FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

DOCTORAL THESIS SUMMARY

Ondřej Dušek

Novel Methods for Natural Language Generation in Spoken Dialogue Systems

Institute of Formal and Applied Linguistics

Supervisor: Ing. Mgr. Filip Jurčíček, Ph.D.

Study Program:	Computer Science
Specialization:	Computational Linguistics

Prague 2017

The results comprised in this thesis were obtained during the candidate's doctoral studies at the Faculty of Mathematics and Physics, Charles University in the years 2012–2017.

Candidate: Ondřej Dušek

Supervisor: Ing. Mgr. Filip Jurčíček, Ph.D.

Department: Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics
Charles University
Malostranské náměstí 25
118 00 Praha 1, Czech Republic

Reviewers: Doc. Ing. Pavel Ircing, Ph.D.
Department of Cybernetics
Faculty of Applied Sciences
University of West Bohemia in Pilsen
Univerzitní 8
306 14 Plzeň, Czech Republic

Doc. Ing. Zdeněk Žabokrtský, Ph.D.
Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics
Charles University
Malostranské náměstí 25
118 00 Praha 1, Czech Republic

The defense of this thesis will be held on June 12, 2017 from 10:00 a.m. before the thesis defense committee for Specialization 4I3 (Computational Linguistics) at the Faculty of Mathematics and Physics, Charles University, Malostranské náměstí 25, Praha 1, in the room S1 (4th floor).

4I3 Board Chair: Doc. Ing. Zdeněk Žabokrtský, Ph.D.
Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics
Charles University
Malostranské náměstí 25
118 00 Praha 1, Czech Republic

The thesis is available for perusal at the Student Affairs Department of Faculty of Mathematics and Physics, Ke Karlovu 3, Praha 2.

This summary has been distributed on May 29, 2017.

FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

AUTOREFERÁT DISERTAČNÍ PRÁCE

Ondřej Dušek

Nové metody generování promluv v dialogových systémech

Ústav formální a aplikované lingvistiky

Vedoucí práce: Ing. Mgr. Filip Jurčíček, Ph.D.

Studijní program:	Informatika
Studijní obor:	Matematická lingvistika

Praha 2017

Disertační práce byla vypracována na základě výsledků získaných během doktorského studia uchazeče na Matematicko-fyzikální fakultě Univerzity Karlovy v letech 2012–2017.

Uchazeč:	Ondřej Dušek
Vedoucí práce:	Ing. Mgr. Filip Jurčíček, Ph.D.
Školící pracoviště:	Ústav formální a aplikované lingvistiky Matematicko-fyzikální fakulta Univerzita Karlova Malostranské náměstí 25 118 00 Praha 1
Oponenti:	Doc. Ing. Pavel Ircing, Ph.D. Katedra kybernetiky Fakulta aplikovaných věd Západočeská univerzita v Plzni Univerzitní 8 306 14 Plzeň
	Doc. Ing. Zdeněk Žabokrtský, Ph.D. Ústav formální a aplikované lingvistiky Matematicko-fyzikální fakulta Univerzita Karlova Malostranské náměstí 25 118 00 Praha 1

Obhajoba práce se koná dne 12. června 2017 od 10:00 hodin před komisí pro obhajoby disertační prací v oboru 4I3 (Matematická lingvistika) v budově Matematicko-fyzikální fakulty Univerzity Karlovy, Malostranské náměstí 25, Praha 1, v místnosti S1 (4. patro).

Předseda RDSO 4I3:	Doc. Ing. Zdeněk Žabokrtský, Ph.D. Ústav formální a aplikované lingvistiky Matematicko-fyzikální fakulta Univerzita Karlova Malostranské náměstí 25 118 00 Praha 1

S disertační prací je možno se seznámit na Studijním oddělení Matematicko-fyzikální fakulty, Ke Karlovu 3, Praha 2.

Autoreferát byl rozeslán dne 29. května 2017.

Contents

Table of Contents **v**

1 Introduction **1**
1.1 Objectives and Contributions 1

2 State of the Art: Adaptive Methods in NLG **2**
2.1 The NLG Pipeline 2
2.2 Handcrafted and Trainable Methods 3
2.3 NLG Training Datasets 4

3 Decomposing the Problem **4**
3.1 The Input Meaning Representation 4
3.2 Using Unaligned Data 5
3.3 Delexicalization 5
3.4 Separating the Stages 6
3.5 Evaluation Metrics 7

4 Experiments in Surface Realization **8**
4.1 Constructing a Rule-based Surface Realizer for English 8
4.2 Statistical Morphology Generation 9

5 Perceptron-based Sentence Planning **10**
5.1 Sentence Planner Architecture 11
5.2 Experiments 12

6 Sequence-to-Sequence Generation Experiments **13**
6.1 The Seq2seq Generation Model 13
6.2 Experiments 15

7 Generating User-adaptive Outputs **16**
7.1 Collecting a Context-Aware NLG Dataset 16
7.2 Context-aware Seq2seq Generator Extensions 17
7.3 Experiments 18

8 Generating Czech **18**
8.1 Creating an NLG Dataset for Czech 19
8.2 Generator Extensions 20
8.3 Experiments 20

9 Conclusions **21**

References 24

List of Abbreviations 29

List of Author's Publications 30

1 Introduction

Natural language generation (NLG), a conversion of an abstract meaning representation into a natural language utterance, is an integral part of various natural language processing (NLP) applications, including spoken dialogue systems (SDSs) – computer interfaces allowing users to perform various tasks or request information using spoken dialogue. In SDSs, the task of NLG is to convert an abstract representation of the system's response into a natural language sentence, which is read to the user using a text-to-speech synthesis module. NLG is thus responsible for accurate, comprehensible, and natural presentation of information provided by the SDS and has a significant impact on the overall perception of the system by the user.

The main motivation for this work has been the relative lack of statistical approaches in NLG for SDSs that are practically usable: The adoption of statistical NLG in SDSs mostly remained limited until very recently, and the NLG component was often reduced to a simple template-filling approach. Although statistical approaches to NLG have advanced greatly during the past year or two with the advent of neural network (NN) based systems, they still leave room for improvement in terms of naturalness, adaptability, and linguistic insight.

1.1 Objectives and Contributions

The main aim of the present thesis is to explore the usage of statistical methods in NLG for SDSs and advance the state-of-the art in naturalness and adaptability. We focus on enabling fast reuse in new domains and languages, and we aim at adapting the structure and lexical choice in generated sentences to the communication goal, to the current situation in the dialogue, and to the particular user (e.g., by aligning vocabulary to the expressions uttered by the user). This work thus not only brings a radical improvement over NLG systems based on handwritten rules or domain-specific templates, but also represents an important contribution to recent works in statistical NLG by experimenting with deep-syntactic generation, multilingual NLG, and user-adaptive models.

Our experiments, and also the main contributions of this thesis, proceed along the following key objectives:

A) Generator easily adaptable for different domains. We create a generator that can be fully and easily retrained from data for a given domain. Unlike previous methods, our generator does not require fine-grained alignments between elements of the input meaning representation and output words and phrases. We will show two different novel approaches to NLG trainable from unaligned data.

B) Generator easily adaptable for different languages. We adapt a rule-based surface realizer to a new language and simplify it by introducing statistical components. In addition, we experiment with fully statistical NN-based NLG on both English and Czech for the first time.

C) Generator that adapts to the user. We create a first fully trainable context-aware NLG system that is able to adapt the generated responses to the wording and syntax of the user's requests.

D) Comparing different NLG system architectures. We experiment with both major approaches used in modern NLG systems, pipeline (separating high-level sentence structuring from surface grammatical rules) and joint (end-to-end), and we compare their results on the same dataset.

E) Dataset availability for NLG in SDSs. We address the limited availability of datasets for NLG in task-oriented SDSs by collecting and publicly releasing two different novel datasets: the first dataset for training context-aware NLG systems and the first Czech NLG dataset.

2 State of the Art: Adaptive Methods in NLG

In this chapter, we give an introduction into the problem of NLG and briefly discuss previous architectures and available training datasets.

In general, NLG is defined as the task of presenting information in natural language: Given input data and a communication goal (e.g., to describe the data or receive user reaction), the system should produce a natural language string that is relevant, well-formed, grammatically correct, and fluent (Dale et al., 1998).

2.1 The NLG Pipeline

The standard "textbook" description of an NLG system (Reiter and Dale, 2000) involves a pipeline consisting of three main phases:

1. *Content planning* – selecting relevant content from the input data and basic structuring of this content,
2. *Sentence planning* (also called *microplanning*) – detailed sentence shaping and expression selection,
3. *Surface realization* – linearization of the sentence plan according to the grammar of the target language.

Most NLG systems follow the standard pipeline more or less closely, but only a few of them implement it as a whole. Many generators focus only on one of the phases while using a very basic implementation of the other or leaving it out completely. NLG systems in SDSs often let the dialogue manager handle content planning and focus only on sentence planning and surface realization.

Some NLG systems choose to replace the pipeline with a joint, end-to-end architecture (e.g., Angeli et al., 2010; Mairesse et al., 2010). Both approaches can offer their own advantages: Dividing the problem of NLG into several subtasks makes the individual subtasks simpler. A sentence planner can abstract away from complex surface syntax and morphology and only concern itself with a high-level sentence structure. It is also possible to reuse third-party modules for parts of the generation pipeline (Walker et al., 2001). On the other hand, the problem of pipeline approaches in general is error propagation. In addition, joint methods do not need to model intermediate structures explicitly (Konstas and Lapata, 2013).

2.2 Handcrafted and Trainable Methods

Traditional NLG systems are based on procedural rules (Bangalore and Rambow, 2000; Belz, 2005; Ptáček and Žabokrtský, 2007), template filling (Rudnicky et al., 1999; van Deemter et al., 2005), or grammars in various formalisms. Such rule-based generators are still used frequently today. Their main advantages are implementation simplicity and speed, but many rule-based systems struggle to achieve high coverage in larger domains (White et al., 2007) and are not easy to adapt for different domains and/or languages. Rule-based systems also tend to exhibit little variation in the output, which makes them appear repetitive and unnatural.

Various approaches have been taken to make NLG output more flexible and natural as well as to simplify its reuse in new domains. While statistical methods and trainable modules in NLG are not new (cf. Langkilde and Knight, 1998), their adoption has been slower than in most other subfields of NLP. Several research paths were pursued for statistical NLG in the last decade; most of them focus on just one of the generation stages or on enhancing the capabilities of an existing rule-based generator, e.g., by introducing trainable parameters (Paiva and Evans, 2005; Mairesse and Walker, 2008). Fully trainable statistical NLG (Mairesse et al., 2010; Angeli et al., 2010) has been rare. Only in the past year or two, new fully trainable NN-based generators (e.g., Wen et al., 2015b,a, but also work described in this thesis) have been dominating the field.

2.3 NLG Training Datasets

The number of publicly available datasets suitable for NLG experiments is rather small, compared to other areas of NLP. Publicly available datasets are more common in text-based NLG than in NLG for SDSs (Sripada et al., 2003; Wong and Mooney, 2007; Liang et al., 2009). However, most of text-based NLG sets assume a content selection step, which is not applicable to our work.

Publicly available corpora for NLG in SDSs have been up until now very scarce: Mairesse et al. (2010) published a dataset of 404 restaurant recommendations, which includes detailed semantic alignments (see Section 3.2). Wen et al. (2015b,a) present two similar sets for restaurant and hotel information domains, both containing over 5,000 instances but lots of repetition. Similar but larger and more diverse datasets for laptop and TV recommendation domains have been released recently by Wen et al. (2016b), who focus on domain adaptation.

3 Decomposing the Problem

This chapter provides a methodological background for all our experiments in Chapters 4 through 8: it is concerned with a closer definition of the task that we are solving, as well as with defining some of the basic aims and features common to all NLG systems developed in the course of this thesis.

3.1 The Input Meaning Representation

Throughout our experiments in this thesis, we use a version of the dialogue act (DA) meaning representation (Young et al., 2010; Jurčíček et al., 2014; Wen et al., 2015a). Here, a DA is simply a list of triplets (DA items) in the following form:

- *DA type* – the type of the utterance or a dialogue act per se, e.g., *hello*, *inform*, or *request*.

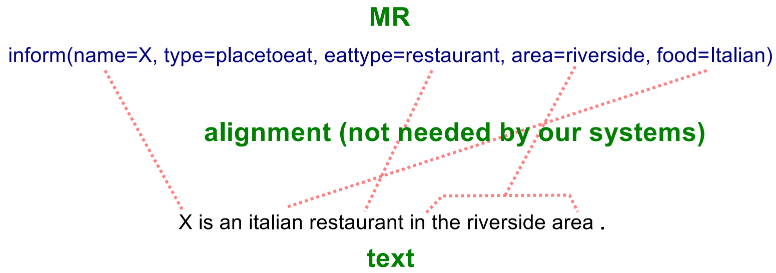

Figure 3.1: A training data instance for NLG from dialogue acts, with manual fine-grained alignments, which are not needed for our generators.

- *slot* – the slot (domain attribute) that the DA is concerned with, e.g., *departure_time* or *price_range*.
- *value* – the particular value of the slot in the DA item.

The latter two members of the triplet can be optional (or null). For instance, the DA type *hello* does not use any slots or values, and the DA type *request* uses slots but not values since it is used to request a value from the user. DA items with identical DA type are joined in figures for brevity (see Figure 3.1).

3.2 Using Unaligned Data

In all our experiments, we use *unaligned* pairs of input DAs and output sentences. This simplifies training data acquisition: Previous NLG systems usually required a separate training data alignment step (Mairesse et al., 2010; Konstas and Lapata, 2013), and this is now no longer needed since our sentence planners learn alignments jointly with learning to generate (see Figure 3.1). In addition, alignments are not decided by hard, binary decisions, which allows for a more fine-grained modeling.

3.3 Delexicalization

In all our experiments, we use *delexicalization* – the replacing of some values, such as restaurant names or time constants, with placeholders (see Figure 3.2). The generator then only works with these placeholders, which are replaced with the respective values in a simple postprocessing stage. This helps to reduce data sparsity issues and improves generalization to unseen slot values since

inform(name="Gourmet Burger Kitchen", type=placetoeat,
eattype=restaurant, area="city centre", near="Tatties (Trinity Street)",
food="Cafe food", food=English)

Gourmet Burger Kitchen is an English and cafe food restaurant in the city centre near Tatties (Trinity Street).

inform(name=**X-name**, type=placetoeat, eattype=restaurant,
area="city centre", near=**X-near**, food="Cafe food", food="English")

X-name is an English and cafe food restaurant in the city centre near **X-near**.

Figure 3.2: Delexicalization example (from the BAGEL dataset).

From top to bottom: lexicalized DA, lexicalized sentence, delexicalized DA, delexicalized sentence. Placeholders in delexicalized items are highlighted.

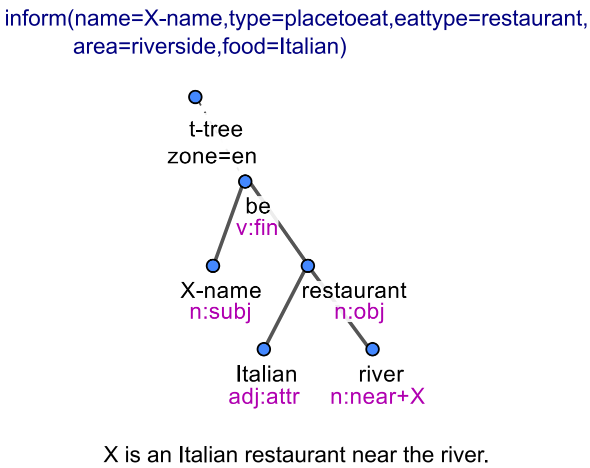

Figure 3.3: Example t-tree (middle, t-lemmas in black and formemes in purple), with the corresponding DA (top) and natural language paraphrase (bottom).

the possible number of values for some slots is unbounded in theory, and most values are only seen once or never in the training data.

Note that delexicalization is different from using full, fine-grained semantic alignments (see Section 3.2) and can easily be obtained automatically using simple string replacement rules as the values to be delexicalized occur verbatim in training data (possibly in an inflected form for Czech, see Chapter 8).

3.4 Separating the Stages

We will explore both approaches to NLG sketched in Section 2.1: two-step generation with separate sentence planning and surface realization steps and joint, end-to-end, one-step direct generation. We believe that both avenues have their own advantages and disadvantages (cf. Section 2.1), and that both of them should be explored.

We opted for using sentence plans in the form of simplified deep syntactic trees (tectogrammatical trees or t-trees) based on the Functional Generative Description (Sgall et al., 1986) as the intermediate data representation between the stages (sentence plan) due to three main reasons: t-tree surface realization is viable and straightforward (Ptáček and Žabokrtský, 2007; Žabokrtský et al., 2008; see Chapter 4), there are efficient algorithms for projective tree structures such as t-trees, and automatic domain-independent parsers into t-trees for several languages are available (Popel and Žabokrtský, 2010).

The t-tree sentence plan structure is a deep-syntactic dependency tree that only contains nodes for content words (nouns, full verbs, adjectives, adverbs) and coordinating conjunctions (see Figure 3.3). The nodes maintain surface word order. Each node has several attributes; the most important ones for our experiments are *t-lemma* or deep lemma (base word form of the content word) and *formeme* (a morphosyntactic label describing the word form).

3.5 Evaluation Metrics

Several different approaches have been applied to evaluating NLG (Hastie and Belz, 2014; Gkatzia and Mahamood, 2015): intrinsic, further divided into automatic scores and human ratings, and extrinsic, such as users' success in completing a task based on information provided by the NLG output. In this work, we limit ourselves to evaluating our systems intrinsically.

Automatic intrinsic NLG evaluation typically uses metrics developed for machine translation (MT) which are based on word-by-word comparisons against reference texts, measuring word overlap. This approach is cheap and fast, but correspondence to human judgments has been disputed (Stent et al., 2005; Callison-Burch et al., 2006). Manual human evaluation provides a more accurate estimate of an NLG system's performance, but requires much more resources. Both approaches are therefore combined in practice.

For automatic metrics, we use BLEU (Papineni et al., 2002) and NIST (Doddington, 2002) to evaluate our experiments, two of the oldest and arguably the most widely used metrics for NLG. In addition, we apply a complementary metric that is only applicable to delexicalized NLG: *slot error rate* which estimates the number of semantic errors based on counting DA value placeholders in the generated output (Wen et al., 2015a). For human evaluation, our task is to decide which system variant will provide outputs preferable to users. Therefore, we focus on direct comparisons of outputs generated for the same input DA, asking users which variant is better/preferred. This promises more consistent and efficient ratings than using Likert scales or multiple judgment criteria (Callison-Burch et al., 2007; Koehn, 2010, p. 220).

4 Experiments in Surface Realization

This chapter is an account of our own experiments with surface realization – generating natural language sentences from t-trees (cf. Section 3.4). Based on a similar module for Czech, we developed a new general-domain, mostly rule-based surface realizer for English, which is used in our experiments with full generation from DAs in Chapters 5 and 6. We also introduced into the realizer pipeline a new statistical module for morphological inflection (called *Flect*) and show that it improves on dictionary-based modules.

4.1 Constructing a Rule-based Surface Realizer for English

Our English surface realizer was developed within the Treex NLP framework (Popel and Žabokrtský, 2010), where it mostly adapts Czech realizer pipeline modules (Žabokrtský et al., 2008; Popel, 2009, p. 84ff.) and shares their language-independent code components. It starts from a copy of the input t-tree, gradually transforming it into a surface dependency tree, which is then linearized (see Figure 4.1). It handles all the important surface language phenomena: auxiliary words, inflection, word order, agreement, punctuation, and capitalization.

To evaluate the realizer on a broad domain, we ran a round-trip test: We first automatically analyzed English texts into t-trees using Treex, then ran our surface realizer to regenerate texts and evaluated the results using BLEU score (Papineni et al., 2002) against the originals. On texts from the Prague Czech-English Dependency Treebank 2.0 (Hajič et al., 2012), the realizer reached a BLEU of 77.47%. This score is relatively high given that the original is used as the only reference and even minor deviations are penalized.

Our realizer has been successfully applied in our NLG experiments in Chapters 5 and 6 as well as in TectoMT translation systems translating into English from Czech, Dutch, Spanish, and Basque (Rosa et al., 2015; Popel et al., 2015).

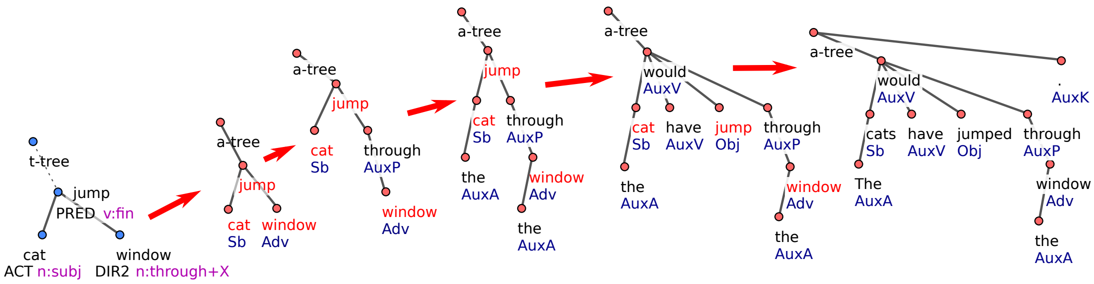

Figure 4.1: Rule-based surface realization pipeline example.

The t-tree for the sentence "The cats would have jumped through the window." is gradually transformed into a surface dependency tree (a-tree). Uninflected words are shown in red in a-trees, dependency labels are shown in blue. From the left: (1) morphological attributes are determined, word order and agreement are enforced. (2 and 3) prepositions and articles are added. (4) auxiliary verbs are added. (5) punctuation is added, words are inflected, and sentence start is capitalized.

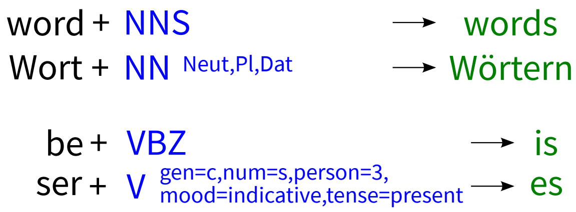

Figure 4.2: The task of morphological generation is to create a fully inflected form (right) from a base word form and morphological information (left).

4.2 Statistical Morphology Generation

To simplify surface realizer development, we introduced a new statistical module for word inflection generation, i.e., deducing the inflected word form given its lemma (base form) and the desired morphological properties (see Figure 4.2). There are three traditional approaches to this: avoiding inflection altogether, rule-based methods, and dictionary-based methods. Avoiding inflection often leads to unnatural results, rules have scalability issues, and dictionary-based methods cannot generalize to previously unseen word forms. Our solution, dubbed *Flect*, manages to produce natural inflection and is easily trainable for different languages and capable of generalizing to unseen inputs.

Similarly to Bohnet et al. (2010) and Durrett and DeNero (2013), we reformulate the task of finding the correct word form as a multiclass classification problem. Instead of finding the desired word form directly (which would in-

Accuracy (%)	English	Czech	German	Spanish	Catalan	Japanese
Baseline	98.94	92.88				
Flect	99.56	99.45	96.46	99.01	98.72	99.94

Table 4.1: Morphology generation results on CoNLL 2009 datasets.

The table shows a percentage of correctly predicted inflected word forms. *Baseline* is a simple dictionary learned from the same data, where unknown words are left uninflected.

duce an explosion of possible target classes), the classifier is trained to find the correct inflection pattern: *lemma-form edit scripts* – rules describing how to transform the base form into the inflected form – are used as target classes.

We used the LIBLINEAR logistic regression classifier (Fan et al., 2008) with the following main feature types: lemma, part-of-speech tag, other morphological features, and lemma suffixes of up to 4 characters. The last feature type allows the classifier to generalize to unknown lemmas since inflection depends mostly on suffixes in many languages.

We evaluated our Flect morphology generator on six languages using the CoNLL 2009 Shared Task data sets (Hajič et al., 2009), and compared it to a simple dictionary baseline for English and Czech (see Table 4.1). We can see that Flect is able to predict the majority of word forms correctly and significantly improves over a dictionary baseline by generalizing to word forms unseen in the training set. The lower score for German is caused partly by insufficient information in the morphological tags.

We also integrated Flect into our English surface realizer, where it replaced a handcrafted morphological dictionary (Straková et al., 2014), gaining over 3.5% BLEU improvement in the round trip test described in Section 4.1.

5 Perceptron-based Sentence Planning

In this chapter, we present our first experiments with a novel, fully trainable approach to sentence planning based on A*-search and perceptron ranking. This approach has since been superseded by a more flexible and better-performing NN-based generator (see Chapter 6), but it advanced the state-of-the art as the first approach where fine-grained semantic alignments were not required for training (see Section 3.2) – our sentence planner includes alignment learning directly into the training process. In addition, unlike most previous approaches

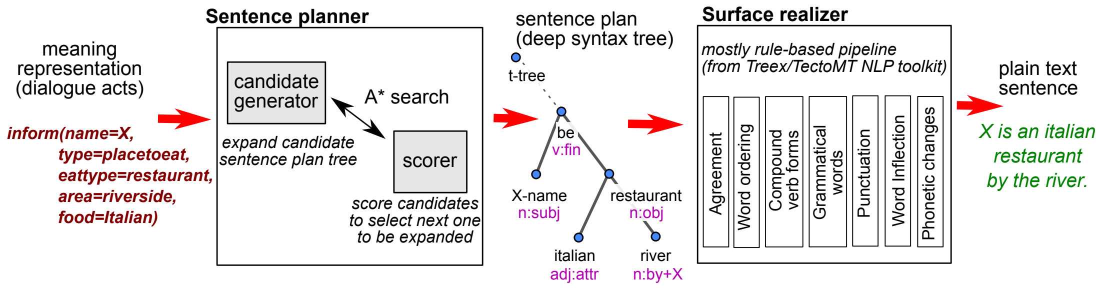

Figure 5.1: Overall structure of our generator.

to trainable sentence planning (e.g., Walker et al., 2001; Stent et al., 2004), our system does not require a handcrafted base module.

The overall schema of the whole generation procedure is depicted in Figure 5.1. First, the sentence planner, which is described in this chapter, generates t-tree sentence plans from the input DAs (see Section 3.4). We then apply the surface realizer described in Chapter 4 to convert the sentence plans to plain text sentences.

5.1 Sentence Planner Architecture

The sentence planner is based on a variant of the A* algorithm (Hart et al., 1968; Och et al., 2001; Koehn et al., 2003). It starts from an empty sentence plan tree and tries to find a path to the complete, optimal sentence plan by iteratively adding nodes to the currently "most promising" incomplete sentence plan. It uses the following two subcomponents to guide the search:

- a *candidate generator* that incrementally generates new candidate sentence plan trees (expanding incomplete sentence plans by adding new nodes),
- a *scorer/ranker* that scores the appropriateness of the sentence plan trees for the input DA and selects the next sentence plan tree to be expanded.

At each step, expansions of the currently best-ranking sentence plan tree are created by adding one node of all viable types and in all viable positions. The expansions are subsequently ranked. The algorithm continues as long as the best-ranking candidate sentence plan score keeps increasing.

The basic scorer for the sentence plan tree candidates is based on the linear perceptron ranker of Collins and Duffy (2002), where the score is computed as a dot product of the features and the corresponding weight vector. Features

Setup	**BLEU**	**NIST**
Basic perceptron updates	54.24	4.643
+ Differing subtree updates	58.70	4.876
+ Future promise	59.89	5.231

Table 5.1: Automatic evaluation on the BAGEL data set
BLEU numbers are shown as percentages. Numbers are averaged over all 10 cross-validation folds.

include the candidate tree shape, nodes and their combinations, as well as conjunctions with items of the input DA. During training, weight vector update is performed if the score of the top-ranking generated tree for a given DA is higher than that of the corresponding the gold-standard tree.

The basic scorer is trained to score full sentence plan trees, but it is also used to score incomplete sentence plans during the decoding, which leads to a bias towards bigger trees. To outweigh this bias, we introduced a novel modification of the perceptron updates to improve scoring of incomplete sentence plans: In addition to updating the weights using the full top-scoring candidate and the gold-standard tree, we also use their *differing subtrees* for extra perceptron updates.

Moreover, to further boost scores of incomplete sentence plans that are expected to further grow, we add a *future promise* term to the sentence plan scores, based on the expected number of children of different node types (with different lemma-formeme combinations).

5.2 Experiments

We performed our experiments on the BAGEL data set (Mairesse et al., 2010) in the restaurant information domain. Note that while the data set contains fine-grained semantic alignment, we do not use it in our experiments. We use 10-fold cross-validation – same as Mairesse et al. (2010) – and evaluate our generator using the automatic BLEU and NIST scores (Papineni et al., 2002; Doddington, 2002). The results are shown in Table 5.1.

Our generator did not achieve the same performance as that of Mairesse et al. (2010) (ca. 67% BLEU). However, our task is substantially harder since the generator also needs to learn the alignment of words and phrases to DA items and determine whether all required information is present on the output (see

Section 3.2). Our differing tree updates clearly bring a substantial improvement over standard perceptron updates; using future promise estimation boosts the scores even further. Both improvements on the full training set are considered statistically significant at 95% confidence level by the paired bootstrap resampling test (Koehn, 2004).

The generator learns to produce meaningful utterances which mostly correspond well to the input DA. It is able to produce original paraphrases and generalizes to previously unseen DAs. On the other hand, the outputs are not free of semantic errors (missing, repeated, or irrelevant information).

6 Sequence-to-Sequence Generation Experiments

With the recent emergence of models based on recurrent neural networks (RNNs) for various tasks in NLP, most notably sequence-to-sequence (seq2seq) models with attention for MT (Cho et al., 2014; Sutskever et al., 2014) and first RNN-based NLG approaches (Wen et al., 2015b,a), we understood the power of this approach and decided to adapt seq2seq generation to our task. Our new generator uses the seq2seq generation technique combined with beam search and an n-best list reranker to suppress irrelevant information in the outputs. The new model is more flexible than most previous solutions including the A*-search-based generator presented in Chapter 5 as it requires neither fine-grained alignments between DA items and words/phrases in training data (Mairesse et al., 2010), nor a handcrafted base generator (Stent et al., 2004), nor handcrafted features (as our A*-search-based generator). In addition, it yields significantly better results than our previous generator.

We improve upon previous RNN-based generators (Wen et al., 2015b,a; Mei et al., 2016) in two ways: First, we are able compare two-step generation (sentence planning and surface realization) with a joint, one-step approach in a single architecture (cf. Section 3.4): our seq2seq generator either generates t-trees, which are subsequently processed by the surface realizer described in Chapter 4, or it produces natural language strings directly. Second, we show that our system can be trained successfully using much less training data than previous RNN-based approaches.

6.1 The Seq2seq Generation Model

Our generator is based on the seq2seq model with attention (Bahdanau et al., 2015), a type of an encoder-decoder RNN architecture operating on variable-

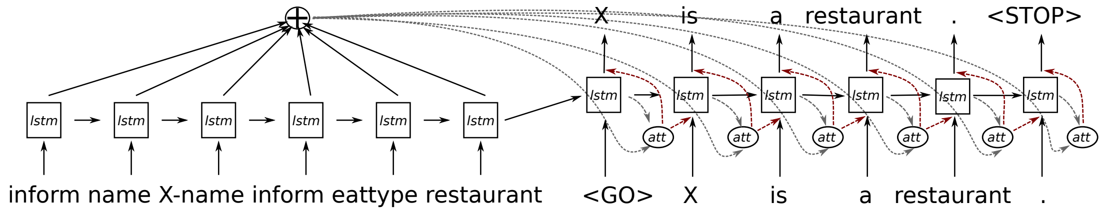

Figure 6.1: The main seq2seq generator with attention.

Left part: encoder, with encoder hidden outputs concatenated to use for the attention model. Right part: decoder; dotted lines indicate data flow in the attention model.

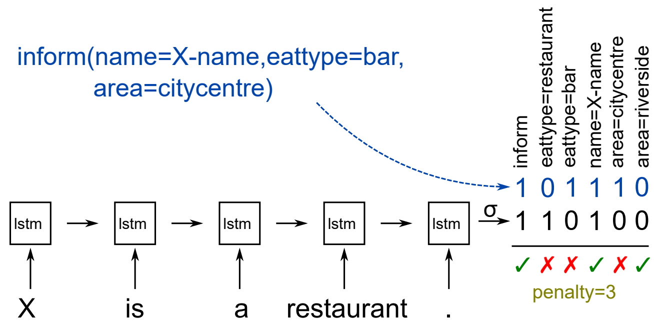

Figure 6.2: The n-best list reranker for system outputs: DA classification (RNN + sigmoid binary classification layer) and comparison with the source DA.

length sequences of tokens (see Figure 6.1). First, its encoder RNN consumes the input token by token and encodes it into a sequence of hidden states (vectors of floating-point numbers). The decoder then generates output tokens one-by-one, using as inputs its own internal state (initialized by the last encoder hidden state and updated in every step), the previously decoded token, and the attention context vector (a weighted sum of all encoder hidden states).

DAs, t-trees, and sentences are represented as sequences of tokens to enable their usage in the sequence-based generator – DAs are encoded as lists of triples "DA type – slot – value", t-trees use a simple bracketed notation. All tokens in turn are represented by their embeddings – vectors of floating-point numbers initialized randomly and trained from data (Bengio et al., 2003).

On top of this basic seq2seq architecture, we use beam search for decoding (Sutskever et al., 2014; Bahdanau et al., 2015) and a reranker that penalizes outputs which miss some information from the input DA or add irrelevant information. The reranker uses a RNN encoder over the outputs and a final sigmoid layer which provides a binary decision on the presence of different DA

Setup	BLEU	NIST	SemErr
Mairesse et al. (2010) with fine-grained alignments	∼67	-	0
Best A*-search-based result (Chapter 5)	59.89	5.231	30
Greedy generation in a 2-step setup with t-trees	55.29	5.144	20
+ Beam search	58.59	5.293	28
+ Reranker	60.44	5.514	19
Greedy direct generation of strings	52.54	5.052	37
+ Beam search	55.84	5.228	32
+ Reranker	62.76	5.669	19

Table 6.1: Results of our seq2seq generator on the BAGEL data set.
NIST, BLEU, and semantic errors in a sample of the output. Beam size is set to 100.

items (DA types, slot-value pairs). This is compared to the items in the input DA and the number of discrepancies for a particular output is used to lower its probability on the output n-best list (see Figure 6.2).

6.2 Experiments

Same as in Chapter 5, we perform our experiments on the BAGEL data set (Mairesse et al., 2010), without using the fine-grained semantic alignments.

The results of our experiments are shown in Table 6.1. We include BLEU and NIST scores and the number of semantic errors (missing, added, or repeated information) counted manually on a sample of the outputs. A manual inspection of the outputs shows that both tree-based and joint setup are able to produce fluent sentences in the domain style for the most part. The occasional errors are of different types in the two setups: while the joint setup confuses semantically close items such as *Italian* and *French* cuisine, the syntax-generating model produces outputs with missing or repeated information more often.

A comparison of the two approaches goes in favor of the joint setup, which offers better performance and does not need an external surface realizer. Both setups surpass the previous best results achieved in Chapter 5.[1]

We also trained our system on the larger restaurant dataset of Wen et al. (2015a) to perform a direct comparison of our system's performance to theirs. Our system performed comparably, offering slightly lower BLEU score (72.7%

[1]The BLEU/NIST differences are statistically significant according to the pairwise bootstrap resampling test (Koehn, 2004).

inform(line=M102, direction="Herald Square", vehicle=bus,
departure_time=9:01am, from_stop="Wall Street")
Take bus line M102 from Wall Street to Herald Square at 9:01am.

is there another option
inform(line=M102, direction="Herald Square", vehicle=bus,
departure_time=9:01am, from_stop="Wall Street")
There is a bus at 9:01am from Wall Street to Herald Square using line M102.

Figure 7.1: A comparison of an ordinary NLG training instance (top) and a context-aware one (bottom).

The context-aware instance includes the preceding user utterance (context), the input DA, and a context-appropriate output sentence (with entrainment highlighted).

vs. 73.1%) but slightly lower number of semantic errors (slot error rate of 0.41% vs. 0.46%).

7 Generating User-adaptive Outputs

In a conversation, speakers are influenced by previous utterances and tend to adapt their way of speaking to each other, reusing lexical items as well as syntactic structure (Reitter et al., 2006). This phenomenon is referred to as *entrainment* or *dialogue alignment*. It occurs naturally and subconsciously, facilitates successful conversations (Friedberg et al., 2012), and forms a natural source of variation in dialogues. There have been several attempts to let SDSs entrain to user utterances (Hu et al., 2014; Lopes et al., 2013, 2015), but all of them are completely or partially rule-based.

In this chapter, we enable our seq2seq system from Chapter 6 to align to the user, thus providing contextually appropriate, more natural, and possibly more successful output. The resulting system is, to our knowledge, the first fully trainable NLG system to support adapting to users' utterances. It improves upon a context-oblivious baseline in terms of both automatic metrics and human judgments.

7.1 Collecting a Context-Aware NLG Dataset

We collected a new NLG dataset for SDSs that is, to our knowledge, the first dataset of its kind to include preceding context (user utterance) with each data

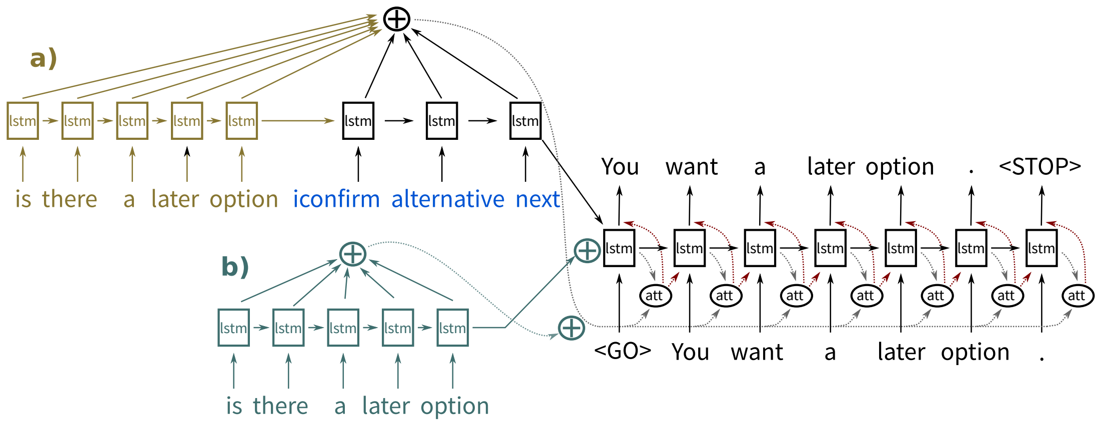

Figure 7.2: Context-aware modifications to the main seq2seq generator.

The base seq2seq model is shown in black, with (a) prepending context highlighted in gold, and (b) context encoder in teal. Note that (a) and (b) are alternatives, they are not used together.

instance (see Figure 7.1).[1] Crowdsourcing was used to obtain both the contextual user utterances and the corresponding system responses to be generated. The dataset contains over 5,500 instances with more than 500 distinct context utterances from the domain of public transport information. It is released under a permissive Creative Commons 4.0 BY-SA license.[2]

7.2 Context-aware Seq2seq Generator Extensions

To allow our seq2seq system from Chapter 6 to entrain to the user and provide naturally variable outputs, we enhanced its architecture in two alternative ways, which condition generation not only on the input DA, but also on the preceding user utterance:

a) *Prepending context.* The tokens of the preceding user utterance are simply prepended to the DA tokens and fed into the encoder (see Figure 7.2).

b) *Context encoder.* We add another, separate encoder for the context utterances. The hidden states of both encoders are concatenated (see Figure 7.2).

[1] To prevent data sparsity issues, we only take into account the immediately preceding user utterance, which we believe has the largest entrainment potential.

[2] Archival version is available at `http://hdl.handle.net/11234/1-1675`, development version at `https://github.com/UFAL-DSG/alex_context_nlg_dataset`.

Setup	BLEU	NIST
Baseline (context not used)	66.41	7.037
n-gram match reranker	68.68	7.577
Prepending context	63.87	6.456
+ n-gram match reranker	69.26	7.772
Context encoder	63.08	6.818
+ n-gram match reranker	69.17	7.596

Table 7.1: BLEU and NIST scores of different generator setups on the test data.

Furthermore, we add an *n-gram match reranker* promoting generator outputs on the k-best list that have a word or phrase overlap with the context utterance.

7.3 Experiments

We use our collected dataset to evaluate the generator extensions described in Section 7.2, applying direct string generation only. Table 7.1 lists our results in terms of the BLEU and NIST metrics. We can see that the n-gram match reranker brings an improvement even if used alone. Both seq2seq model extensions result in lowered scores if used by themselves, but bring in even larger improvements in combination with the n-gram match reranker.

We evaluated the best-performing setting (prepending context with n-gram match reranker) in a blind pairwise preference test against the baseline (cf. Section 3.5) with untrained judges recruited on the CrowdFlower crowdsourcing platform. The judges preferred the context-aware system output in 52.5% cases, slightly but significantly more often than the baseline.[3]

8 Generating Czech

Since NLG systems are typically tested on English, they can exploit its grammar. For instance, many generators are trained on delexicalized data and assume that lexical values can be inserted verbatim into the outputs (see Section 3.3). However, this does not not hold for languages where noun inflection is required, such as Czech.

Unlike most previous works, we test the multilingual capabilities of our generator in an experimental setting: In this chapter, we apply our seq2seq NLG

[3]Differences have been confirmed at 99% statistical significance level by pairwise bootstrap resampling (Koehn, 2004) for both BLEU/NIST scores and human judgments.

inform(name="Café Savoy", food=Mexican)

Café Savoy	*nabízí*	*mexická*	*jídla.*
Café Savoy$_{nominative}$	offers	Mexican	foods.

inform(name="Café Savoy", price_range=moderate)

Kavárna Savoy	*je*	*hezká*	*restaurace*	*se*	*středními*	*cenami.*
Café Savoy$_{nominative}$	is	a nice	restaurant	with	moderate	prices.

inform(name="Café Savoy", phone=293808716)

Telefonní	*číslo*	*do*	***Kavárny Savoy***	*je*	*293270464*
The phone	number	to	Café Savoy$_{genitive}$	is	293270464

Figure 8.1: Examples from our dataset showing three different surface forms for the DA slot value "Café Savoy" (with two synonymous lemmas, "café" and "kavárna").

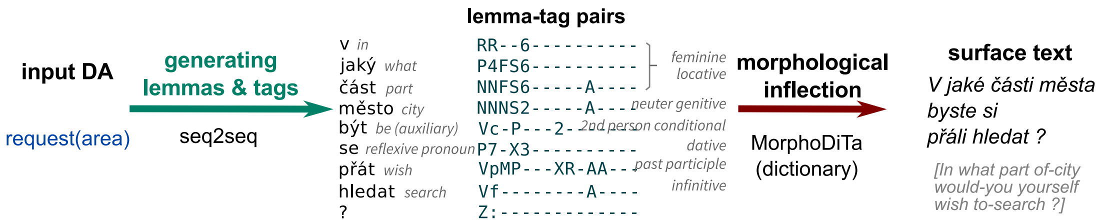

Figure 8.2: Lemma-tag generation: the seq2seq model produces lemmas and morphological tags, which are realized as word forms by a morphological dictionary.

system to Czech, introduce a few improvements, and show that our method produces mostly fluent and relevant outputs.

8.1 Creating an NLG Dataset for Czech

Since no suitable dataset existed for Czech NLG (same as most other non-English languages), we needed to create a new one. To reduce costs, speed up the process, and work around the lack of Czech speakers on crowdsourcing platforms (Pavlick et al., 2014), we localized an existing English set – Wen et al. (2015a)'s 5,000 instances on restaurant information – and had it translated by freelance translators. We released the data under the Creative Commons 4.0 BY-SA license.[1] The result shows that DA slot values, such as restaurant names, have more possible lexical realizations and need to be inflected (see Figure 8.1).

[1]The set can be downloaded from `http://hdl.handle.net/11234/1-2123`, a development version is available at `https://github.com/UFAL-DSG/cs_restaurant_dataset`.

8.2 Generator Extensions

We use the seq2seq approach described in Chapter 6 as the base of our experiments and add the following extensions to better accommodate for Czech:

Input DA handling. As DA slot values may influence output shape (e.g., require a specific preposition), we experiment with *lexically-informed* generation (Sharma et al., 2016): the input DA is lexicalized and values are taken into account during generation, but the output still contains placeholders and lexicalization is performed separately (to avoid data sparsity problems).

Lemma-tag generation. This is a third generator mode in addition to the two-step approach with t-trees and a joint end-to-end setup. The seq2seq model generates an interleaved sequence of lemmas (base word forms) and morphological tags (see Figure 8.2), and the MorphoDiTa morphological dictionary (Straková et al., 2014) maps them to inflected word forms. This should reduce data sparsity by abstracting away from word inflection while still allowing the seq2seq model to have nearly full control of the output.

Lexicalization. We implement four different approaches to selecting one of the multiple possible surface forms for a DA slot value (see Figure 8.1): a random baseline, a baseline selecting the most frequent surface form, an n-gram language model (LM), and a RNN-based LM. Both language models estimate the probability of possible surface forms based on preceding tokens in the sentence.

8.3 Experiments

In our experiments on our restaurant datasets, all 24 system variants learned to produce mostly fluent outputs with little to no semantic errors. We could see based on BLEU/NIST that the lemma-tag and direct generation setups perform better than the tree-based setup and RNN LM outperforms other lexicalization methods. We selected 7 setups for human evaluation (see Table 8.1): the best-performing lexically-informed and delexicalized setups, plus selected contrastive setups with just one setting different from the overall best setup (lexically-informed lemma-tag generation with RNN LM lexicalization).

The human evaluation is based on subjective preference ranking, same as in Chapter 7. We used a multi-way ranking of system outputs (Bojar et al., 2016),

Setup input DAs	generator mode	lexicalization	BLEU	NIST
delexicalized	joint (direct to strings)	RNN LM	19.54	4.273
delexicalized	lemma-tag	RNN LM	18.51	4.162
lexically informed	joint (direct to strings)	RNN LM	17.93	4.094
lexically informed	lemma-tag	most frequent	20.86	4.427
lexically informed	lemma-tag	n-gram LM	20.54	4.399
lexically informed	lemma-tag	RNN LM	**21.18**	**4.448**
lexically informed	two-step with t-trees	RNN LM	17.62	4.112

Table 8.1: Performance of selected generator setups in terms BLEU and NIST.

Setup input DAs	generator mode	lexicalization	True Skill	Rank
delexicalized	joint (direct to strings)	RNN LM	0.511	1
delexicalized	lemma-tag	RNN LM	0.479	2-4
lexically informed	lemma-tag	RNN LM	0.464	2-4*
lexically informed	lemma-tag	most frequent	0.462	2-4
lexically informed	joint (direct to strings)	RNN LM	0.413	5
lexically informed	two-step with t-trees	RNN LM	0.343	6-7
lexically informed	lemma-tag	n-gram LM	0.329	6-7

Table 8.2: Human rating results (best BLEU/NIST system marked with "*").

which is converted to pairwise system comparisons and evaluated using the TrueSkill algorithm (Sakaguchi et al., 2014).

Since users preferred a different system (delexicalized joint generation with RNN LM) than the best one in terms of BLEU/NIST, we performed a small-scale expert comparison of both systems' performance, which showed that both setups perform very comparably, but the human-preferred system fares slightly better. The results thus come out rather in favor of the simplest generator setups. On the other hand, the RNN-based surface form selection clearly pays off.

9 Conclusions

The main contributions of our thesis addressing the individual objectives set in Chapter 1 are as follows:

A) Generator easily adaptable for different domains. In Chapter 5, we developed an A*-search-based NLG system that is trainable from pairs of natural language sentences and corresponding dialogue acts, without the need for fine-grained semantic alignments, thus greatly simplifying training data collection for NLG. It was the first NLG system to learn alignments jointly with sentence planning. This system has then been superseded by a new, seq2seq-based one in terms of both speed and output quality, as described in Chapter 6. The seq2seq-based system reached new state-of-the-art without fine-grained alignments on the small BAGEL dataset (Mairesse et al., 2010), using much less training data than other RNN-based approaches. The two NLG systems were described in (Dušek and Jurčíček, 2015) and (Dušek and Jurčíček, 2016b), respectively.

B) Generator easily adaptable to different languages. We developed a simple, domain-independent surface realizer from the t-trees deep syntax formalism (see Section 3.4) for English, similar to an older Czech realizer (Žabokrtský et al., 2008). We simplified the creation of new t-tree realizers by creating a novel statistical morphological inflection module which generalizes to previously unseen word forms (see Chapter 4). The English realizer was described in (Dušek et al., 2015), and we reported on the morphological inflection module in (Dušek and Jurčíček, 2013). Parts of the realizer were later reused in machine translation (Popel et al., 2015; Aranberri et al., 2016).

In Chapter 8, we applied our seq2seq-based generator to Czech, addressing problems not present in English – larger vocabulary and the need to inflect proper names (DA slot values). We show that our seq2seq-based generator is able to produce mostly correct and fluent sentence structures without any significant changes, apart from proper name inflection, where our RNN-LM-based module significantly outperforms a strong baseline.

C) Generator that adapts to the user. Mimicking human behavior in dialogue, where interlocutors adapt their wording and syntax to each other, we extended our seq2seq generator in Chapter 7 to reflect not only the input DA, but also the previous user request, thus enabling it to create responses appropriate in the preceding dialogue context and providing it with a natural source of variation. The context-aware generator achieved a small but statistically significant performance improvement over the context-oblivious baseline. This result has been described in (Dušek and Jurčíček, 2016c).

D) Comparing different NLG system architectures. In Chapters 6 and 8, we compare two different NLG architectures: a two-step pipeline using separate sentence planning and surface realization modules and a joint setup generating surface strings directly. We are able to use the same seq2seq model for both setups, generating t-trees (deep syntax postprocessed by a surface realizer) or surface word forms (in an end-to-end fashion). In Chapter 8, we experiment with seq2seq generation of Czech lemma-tag sequences (base word forms and morphological categories), which are subsequently postprocessed by a morphological dictionary. We show that the seq2seq models learn to generate valid t-trees and lemma-tag sequences successfully. However, the direct, end-to-end setup reaches superior performance in our domains. Experiments for English from Chapter 6 were described in (Dušek and Jurčíček, 2016b).

E) Dataset availability for NLG in SDSs. To perform our experiments in Chapters 7 and 8, we have created two novel datasets for NLG, which are freely available under a permissive license:[1] the first NLG dataset for Czech, which is also the biggest freely available non-English NLG dataset, and the first NLG dataset using preceding dialogue context and specifically targeted at adapting system responses to the user. The latter set is also described in (Dušek and Jurčíček, 2016a).

In sum, our work constitutes significant advances along all of the preset objectives. In a few aspects, it leaves room for improvement in future work as some of the experiments on dialogue alignment and Czech generation were rather limited. Nevertheless, our generator is fully functional and usable in practice, within a spoken dialogue system or in a standalone setting. It is freely available for download from GitHub.[2]

In future work, we would like to widen the user adaptation experiment by taking the whole dialogue into account. We also plan to work on removing the need for delexicalizing proper names to further simplify portability of NLG systems to other domains and languages. In the long term, we see the future of NLG in interactive systems in end-to-end solutions incorporating language understanding, dialogue management, and response generation (Wen et al., 2016a; Williams et al., 2017).

[1] Available at `https://github.com/UFAL-DSG/alex_context_nlg_dataset`, `https://github.com/UFAL-DSG/cs_restaurant_dataset` under the Creative Commons 4.0 BY-SA license.

[2] Available at `https://github.com/UFAL-DSG/tgen` under the Apache 2.0 license.

Bibliography

Angeli, G. – Liang, P. – Klein, D. A simple domain-independent probabilistic approach to generation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, p. 502–512, Cambridge, MA, USA, 2010.

Aranberri, N. – Labaka Intxauspe, G. – Jauregi, O. – Ilarraza Sánchez, A. – Alegría Loinaz, I. – Agirre Bengoa, E. Tectogrammar-based machine translation for English-Spanish and English-Basque. *Procesamiento del Lenguaje Natural*. 2016, 56, p. 73–80.

Bahdanau, D. – Cho, K. – Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. In *International Conference on Learning Representations*, San Diego, CA, USA, 2015. arXiv:1409.0473.

Bangalore, S. – Rambow, O. Exploiting a probabilistic hierarchical model for generation. In *Proceedings of the 18th conference on Computational linguistics-Volume 1*, p. 42–48, Saarbrücken, Germany, 2000.

Belz, A. Statistical generation: Three methods compared and evaluated. In *Proceedings of the 10th European Workshop on Natural Language Generation (ENLG'05)*, p. 15–23, Helsinki, Finland, 2005.

Bengio, Y. – Ducharme, R. – Vincent, P. – Jauvin, C. A Neural Probabilistic Language Model. *Journal of Machine Learning Research*. 2003, 3, p. 1137–1155.

Bohnet, B. – Wanner, L. – Mille, S. – Burga, A. Broad coverage multilingual deep sentence generation with a stochastic multi-level realizer. In *Proceedings of the 23rd International Conference on Computational Linguistics*, p. 98–106, Beijing, China, 2010.

Bojar, O. – Chatterjee, R. – Federmann, C. – Graham, Y. – Haddow, B. – Huck, M. – Yepes, A. J. – Koehn, P. – Logacheva, V. – Monz, C. – others. Findings of the 2016 conference on machine translation (WMT16). In *Proceedings of the First Conference on Machine Translation (WMT), Volume 2: Shared Task Papers*, p. 131–198, Berlin, Germany, 2016.

Callison-Burch, C. – Osborne, M. – Koehn, P. Re-evaluating the role of BLEU in machine translation research. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, p. 249–256, Trento, Italy, 2006.

Callison-Burch, C. – Fordyce, C. – Koehn, P. – Monz, C. – Schroeder, J. (Meta-) evaluation of machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, p. 136–158, Prague, Czech Republic, 2007.

Cho, K. – Merrienboer, B. – Gulcehre, C. – Bahdanau, D. – Bougares, F. – Schwenk, H. – Bengio, Y. 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)*, p. 1724–1734, Doha, Qatar, 2014. arXiv:1406.1078.

Collins, M. – Duffy, N. New Ranking Algorithms for Parsing and Tagging: Kernels over Discrete Structures, and the Voted Perceptron. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, p. 263–270, Pennsylvania, PA, USA, 2002.

Dale, R. – Scott, D. – Di Eugenio, B. Introduction to the Special Issue on Natural Language Generation. *Computational Linguistics*. September 1998, 24, 3, p. 346–353.

Doddington, G. Automatic Evaluation of Machine Translation Quality Using N-gram Co-occurrence Statistics. In *Proceedings of the Second International Conference on Human Language Technology Research*, p. 138–145, San Francisco, CA, USA, 2002.

Dušek, O. – Jurčíček, F. Robust multilingual statistical morphological generation models. In *Proceedings of the ACL Student Research Workshop*, p. 158–164, Sofia, 2013.

Dušek, O. – Jurčíček, F. Training a Natural Language Generator From Unaligned Data. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, p. 451–461, Beijing, China, 2015.

Dušek, O. – Gomes, L. – Novák, M. – Popel, M. – Rosa, R. New Language Pairs in TectoMT. In *Proceedings of the 10th Workshop on Machine Translation*, p. 98–104, Lisbon, Portugal, 2015.

Dušek, O. – Jurčíček, F. A Context-aware Natural Language Generation Dataset for Dialogue Systems. In *Proceedings of RE-WOCHAT: Workshop on Collecting and Generating Resources for Chatbots and Conversational Agents – Development and Evaluation*, p. 6–9, Portorož, Slovenia, 2016a.

Dušek, O. – Jurčíček, F. 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*, Berlin, Germany, 2016b. arXiv:1606.05491.

Dušek, O. – Jurčíček, F. A Context-aware Natural Language Generator for Dialogue Systems. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, p. 185–190, Los Angeles, CA, USA, 2016c.

Durrett, G. – DeNero, J. Supervised Learning of Complete Morphological Paradigms. In *Proceedings of NAACL-HLT 2013*, p. 1185–1195, Atlanta, GA, USA, 2013.

Fan, R. E. – Chang, K. W. – Hsieh, C. J. – Wang, X. R. – Lin, C. J. LIBLINEAR: A library for large linear classification. *The Journal of Machine Learning Research*. 2008, 9, p. 1871–1874.

Friedberg, H. – Litman, D. – Paletz, S. B. Lexical entrainment and success in student engineering groups. In *IEEE Spoken Language Technology Workshop*, p. 404–409, Miami, FL, USA, 2012.

Gkatzia, D. – Mahamood, S. A Snapshot of NLG Evaluation Practices 2005 - 2014. In *Proceedings of the 15th European Workshop on Natural Language Generation (ENLG)*, p. 57–60, Brighton, England, UK, 2015.

Hajič, J. – Ciaramita, M. – Johansson, R. – Kawahara, D. – Martí, M. A. – Màrquez, L. – Meyers, A. – Nivre, J. – Padó, S. – Štěpánek, J. – others. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, p. 1–18, Boulder, CO, USA, 2009.

Hajič, J. et al. Announcing Prague Czech-English Dependency Treebank 2.0. In *Proceedings of LREC*, p. 3153–3160, Istanbul, Turkey, 2012.

Hart, P. E. – Nilsson, N. J. – Raphael, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*. 1968, 4, 2, p. 100–107.

Hastie, H. – Belz, A. A comparative evaluation methodology for NLG in interactive systems. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, p. 4004–4011, Reykjavík, Iceland, 2014.

Hu, Z. – Halberg, G. – Jimenez, C. – Walker, M. Entrainment in pedestrian direction giving: How many kinds of entrainment. In *Proceedings of the IWSDS'2014 Workshop on Spoken Dialog Systems*, p. 90–101, Napa, CA, USA, 2014.

Jurčíček, F. – Dušek, O. – Plátek, O. – Žilka, L. Alex: A Statistical Dialogue Systems Framework. In Sojka, P. – Horák, A. – Kopeček, I. – Pala, K. (Ed.) *Text, Speech and Dialogue: 17th International Conference, TSD*, Lecture Notes in Artificial Intelligence, p. 587–594, Brno, Czech Republic, 2014.

Koehn, P. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, p. 388–395, Barcelona, Spain, 2004.

Koehn, P. – Och, F. J. – Marcu, D. Statistical phrase-based translation. In *Proceedings of NAACL-HLT - Volume 1*, p. 48–54, Edmonton, Canada, 2003.

Koehn, P. *Statistical machine translation*. Cambridge University Press, 2010.

Konstas, I. – Lapata, M. A Global Model for Concept-to-Text Generation. *Journal of Artificial Intelligence Research*. 2013, 48, p. 305–346.

Langkilde, I. – Knight, K. Generation that exploits corpus-based statistical knowledge. In *Proceedings of the 36th Annual Meeting of the ACL and 17th International Conference on Computational Linguistics-Volume 1*, p. 704–710, Montréal, Canada, 1998.

Liang, P. – Jordan, M. I. – Klein, D. Learning semantic correspondences with less supervision. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, p. 91–99, Singapore, 2009.

Lopes, J. – Eskenazi, M. – Trancoso, I. Automated two-way entrainment to improve spoken dialog system performance. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 8372–8376, 2013.

Lopes, J. – Eskenazi, M. – Trancoso, I. From rule-based to data-driven lexical entrainment models in spoken dialog systems. *Computer Speech & Language*. 2015, 31, 1, p. 87–112.

Mairesse, F. – Walker, M. Trainable generation of big-five personality styles through data-driven parameter estimation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies*, p. 165–173, Columbus, OH, USA, 2008.

Mairesse, F. – Gašić, M. – Jurčíček, F. – Keizer, S. – Thomson, B. – Yu, K. – Young, S. Phrase-based statistical language generation using graphical models and active learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, p. 1552–1561, Uppsala, Sweden, 2010.

Mei, H. – Bansal, M. – Walter, M. R. What to talk about and how? Selective Generation using LSTMs with Coarse-to-Fine Alignment. In *The 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, p. 720–730, San Diego, CA, USA, 2016. arXiv: 1509.00838.

Och, F. J. – Ueffing, N. – Ney, H. An Efficient A* Search Algorithm for Statistical Machine Translation. In *Proceedings of the Workshop on Data-driven Methods in Machine Translation - Volume 14*, p. 1–8, Toulouse, France, 2001.

Paiva, D. S. – Evans, R. Empirically-based control of natural language generation. In *Proceedings of the 43rd Annual Meeting of ACL*, p. 58–65, Stroudsburg, PA, USA, 2005. ACL.

Papineni, K. – Roukos, S. – Ward, T. – Zhu, W.-J. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, p. 311–318, Pennsylvania, PA, USA, 2002.

Pavlick, E. – Post, M. – Irvine, A. – Kachaev, D. – Callison-Burch, C. The language demographics of Amazon Mechanical Turk. *Transactions of the Association for Computational Linguistics*. 2014, 2, p. 79–92.

Popel, M. – Žabokrtský, Z. TectoMT: modular NLP framework. In *Proceedings of IceTAL, 7th International Conference on Natural Language Processing*, p. 293–304, Reykjavík, 2010.

Popel, M. *Ways to Improve the Quality of English-Czech Machine Translation*. Master's thesis, Charles University in Prague, 2009.

Popel, M. et al. Report on the Second MT Pilot and Its Evaluation. Technical Report Deliverable D2.8, QTLeap, EC FP7 Project no. 610516, 2015.

Ptáček, J. – Žabokrtský, Z. Dependency-Based Sentence Synthesis Component for Czech. In *Proceedings of 3rd International Conference on Meaning-Text Theory*, 69 / *Wiener Slawistischer Almanach*, p. 407–415. Verlag Otto Sagner, 2007.

Reiter, E. – Dale, R. *Building Natural Language Generation Systems*. Cambridge University Press, 2000.

Reitter, D. – Keller, F. – Moore, J. D. Computational Modelling of Structural Priming in Dialogue. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, p. 121–124, New York City, NY, USA, 2006.

Rosa, R. – Dušek, O. – Novák, M. – Popel, M. Translation Model Interpolation for Domain Adaptation in TectoMT. In *Proceedings of the 1st Deep Machine Translation Workshop*, p. 89–96, Prague, Czech Republic, 2015. ÚFAL MFF UK.

Rudnicky, A. I. – Thayer, E. H. – Constantinides, P. C. – Tchou, C. – Shern, R. – Lenzo, K. A. – Xu, W. – Oh, A. Creating natural dialogs in the Carnegie Mellon Communicator system. In *Proceedings of the 6th European Conference on Speech Communication and Technology*, p. 1531–1534, Lisbon, Portugal, 1999.

Sakaguchi, K. – Post, M. – Van Durme, B. Efficient elicitation of annotations for human evaluation of machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, p. 1–11, Baltimore, MD, USA, 2014.

Sgall, P. – Hajičová, E. – Panevová, J. *The meaning of the sentence in its semantic and pragmatic aspects*. D. Reidel, 1986.

Sharma, S. – He, J. – Suleman, K. – Schulz, H. – Bachman, P. Natural Language Generation in Dialogue using Lexicalized and Delexicalized Data. *arXiv:1606.03632 [cs]*. June 2016.

Sripada, S. G. – Reiter, E. – Hunter, J. – Yu, J. Exploiting a parallel text-data corpus. In *Proceedings of the Corpus Linguistics 2003 conference*, p. 734–743, Lancaster, England, UK, 2003.

Stent, A. – Prasad, R. – Walker, M. Trainable sentence planning for complex information presentation in spoken dialog systems. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, p. 79–86, Barcelona, Spain, 2004.

Stent, A. – Marge, M. – Singhai, M. Evaluating evaluation methods for generation in the presence of variation. In *International Conference on Intelligent Text Processing and Computational Linguistics*, p. 341–351, Mexico City, Mexico, 2005.

Straková, J. – Straka, M. – Hajič, J. Open-Source Tools for Morphology, Lemmatization, POS Tagging and Named Entity Recognition. In *Proceedings of 52nd Annual*

Meeting of the Association for Computational Linguistics: System Demonstrations, p. 13–18, Baltimore, MA, USA, 2014.

SUTSKEVER, I. – VINYALS, O. – LE, Q. V. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, p. 3104–3112, Montréal, Canada, 2014. arXiv:1409.3215.

DEEMTER, K. – KRAHMER, E. – THEUNE, M. Real vs. template-based natural language generation: a false opposition? *Computational Linguistics*. 2005, 31, 1, p. 15–24.

WALKER, M. A. – RAMBOW, O. – ROGATI, M. SPoT: a trainable sentence planner. In *Proceedings of 2nd meeting of NAACL*, p. 1–8, Stroudsburg, PA, USA, 2001. ACL.

WEN, T.-H. – GAŠIĆ, M. – MRKŠIĆ, N. – SU, P.-H. – VANDYKE, D. – YOUNG, S. Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, p. 1711–1721, Lisbon, Portugal, 2015a.

WEN, T.-H. – GASIC, M. – KIM, D. – MRKSIC, N. – SU, P.-H. – VANDYKE, D. – YOUNG, S. 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*, p. 275–284, Prague, Czech Republic, 2015b.

WEN, T.-H. – GASIC, M. – MRKSIC, N. – ROJAS-BARAHONA, L. M. – SU, P.-H. – ULTES, S. – VANDYKE, D. – YOUNG, S. A Network-based End-to-End Trainable Task-oriented Dialogue System. *arXiv:1604.04562 [cs, stat]*. April 2016a.

WEN, T.-H. – GASIC, M. – MRKSIC, N. – ROJAS-BARAHONA, L. M. – SU, P.-H. – VANDYKE, D. – YOUNG, S. Multi-domain Neural Network Language Generation for Spoken Dialogue Systems. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, p. 120–129, San Diego, CA, USA, 2016b. arXiv: 1603.01232.

WHITE, M. – RAJKUMAR, R. – MARTIN, S. Towards broad coverage surface realization with CCG. In *Proceedings of the Workshop on Using Corpora for NLG: Language Generation and Machine Translation (UCNLG+MT)*, p. 22–30, Copenhagen, Denmark, 2007.

WILLIAMS, J. D. – ASADI, K. – ZWEIG, G. Hybrid Code Networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. *arXiv:1702.03274 [cs]*. February 2017.

WONG, Y. W. – MOONEY, R. J. Generation by inverting a semantic parser that uses statistical machine translation. In *Proceedings of Human Language Technologies: The Conference of the North American Chapter of the ACL (NAACL-HLT-07)*, p. 172–179, Prague, Czech Republic, 2007.

YOUNG, S. – GAŠIĆ, M. – KEIZER, S. – MAIRESSE, F. – SCHATZMANN, J. – THOMSON, B. – YU, K. The Hidden Information State model: A practical framework for POMDP-based spoken dialogue management. *Computer Speech & Language*. 2010, 24, 2, p. 150–174.

ŽABOKRTSKÝ, Z. – PTÁČEK, J. – PAJAS, P. TectoMT: highly modular MT system with tectogrammatics used as transfer layer. In *Proceedings of the Third Workshop on Statistical Machine Translation*, p. 167–170, Columbus, OH, USA, 2008.

List of Abbreviations

DA	Dialogue act
LM	Language model
MT	Machine translation
NLG	Natural language generation
NLP	Natural language processing
NN	Neural network
RNN	Recurrent neural network
SDS	Spoken dialogue system
seq2seq	Sequence-to-sequence

List of Author's Publications

Publications are divided by type and sorted in reverse chronological order. The number of citations is given according to Google Scholar as of May 27, 2017 and includes self-citations.

Major Conference Papers

Dušek, O. – Jurčíček, F. 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 (Volume 2: Short Papers)*, p. 45–51, Berlin, Germany, 2016a. Citations: 3.

Dušek, O. – Jurčíček, F. A Context-aware Natural Language Generator for Dialogue Systems. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, p. 185–190, Los Angeles, CA, USA, 2016b. Citations: 1.

Dušek, O. – Jurčíček, F. Training a Natural Language Generator From Unaligned Data. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, p. 451–461, Beijing, China, 2015. Citations: 8.

Dušek, O. – Plátek, O. – Žilka, L. – Jurčíček, F. Alex: Bootstrapping a Spoken Dialogue System for a New Domain by Real Users. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, p. 79–83, Philadelphia, PA, USA, 2014. Citations: 3.

Dušek, O. – Jurčíček, F. Robust Multilingual Statistical Morphological Generation Models. In *51st Annual Meeting of the Association for Computational Linguistics Proceedings of the Student Research Workshop*, p. 158–164, Sofia, Bulgaria, 2013. Citations: 6.

Other Conference Papers

Dušek, O. – Jurčíček, F. A Context-aware Natural Language Generation Dataset for Dialogue Systems. In *Proceedings of RE-WOCHAT: Workshop on Collecting and Generating Resources for Chatbots and Conversational Agents – Development and Evaluation*, p. 6–9, Portorož, Slovenia, 2016. Citations: 1.

Bojar, O. – Dušek, O. – Kocmi, T. – Libovický, J. – Novák, M. – Popel, M. – Sudarikov, R. – Variš, D. CzEng 1.6: Enlarged Czech-English Parallel Corpus with Processing Tools Dockered. In *Proceedings of the 19th International Conference on Text, Speech and Dialogue*, Brno, Czech Republic, 2016. Citations: 9.

Rosa, R. – Popel, M. – Bojar, O. – Mareček, D. – Dušek, O. Moses & Treex Hybrid MT Systems Bestiary. In Hajič, J. – Noord, G. – Branco, A. (Ed.) *Proceedings of the 2nd Deep Machine Translation Workshop*, p. 1–10, Lisbon, Portugal, 2016.

Sudarikov, R. – Bojar, O. – Dušek, O. – Holub, M. – Kríž, V. Verb Sense Disambiguation in Machine Translation. In *Sixth Workshop on Hybrid Approaches to Translation (HyTra-6)*, p. 42–50, Osaka, Japan, 2016.

Dušek, O. – Gomes, L. – Novák, M. – Popel, M. – Rosa, R. New Language Pairs in TectoMT. In *Proceedings of the 10th Workshop on Machine Translation*, p. 98–104, Lisbon, Portugal, 2015a. Citations: 9.

Dušek, O. – Fučíková, E. – Hajič, J. – Popel, M. – Šindlerová, J. – Urešová, Z. Using Parallel Texts and Lexicons for Verbal Word Sense Disambiguation. In *Proceedings of the Third International Conference on Dependency Linguistics*, p. 82–90, Uppsala, Sweden, 2015b. Citations: 5.

Urešová, Z. – Dušek, O. – Fučíková, E. – Hajič, J. – Šindlerová, J. Bilingual English-Czech Valency Lexicon Linked to a Parallel Corpus. In *Proceedings of LAW IX - The 9th Linguistic Annotation Workshop*, p. 124–128, Denver, CO, USA, 2015.

Rosa, R. – Dušek, O. – Novák, M. – Popel, M. Translation Model Interpolation for Domain Adaptation in TectoMT. In Hajič, J. – Branco, A. (Ed.) *Proceedings of the 1st Deep Machine Translation Workshop*, p. 89–96, Prague, Czech Republic, 2015. Citations: 1.

Dušek, O. – Hajič, J. – Urešová, Z. Verbal Valency Frame Detection and Selection in Czech and English. In *The 2nd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, p. 6–11, Baltimore, MD, USA, 2014a. Citations: 3.

Urešová, Z. – Dušek, O. – Hajič, J. – Pecina, P. Multilingual Test Sets for Machine Translation of Search Queries for Cross-Lingual Information Retrieval in the Medical Domain. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, Reykjavík, Iceland, 2014. Citations: 6.

Majchráková, D. – Dušek, O. – Hajič, J. – Karčová, A. – Garabík, R. Semi-automatic Detection of Multiword Expressions in the Slovak Dependency Treebank. In *Proceedings of the First International Conference Computational Linguistics in Bulgaria*, p. 32–39, Sofia, Bulgaria, 2014.

Korvas, M. – Plátek, O. – Dušek, O. – Žilka, L. – Jurčíček, F. Free English and Czech telephone speech corpus shared under the CC-BY-SA 3.0 license. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, Reykjavík, Iceland, 2014.

Jurčíček, F. – Dušek, O. – Plátek, O. – Žilka, L. Alex: A Statistical Dialogue Systems Framework. In *Text, Speech and Dialogue: 17th International Conference*, p. 587–594, Brno, Czech Republic, 2014a. Citations: 3.

Jurčíček, F. – Dušek, O. – Plátek, O. A Factored Discriminative Spoken Language Understanding for Spoken Dialogue Systems. In *Text, Speech and Dialogue: 17th International Conference*, p. 579–586, Brno, Czech Republic, 2014b. Citations: 1.

Dušek, O. – Hajič, J. – Hlaváčová, J. – Novák, M. – Pecina, P. – Rosa, R. – Tamchyna, A. – Urešová, Z. – Zeman, D. Machine Translation of Medical Texts in the Khresmoi Project. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, p. 221–228, Baltimore, MD, USA, 2014b. Citations: 14.

Dušek, O. Towards a Truly Statistical Natural Language Generator for Spoken Dialogues. In *Week of Doctoral Students '13 Proceedings of Contributed Papers*, p. 12–17, Prague, Czech Republic, 2013.

Dušek, O. – Žabokrtský, Z. – Popel, M. – Majliš, M. – Novák, M. – Mareček, D. Formemes in English-Czech Deep Syntactic MT. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, p. 267–274, Montreal, Canada, 2012. Citations: 11.

Rosa, R. – Mareček, D. – Dušek, O. DEPFIX: A System for Automatic Correction of Czech MT Outputs. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, p. 362–368, Montreal, Canada, 2012a. Citations: 36.

Rosa, R. – Dušek, O. – Mareček, D. – Popel, M. Using Parallel Features in Parsing of Machine-Translated Sentences for Correction of Grammatical Errors. In *Proceedings of Sixth Workshop on Syntax, Semantics and Structure in Statistical Translation*, p. 39–48, Jeju, Korea, 2012b. Citations: 8.

Bojar, O. – Žabokrtský, Z. – Dušek, O. – Galuščáková, P. – Majliš, M. – Mareček, D. – Maršík, J. – Novák, M. – Popel, M. – Tamchyna, A. The Joy of Parallelism with CzEng 1.0. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, p. 3921–3928, Istanbul, Turkey, 2012. Citations: 43.

Journal Papers

Dušek, O. Srovnání německých a českých valenčních slovníků. *Ročenka Kruhu moderních filologů*. 2015, 25, p. 63–71.

Zeman, D. – Dušek, O. – Mareček, D. – Popel, M. – Ramasamy, L. – Štěpánek, J. – Žabokrtský, Z. – Hajič, J. HamleDT: Harmonized Multi-Language Dependency Treebank. *Language Resources and Evaluation*. 2014, 48, 4, p. 601–637. Citations: 35.

Pecina, P. – Dušek, O. – Goeuriot, L. – Hajič, J. – Hlaváčová, J. – Jones, G. – Kelly, L. – Leveling, J. – Mareček, D. – Novák, M. – Popel, M. – Rosa, R. – Tamchyna, A. – Urešová, Z. Adaptation of machine translation for multilingual information retrieval in the medical domain. *Artificial Intelligence in Medicine*. July 2014, 61, 3, p. 165–185. Citations: 25.

Tamchyna, A. – Dušek, O. – Rosa, R. – Pecina, P. MTMonkey: A Scalable Infrastructure for a Machine Translation Web Service. *The Prague Bulletin of Mathematical Linguistics*. 2013, 100, p. 31–40. Citations: 6.

Theses

Dušek, O. *Zum Vergleich der tschechischen und deutschen Valenzwörterbücher*. Master's Thesis, Charles University, Faculty of Arts, Prague, 2013. Supervisor: Vít Dovalil. Citations: 1.

Dušek, O. *Deep automatic analysis of English*. Master's Thesis, Charles University, Faculty of Mathematics and Physics, Prague, 2010. Supervisor: Jan Hajič.

Dušek, O. *BashCommander*. Bachelor's Thesis, Charles University, Faculty of Mathematics and Physics, Prague, 2007. Supervisor: Jan Kofroň.