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
January 17, 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

//TODO Write what is NLP

Chapter 2

Dialogue systems

Dialogue system is a computer system to communicate with a human. Nowadays you meet dialog systems everywhere. A lot of devices have incorporated goal-oriented spoken dialogue systems, such as Yandex's Alisa, Apple's Siri, Microsoft's Cortana, Amazon Alexa, and Google Assistant. Dialogue systems are also used in cars (hands-free car-specific functions, Android Auto, Apple CarPlay, vendor-specific solutions), web (search assistants (IKEA), Facebook Messenger and Telegram chatbots), robots, computer games, research systems (skylar.speech.cs.cmu.edu) etc, because a conversation is a natural way for people to get information.

Basic Dialogue System Types:

- Task-oriented
  - focused on completing a certain task(s)
- Non-task-oriented
  - chitchat
  - gaming the Turing test

Communication Domains:
„**Domain**“ is a conversation topic or an area of interest

- Single/Closed-domain is one well-defined area
- Multi-domain is joining several single-domain systems
- Open-domain "responds to anything"

Exists several **modes of communication**:

- Text
- Voice
- Multimodal
  - voice/text + graphics
  - additional modalities: video - gestures, mimics; touch

Dialogue initiative:

- system-initiative
  - system asks questions, user must reply in oreder to progress
  - least natural
  - „form-filling“ („Hello, please enter your e-mail“)
- user-initiative
  - user asks, machine responds („Siri, set the timer for 5 minutes“)
- mixed-initiative
  - system and user both can ask and react to queries
  - most natural

Dialogue system architecture is illustrated in Figure 2.1. 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.[2]

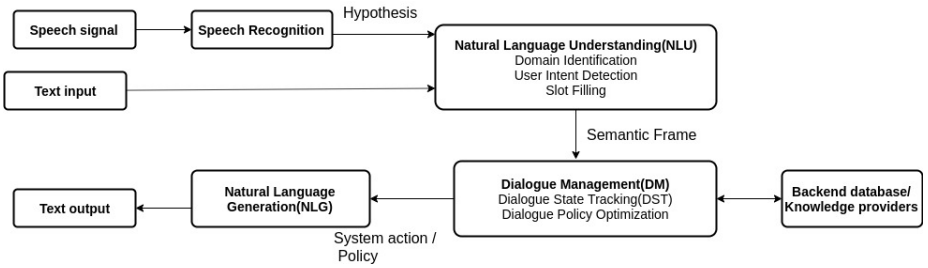

Figure 2.1: Dialogue system architecture.

Chapter 3

Natural Language Generation

Natural Language Generation is a subsection of Natural Language Processing (NLP). NLG uses different methods and tricks to convert data to human language.

3.1 Architecture of NLG

Reiter and Dale in the article [6] proposed architecture splitted into 3 stages, which is illustrated in Figure 3.1.

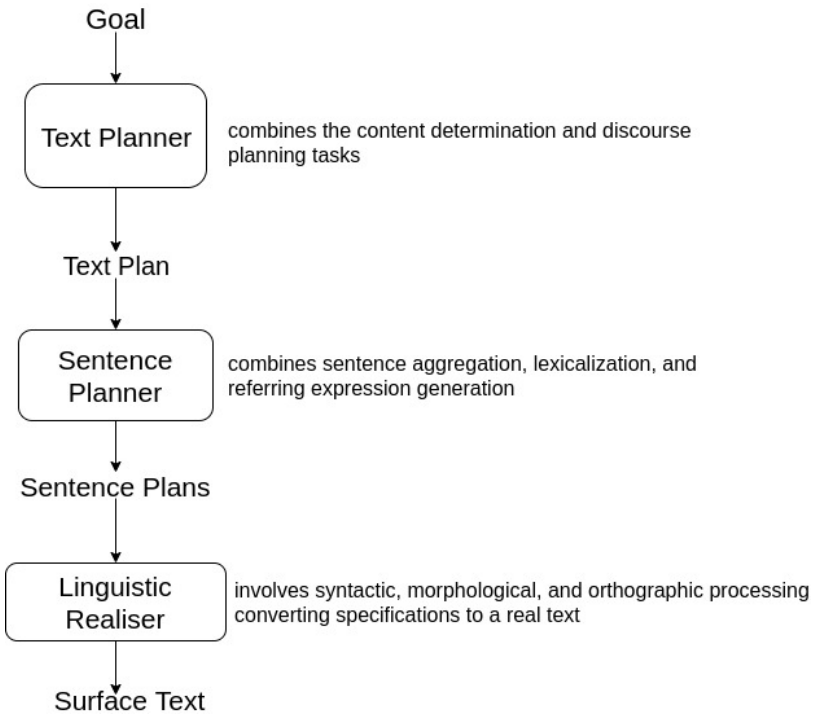

Figure 3.1: NLG architecture.

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

3.2.1 Template-based approach

Template-based systems map their non-linguistic input directly to the linguistic surface structure. This linguistic structure may contain gaps. Well-formed outputs does not cointain gaps. The template-based system selects a proper response for the current conversation from a repository with response selection algorithms.

This approach has a long development way from simple gap-filling to word-level grammatical functions:

Gap-Filling Approach is one of the oldest approaches, that can automatically fill gaps in text, that have a predefined structure, with a small amount of data retrieved from a spreadsheet row, database table entry, etc. This approach is a quite limited.

Scripts or Rules-Producing Text is an approach, that expands basic gap-filling systems with general-purpose programming constructs via a scripting language or by using business rules. These systems still lack linguistic capabilities and cannot generate high-quality text reliably.

Word-Level Grammatical Functions was adding to template-based systems to deal with morphology, morphophonology and orthography. These functions made it easier to generate grammatically correct texts and to write complex template systems.

Advantages of template-based systems

Template-based systems have a number of advantages: the output produced by this approach is likely to be grammatically correct and not contain unexpected generation errors; process of sentence generation is fully controlled; these models are robust and reliable because they consist of clearly defined rules.

Disadvantages of template-based systems

Despite all the above, template-based approach has some disadvantages. First of all, these models requires 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; and finally a template-based model is not able to learn and is not able to adapt to the user.

The gaps represented by [**train**] and [**town**] in example 3.1 are filled by looking up the relevant information in a table. More information you can find in [6].

3.2.2 Corpus-based/Dynamic approach

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

Example
„The 306 train leaves Aberdeen at 11:20 am“
Semantic representation: $Departure(train_{306}, location_{abdn}, time_{1120})$
Template: [train] *is leaving* [town] *now*

Table 3.1: The example of gaps in the sentence.

semantic units and their respective surface realization. In corpus-based systems include statistical and machine learning approaches.

One of the first approaches in corpus-based methods was **Dynamic Sentence Generation**, that 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 was **Dynamic Document Creation**, what can produce relevant and well-structured document.

Advantages of corpus-based approach

Corpus-based models have some advantages: they 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 varioations of output.

Disadvantages of corpus-based approach

Unfortunately corpus-based systems have some disadvantages. It is necessary to have a good corpus, it should contain a large amount of data and on a variety of topics to get a sensible output. Even if you have a good corpus, process of text generation is not fully controlled and the output can be grammatically incorrect.

More information you can find in [4] [1] [3].

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.

One of the first model's implementation was **Markov chain**. 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).

The next model, what is used in NLG, is **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 3.2, 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 [10] and a training model with semantic aggregation [8].

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 [5].

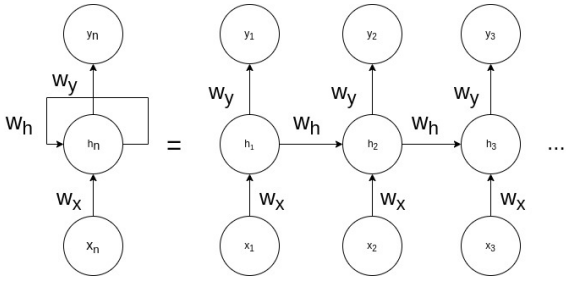

Figure 3.2: Architecture of recurrent neural network.

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 3.3 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.

Transformer is a relatively new architecture, which proposes a „self-attention mechanism". This model was introduced in the paper „Attention is all you need" [9] 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 3.4. 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

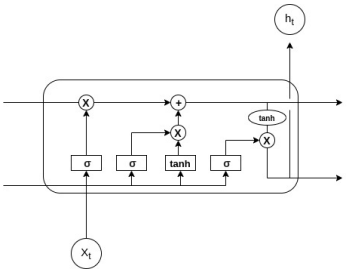

Figure 3.3: A cell in an LSTM network.

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.

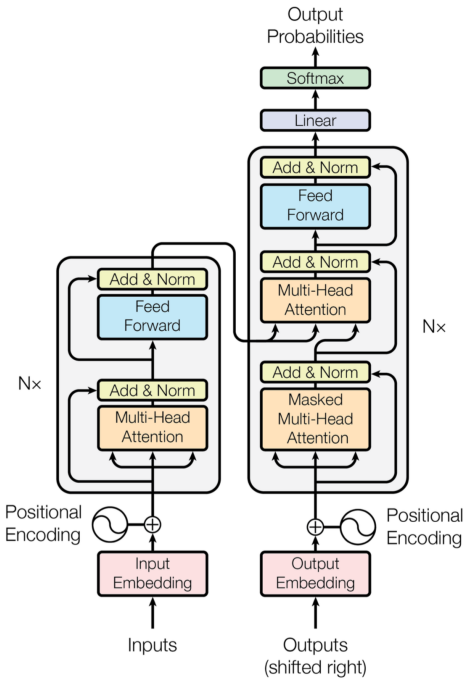

Figure 3.4: The architecture of the Transformer[9]

NLG problems

The NLG component converts an abstract dialogue action into natural language surface utterances. As noticed in [7], 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.)
- **fluency** (syntactic correctness)

Example	
1	I bought new dress on Tuesday.
2	I got new dress on Tuesday.
3	On Tuesday I bought new dress.
4	On new Tuesday dress I bought

Table 3.2: The example of sentences' variation

- **readability** (efficacy in context)
- **variation** (there are 2 basic forms of variation: *word choice variation* and *word order variation* for enriching speech.)

It is very important to have all factors in generator, because sentences that are both adequate and fluent may still not be adequate or fluent in a particular context, so they are not readable. In example 3.2 the sentence number 4 shows, that a variation can be syntactically incorrect or unclear.

Chapter 4

Other

Oh and Rudnicky showed that stochastic generation benefits from two factors:

- it takes advantage of the practical language of a domain expert instead of the developer
- it restates the problem in terms of classification and labeling, where expertise is not required for developing a rule-based generation system

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: more information in article [] //TODO: Info about datasets //TODO: NLP vs Computation linguistic

//TODO: In non task oriented dialogue systems it is very difficult to use template-based generation [4]

Bibliography

[1] Chen, H., Liu, X., Yin, D. and Tang, J. *A Survey on Dialogue Systems: Recent Advances and New Frontiers.* [Online; visited 17.11.2019]. Available at: https://arxiv.org/pdf/1711.01731.pdf.

[2] 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.

[3] Manishina, E. *Data-driven natural language generation using statistical machine translation and discriminative learning.* [Online; visited 23.11.2019]. Available at: https://tel.archives-ouvertes.fr/tel-01398776.

[4] Oh, A. H. and Rudnicky, A. I. *Dialog Annotation for Stochastic Generation.* [Online; visited 10.11.2019]. Available at: https://acl-arc.comp.nus.edu.sg//~antho/W/W00/W00-0306.pdf.

[5] 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.

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.