# Neural Generation for Czech: Data and Baselines

**Ondřej Dušek** and **Filip Jurčíček**
Charles University, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Prague, Czech Republic
{odusek,jurcicek}@ufal.mff.cuni.cz

## Abstract

We present the first dataset targeted at end-to-end NLG in Czech in the restaurant domain, along with several strong baseline models using the sequence-to-sequence approach. While non-English NLG is under-explored in general, Czech, as a morphologically rich language, makes the task even harder: Since Czech requires inflecting named entities, delexicalization or copy mechanisms do not work out-of-the-box and lexicalizing the generated outputs is non-trivial.

In our experiments, we present two different approaches to this this problem: (1) using a neural language model to select the correct inflected form while lexicalizing, (2) a two-step generation setup: our sequence-to-sequence model generates an interleaved sequence of lemmas and morphological tags, which are then inflected by a morphological generator.

## 1 Introduction

While most current neural NLG systems do not explicitly contain language-specific components and are thus capable of multilingual generation in principle, there has been little work to test these capabilities experimentally. This goes hand in hand with the scarcity of non-English training datasets for NLG – the only data-to-text NLG set known to us is a small sportscasting Korean dataset (Chen et al., 2010),[1] which only contains a limited number of named entities, reducing the need for their inflection.

Since most generators are only tested on English, they do not need to handle grammar complexities not present in English. A prime example is the delexicalization technique used by most current generators (e.g., Oh and Rudnicky, 2000; Mairesse et al., 2010; Wen et al., 2015a,b; Juraska et al., 2018): It is generally assumed that attribute (slot) values from the input meaning representation (MR) can be replaced by placeholders during generation and inserted into the output verbatim. Delexicalization or an analogous technique, such as a copy mechanism (Gu et al., 2016; Gehrmann et al., 2018), is required for most generation scenarios to allow generalization to unseen entity names: sets of entities are open (potentially infinite and subject to change) while training data is scarce. However, the verbatim insertion assumption does not hold for languages with extensive noun inflection – attribute values need to be inflected here to produce fluent outputs (see Figure 1).

This paper presents the following contributions:

- We create a novel dataset for Czech delexicalized generation; this extends the typical task of data-to-text NLG by requiring attribute value inflection (Section 2). We choose Czech as an example of a morphologically complex language (Cotterell et al., 2018) with a large set of NLP tools readily available (e.g. Popel and Žabokrtský, 2010; Straková et al., 2014; Straka and Straková, 2017).
- We present baseline models based on the TGen sequence-to-sequence (seq2seq) system (Dušek and Jurčíček, 2016), with two novel extensions to the model for our task (Section 3):
  - A model for lexicalization, i.e., selecting the correct inflected surface form for a slot value, based on a recurrent neural network language model (RNN LM);
  - A new generation mode, where the seq2seq generator produces interleaved sequences of lemmas (base word forms) and morphological tags that are postprocessed using a morphological generator.
- Using both automatic and manual evaluation in Section 4, we show that our extensions improve

[1] http://www.cs.utexas.edu/users/ml/clamp/sportscasting/

*Proceedings of The 12th International Conference on Natural Language Generation*, pages 563–574,
Tokyo, Japan, 28 Oct - 1 Nov, 2019. ©2019 Association for Computational Linguistics

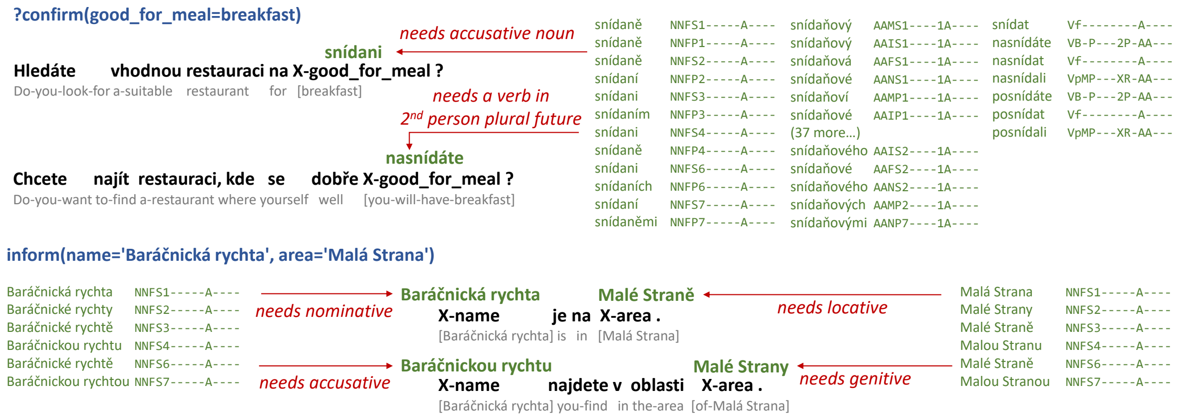
?confirm(good_for_meal=breakfast)
needs accusative noun
snídani
Hledáte vhodnou restauraci na X-good_for_meal ?
Do-you-look-for a-suitable restaurant for [breakfast]
needs a verb in
2nd person plural future
nasnídáte
Chcete najít restauraci, kde se dobře X-good_for_meal ?
Do-you-want to-find a-restaurant where yourself well [you-will-have-breakfast]
snídaně NNFS1-----A----
snídaně NNFP1-----A----
snídaně NNFS2-----A----
snídaní NNFP2-----A----
snídani NNFS3-----A----
snídaním NNFP3-----A----
snídani NNFS4-----A----
snídaně NNFP4-----A----
snídani NNFS6-----A----
snídaních NNFP6-----A----
snídaní NNFS7-----A----
snídaněmi NNFP7-----A----
snídaňový AAMS1----1A----
snídaňový AAIS1----1A----
snídaňová AAFS1----1A----
snídaňové AANS1----1A----
snídaňoví AAMP1----1A----
snídaňové AAIP1----1A----
(37 more...)
snídaňového AAIS2----1A----
snídaňové AAFS2----1A----
snídaňového AANS2----1A----
snídaňových AAMP2----1A----
snídaňovými AANP7----1A----
snídat Vf--------A----
nasnídáte VB-P---2P-AA---
nasnídat Vf--------A----
nasnídali VpMP---XR-AA---
posnídáte VB-P---2P-AA---
posnídat Vf--------A----
posnídali VpMP---XR-AA---
inform(name='Baráčnická rychta', area='Malá Strana')
Baráčnická rychta NNFS1-----A----
Baráčnické rychty NNFS2-----A----
Baráčnické rychtě NNFS3-----A----
Baráčnickou rychtu NNFS4-----A----
Baráčnické rychtě NNFS6-----A----
Baráčnickou rychtou NNFS7-----A----
needs nominative
Baráčnická rychta
Malé Straně
needs locative
X-name je na X-area .
[Baráčnická rychta] is in [Malá Strana]
needs accusative
Baráčnickou rychtu
Malé Strany
needs genitive
X-name najdete v oblasti X-area .
[Baráčnická rychta] you-find in the-area [of-Malá Strana]
Malá Strana NNFS1-----A----
Malé Strany NNFS2-----A----
Malé Straně NNFS3-----A----
Malou Stranu NNFS4-----A----
Malé Straně NNFS6-----A----
Malou Stranou NNFS7-----A----

Figure 1: Example of delexicalized generation in Czech. Input MRs are shown in bold blue, corresponding target (delexicalized) outputs in bold black, with "X-" marking slot value placeholders. English glosses are shown below each word in gray. Appropriate inflected forms to be filled into slot placeholders are shown in bold green, with lists of all possible forms along with their morphological tags (Hajič, 2004). Note that the surface form for "X-good_for_meal" can even have different parts-of-speech (left column: noun, middle: adjective, right: verb forms).

over the base model, but do not solve the task completely.

We propose improvements for future work in Section 6. Our dataset and all experimental code are released on GitHub.[2]

## 2 Dataset

Our goal was to create a dataset comparable in size and domain to existing English data-to-text NLG datasets used in experiments with neural systems. Since there are few to none Czech speakers on crowdsourcing platforms (Pavlick et al., 2014; Dušek et al., 2014), we were not able to use them for data collection. Recruiting freelance translators seemed easier than training annotators; therefore, we turned to localizing and translating an existing dataset instead of creating a new one from scratch. We chose the restaurant dataset of Wen et al. (2015b) due to its manageable, yet non-trivial size and the familiarity of the domain (cf. Mairesse et al., 2010; Dušek et al., 2019). The original dataset contains 5,192 MR-sentence pairs, where MRs come in the form of dialogue acts (DAs). A DA consists of DA type (e.g., *request*, *confirm*, *inform*) and a list of slots (attributes) and their values (e.g., *name*, *price_range*, *address*, *area*). There are 8 different DA types and 12 slots in the dataset. All slots except the binary *kids_allowed* are delexicalized during generation (cf. Figure 1).

***Ananta*** – feminine noun, inflected (nom: *Ananta*, gen: *Ananty*, dat, loc: *Anantě*, acc: *Anantu*, inst: *Anantou*)

***BarBar*** – masculine inanimate noun, inflected (nom, acc: *BarBar*, gen, dat, loc: *BarBaru*, inst: *BarBarem*)

***Café Savoy*** – neuter noun, not inflected

***Místo*** – neuter noun, inflected (nom, acc: *Místo*, gen: *Místa*, dat: *Místu*, loc: *Místě*, inst: *Místem*)

***U Konšelů*** – prepositional phrase, not inflected

Figure 2: Examples of restaurant names from the localized data with different morphosyntactic behavior (*nom* = nominative, *gen* = genitive, *dat* = dative, *acc* = accusative, *loc* = locative, *inst* = instrumental).

### 2.1 Localizing the Data

We first needed to localize the dataset, replacing the original setting of San Francisco with a Czech one. In particular, we aimed at using domestic entity names (DA slot values) that need to be inflected since foreign names are often kept uninflected in Czech, using less fluent and conspicuous grammatical constructions to avoid inflection.[3]

We localized the following slots in both DAs and texts from the dataset: restaurant names, areas, food types, street addresses, and landmarks. We

[2] Dataset: https://github.com/UFAL-DSG/cs_restaurant_dataset, code: https://github.com/UFAL-DSG/tgen.

[3] This is not to say that we avoided using any foreign words in the localization process. Since foreign restaurant names are quite common in Czechia, we also included some of them in the localized data.

used a list of randomly chosen restaurant names from the Prague city center as well as lists of Prague neighborhoods, streets, and landmarks. The resulting sentences contain mostly factually inaccurate, yet meaningful utterances about restaurants in Prague.

The localized lists are quite short, with just 15 different restaurant names and a similar number of landmarks, streets, and neighborhoods. While much longer lists would be needed for a real-world scenario, this is sufficient to cover most common classes of names with different inflection patterns and/or syntactic behavior (see Figure 2).

### 2.2 Translation

We recruited six translators and asked them to translate all unique texts in the localized dataset. They were given the following instructions:

- translate the utterances in isolation,
- use fluent, spoken-style Czech,
- strive to preserve the facts but not necessarily all nuances of the original,
- use varying synonyms (as long as they belong to casual, fluent Czech), including for entity names or slot values (such as price ranges or meal types),
- inflect entity names as needed,
- use formal address (or plural) when addressing the user, and use the female form in the first person for self-references.[4]

All rules but the last one aim at obtaining a varied and fluent dataset; the last rule strives for consistency. Note that the translators were not given the input DAs – these carry no more information than the corresponding English sentences, and we assume that they would only confuse the translators and could hurt the fluency of the results.

### 2.3 Consistency Checks and Deduplication

We checked the translated Czech texts for the presence of all required slot values. We took the following iterative, partially automatic approach:

1. Create a list of possible inflected surface forms for all slot values in the dataset. We used the morphological generator of Straková et al. (2014) to inflect the surface forms automatically and manually checked for errors.
2. Given a DA and a translated sentence, check (using an automatic script) that the sentence contains surface forms for all slots in the DA.
3. Given a sentence found by the script to miss a value, check if it contains an alternative surface form not included in the list from Step 1. If so, add this alternative surface form to the list.
4. If the translated sentence does not contain any mention of the DA value, fix the translation.
5. Repeat from Step 2 until there are no missing DA value mentions in the whole set.

Note that these checks result not only in greater consistency of the dataset, but also in a list of possible surface realizations for all slot values in the dataset. We store this list including morphological information provided by the tagger (with manually corrected errors), and we use it for lexicalization (see Section 3).[5]

### 2.4 Duplicate Sentence Handling

If the exact lexicalization is not taken into account, the original dataset of Wen et al. (2015b) contains a lot of duplicate texts – the total number of DA-text pairs is 5,192, but only 2,648 are unique. Therefore, we chose to only translate unique texts, in order to speed up the translation process and lower the costs, albeit at a cost of a lower-quality result. We ensured that the translations preserve the same number of unique sentences by modifying any duplicate translations, manually replacing selected words or phrases with synonyms.

After the dataset was translated, we expanded it to obtain the same number of instances and the same distribution of different DAs as in the original. Given a delexicalized DA, a list of corresponding translated sentences, and the target number of corresponding sentences to match the original set, we sampled additional copies of the existing translations to match the number of originals. To estimate probabilities of the individual translations for the sampling, we used a 5-gram LM[6] trained on lemmatized and delexicalized translations (see Figure 3 for details). We obtained LM scores for all

[4] Czech grammar requires a selection between formal an informal address whenever using a verb in the 2nd person (Naughton, 2005, p. 134ff.). For verbs with past tense or conditional and in any person, gender must be selected (Naughton, 2005, p. 140ff.). Here we opted for a feminine form whenever the system addresses itself, and formal address (mostly homonymous with plural) when addressing the user.

[5] We treat multiword slot values as single tokens in our surface form list. We assign them a morphological tag that fits the whole expression best, e.g., a noun tag for noun phrases.

[6] We used the implementation in the KenLM toolkit (Heafield, 2011).

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *mít* | *pro* | *ty* | *vhodný* | *restaurace* | . | *jeho* | *název* | *být* | *X-name* | *a* | *moci* | *se* | *dát* | *X-food* | *kuchyně* | . |
| Mám | pro | Vás | vhodnou | restauraci | . | Její | název | je | Kočár z Vídně | a | můžete | si | dát | českou | kuchyni | . |
| I have | for | you | a suitable | restaurant | . | Its | name | is | Kočár z Vídně | and | you can | yourself | give | Czech | cuisine | . |

‘I have a suitable restaurant for you. Its name is Kočár z Vídně and you can have Czech cuisine.’

Figure 3: Lemmatized and delexicalized form of the translations for LM scoring. Top: lemmatized and delexicalized Czech used for the LM; middle: original Czech sentence including lexicalization; bottom: English word-by-word gloss. An English translation is shown below the example.

| | **English** | **Czech** |
|---|---|---|
| Number of instances | 5,192 | 5,192 |
| Unique delexicalized instances | 2,648 | 2,752 |
| Unique delexicalized DAs | 248 | 248 |
| Unique lemmas (in delexicalized set) | 399 | 532 |
| Unique word forms (in delexicalized set) | 455 | 962 |
| Average lexicalizations per slot value | 1 | 3.84 |

Table 1: Statistics of our translated Czech dataset and a comparison to the English original of Wen et al. (2015b). The average lexicalizations per slot value shows the number of different surface lexical forms per slot value, as it appears in the dataset. Numerals were disregarded when computing this value.

| Part | **Train** | **Dev** | **Test** |
|---|---|---|---|
| Unique delexicalized DAs | 144 | 51 | 53 |
| Total number of instances | 3,569 | 781 | 842 |

Table 2: Dataset split statistics.

translations, used softmax to obtain a probability distribution, and sampled additional copies from this distribution. This ensures that translations using more frequent phrasing are more likely to be used multiple times in the set.

We then relexicalized the sampled copies: We randomly changed DA slot values and replaced their surface forms in the text using the surface forms list, checking for roughly corresponding morphology. Since the morphological information used by this approach was rather crude (e.g., noun/adjective gender was not taken into account), disfluencies ensued in some cases. Therefore, we manually corrected all relexicalized sentences, changing inflection or wording where needed.

### 2.5 Dataset Statistics

The final Czech set contains the same number of instances as the English original, copies the DA distribution of the original, and contains a slightly higher number of unique delexicalized sentences due to post-expansion corrections (see Section 2.4). A statistics of the dataset size is shown in Table 1, with a comparison to the original English set. We can see that while the number of unique word lemmas (disregarding restaurant and place names) is slightly higher in the Czech set, the number of unique inflected word forms is more than twice as high. It is also clear that using slot values verbatim in the text is not possible in the Czech set as the number of possible lexical realizations for each value is much higher than one.

### 2.6 Data Split

The original dataset of Wen et al. (2015b), which used a sequential 3:1:1 split into training, development and test parts, suffered from a lot of overlap in terms of delexicalized DAs between the sections. This means that a system can perform quite well on this dataset and still be unable to generalize to unseen DAs (Lampouras and Vlachos, 2016). To make testing systems’ generalization capabilities possible on our Czech dataset, we opted for a different data split. We roughly keep the same 3:1:1 size proportion (see Table 2), but we make sure no delexicalized DA appears in two different parts. On the other hand, we ensure that most DA types (*inform*, *confirm* etc.) are represented in all data parts, so the system has access to all general types of sentences during training.[7]

## 3 Model

We use TGen (Dušek and Jurčíček, 2016) in our experiments, which is a freely available NLG system based on the seq2seq model with attention (Bahdanau et al., 2015).

The seq2seq model consists of the encoder, the decoder, and the attention model. Both the encoder and decoder are recurrent neural networks (RNN)

[7] This is impossible to achieve for the *goodbye* and *?reqmore* DA types (i.e., goodbyes and asking if the user needs anything else). These DA types never appear with slots and thus only have one corresponding DA. We keep the corresponding instances in the training set.

with LSTM cells (Hochreiter and Schmidhuber, 1997). The encoder takes the input DA as a sequence of triples "DA type – slot – value"[8] and produces a sequence of hidden states. The last hidden state is used to initialize the decoder, all hidden states serve as input into the attention model. The attention model produces their weighted combination for each decoder step using a 1-layer fully connected network. The decoder generates output tokens one-by-one using the previously generated token and the attention model as inputs.

In addition to the basic seq2seq model, TGen adds beam search and a reranker for the candidate outputs on the generation beam that checks if the input semantics is preserved. The reranker encodes a candidate output using an LSTM RNN and produces a binary classification of DA types and slot-value pairs present. The number of differences against the input DA is used as penalty.

### 3.1 Basic Extensions

We added two features fairly standard in seq2seq-based models but absent from TGen:

- Bidirectional encoder (Bahdanau et al., 2015) – the input sequence is encoded in both directions and the resulting hidden states are joined. We added this for both the main seq2seq generator and the reranker.
- Dropout (Hinton et al., 2012) – this zeroes out certain connections within the network with a given probability during the training process; it serves as regularization feature. We use this in the main generator only.

We use these extensions in all our setups as they improved results in our preliminary experiments.

### 3.2 Lemma-tag Generation Mode

Dušek and Jurčíček (2016) experiment with generating syntactic trees and realizing them using an external surface realizer; they report slightly worse performance than generating tokens directly.

In order to fight data sparsity coming from the rich morphology of Czech, we decided to explore the middle ground between syntactic trees and full word-form generation: generating base forms (lemmas) and morphological tags that indicate how the form should be inflected. We train TGen to simply generate an interleaved sequence of lemmas and tags (see Figure 4), which are then postprocessed using the dictionary-based morphological generator of Straková et al. (2014) to obtain the inflected word forms.

In the lemma-tag mode, the set of possible output tokens is reduced compared to direct token generation, but the postprocessing step is much simpler than using a full syntactic surface realizer. Moreover, the generated morphological tags following slot placeholders can be used to limit the scope of possible surface forms during lexicalization (see Section 3.3).

This approach is inspired by similar approaches in phrase-based MT (Bojar, 2007; Toutanova et al., 2008; Fraser, 2009) and was developed in parallel to recent similar experiments with two-step neural MT (Nadejde et al., 2017; Tamchyna et al., 2017). We compare the lemma-tag generation mode against the TGen default direct word-form generation mode in our experiments.

### 3.3 Lexicalization

We experiment with three different approaches for selecting the surface form for a DA slot value placeholder from a set of applicable ones – two very straightforward baselines requiring no training and our proposed solution based on a neural LM:

- *Random baseline.* This selects a surface form at random. This approach is certainly not suitable for a real application, we only use it for comparison.
- *Most frequent baseline.* Here, the applicable surface form that occurs overall most frequently in the training data is selected. This represents a stronger baseline than the random method.
- *RNN-based language model.* Our main solution attempts to choose the best surface form using a bidirectional LSTM RNN-based LM (Mikolov et al., 2010), trained to predict a token probability distribution given all previous and all following tokens. During decoding, the RNN LM estimates the probabilities of all applicable surface forms, and we select the most probable surface form for the output.

When selecting a surface form during direct word-form generation, all possible forms for the given slot value are considered. In the lemma-tag mode (Section 3.2), only forms matching the morphological tag following the slot placeholder are considered (cf. Figure 4) – first the ones matching perfectly, with backoffs to coarse part-of-speech or all possible forms.

[8] DA type is repeated for each slot-value pair.

| | | | | | |
|---|---|---|---|---|---|
| *hledat* | VB-P---2P-AA--- | *vhodný* | AAFS4----1A---- | *restaurace* | NNFS4-----A---- |
| search | verb, 2nd person present formal | suitable | adjective, fem sg acc | restaurant | noun, fem sg acc |
| *na* | RR--4---------- | *X-good_for_meal* | NNFS4-----A---- | *?* | Z:------------- |
| for | preposition, acc | slot placeholder | noun, fem sg acc | ? | final punctuation |

Figure 4: Example interleaved lemma-tag sequence for the input DA *?confirm(good_for_meal=breakfast)*, the first output from Figure 1 (acc = accusative, fem = feminine, sg = singular; cf. (Hajič, 2004) for tagset details). Note that the morphological tag for the slot placeholder is included and can be used during lexicalization (cf. Section 3.3).

### 3.4 Lexicalized Input DAs

Some slot values in our dataset may require certain morphosyntactic structure of their neighborhood. This is the case for restaurant counts: Czech cardinal numerals 2-4 behave as adjectives, while higher numerals behave as nouns and take the counted quantity as genitive object. The correct nominative forms when counting restaurants are then "2 restaurac*e*", but "5 restaurac*í*" (Naughton, 2005, p. 113ff.). Another example are area names requiring different prepositions for location – the correct form for "in Malá Strana" is "*na* Malé Straně", but for "in Karlín", it is "*v* Karlíně" (Naughton, 2005, p. 202).

Therefore, inspired by Sharma et al. (2017), we test using fully lexicalized input DAs with the main generator to check if it learns to produce more appropriate structure for concrete values (while still producing delexicalized output).[9] We compare this setup against the default with delexicalized DAs.

## 4 Experiments

### 4.1 Experimental Setup

We test all combinations of the features described in Section 3:

- Direct token vs. lemma-tag generation
- Random / most-frequent / RNN LM lexicalizer
- Delexicalized vs. lexicalized input DAs

We train the resulting 12 model variants using the Adam optimizer (Kingma and Ba, 2015) to minimize cross entropy on the training set; this approach is used for all parts of the system: the main seq2seq generator, the reranker, and the RNN LM lexicalizer. After each training data pass, we validate the models and keep the best-performing parameters. We use BLEU score (Papineni et al., 2002), classification error, and LM perplexity as the respective validation criteria. We set hyperparameters based on TGen defaults for other datasets and a few experiments on the development set.[10]

Training the baseline lexicalizers is trivial: the random baseline does not require any training, it simply uses the list of possible surface forms; the most frequent baseline just memorizes surface form frequencies in the training data.

To reduce the effect of random initialization, we train five runs using different random seeds and use results of all of them for evaluation. In addition, we fix the random seeds so that identical seq2seq generators and rerankers are used in setups that only differ in the lexicalization method.

### 4.2 Metrics

We use the suite of word-overlap-based automatic metrics from the E2E NLG Challenge (Dušek et al., 2019),[11] supporting BLEU (Papineni et al., 2002), NIST (Doddington, 2002), ROUGE-L (Lin, 2004), METEOR (Lavie and Agarwal, 2007) and CIDEr (Vedantam et al., 2015). Although multiple texts often correspond to the same delexicalized DA, we treat each instance individually both in training and testing since the particular slot values influence the shape of the whole sentence (see Sections 2.4 and 3.4). This means that only a single reference output per instance is available to be used with automatic metrics (see Section 4.3).

[9] We exploit the fact that the number of possible values for different slots in the dataset is relatively small (cf. Section 2); morphosyntactic classes of the values would need to be used if the number of values was higher.

[10] The main generator uses embedding and LSTM cell size 200, learning rate 0.005, dropout rate 0.5, and batch size 20. At least 50 and up to 1000 training data passes are used, with early stopping if the top 10 validation BLEU scores do not change for 50 passes. Beam size 20 is used for decoding.

The reranker uses embedding and LSTM cell size 50, no dropout, learning rate 0.001, and batch size 20. Training runs for 100 passes, performance is validated starting with pass 10. The reranker is validated both on training and development data; classification error on the development set is given 10 times more weight than training set error.

The RNN LM lexicalizer uses the same parameters as the reranker, with training for 50 passes maximum and validation (on development data only) starting after the first pass.

[11] https://github.com/tuetschek/e2e-metrics

| Input DAs | Generator mode | Lexicalizer | BLEU | NIST | METEOR | ROUGE-L | CIDEr | SER |
|---|---|---|---|---|---|---|---|---|
| Delexicalized | Word forms | Random | $15.51^{\ddagger}$ | 3.7352 | 18.60 | 35.00 | 1.3922 | **00.70** |
| | | Most frequent | $20.28^{\ddagger}$ | 4.5192 | 22.69 | 40.92 | 1.9399 | **00.70** |
| | | RNN LM | $20.74^{*}$ | 4.5096 | 22.61 | 40.72 | 1.9924 | **00.70** |
| | Lemma-tag | Random | $19.66^{\dagger}$ | $4.4884^{\dagger\ddagger}$ | 22.19 | 41.42 | 1.8844 | 01.85 |
| | | Most frequent | $21.21^{\dagger\ddagger}$ | $4.6900^{\dagger\ddagger}$ | 23.07 | 42.62 | 2.0983 | 01.85 |
| | | RNN LM | **21.96**$^{*\dagger\ddagger}$ | **4.7720**$^{*\dagger\ddagger}$ | **23.32** | **42.95** | **2.1783** | 01.85 |
| Lexicalized | Word forms | Random | 14.70 | 3.7595 | 18.29 | 35.64 | 1.3712 | 02.30 |
| | | Most frequent | 19.73 | 4.5618 | 22.45 | 41.71 | 1.9473 | 02.30 |
| | | RNN LM | $20.48^{*}$ | $4.6060^{*\ddagger}$ | 22.55 | 41.66 | 2.0192 | 02.30 |
| | Lemma-tag | Random | $18.92^{\dagger}$ | $4.3501^{\dagger}$ | 21.76 | 40.55 | 1.8014 | 03.08 |
| | | Most frequent | 19.44 | 4.4453 | 22.22 | 41.26 | 1.8801 | 03.08 |
| | | RNN LM | $20.42^{*}$ | $4.5460^{*}$ | 22.56 | 41.73 | 1.9796 | 03.08 |

Table 3: Automatic metrics results. See Section 4.2 for metrics; scores are averaged over 5 different random initializations, all scores except for NIST and CIDEr are percentages. * = significantly better than the corresponding most frequent baseline lexicalizer, † = significantly better than the corresponding word forms mode, ‡ = significantly better than the corresponding (de)lexicalized input DAs. Significance was assessed using pairwise bootstrap resampling (Koehn, 2004), $p < 0.01$.

| Input DAs | Generator mode | Lexicalizer | S | R | F | I | L | F+I+L | Σ |
|---|---|---|---|---|---|---|---|---|---|
| Delexicalized | Word forms | Most frequent | **8** | **0** | **5** | 11 | 57 | 73 | 81 |
| | | RNN LM | **8** | **0** | **5** | 11 | 25 | 41 | 49 |
| | Lemma-tag | Most frequent | 12 | 2 | **5** | 11 | 45 | 61 | 75 |
| | | RNN LM | 12 | 2 | **5** | 11 | 6 | 22 | 36 |
| Lexicalized | Word forms | Most frequent | 14 | 5 | 14 | 6 | 34 | 54 | 73 |
| | | RNN LM | 14 | 5 | 14 | 6 | 10 | 30 | 49 |
| | Lemma-tag | Most frequent | 15 | 4 | 6 | **4** | 34 | 44 | 63 |
| | | RNN LM | 15 | 4 | 6 | **4** | **4** | **14** | **33** |

Table 4: Manual evaluation results on 100 sampled sentences – absolute numbers of different types of errors (S = semantic errors, R = repetition, F = fluency problems except lexicalization, I = impossible to lexicalize correctly with the given value, L = lexicalization errors). All error types are exemplified in Figure 5.

In addition to word-overlap metrics, we use the slot error rate (SER; Wen et al., 2015b) to evaluate semantic accuracy of the outputs. This metric counts slot placeholders in the output before lexicalization and compares them to slots in the input DA. It reliably measures the amount of missed/added content in all delexicalized slots (cf. Section 2), but the non-delexicalized binary *kids_allowed* slot is ignored.

### 4.3 Results

The automatic metrics scores for all setups are shown in Table 3. In terms of generator mode, using lemma-tag generation significantly[12] improves word-overlap metrics over direct token generation in the delexicalized input setting. However, it also leads to an increased SER. The RNN LM brings a significant[12] improvement over both baselines in all setups; the very low performance of the random baseline only documents that inflection indeed matters for slot values. The lexicalized input DAs did not bring improvement over the delexicalized setting – lexicalized setups seem to perform slightly worse in terms of both word-overlap metrics and SER.

### 4.4 Manual Error Analysis

To obtain a deeper insight into the results and account for automatic metrics' inaccuracy (Novikova et al., 2017; Reiter, 2018), we performed a detailed manual error analysis on a sample of 100 outputs produced by all systems except the ones with random baseline lexicalizers, which clearly perform poorly. This was a blind annotation of semantic and fluency errors; it is not a preference rating. We categorized multiple error types; the results are shown in Table 4.

The analysis confirmed that lexicalized input DAs cause more semantic errors (both missed slots and repetition). On the other hand, the outputs were more fluent in this setting, which is not apparent with automatic metrics. Lemma-tag generation also improves fluency overall, at the cost of increasing the number of semantic errors. The RNN LM lexicalizer leads to significant reduction of lexicalization errors compared to the most frequent

[12] BLEU and NIST differences are statistically significant ($p < 0.01$) according to bootstrap resampling (Koehn, 2004).

| | | |
|---|---|---|
| **Input DA** | | *inform(food=Turkish, name="Green Spirit", price_range=expensive)* |
| **Systems** | Delex. input DAs, any mode<br>Most frequent | Green Spirit je drahé turecká restaurace.<br>is expensive Turkish restaurant |
| | Delex. input DAs, any mode<br>RNN LM | Green Spirit je drahá turecká restaurace.<br>is expensive Turkish restaurant |
| | Lex. DAs, word forms<br>Most frequent | Green Spirit je drahé turecká restaurace<br>is expensive Turkish restaurant |
| | Lex. DAs, word forms<br>RNN LM | Green Spirit je drahá turecká restaurace<br>is expensive Turkish restaurant |
| | Lex. DAs, lemma-tag<br>Most frequent | Green Spirit je drahé restaurace, která podává turecká jídla .<br>expensive restaurant which serves Turkish meals |
| | Lex. DAs, lemma-tag<br>RNN LM | Green Spirit je drahá restaurace, která podává turecká jídla .<br>is expensive restaurant which serves Turkish meals |
| **Input DA** | | *inform(area=dont_care, count=218, food=dont_care, price_range=dont_care, type=restaurant)* |
| **Systems** (any lexicalizer) | Delex. input DAs<br>Word forms mode | Pokud vám nezáleží na druhu jídla , našla jsem 218 restaurace v různých cenových skupinách. *(missing area)*<br>if you don't-care about type of-food I-found restaurants in various price ranges |
| | Delex. input DAs<br>Lemma-tag mode | Pokud vám nezáleží na druhu jídla , našla jsem 218 restaurací v různých cenových skupinách. *(missing area)*<br>if you don't-care about type of-food I-found restaurants in various price ranges |
| | Lex. input DAs<br>Word forms mode | V úvahu připadají 218 restaurací , pokud vám nezáleží na druhu jídla ~~, pokud vám nezáleží na druhu jídla~~ .<br>into consideration come restaurants if you don't-care about type of-food if you don't-care about type of-food<br>*(missing area, price range)* |
| | Lex. input DAs<br>Lemma-tag mode | Mám tu 218 restaurací , pokud vám nezáleží na druhu cenových skupinách. *(missing area, food type)*<br>I-have here restaurants if you don't-care about type price ranges |

Figure 5: Examples from manual error analysis. Errors are marked with color and underlining: *semantic errors*, ~~repetition~~, fluency, impossible to lexicalize correctly, lexicalization (cf. Table 4). In the top example, the RNN LM lexicalizer is able to select the correct feminine singular form, while the most frequent baseline selects a neuter form. In the bottom example, systems with lexicalized input DAs make more semantic errors. The lemma-tag mode is able to select a more appropriate syntactic structure for the numeral 218.

baseline, especially in combination with lemma-tag generation (see top example in Figure 5). None of the systems produce perfect output; they seem to struggle especially with DAs that are very different from the ones found in the training set and/or occur less frequently (see bottom example in Figure 5, cf. Section 2.6). We believe that an increased amount of training data could improve the situation.

## 5 Related Work

NLG experiments for non-English languages are relatively rare and fully trainable approaches even rarer. Our work is, to our knowledge, the first application of neural NLG to a non-English language for data-to-text generation.

Most works concerned with multiple languages focus on surface realization. There have been a few approaches using handcrafted grammars (Bateman, 1997; Allman et al., 2012). The procedural SimpleNLG realizer (Gatt and Reiter, 2009) has also been ported into multiple languages (Bollmann, 2011; Vaudry and Lapalme, 2013; de Oliveira and Sripada, 2014; Mazzei et al., 2016; Ramos-Soto et al., 2017; Cascallar-Fuentes et al., 2018; Chen et al., 2018; de Jong and Theune, 2018). Further works using multilingual rule-based surface realization pipelines were developed in the context of machine translation (Aikawa et al., 2001; Žabokrtský et al., 2008; Dušek et al., 2015). Bohnet et al. (2010) created the first statistical multilingual realizer based on a pipeline of SVMs, the recent surface realization challenge (Mille et al., 2018) then features further fully trainable realizers tested on multiple languages, including neural models.

In data-to-text generation, the recent work of Moussallem et al. (2018) is applied to Portuguese, but is largely rule-based. The works of Chen et al. (2010) and Kim and Mooney (2010) represent the only data-to-text end-to-end NLG system with multilingual experiments known to us; they generate English and Korean sport commentary sentences using an inverted (non-neural) semantic parser. Our dataset is ca. 2.5 times larger and more complex, given the slot value inflection.

Other works on neural non-English NLG solve in fact different tasks from ours: Chinese poetry generation (Zhang and Lapata, 2014; Yi et al., 2017; Wang et al., 2016), non-task-oriented response generation in chatbots (Xing et al., 2016, 2017), or morphological inflection (e.g. Faruqui et al., 2016; Kann and Schütze, 2016).

## 6 Conclusions and Future Work

We presented the first dataset targeted at end-to-end neural non-English NLG, containing Czech texts from the restaurant domain. We show that the task of data-to-text NLG here is harder as slot values require morphological inflection. We apply to our data the freely available, state-of-the-art TGen NLG system (Dušek and Jurčíček, 2016) based on the seq2seq architecture, and we implement two extensions for Czech: (1) an RNN LM model to select the correct inflected surface form for slot values and (2) lemma-tag generation mode, where the generator produces an interleaved sequences of base form and morphological tags, which are postprocessed by a morphological generator. We also experiment with lexicalized and delexicalized slot values in generator inputs. Using both automatic metrics and manual analysis, we show that the RNN LM brings clear benefits. The lemma-tag mode and lexicalized inputs improve fluency but hurt semantic accuracy of the outputs. We release our dataset dataset and all experimental code on GitHub.[13]

In future work, we will collect a large unannotated dataset and pretrain the generator (Chen et al., 2019). We believe that this will lead to increased output fluency and accuracy. We are also considering using machine translation to obtain more synthetic training data points.

## Acknowledgments

This research was supported by the Charles University project PRIMUS/19/SCI/10 and by the Ministry of Education, Youth and Sports of the Czech Republic under the grant agreement LK11221. This work used using language resources distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2015071).

## References

T. Aikawa, M. Melero, L. Schwartz, and A. Wu. 2001. Generation for multilingual MT. In *Proceedings of the MT-Summit*, pages 9–14, Santiago de Compostela, Spain.

T. Allman, S. Beale, and R. Denton. 2012. Linguist's Assistant: A Multi-Lingual Natural Language Generator based on Linguistic Universals, Typologies, and Primitives. In *INLG 2012 Proceedings of the Seventh International Natural Language Generation Conference*, pages 59–66, Utica, IL, USA.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *3rd International Conference on Learning Representations (ICLR2015)*, San Diego, CA, USA. arXiv:1409.0473.

J. A. Bateman. 1997. Enabling technology for multilingual natural language generation: the KPML development environment. *Natural Language Engineering*, 3(1):15–55.

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

Ondřej Bojar. 2007. English-to-Czech factored machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation – StatMT '07*, pages 232–239, Prague, Czech Republic.

M. Bollmann. 2011. Adapting SimpleNLG to German. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 133–138, Nancy, France.

Andrea Cascallar-Fuentes, Alejandro Ramos-Soto, and Alberto Bugarín Diz. 2018. Adapting SimpleNLG to Galician language. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 67–72, Tilburg, The Netherlands.

David L. Chen, Joohyun Kim, and Raymond J. Mooney. 2010. Training a multilingual sportscaster: Using perceptual context to learn language. *Journal of Artificial Intelligence Research*, 37:397–435.

Guanyi Chen, Kees van Deemter, and Chenghua Lin. 2018. SimpleNLG-ZH: a Linguistic Realisation Engine for Mandarin. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 57–66, Tilburg, The Netherlands.

Zhiyu Chen, Harini Eavani, Yinyin Liu, and William Yang Wang. 2019. Few-shot NLG with Pretrained Language Model. *arXiv:1904.09521 [cs]*.

Ryan Cotterell, Sebastian J. Mielke, Jason Eisner, and Brian Roark. 2018. Are All Languages Equally Hard to Language-Model? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 536–541, New Orleans, LA, USA.

George Doddington. 2002. Automatic Evaluation of Machine Translation Quality Using N-gram Co-occurrence Statistics. In *Proceedings of the Second International Conference on Human Language*

[13] Dataset: `https://github.com/UFAL-DSG/cs_restaurant_dataset`, code: `https://github.com/UFAL-DSG/tgen`.

*Technology Research*, HLT '02, pages 138–145, San Francisco, CA, USA.

Ondřej Dušek, Ondřej Plátek, Lukáš Žilka, and Filip Jurčíček. 2014. 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*, pages 79–83, Philadelphia, PA, USA.

Ondřej Dušek, Luís Gomes, Michal Novák, Martin Popel, and Rudolf Rosa. 2015. New Language Pairs in TectoMT. In *Proceedings of the 10th Workshop on Machine Translation*, pages 98–104, Lisbon, Portugal.

Ondřej Dušek and Filip Jurčíček. 2016. Sequence-to-Sequence Generation for Spoken Dialogue via Deep Syntax Trees and Strings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 45–51, Berlin. arXiv:1606.05491.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2019. Evaluating the State-of-the-Art of End-to-End Natural Language Generation: The E2E NLG Challenge. *Computer Speech & Language*, 59:123–156. arXiv:1901.07931.

Manaal Faruqui, Yulia Tsvetkov, Graham Neubig, and Chris Dyer. 2016. Morphological Inflection Generation Using Character Sequence to Sequence Learning. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 634–643, San Diego, CA, USA. arXiv:1512.06110.

Alexander Fraser. 2009. Experiments in Morphosyntactic Processing for Translating to and from German. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 115–119, Athens, Greece.

A. Gatt and E. Reiter. 2009. SimpleNLG: A realisation engine for practical applications. In *Proceedings of the 12th European Workshop on Natural Language Generation*, pages 90–93.

Sebastian Gehrmann, Falcon Z. Dai, Henry Elder, and Alexander M. Rush. 2018. End-to-End Content and Plan Selection for Data-to-Text Generation. In *Proceedings of the 11th International Conference on Natural Language Generation*, Tilburg, The Netherlands. arXiv:1810.04700.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O. K. Li. 2016. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1631–1640, Berlin, Germany. arXiv:1603.06393.

Jan Hajič. 2004. *Disambiguation of rich inflection: computational morphology of Czech*. Karolinum, Praha.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, UK.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv:1207.0580 [cs]*.

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

Ruud de Jong and Mariët Theune. 2018. Going Dutch: Creating SimpleNLG-NL. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 73–78, Tilburg, The Netherlands.

Juraj Juraska, Panagiotis Karagiannis, Kevin K. Bowden, and Marilyn A. Walker. 2018. A Deep Ensemble Model with Slot Alignment for Sequence-to-Sequence Natural Language Generation. In *NAACL*, New Orleans, LA, USA. arXiv:1805.06553.

Katharina Kann and Hinrich Schütze. 2016. Single-Model Encoder-Decoder with Explicit Morphological Representation for Reinflection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 555–560, Berlin, Germany. arXiv:1606.00589.

Joohyun Kim and Raymond J. Mooney. 2010. Generative Alignment and Semantic Parsing for Learning from Ambiguous Supervision. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 543–551, Beijing, China.

Diederik Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, CA, USA. arXiv:1412.6980.

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

Gerasimos Lampouras and Andreas Vlachos. 2016. Imitation learning for language generation from unaligned data. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1101–1112, Osaka, Japan.

Alon Lavie and Abhaya Agarwal. 2007. Meteor: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, Prague, Czech Republic.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, pages 74–81. Barcelona, Spain.

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

Alessandro Mazzei, Cristina Battaglino, and Cristina Bosco. 2016. SimpleNLG-IT: Adapting SimpleNLG to Italian. In *The 9th International Natural Language Generation conference*, pages 184–192, Edinburgh, Scotland, UK.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Interspeech*, pages 1045–1048, Makuhari, Japan.

Simon Mille, Anja Belz, Bernd Bohnet, Yvette Graham, Emily Pitler, and Leo Wanner. 2018. The First Multilingual Surface Realisation Shared Task (SR'18): Overview and Evaluation Results. In *Proceedings of the First Workshop on Multilingual Surface Realisation*, pages 1–12, Melbourne, Australia.

Diego Moussallem, Thiago Castro Ferreira, Marcos Zampieri, Maria Claudia Cavalcanti, Geraldo Xexéo, Mariana Neves, and Axel-Cyrille Ngonga Ngomo. 2018. RDF2pt: Generating Brazilian Portuguese Texts from RDF Data. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan. arXiv:1802.08150.

Maria Nadejde, Siva Reddy, Rico Sennrich, Tomasz Dwojak, Marcin Junczys-Dowmunt, Philipp Koehn, and Alexandra Birch. 2017. Predicting Target Language CCG Supertags Improves Neural Machine Translation. In *Proceedings of the Second Conference on Machine Translation*, pages 68–79, Copenhagen, Denmark.

James Naughton. 2005. *Czech : an essential grammar*. Routledge, London.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. Why We Need New Evaluation Metrics for NLG. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2243–2253, Copenhagen, Denmark. arXiv:1707.06875.

A. H. Oh and A. I. Rudnicky. 2000. Stochastic language generation for spoken dialogue systems. In *Proceedings of the 2000 ANLP/NAACL Workshop on Conversational systems-Volume 3*, page 27–32, Seattle, WA, USA.

Rodrigo de Oliveira and Somayajulu Sripada. 2014. Adapting SimpleNLG for Brazilian Portuguese realisation. In *Proceedings of the 8th International Natural Language Generation Conference (INLG)*, pages 93–94, Philadelphia, PA, USA.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Ellie Pavlick, Matt Post, Ann Irvine, Dmitry Kachaev, and Chris Callison-Burch. 2014. The language demographics of Amazon Mechanical Turk. *Transactions of the Association for Computational Linguistics*, 2:79–92.

Martin Popel and Zdeněk Žabokrtský. 2010. TectoMT: modular NLP framework. In *Proceedings of IceTAL, 7th International Conference on Natural Language Processing*, pages 293–304, Reykjavík, Iceland.

Alejandro Ramos-Soto, Julio Janeiro-Gallardo, and Alberto Bugarín Diz. 2017. Adapting SimpleNLG to Spanish. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 144–148, Santiago de Compostela, Spain.

Ehud Reiter. 2018. A Structured Review of the Validity of BLEU. *Computational Linguistics*, 44(3):1–8.

Shikhar Sharma, Jing He, Kaheer Suleman, Hannes Schulz, and Philip Bachman. 2017. Natural Language Generation in Dialogue using Lexicalized and Delexicalized Data. In *ICLR Workshop Track*, Toulon, France. arXiv:1606.03632.

Milan Straka and Jana Straková. 2017. Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada.

Jana Straková, Milan Straka, and Jan Hajič. 2014. 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*, pages 13–18, Baltimore, MD, USA.

Aleš Tamchyna, Marion Weller-Di Marco, and Alexander Fraser. 2017. Modeling Target-Side Inflection in Neural Machine Translation. In *Proceedings of the Conference on Machine Translation (WMT), Volume 1: Research Papers*, Copenhagen, Denmark. arXiv:1707.06012.

Kristina Toutanova, Hisami Suzuki, and Achim Ruopp. 2008. Applying morphology generation models to machine translation. In *Proceedings of ACL-08: HLT*, pages 514–522, Columbus, OH, USA.

Pierre-Luc Vaudry and Guy Lapalme. 2013. Adapting SimpleNLG for bilingual English-French realisation. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 183–187, Sofia, Bulgaria.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. CIDEr: Consensus-Based Image Description Evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4566–4575, Boston, MA, USA. arXiv:1411.5726.

Zhe Wang, Wei He, Hua Wu, Haiyang Wu, Wei Li, Haifeng Wang, and Enhong Chen. 2016. Chinese Poetry Generation with Planning based Neural Network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1051–1060, Osaka, Japan. arXiv:1610.09889.

Tsung-Hsien Wen, Milica Gasic, Dongho Kim, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015a. 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*, pages 275–284, Prague, Czech Republic.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015b. Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721, Lisbon, Portugal.

Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2016. Topic Augmented Neural Response Generation with a Joint Attention Mechanism. *arXiv:1606.08340 [cs]*.

Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2017. Topic Aware Neural Response Generation. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, pages 3351–3357, San Francisco, CA, USA.

Xiaoyuan Yi, Ruoyu Li, and Maosong Sun. 2017. Generating Chinese Classical Poems with RNN Encoder-Decoder. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 211–223, Nanjing, China. arXiv:1604.01537.

Xingxing Zhang and Mirella Lapata. 2014. Chinese poetry generation with recurrent neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 670–680, Doha, Qatar.

Z. Žabokrtský, J. Ptáček, and P. Pajas. 2008. TectoMT: highly modular MT system with tectogrammatics used as transfer layer. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 167–170, Columbus, OH, USA.