BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

STYLIZED NATURAL LANGUAGE GENERATION IN DIALOGUE SYSTEMS
GENEROVÁNÍ STYLIZOVANÉHO LIDSKÉHO JAZYKA V DIALOGOVÝCH SYSTÉMECH

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR   **KSENIA BOLSHAKOVA**
AUTOR PRÁCE

SUPERVISOR   **Ing. MARTIN FAJČÍK**
VEDOUCÍ PRÁCE

BRNO 2020

Abstract

Do tohoto odstavce bude zapsán výtah (abstrakt) práce v anglickém jazyce.

Abstrakt

Do tohoto odstavce bude zapsán výtah (abstrakt) práce v českém (slovenském) jazyce.

Keywords

Sem budou zapsána jednotlivá klíčová slova v anglickém jazyce, oddělená čárkami.

Klíčová slova

Sem budou zapsána jednotlivá klíčová slova v českém (slovenském) jazyce, oddělená čárkami.

Reference

BOLSHAKOVA, Ksenia. *Stylized Natural Language Generation in Dialogue Systems*. Brno, 2020. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Ing. Martin Fajčík

Stylized Natural Language Generation in Dialogue Systems

Declaration

I hereby declare that this Bachelor's thesis was prepared as an original work by the author under the supervision of Mr. X The supplementary information was provided by Mr. Y I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

. . . .
Ksenia Bolshakova
February 8, 2020

Acknowledgements

Here it is possible to express thanks to the supervisor and to the people which provided professional help (external submitter, consultant, etc.).

Contents

Chapter 1

Introduction

Dialogue system (i.e. conversational agent) is a computer system which interacts with a human in natural language. These systems are used in cars (hands-free car-specific functions, Android Auto, Apple CarPlay, vendor-specific solutions), web, robots, computer games etc, because a conversation is a natural way for people to get information.

Natural Language Generation (NLG) is an important component of dialogue systems, which has a significant impact on system quality, because NLG goal is to imitate human behaviour. The conversational agent can be classified into task-oriented (i.e. goal-oriented), which focused on completing a certain tasks and adhere to a determined script for each stage of the conversation, and non-task-oriented, which do not have a stated goal to work towards. A lot of devices have incorporated goal-oriented dialogue systems, such as Yandex's Alisa, Apple's Siri, Microsoft's Cortana, Amazon Alexa, and Google Assistant. Goal-oriented dialogue acts make conversations more interpetable and controllable. On the other hand, they also hinder scaling such systems to new domains (i.e. conversation topic). To escape from the limitation, recent interest of research started moving to non-task-oriented chitchat dialogues (chatbots). Chitchat dialogues are systems designed to mimic the unstructured human-human conversation. This kind of conversational agent often have an entertainment value, such as Cleverbot, Microsoft's XiaoIce system etc. Chatbots have also been used for testing theories of psychological counseling.

The ability to communicate freely in a natural language is one of the hallmarks of human intelligence, and is likely one of the requirements for true artificial intelligence. Many researches work on open-ended (i.e. there is a huge range of appropriate outputs given the input) chitchat dialogues to explore this aspect of intelligence, because in goal-oriented dialogue systems there is a relatively narrow range of correct outputs given the input. Creating a non-task-oriented agent is a challenge for researches, because there are a lot of topics of conversations as well as user reactions and responses to them. Such bots are not able to generate meaningful conversations. Their replies are often too generic, because non-specific responses sound quite natural (e.g. "Ok, I see", "I don't know"). There are still a lot of problems in the Natural Language Generation, such as response-relatedness, semantic errors, repetition etc., which will be described in more detail in the chapture 2. In human-human conversation not only these attributes are important, but also your partner's style of communication, what phrases he is using to make your dialogue more diverse and interesting, because it is unlikely that someone will want to communicate for a long time with a person who answers the same thing all the time.

The main purpose of this thesis was to create NLG model, which will be able to generate text in different styles. //TODO

Chapter 2

NLG problems in dialogue systems

In a book [1] Natural Language Generation is defined as "the process by which thought is rendered into language". NLG approaches can be grouped into two categories, one focuses on generating text using templates or (linguistic) rules, the other uses corpus-based statistical methods (corpora is a collection of texts) [5].

2.1 Template-based approach

Until recently Natural Language Generation component of a dialog system used primarily hand-coded generation templates, which represented model sentences in a natural language mapped to a particular semantic content. The template-based system selects a proper response for the current conversation from a repository with response selection algorithms. Templates are often designed for a specific task in a given domain [4]. Example of template-based system is shown in the Table 2.1.

Advantages of template-based approach

The output produced by this approach is likely to be grammatically correct and not contain unexpected generation errors. The process of sentence generation is fully controlled, these models are robust and reliable because they consist of clearly defined rules.

Disadvantages of template-based approach

These models require time and human resources to deploy a real dialogue system, because templates are constructed manually, and the number of templates grows quickly (using different templates for singular and plural versions). These systems are not able to handle unknown inputs. Templates often sound unnatural due to their generic structures. Template-based systems cannot make variation in output, it is just concatenation of strings. This approach is not flexible, because it has limits to use templates in other domains. Template-based model is not able to learn and is not able to adapt to the user.

2.2 Corpus-based approach

Corpus-based system dominates the NLG community, special in the case of open-ended tasks, where it is almost impossible to hand-craft the templates for all possible combi-

Example:
User's input:
"I'm going to travel from Moscow on April 2."
Template:
What time would you like to travel from {*departure_city*} on {*departure_date*}?
Agent's output:
"What time would you like to travel from Moscow on April 2?"

Table 2.1: The example of template-based approach.

nations of semantic units. Corpus-based systems include statistical and machine learning approaches to resolve it [8].

One of the first approaches in corpus-based methods is **Dynamic Sentence Generation**, which dynamically creates sentences from representations of the meaning to be conveyed by the sentence and/or its desired linguistic structure. It allows do not write code for every boundary case and includes aggregation, reference, ordering and connectives to optimise sentences.

Next level of corpus-based approaches is **Dynamic Document Creation**, what can produce relevant and well-structured document.

Advantages of corpus-based approach

Corpus-based models have ability to generate more proper responses that could have never appeared in the corpus; it is possible to mimick the language of a real domain expert and use this models for open-domain dialogue systems; dynamic approach is able to learn and to handle unknown inputs, it is also has a lot of possible variations of output.

Disadvantages of corpus-based approach

It is necessary to have a corpus, which contains a large amount of data and on a variety of topics to get a sensible output. Even if you have the corpus, process of text generation is not fully controlled and the output can be incorrect or does not make a sence. This approach still has a lot of problems, what will be described in more detail in //TODO.

2.3 Language Models

In corpus-based system natural language generation uses **Language Models(LMs)** to generate sequences of texts. LM is a probabilistic model which learns to predict the probability of a sequence of words. The equation 2.1 represents the language model, where W is a sequence and $w_1, w_2, ..., w_n$ are words in this sequence. The language model provides a context for distinguishing words and phrases that sound the same. For example the phrases "but her" and "butter" sound the same, but mean different things.

$$P(W) = P(w_1, w_2, ..., w_n) \tag{2.1}$$

The **Chain rule** (equation 2.2) is used to calculate the joint probability of a sentence by using the conditional probability (equation 2.3) of a word given previous words.

$$P(w_1, w_2, ..., w_n) = \prod_i P(w_i | w_1, w_2, ..., w_{i-1}) \tag{2.2}$$

Hi	1-gram
New York	2-gram
The Three Musketeers	3-gram
She is studying IT	4-gram

Table 2.2: The example of N-grams.

$$P(A|B) = P(A \cap B)/P(B) \tag{2.3}$$

In equation 2.3 $P(A \cap B)$ is the probability that both events A and B occur.

Figure 2.1: Example of a word prediction by using chain rule.

An example in the Figure 2.1 shows how to predict probability of a word given previous words. A subsequence (context) may consist of a very large number of words and the likelihood that such subsequence is found in a corpus is very small. It is a main problem in language models, which is called **data sparsity**.

Data sparsity is the phenomenon of not observing enough data in a corpus to model language accurately. The solution to resolve this issue is to make the assumption that the probability of a word depends only on the previous n words and use **N-gram model** (N-gram is a sequence of N words).

The n-gram "She is studying IT" from the Table 2.2 does not occur as often in texts of corpus as n-grams "Hi", "New York" and "The Three Musketeers". Knowing a probability to the occurrence of an N-gram in a sequence of words can be useful, because it can help to decide which N-grams can be chunked together to from single entities (like "New York" chuncked together as one word). It can also help make next word predictions. For example, "tea" is more likely than "ball" in the phrase "I would like to drink".

2.3.1 Dialogue system architecture

Dialogue system architecture is illustrated in Figure 2.2. This architecture consists from Natural Language Understanding (NLU), dialogue management (DM), and Natural Language Generation (NLG).

NLU extracts the meaning from the user utterance and converts into a structured semantic representation. Natural Language Understanding traditionally consists of domain identification and intent prediction, which are framed as utterance classification problems, and slot filling, framed as a sequence tagging task.

DM plays two roles,tracking the dialogue state and performing the dialogue policy (i.e., telling the agent how to act given the dialogue state.)

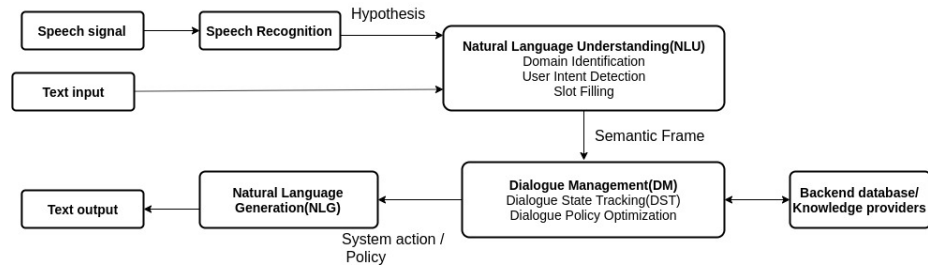**NLG** transforms structured data into natural language.[3]

Figure 2.2: Dialogue system architecture.

Chapter 3

Natural Language Generation

Natural Language Generation is a subsection of Natural Language Processing (NLP).

3.1 Architecture of NLG

Reiter and Dale in the article [7] proposed architecture splitted into 3 stages, which is illustrated in Figure 3.1.
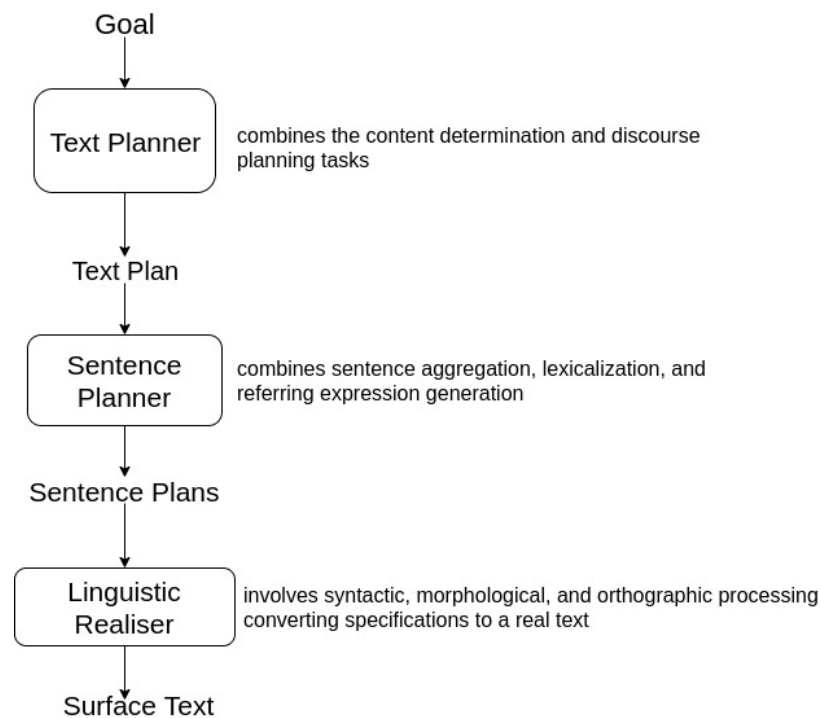

Figure 3.1: NLG architecture.

This pipeline shows the milestones of natural language generation, however, specific steps and approaches, as well as the models used, can vary significantly with the technology development.

NLG approaches can be grouped into two categories, one focuses on generating text using templates or rules (linguistic) methods, the other uses corpus-based statistical/dynamic methods, where **corpus** is a collection of texts.

3.2   Categories of Natural Language Generation approaches

This section describes categories of NLG. More information you can find in [8] [2] [4].

The gaps represented by [**train**] and [**town**] in example 2.1 are filled by looking up the relevant information in a table. More information you can find in [7].

3.3   Model components for Natural Language Generation

NLG evolution from templates to dynamic generation of sentences took a lot of time and models developed along with it. Corpus-based generation uses a generative probabilistic model what can be implemented in many ways.

3.3.1   Markov Chain

Markov chain is one of the first model's implementation in NLG. It is a stochastic model, that is used to describe the next event in a sequence given the previous event only by probabilistic weighting. It does not track all the previous states in a sequence to conclude what is the next possible state. In our case state is an n-gram(contiguous sequence of n items from text or speech). Markov chain is shown in 3.2.
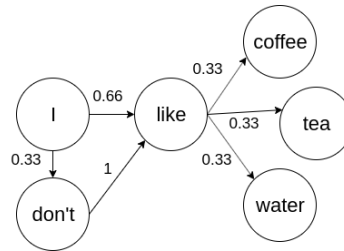

Figure 3.2: Markov chain

3.3.2   Recurrent neural network (RNN)

Neural networks are computing systems that are inspired by biological neural networks. A „neuron" is a mathematical function that classifies and collects information according to a specific architecture. A neural network contains layers of interconnected „neurons". RNN takes series of input with no predetermined limit on size. Recurrent neural network passes each item of the sequence through a feedforward network and stores the information from the previous step. In each iteration this model calculates the probability of the next item with using information in the memory.

The architecture of RNN is illustrated in Figure 3.3, where coefficient **h** is a hidden layer, **x** is an input and **y** is an output. Coefficient **w** is a weight, what is transformed to produce a sensible output.

The RNN-based models have been used for NLG as an end-to-end training network [12] and a training model with semantic aggregation [10].

Nowadays RNN networks almost are not used in NLG, because it has problems with vanishing and exploding gradient. The long-term information travels sequentially through all cells before getting to the present processing cell. This means, that it can be easily corrupted by being multiplied many time by small or big numbers. Other problem of RNN is that they are not hardware friendly, it takes a lot of resources to train and run this network. More information about difficulties of training RNN you can find in [6].
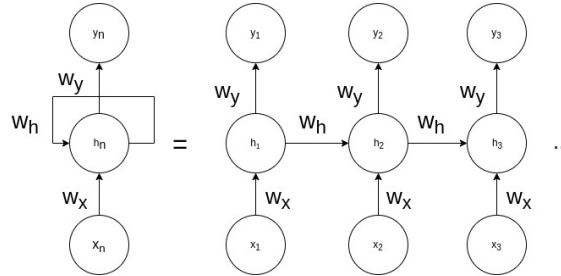

Figure 3.3: Architecture of recurrent neural network.

3.3.3 Long short-term memory (LSTM)

LSTM networks are a special kind of RNN, capable of learning long-term dependencies. All recurrent neural networks have the form of a chain of repeating modules of neural network. LSTM network also has this chain structure, but the repeating module has a different structure.

In Figure 3.4 each line carries an entire vector from the output of one node to the inputs of others. The circles represent pointwise operations, while rectangles are learned neural network layers. Lines merging denote concatenation, while a line forking denote its content being copied and the copies going to different locations.

This model resolved a problem with vanishing gradient, but still the capacity of the LSTM memory is limited, because of inherently complex sequential words' paths from the previous unit to the current unit. The same complexity results in high computational requirements that make LSTM difficult to train or parallelize.
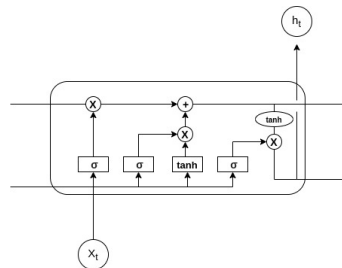

Figure 3.4: A cell in an LSTM network.

3.3.4 Transformer

Transformer is a relatively new architecture, which proposes a „self-attention mechanism". This model was introduced in the paper „Attention is all you need" [11] in 2017. The advantage of this model is a constant number of operations, which simplifies the parallelization

of processes. The Transformer uses the representation of all words in context without compressing all information into a single fixed-length representation that allows the system to handle longer sentences without the skyrocketing of computational requirements.

The architecture of the Transformer is illustrated in Figure 3.5. The Transformer consists of a stack of encoders (on the left) for processing inputs of any length and another set of decoders (on the right) to output the generated sentences. The inputs and output are first embedded into an n-dimensional space since we cannot use strings directly. The Transformer cannot remember how sequences are fed into a model, because of it positions are added to the embedded representation (n-dimensional vector) of each word.
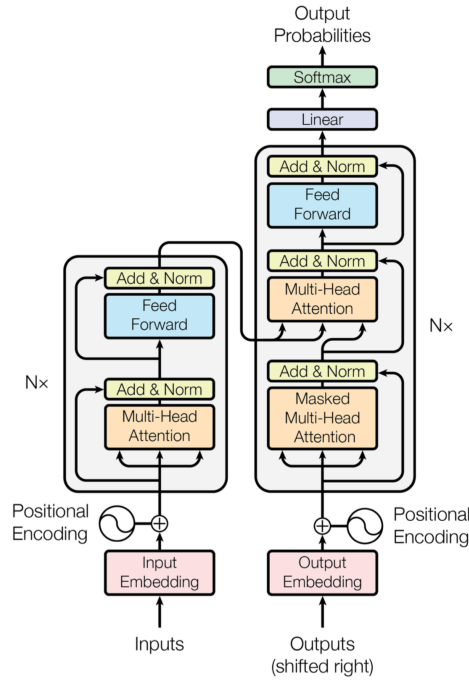

Figure 3.5: The architecture of the Transformer[11]

3.4 NLG problems

The NLG component converts an abstract dialogue action into natural language surface utterances. As noticed in [9], the main task of NLG is to select, inflect and order words to communicate the input meaning as completely, clearly and fluently as possible in context. That's why it is necessary to control not only correctness of output but also if output is appropriate or felicitous in a given context. A good generator usually relies on several factors:

- **adequacy** (a sentence that is ambiguous or not contains communicates meaning in the input, is **not** adequate)

- **syntactic correctness**

- **repetition** (self-repetition across utterances and with utterances, repeating the conversational partner)

Example	
1	I bought movie tickets on Tuesday.
2	I got movie tickets on Tuesday.
3	On Tuesday I bought movie tickets.
4	On movie Tuesday tickets I bought.
5	I bought tickets for the Tuesday movie.

Table 3.1: The example of sentences' variation

Problems of syntactic correctness	
1	They IS going to school.
2	It depends AT you.
Problems of response-relatedness	
1	-Do you go get coffee often?
	-I am a musician.
Problems of repetition	
1	-Yes, I'm studying law at the moment.
	- Good.
	- I like playing the piano.
	- Good.
Problems of adequacy	
1	More phones have games on them than this one.
2	Why a mouse when it spins?

Table 3.2: The example of NLG problems

- **response-relatedness** (efficacy in context)

- **variation** (there are 2 basic forms of variation: *word choice variation* and *word order variation* for enriching speech)

An example 3.1 shows all types of variation. Sometimes this factor can be syntactically incorrect or unclear, what you can see in the forth sentence. In fifth sentnecne a variation changed the meaning of part of the sentence. In addition, the variation may add or remove meaning possibilities.

The table 3.2 presents examples of NLG problems that were described above. Sentences 1 and 2 in examples of problems of adequacy show that a sentence can be grammatically and syntactically composed correctly, but without giving any meaning. In example of response-relatedness the answer to the question does not make a sense and it spoils the impression of the conversation. A communication will be boring if NLG model has problems with repetition.

All of these factors influence success of NLG evaluation.

Chapter 4

Datasets and Evaluation methods

A good corpus is very important for successful Natural Language Generation. Dialogue systems require training data in the format of people text conversation, for example, non-fiction or movie reviews are not suitable for this. Large volumes of training data improves the decision-making ability of NLG model, so those models can use it to figure out patterns. Quality is more important for training data than the quantity of data points. Unfortunately, there are not a lot of datasets available for training NLG models, due to the high cost of creating quality datasets.

In my bachelor thesis I am using 2 different dataset (Twitter data and Persona-Chat).

4.0.1 Persona-Chat

Persona 1	Persona 2
I like to ski	I am an artist
My wife does not like me anymore	I have four children
I have went to Mexico 4 times this year	I recently got a cat
I hate Mexican food	I enjoy walking for exercise
I like to eat cheetos	I love watching Game of Thrones

[PERSON 1:] Hi
[PERSON 2:] Hello ! How are you today ?
[PERSON 1:] I am good thank you , how are you.
[PERSON 2:] Great, thanks ! My children and I were just about to watch Game of Thrones.
[PERSON 1:] Nice ! How old are your children?
[PERSON 2:] I have four that range in age from 10 to 21. You?
[PERSON 1:] I do not have children at the moment.
[PERSON 2:] That just means you get to keep all the popcorn for yourself.
[PERSON 1:] And Cheetos at the moment!
[PERSON 2:] Good choice. Do you watch Game of Thrones?
[PERSON 1:] No, I do not have much time for TV.
[PERSON 2:] I usually spend my time painting: but, I love the show.

Figure 4.1: Example dialogue from the Persona-Chat dataset [13]

Persona-Chat models normal conversation when 2 people meet for the first meet and try to get know each other better. The aim of the dialogue is to learn about interests of another person, find common ground and discuss their hobbies. The task involves both asking and answering questions.

Average length of your persona description:	6.331842242600462
Average length of partner's persona description:	6.321163335493196
Average length of the first person's utterances:	11.418615621053272
Average length of the second person's utterances:	11.929411585690591
Number of your persona description's sentences:	40239
Number of partner's persona description's sentences:	40126
Number of the first person's utterances:	65719
Number of the second person's utterances:	65719
Number of dialogues	8938

Table 4.1: Persona-Chat statistics

Original Persona	Revised Persona
I love the beach.	To me, there is nothing like a day at the seashore.
My dad has a car dealership	My father sales vehicles for a living.
I just got my nails done	I love to pamper myself on a regular basis.
I am on a diet now	I need to lose weight.
Horses are my favorite animal.	I am into equestrian sports.
I play a lot of fantasy videogames.	RPGs are my favorite genre.
I have a computer science degree.	I also went to school to work with technology.
My mother is a medical doctor	The woman who gave birth to me is a physician.
I am very shy.	I am not a social person.
I like to build model spaceships.	I enjoy working with my hands.

Figure 4.2: Example of original and revised personas [13]

Persona-Chat dataset consists of small conversations between 2 crowdworkers from Amazon Mechanical Turk who were randomly paired and asked to act the part of a given provided persona (randomly assigned, and created by another set of crowdworkers). The data collection consists of persona chat (each dialogue has 6-8 turns), personas (set of 1155 possible personas, each consisting of at least 5 profile sentences), revised personas to avoid word overlap, because crowdworkers sometimes could repeat profile information in a chat(example you can see in the Figure 4.2). In turn-based dialogue each message consists of a maximum of 15 words. All statistics are presented in the Table 4.1 Example of Persona-Chat dialogue you can see in Figure 4.1.

4.0.2 Twitter

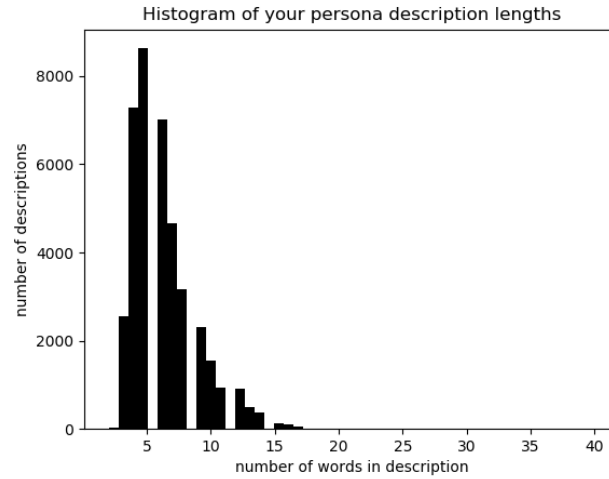Twitter dataset contains message-response pairs from Twitter. Example of these data you can see in the Table 4.2.

Figure 4.3: Histogram of your persona description lengths

Example	
Person 1	yeah i'm preparing myself to drop a lot on this man, but definitely need something reliable
Person 2	yeah dude i would definitely consider a daniel defence super reliable and they are just bad ass
Person 1	besides if trump say his condolences it won't sound genuine, ex: (dwayne wade cousin) it will sound all political and petty
Person 2	yea you right. but we do live in a world where republicans will harass obama about a birth certificate but won't say

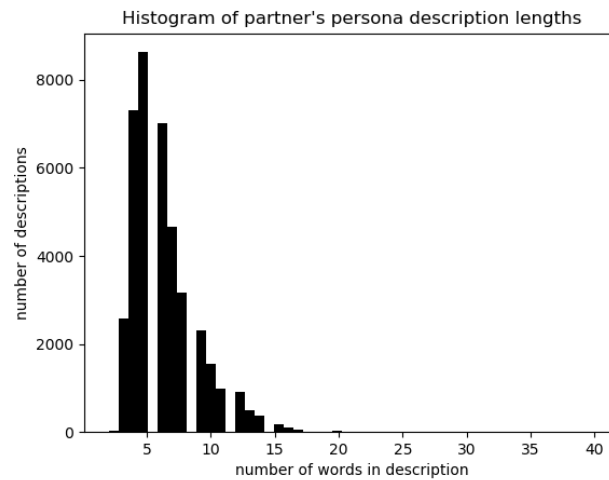Table 4.2: The example of Twitter message-response pairs

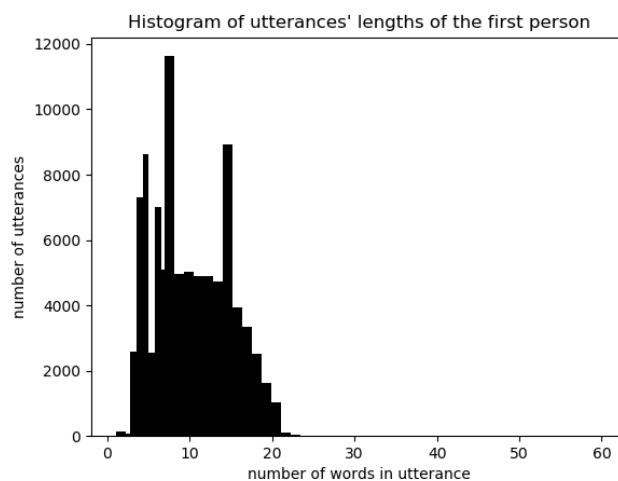Figure 4.4: Histogram of partner's persona description lengths

14

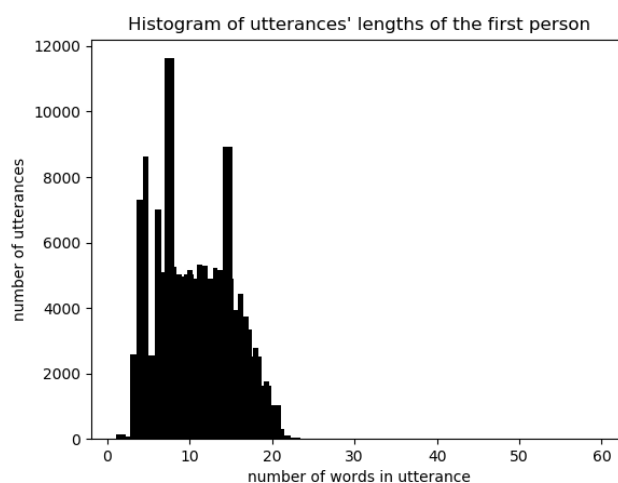Figure 4.5: Histogram of utterances' lengths of the first person

Figure 4.6: Histogram of utterances' lengths of the second person

Chapter 5

Other

Non-task-oriented dialogue system

The aim of task-oriented dialogue systems is to complete specific tasks fo user, non-task-oriented dialogue systems focus on conversing with human on open domains.

//TODO: Info about datasets

//TODO: Info about evaluation metrics

//TODO: Description of baseline

//TODO: NLP vs Computation linguistic

Bibliography

[1] ALDER, H. *Handbook of NLP: A manual for professional communicators.* Routledge, 2017.

[2] CHEN, H., LIU, X., YIN, D. and TANG, J. A survey on dialogue systems: Recent advances and new frontiers. *Acm Sigkdd Explorations Newsletter.* ACM New York, NY, USA. 2017, vol. 19, no. 2, p. 25–35.

[3] CHEN, Y.-N. and GAO, J. *Open-Domain Neural Dialogue Systems.* [Online; visited 10.11.2019]. Available at: https://www.aclweb.org/anthology/I17-5003.pdf.

[4] MANISHINA, E. *Data-driven natural language generation using statistical machine translation and discriminative learning.* Dissertation.

[5] OH, A. H. and RUDNICKY, A. I. Stochastic natural language generation for spoken dialog systems. *Computer Speech & Language.* Elsevier. 2002, vol. 16, 3-4, p. 387–407.

[6] PASCANU, R., MIKOLOV, T. and BENGIO, Y. *On the difficulty of training Recurrent Neural Networks.* [Online; visited 24.11.2019]. Available at: https://arxiv.org/pdf/1211.5063.pdf.

[7] REITER, E. and DALE, R. *Building Applied Natural Language Generation.* [Online; visited 17.11.2019]. Available at: https://pdfs.semanticscholar.org/728e/18fbf00f5a80e9a070db4f4416d66c7b28f4.pdf.

[8] RUDNICKY, A. and OH, A. H. Dialog annotation for stochastic generation. Carnegie Mellon University. 2002.

[9] STENT, A., MARGE, M. and SINGHAI, M. *Evaluating Evaluation Methods for Generation in the Presence of Variation.* [Online; visited 17.11.2019]. Available at: http://www.cs.cmu.edu/~mrmarge/cicling04eval.pdf.

[10] TRAN, V.-K. and NGUYEN, L.-M. *Neural-based Natural Language Generation in Dialogue using RNN Encoder-Decoder with Semantic Aggregation.* [Online; visited 24.11.2019]. Available at: https://arxiv.org/pdf/1706.06714.pdf.

[11] VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L. et al. *Attention Is All You Need.* [Online; visited 24.11.2019]. Available at: https://arxiv.org/pdf/1706.03762.pdf.

[12] WEN, T.-H., VANDYKE, D., MRKŠIĆ, N., GAŠIĆ, M., ROJAS BARAHONA, L. M. et al. *A Network-based End-to-End Trainable Task-oriented Dialogue System.* [Online; visited 24.11.2019]. Available at: https://www.aclweb.org/anthology/E17-1042.pdf.

[13] ZHANG, S., DINAN, E., URBANEK, J., SZLAM, A., KIELA, D. et al. *Personalizing Dialogue Agents: I have a dog, do you have pets too?* [Online, visited 20.01.2020]. Available at: https://arxiv.org/pdf/1801.07243.pdf.