answering track. *Natural Language Engineering*, 7(4):361–378. 90

Walker, M. A., Litman, D. J., Kamm, C. A., and Abella, A. (1997). PARADISE: A framework for evaluating spoken dialogue agents. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics (EACL'97)*, pages 271–280, Madrid, Spain. 45

Wallace, R., Tomabechi, H., and Aimless, D. (2003). Chatterbots go native: Considerations for an eco-system fostering the development of artificial life forms in a human world. *Published online: http://www. pandorabots. com/pandora/pics/chatterbotsgonative. doc*. 15

Wan, S., Lan, Y., Guo, J., Xu, J., Pang, L., and Cheng, X. (2016). A deep architecture for semantic matching with multiple positional sentence representations. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI'16)*, pages 2835–2841, Phoenix, AZ, USA. 31, 64, 67, 68, 82

Wang, M., Lu, Z., Li, H., and Liu, Q. (2015). Syntax-based deep matching of short texts. In *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI'15)*, pages 1354–1361, Buenos Aires, Argentina. 27, 29

Wang, S. and Jiang, J. (2016). Learning natural language inference with lstm. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL'16)*, pages 1442–1451, San Diego, CA, USA. 31, 63, 64, 67, 68, 82

Wang, W., Yang, N., Wei, F., Chang, B., and Zhou, M. (2017). Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL'17)*, pages 189–198, Vancouver, Canada. 33, 81

Ward, N. G. and DeVault, D. (2016). Challenges in building highly-interactive dialog systems. *AI Magazine*, 37(4):7–18. 22

Weizenbaum, J. (1966). Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45. 3, 15, 21

Wen, T.-H., Gašić, M., Kim, D., Mrkšić, N., Su, P.-H., Vandyke, D., and Young, S. (2015). Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue (ACL'15)*, pages 275–284, Prague, Czech Republic. 24

Weston, J., Dinan, E., and Miller, A. (2018). Retrieve and refine: Improved sequence generation models for dialogue. In *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI (SCAI'18)*, pages 87–92, Brussels, Belgium. 34

Wieting, J., Bansal, M., Gimpel, K., and Livescu, K. (2016). Towards universal paraphrastic sentence embeddings. In *4th International Conference on Learning Representations, (ICLR'16)*, San Juan, Puerto Rico. 43

Williams, J., Raux, A., Ramachandran, D., and Black, A. (2013). The dialog state tracking challenge. In *Proceedings of the Special Interest Group on Discourse and Dialogue (SIGDial'13)*, pages 404–413, Metz, France. 50

Wu, W., Lu, Z., and Li, H. (2013). Learning bilinear model for matching queries and documents. *Journal of Machine Learning Research*, 14:2519–2548. 27

Wu, Y., Li, Z., Wu, W., and Zhou, M. (2018a). Response selection with topic clues for retrieval-based chatbots. *Neurocomputing*, 316:251–261. 66, 86, 87

Wu, Y., Wu, W., Li, Z., and Zhou, M. (2016). Topic augmented neural network for short text conversation. *ArXiv*. 29

Wu, Y., Wu, W., Li, Z., and Zhou, M. (2018b). Learning matching models with weak supervision for response selection in retrieval-based chatbots. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL'18)*, pages 420–425, Melbourne, Australia. 58

Wu, Y., Wu, W., Xing, C., Zhou, M., and Li, Z. (2017). Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL'17)*, pages 496–505, Vancouver, Canada. 28, 32, 33, 43, 51, 56, 57, 58, 63, 67, 68, 77, 78, 79, 81, 82, 86, 87

Xing, C., Wu, W., Wu, Y., Liu, J., Huang, Y., Zhou, M., and Ma, W.-Y. (2017). Topic aware neural response generation. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI'17)*, San Francisco, CA, USA. 26

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y. (2015a). Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning (ICML'15)*, pages 2048–2057, Lille, France. 39, 40

Xu, K., Ba, J. L., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R. S., and Bengio, Y. (2015b). Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning (ICML'15)*, pages 2048–2057, Lille, France. 38

Xu, X., Dušek, O., Konstas, I., and Rieser, V. (2018). Better conversations by modeling, filtering, and optimizing for coherence and diversity. In *Proceedings of the 2018*

Conference on Empirical Methods in Natural Language Processing (EMNLP'18), pages 3981–3991, Brussels, Belgium. 102

Xu, Z., Liu, B., Wang, B., Sun, C., and Wang, X. (2017). Incorporating loose-structured knowledge into conversation modeling via recall-gate lstm. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'17)*, pages 3506–3513, Anchorage, AK, USA. 29, 66, 68, 87

Xue, X., Jeon, J., and Croft, W. B. (2008). Retrieval models for question and answer archives. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '08)*, pages 475–482, Singapore, Singapore. 27

Yan, R., Song, Y., and Wu, H. (2016). Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '16)*, pages 55–64, Pisa, Italy. 31, 80, 82

Yang, L., Qiu, M., Qu, C., Guo, J., Zhang, Y., Croft, W. B., Huang, J., and Chen, H. (2018). Response ranking with deep matching networks and external knowledge in information-seeking conversation systems. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '18)*, pages 245–254, Ann Arbor, MI, USA. 9, 32, 36, 43, 51, 54, 58, 59, 63, 77

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., and Hovy, E. (2016). Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL'16)*, pages 1480–1489, San Diego, CA, USA. 41

Yoshino, K., Hori, C., Perez, J., D'Haro, L. F., Polymenakos, L., Gunasekara, C., Lasecki, W. S., Kummerfeld, J., Galley, M., Brockett, C., Gao, J., Dolan, B., Gao, S., Marks, T. K., Parikh, D., and Batra, D. (2018). Dialog system technology challenge 7. In *Proceedings of the 2nd Conversational AI workshop at NIPS (ConvAI'18)*, Montreal, Canada. 52, 86

Young, S., Gašić, M., Thomson, B., and Williams, J. D. (2013). Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179. 24

Young, T., Cambria, E., Chaturvedi, I., Zhou, H., Biswas, S., and Huang, M. (2018). Augmenting end-to-end dialogue systems with commonsense knowledge. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI'18)*, pages 4970–4977, New Orlean, LA, USA. 36

Zhang, X., Su, J., Qin, Y., Liu, Y., Ji, R., and Wang, H. (2018a). Asynchronous bidirectional decoding for neural machine translation. In *Thirty-Second AAAI Conference on Artificial Intelligence (AAAI'18)*, pages 5698–5705, New Orleans, LA, USA. 26

Zhang, X., Zhao, J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS'15)*, pages 649–657, Montreal, Canada. 96

Zhang, Y., Galley, M., Gao, J., Gan, Z., Li, X., Brockett, C., and Dolan, B. (2018b). Generating informative and diverse conversational responses via adversarial information maximization. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS'18)*, pages 1815–1825, Montreal, Canada. 24

Zhang, Z., Li, J., Zhu, P., Zhao, H., and Liu, G. (2018c). Modeling multi-turn conversation with deep utterance aggregation. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING'18)*, pages 3740–3752, Santa Fe, NM, USA. 32, 51, 55, 63, 67, 77, 81, 82, 101

Zhao, W. X., Jiang, J., Weng, J., He, J., Lim, E.-P., Yan, H., and Li, X. (2011). Comparing twitter and traditional media using topic models. In *Proceedings of the 33rd European Conference on Advances in Information Retrieval (ECIR'11)*, pages 338–349, Dublin, Ireland. 29

Zhou, X., Dong, D., Wu, H., Zhao, S., Yu, D., Tian, H., Liu, X., and Yan, R. (2016). Multi-view response selection for human-computer conversation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP'16)*, pages 372–381, Austin, TX, USA. 32, 81, 82

Zhou, X., Li, L., Dong, D., Liu, Y., Chen, Y., Zhao, W. X., Yu, D., and Wu, H. (2018). Multi-turn response selection for chatbots with deep attention matching network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL'18)*, pages 1118–1127, Melbourne, Australia. 32, 43, 63, 67, 77, 81, 82, 86, 101

Zhufeng, P., Yan, W., Kun, B., Lianqiang, Z., and Xiaojiang, L. (2019). Improving open-domain dialogue systems via multi-turn incomplete utterance restoration. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'19)*. 61

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

UNIVERSITÉ DE NANTES

Titre: Sélection de Réponses pour les Systèmes de Dialogue Basés sur la Recherche de Réponse de Bout en Bout

Mot clés : Apprentissage profond, systèmes de dialogue orientés tâche, chatbots, systèmes de recherche, réseaux de neurones.

Resumé : Le besoin croissant en assistance humaine a poussé les chercheurs à développer des systèmes de dialogue automatiques, intelligents et infatigables qui conversent avec les humains dans un langage naturel pour devenir soit leurs assistants virtuels ou leurs compagnons. L'industrie des systèmes de dialogue est devenue populaire cette dernière décennie, ainsi, plusieurs systèmes ont été développés par des industriels comme des académiques. Dans le cadre de cette thèse, nous étudions les systèmes de dialogue basés sur la recherche de réponse qui cherchant la réponse la plus appropriée à la conversation parmi un ensemble de réponses prédéfini. Le défi majeur de ces systèmes est la compréhension de la conversation et l'identification des éléments qui décrivent le problème et la solution qui sont souvent implicites. La plupart des approches récentes sont basées sur des techniques d'apprentissage profond qui permettent de capturer des informations implicites. Souvent, ces approches sont complexes ou dépendent fortement du domaine. Nous proposons une approche de recherche de réponse de bout en bout, simple, efficace et indépendante du domaine et qui permet de capturer ces informations implicites. Nous effectuons également plusieurs analyses afin de déterminer des pistes d'amélioration.

Title: Response Selection for End-to-End Retrieval-Based Dialogue Systems

Keywords : Deep learning, goal-oriented dialogue systems, chatbots, retrieval-systems, neural networks.

Abstract : The increasing need of human assistance pushed researchers to develop automatic, smart and tireless dialogue systems that can converse with humans in natural language to be either their virtual assistant or their chat companion. The industry of dialogue systems has been very popular in the last decade and many systems from industry and academia have been developed. In this thesis, we study retrieval-based dialogue systems which aim to find the most appropriate response to the conversation among a set of predefined responses. The main challenge of these systems is to understand the conversation and identify the elements that describe the problem and the solution which are usually implicit. Most of the recent approaches are based on deep learning techniques which can automatically capture implicit information. However these approaches are either complex or domain dependent. We propose a simple, end-to-end and efficient retrieval-based dialogue system that first matches the response with the history of the conversation on the sequence-level and then we extend the system to multiple levels while keeping the architecture simple and domain independent. We perform several analyzes to determine possible improvements.