How to Make Neural Natural Language Generation as Reliable as Templates in Task-Oriented Dialogue

Henry Elder
ADAPT Centre
Dublin City University
henry.elder@adaptcentre.ie

Alexander O'Connor
Autodesk, inc.
alex.oconnor@autodesk.com

Jennifer Foster
School of Computing
Dublin City University
jennifer.foster@dcu.ie

Abstract

Neural Natural Language Generation (NLG) systems are well known for their unreliability. To overcome this issue, we propose a data augmentation approach which allows us to restrict the output of a network and guarantee reliability. While this restriction means generation will be less diverse than if randomly sampled, we include experiments that demonstrate the tendency of existing neural generation approaches to produce dull and repetitive text, and we argue that reliability is more important than diversity for this task. The system trained using this approach scored 100% in semantic accuracy on the E2E NLG Challenge dataset, the same as a template system.

1 Introduction

The goal of task oriented dialogue is to help a user achieve a narrow goal, such as booking a restaurant or movie ticket. The final step of a conversational interface is generating a response to the user; more specifically, performing surface realization of some structured data containing relevant information.

Research into neural NLG systems for the surface realization task is popular because such systems may have advantages over the dominant rule and template-based systems: neural NLG systems trained on datasets may be both easier to maintain and to scale to new domains, as well as generating more natural responses (Wen et al., 2015; Guo and Zhao, 2017). But neural NLG systems are not without problems. They are widely considered too unreliable for business applications; they have a tendency to *hallucinate* facts, unsupported by the structured data they were given (Wiseman et al., 2017).

A less well known issue is the template-like generation of neural NLG systems (Wei et al., 2019). Figure 1 highlights this issue; neural NLG systems (TGen and Slug2Slug) are far less diverse than the training data (E2E Dataset) in their usage of surface forms that express an attribute. Intuitively, one might expect that a neural NLG system trained on a dataset with 75 different surface forms to express an attribute would use a wide variety of them – instead we see only the top two most common surface forms in use.

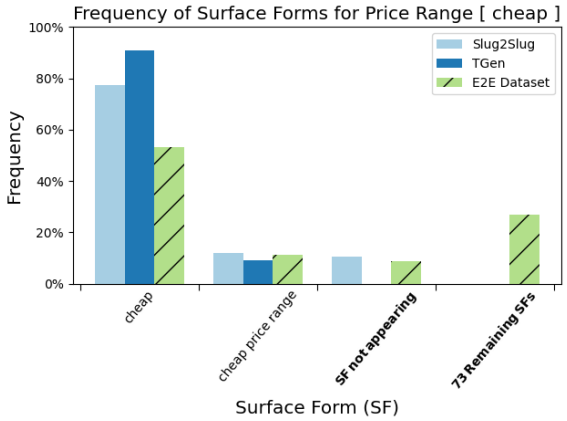

Figure 1: Surface forms used to express the attribute-value pair: PriceRange[Cheap]. **SF not appearing**: no surface form was found in the utterance, **73 Remaining SFs**: surface forms other than *cheap* and *cheap price range*

We highlight the issue of lack of diversity, not to provide a specific solution to it, but rather to provide some context for our proposal which relates to *reliability* of neural NLG systems. Given that our main goal is reliability, we wondered if there were some way to *lean into* the blandness and lack of diversity of existing neural systems.

We propose a data-oriented and model-agnostic solution. Using the E2E NLG Challenge dataset (Dušek et al., 2019b), we experiment with an augmented input sequence that includes the surface form of each attribute-value pair. By including the surface form in the input sequence, we can use a restricted decoding strategy when generating an

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 2877–2888,
November 16–20, 2020. ©2020 Association for Computational Linguistics

utterance. This guarantees reliability. By sacrificing a small amount of unconstrained diversity, we are able to achieve 100% semantic accuracy on the E2E dataset.

2 Frequency of Surface Forms

To compare the diversity of the E2E training data with that of the generated text, we looked at the surface forms used to express each attribute-value pair. This is enabled by a set of regular expressions released by Dušek et al. (2019a)[1]. The regular expressions capture the entire phrase used to express an attribute-value pair, focusing on the content words and attempting as much as possible to leave out the function words, e.g.

```
(?:(?:price|range).*)?
 (?:inexpensive|cheap)(?:ly)?
 (?:.*(?:price\w|range))?
```

We counted the surface forms used for a given attribute-value pair, in both the dataset and generated text, and plotted them against each other, see Figure 1 and additional figures in the supplementary material. While there was an average of 133 different surface forms for each attribute-value pair in the E2E dataset, the neural systems, on average, only generated 3 of the most common surface forms. This convinced us that the diversity was hardly any better than templates, which by default only use a single surface form per attribute.

3 Method

How can a neural NLG system generate text from a set of attribute-value pairs and ensure that they appear correctly in the generated text? As opposed to templates, which are static, neural NLG systems are statistical generators and provide no inherent guarantees of accuracy. Thus we propose augmenting the input sequence with the surface form of each attribute-value pair. This augmented input sequence enables us to restrict the text that is generated in a way that provides guaranteed accuracy.

Finding Surface Forms The first step in this process is finding the surface forms in a given utterance. We want to find the content words used to describe attribute-value pairs in a human authored sentence. But this is not a straightforward task. Specially designed regular expressions (Dušek et al., 2019a) or heuristics involving dependency relations (Oraby et al., 2019) must be used.

[1] https://github.com/tuetschek/e2e-cleaning

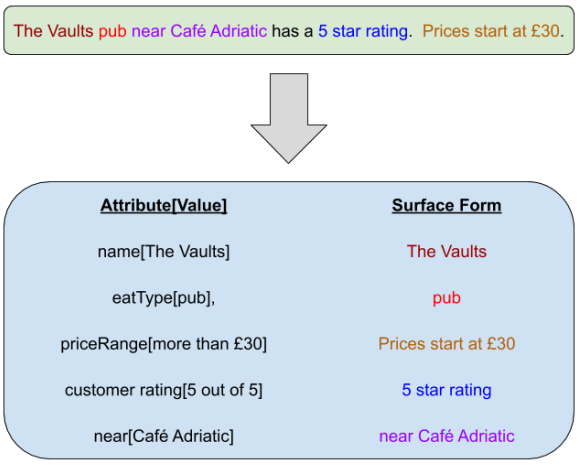

Figure 2: First, use regular expressions to find the surface forms in a target utterance. Then, use the surface forms to construct an augmented input sequence.

Augmented Input Sequence Once the surface forms of each attribute-value pair in a target utterance are found, we add them to the input sequence, as shown in Figure 2. Only the input is altered, the target utterance remains the same.

How can we add surface forms to an input sequence from the validation or test sets without peeking at its target utterance? A simple heuristic we have used is to choose the most common surface form for each attribute-value from the training set.

Restricted Decoding Why do we focus so much on surface forms? Because when surface forms are part of the input, we can add restrictions to the generation strategy, e.g. beam search, which guarantee that all, and only, the surface forms provided have been expressed (Zhong et al., 2017).

Furthermore, by including all the necessary content words in the input sequence, it is possible to limit the vocabulary used during generation to only these content words and a couple of hundred function words. This would enable the use of a constrained softmax (Hu et al., 2015) – an optimization that can greatly speed up the decoding step.

4 Experimental Setup

We performed experiments with the E2E NLG Challenge dataset (Dušek et al., 2019b). It is a task-oriented dialogue dataset, collected using crowd sourcing, focused on the surface realization of attribute-value pairs describing restaurants.

4.1 Applying the Surface Forms Method

To extract the surface form of each attribute-value pair from a target utterance, we used modified regular expressions from Dušek et al. (2019a). The input sequence was constructed in the format of a single token representing an attribute-value pair followed by multiple tokens for the surface form, e.g. **eatType_pub** *pub* **customer_rating_5_out_of_5** *5 star rating*. The order of the attribute-value pairs remained the same as in the original dataset.

If the surface form of an attribute-value pair was not found in the target utterance then a *missing* token was added instead. Any additional attribute-values, those that appeared in the target utterance but not in the input, were ignored. To avoid peeking at the target utterance when adding surface forms to the validation and test sets, the most common surface form for each attribute-value pair from the training set was used.

The task proved to be simple enough for the model that only minimal restricted decoding was necessary. We added a single rule to the beam search: if *restaurant* does not appear in the input then it should not appear in the output.

4.2 Modelling

Our baseline is a sequence-to-sequence model with copy attention, trained on the E2E dataset, using the neural machine translation framework OpenNMT (Klein et al., 2017). To test our method, we trained a model with the same hyperparameters (see the appendix for details) on the surface form augmented version of the E2E dataset.

4.3 Reference Systems

The E2E NLG Challenge organisers released the generated outputs of all participant systems. In our analysis, we compare with three of these systems:

1. the E2E baseline, TGen (Dušek and Jurcicek, 2016), a neural system with a semantic reranker as a final step to improve accuracy
2. the overall winner of E2E, Slug2Slug (Juraska et al., 2018), a neural system, also with a reranker, trained using an augmented dataset in which attribute-value pairs are aligned to individual sentences in the utterance
3. a template based-system, TUDA (Puzikov and Gurevych, 2018), which, by using a set of handwritten templates, was able to express attributes more reliably than all other systems and came in second place in the challenge's human evaluation.

	OK	Added	Missing	A+M
♡ TGen	502 (80%)	14 (2%)	100 (16%)	14 (2%)
♡ Slug2Slug	582 (92%)	0	23 (4%)	25 (4%)
♡ OpenNMT (Baseline)	426 (68%)	13 (2%)	191 (30%)	0
♢ OpenNMT + Surface Forms	630 (100%)	0	0	0
♠ TUDA	630 (100%)	0	0	0

Table 1: Semantic accuracy on the test set. System architectures are coded with colours and symbols: ♡seq2seq, ♢augmented data ♠template-based

	BLEU	NIST	METEOR	ROUGE_L	CIDEr
Validation					
♡ TGen	0.6925	8.4781	0.4703	0.7257	2.3987
♡ Slug2Slug	0.6576	8.0761	0.4675	0.7029	-
♡ OpenNMT (Baseline)	**0.7415**	**8.7010**	**0.4898**	**0.7663**	**2.5999**
♢ OpenNMT + Surface Forms	0.6589	8.4099	0.4372	0.6907	2.2848
♠ TUDA	0.6051	7.5257	0.4678	0.6890	1.6997
Test					
♡ TGen	0.6593	8.6094	0.4483	0.6850	2.2338
♡ Slug2Slug	0.6619	8.6130	0.4454	0.6772	**2.2615**
♡ OpenNMT (Baseline)	**0.6815**	**8.7481**	0.4452	**0.6904**	2.2391
♢ OpenNMT + Surface Forms	0.6283	8.3107	0.4277	0.6682	2.1465
♠ TUDA	0.5657	7.4544	**0.4529**	0.6614	1.8206

Table 2: N-gram overlap metrics for validation and test sets. System architectures are coded with colours and symbols: ♡seq2seq, ♢augmented data, ♠template-based

4.4 Evaluation

To evaluate the performance of our proposed approach we focus on semantic accuracy. Semantic accuracy scoring was also provided by Dušek et al. (2019a). It reports the number of generated utterances that: correctly express all attribute-value pairs (OK), have additional pairs (Added), are missing pairs (Missing), have both missing and added pairs (A+M).

For completeness, we report results from the E2E NLG Challenge's official scoring script, which is comprised of the following n-gram overlap metrics; BLEU (Papineni et al., 2002), NIST (Doddington, 2002), METEOR (Lavie and Agarwal, 2007), ROUGE (Lin, 2004), and CIDEr (Vedantam et al., 2015). The validation and test sets contain multiple human-authored references for each input sequence, which helps to alleviate some of the issues with n-gram overlap metrics.

5 Results

5.1 Semantic Accuracy

Table 1 demonstrates that the semantic accuracy of our proposed method is on par with that of the template system; both achieve 100% accuracy, whereas the neural systems struggle, with the best system, Slug2Slug, only achieving 92%. Our baseline OpenNMT system performs particularly poorly as

♡Blue Spice is a coffee shop in the city centre. ♢Blue Spice is a coffee shop in the city centre. ♠Blue Spice is a coffee shop located in the city centre area.
♡Blue Spice is a coffee shop near Crowne Plaza Hotel with a customer rating of 5 out of 5. ♢Blue Spice is a coffee shop near Crowne Plaza Hotel with a customer rating of 5 out of 5. ♠Blue Spice is a coffee shop located near Crowne Plaza Hotel. It has a customer rating of 5 out of 5.
♡The Cricketers is a family friendly coffee shop near Avalon. It has a customer rating of 1 out of 5. ♢The Cricketers is a family friendly coffee shop near Avalon with a customer rating of 1 out of 5. ♠The Cricketers is a family-friendly coffee shop located near Avalon. It has a customer rating of 1 out of 5.
♡Blue Spice is a Chinese pub located in the city centre near Rainbow Vegetarian Café. It is not family-friendly. ♢Blue Spice is a Chinese pub near Rainbow Vegetarian Café in the city centre. It is not family-friendly. ♠Blue Spice is a pub which serves Chinese food. It is located in the city centre area, near Rainbow Vegetarian Café. It is not family friendly.
♡The Mill is a high priced English pub in the riverside area near Raja Indian Cuisine. It is child friendly. ♢The Mill is a family friendly English pub in the riverside area near Raja Indian Cuisine with a high price range. ♠The Mill is a family-friendly pub which serves English food in the high price range. It is located in the riverside area, near Raja Indian Cuisine.
♡The Cricketers is a Chinese restaurant in the city centre near All Bar One. It has a price range of £20-25 and is not kid friendly and has a high customer rating. ♢The Cricketers is a Chinese restaurant in the city centre near All Bar One. It has a high customer rating, is not family-friendly, and has a price range of £20-25. ♠The Cricketers is a restaurant which serves Chinese food in the price range of £20-25. It has a high customer rating and is located in the city centre area, near All Bar One. It is not family friendly.

Table 3: Examples of generated text are coded with colours and symbols: ♡SLUG2SLUG, ♢OPENNMT + SURFACE FORMS, ♠TUDA

it does not use a semantic reranker. In a business setting, where automated task-oriented dialogue is most likely to be applied, nothing less than 100% accuracy is typically acceptable, especially when it comes to a relatively new technology like deep neural networks.

5.2 N-gram Overlap Metrics

According to the automated results on the E2E validation and test sets, shown in Table 2, semantic accuracy and n-gram overlap metrics have little correlation. The highest scoring system in many of the n-gram metrics, the OpenNMT baseline, is the worst performing in semantic accuracy, while the template system scores highest in METEOR but lowest in all other metrics. Overall, we infer that the n-gram metrics results are ambiguous, making it difficult to draw useful conclusions from them.

5.3 Generated examples

In Table 3, we compare randomly selected examples from Slug2Slug, our Surface Forms system and TUDA. In each of the examples, the systems appear to follow a very similar sentence structure to each other. In the E2E human evaluation for naturalness, Slug2Slug came second while TUDA came eighth, compared with the human evaluation for overall quality in which Slug2Slug came first and TUDA second. Dušek et al. (2019a) hypothesised that the lower performance in naturalness may be linked to sentence length; template systems tend to be slightly longer than neural ones. Slug2Slug has an average utterance length of 24 tokens, while TUDA has an average length of 32 tokens. Our system has an average length of 23, which is closer to that of Slug2Slug. This suggests that our approach has the potential to combine the reliability of template systems with the perceived naturalness of neural ones.

6 Discussion

This is not the first data-focused approach to improving accuracy; Balakrishnan et al. (2019) also proposed a constrained decoding strategy. The

difference between our decoding strategies lies in the guarantees provided. Their approach focuses on an augmented target utterance, as opposed to input sequence, in which special bracket tokens are used to surround surface forms. e.g. **[__ARG_AREA_CITY_CENTRE__** *city centre* **]**. Their constrained decoding strategy guarantees that when an opening bracket is generated, a closing bracket will also be generated. However this provides no guarantee as to what will be contained within the brackets. What sets our method apart is that: we can guarantee the text will actually be generated as requested, we generate shorter sequences (no bracket tokens in the output) and have the option for a restricted vocabulary, which speeds up decoding.

The major weakness of both approaches, however, is the difficulty of extracting surface forms from human-authored text. We were able to avail of the hand-crafted regular expressions of Dusek et al in our E2E experiments, but moving to another dataset would entail a similar exercise. A method to do this automatically would be convenient. Some work has already been done by Oraby et al. (2019), in which dependency trees were used to find adjectives that describe a specific list of food related nouns. In the Surface Realization shared task (Mille et al., 2018), the deep task dataset was created by pruning function words from a dependency tree, leaving only content words remaining.

In our proposed method, surface forms still need to be joined together with function words. We believe neural networks are well suited for this task because they are good at generating natural sounding, though sometimes nonsensical, text. By combining neural generation with constraints based on content words included in the input sequence, we aim to achieve both reliability and naturalness.

An alternate approach, which we did not compare with, is automatic template generation (Biran et al., 2016; Wiseman et al., 2018). However, as with neural generation, when applied to the E2E task it has issues with reliability. Mille and Dasiopoulou (2017) used an automated template generation approach on the E2E shared task and their accuracy score was similar to that of a neural system, 92% (Dušek et al., 2019b), mostly due to missing attributes in templates.

However, the question remains: why pursue this approach when templates perform satisfactorily? We believe that neural NLG systems are easier to maintain, generate more natural text, and, as surface form extraction improves, they also become more scalable: to new domains, languages, and, possibly even, personalization.

In our proposed approach, we purposefully remove a neural NLG system's ability to generate diverse text. While this may seem perverse, we consider reliability to be the most important starting point. Diversity can always be increased later. If augmenting an input sequence with surface forms allows us to restrict decoding and generate utterances that are as reliable as templates, then this is an approach worth investigating further.

Acknowledgments

We thank the anonymous reviewers for their helpful comments. This research is supported by Science Foundation Ireland in the ADAPT Centre for Digital Content Technology. The ADAPT Centre for Digital Content Technology is funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund.

References

Anusha Balakrishnan, Jinfeng Rao, Kartikeya Upasani, Michael White, and Rajen Subba. 2019. Constrained Decoding for Neural NLG from Compositional Representations in Task-Oriented Dialogue. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 831–844, Stroudsburg, PA, USA. Association for Computational Linguistics.

Or Biran, Terra Blevins, and Kathleen McKeown. 2016. Mining Paraphrasal Typed Templates from a Plain Text Corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1913–1923, Stroudsburg, PA, USA. Association for Computational Linguistics.

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 Technology Research*, pages 138–145, San Diego, CA, USA. ®˙.

Ondřej Dušek, David M. Howcroft, and Verena Rieser. 2019a. Semantic Noise Matters for Neural Natural Language Generation. In *Proceedings of the 12th International Conference on Natural Language Generation*, October, pages 421–426, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ondřej Dušek and Filip Jurcicek. 2016. 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*, September, pages 185–190, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2019b. Evaluating the State-of-the-Art of End-to-End Natural Language Generation: The E2E NLG Challenge. *arXiv preprint arXiv:1901.11528*.

Yufeng Guo and Justin Zhao. 2017. Natural Language Generation at Google Research (AI Adventures).

Xiaoguang Hu, Wei Li, Xiang Lan, Hua Wu, and Haifeng Wang. 2015. Improved beam search with constrained softmax for NMT. *Machine Translation Summit XV*, 1(2014):297–309.

Juraj Juraska, Panagiotis Karagiannis, Kevin Bowden, and Marilyn Walker. 2018. A Deep Ensemble Model with Slot Alignment for Sequence-to-Sequence Natural Language Generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 152–162, Stroudsburg, PA, USA. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Stroudsburg, PA, USA. Association for Computational Linguistics.

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*, StatMT '07, pages 228–231, Stroudsburg, PA, USA.

C Y Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the workshop on text summarization branches out (WAS 2004)*, 1, pages 25–26.

S Mille and S Dasiopoulou. 2017. FORGe at E2E 2017.

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 1st Workshop on Multilingual Surface Realisation (MSR), 56th Annual Meeting of the Association for Computational Linguistics*, pages 1–10, Melbourne, Australia.

Shereen Oraby, Vrindavan Harrison, Abteen Ebrahimi, and Marilyn Walker. 2019. Curate and Generate: A Corpus and Method for Joint Control of Semantics and Style in Neural NLG. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5938–5951, Stroudsburg, PA, USA. Association for Computational Linguistics.

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 on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yevgeniy Puzikov and Iryna Gurevych. 2018. E2E NLG Challenge: Neural Models vs. Templates. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 463–471, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. CIDEr: Consensus-based image description evaluation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June:4566–4575.

Bolin Wei, Shuai Lu, Lili Mou, Hao Zhou, Pascal Poupart, Ge Li, and Zhi Jin. 2019. Why Do Neural Dialog Systems Generate Short and Meaningless Replies? a Comparison between Dialog and Translation. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7290–7294. IEEE.

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

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in Data-to-Document Generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2018. Learning Neural Templates for Text Generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3174–3187, Stroudsburg, PA, USA. Association for Computational Linguistics.

Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning. *CoRR*, abs/1709.0.

A Replication Instructions

Dataset The E2E dataset contains a training set of 42,061 pairs of meaning representations and human authored utterances, 4,672 pairs in the development set, and 4,693 in the test set. Download the dataset from https://github.com/tuetschek/e2e-dataset.

We used the delexicalization script provided by the organizers in the TGen repository https://github.com/UFAL-DSG/tgen/tree/master/e2e-challenge. This module replaces the names of restaurants which appeared in the *Name* and *Near* attributes with a generic value, X-Name and X-Near.

Main experiment repository All the experiments are done with python modules and bash scripts. These are available in our *main repository* https://github.com/Henry-E/reliable_neural_nlg

Experiment steps

1. First the delexicalised data is converted into source and target files. It uses modified regular expressions from the e2e-cleaning repository. https://github.com/tuetschek/e2e-cleaning

2. Inside the *scripts/* folder of our main repository there are bash scripts for running the preprocessing required by OpenNMT and the actual training. We use our own fork of OpenNMT; the only changes made were to the beam search decoding code. https://github.com/Henry-E/surface_realization_opennmt-py

3. Using a trained model, a translate script generates text dev or test set. See *scripts/translate_surface_forms.sh* in our main repository.

4. Generated text is still in a raw format and requires relexicalisation and detokenization, see the python module *modules/relex_and_detok.py*.

Full hyperparameter details are available in the main repository. Here is a short synopsis of the model: A sequence-to-sequence model with copy attention, using the adam optimizer with learning rate 0.001, 2 layers, 300 dimension word vectors, 600 dimension LSTM cells, and shared embeddings between encoder and decoder. We train for 20 epochs of the data, this takes 15 minutes using two NVIDIA 1080 Ti gpu cards. We then choose the checkpoint with the highest validation set accuracy. We try to select a checkpoint before overfitting becomes noticeable, usually around the 15 epoch mark.

Evaluation

1. We calculate n-gram metric scores using the E2E-metrics module https://github.com/tuetschek/e2e-metrics

2. To calculate semantic accuracy we use a minimally modified version of the slot_error.py module from https://github.com/tuetschek/e2e-cleaning. We noticed it was incorrectly grouped together attributes in a small number of cases (we saw less than 5). This change improved Slug2Slug's results, as it now showed that it had fewer missing attributes.

System outputs from the E2E NLG Challenge participants can be found at https://github.com/tuetschek/e2e-eval

B Ordering and Relationship of attributes

Note that while we extract the surface forms and include them in the source sequence, they do not appear in the same order in the input as in the target sentence. This adds an extra requirement at test time to provide a reasonable order for the attribute-value pairs, and when an order which has not been seen commonly enough during training time is used during test time errors are likely to occur. Slug2Slug also noted this in their paper. In an experiment, they randomised the order of attribute-value pairs in the input sequence to augment the training data but found that this significantly decreased performance.

We have also omitted discussion of the more complex, but complete, notion of hierarchy of inputs and their relationship to each other, which can be used to give more control over how attributes in a sentence are expressed. This was touched upon in both the Surface Realization Shared Task (Mille et al., 2018) (hierarchical dependency relations link together tokens) and in the Constrained Decoding paper of Balakrishnan et al. (2019) (discourse relations link together attributes-value pairs).

C Frequency Graphs

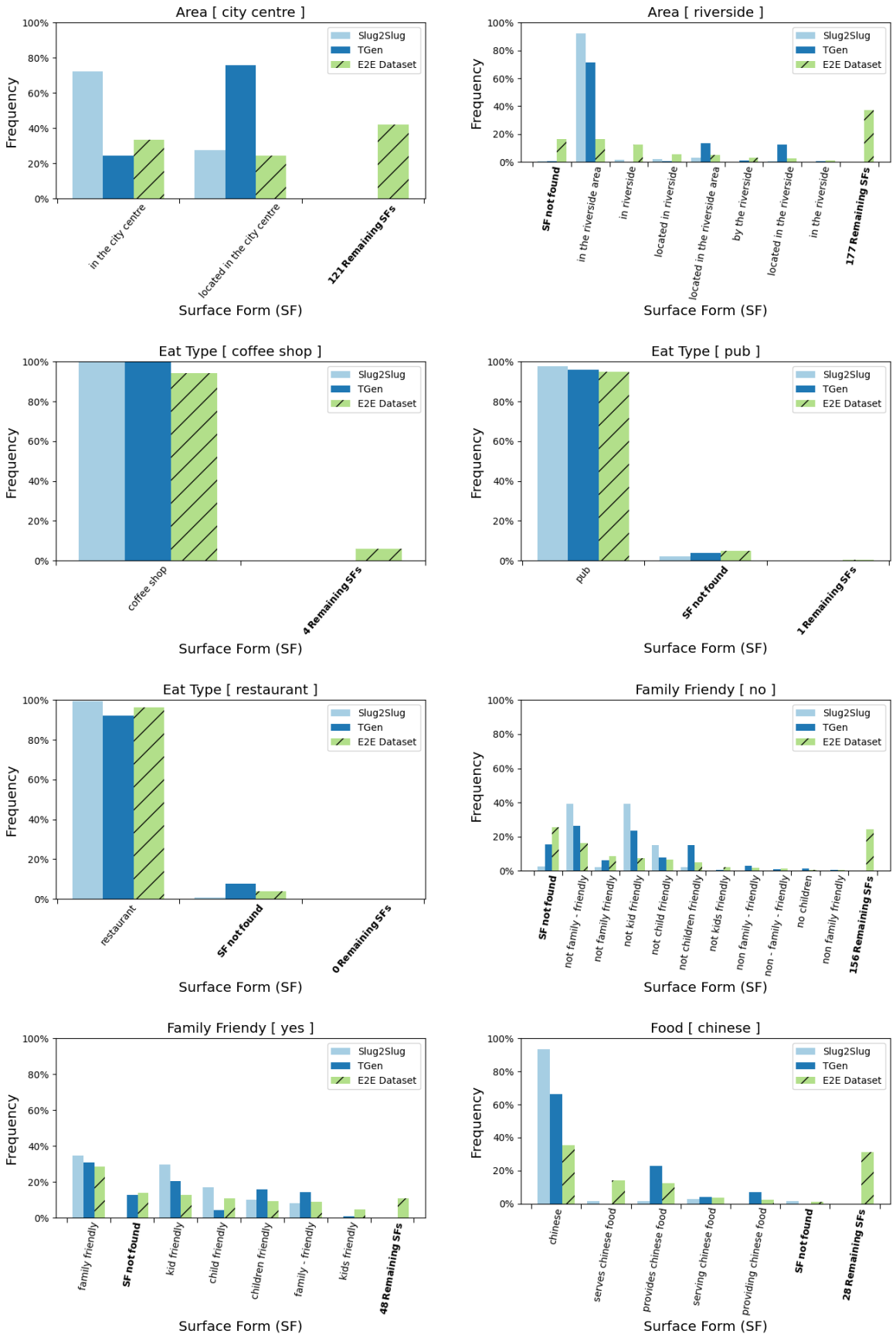

Figure 3: Frequency of surface forms used to express attribute-value pairs. **SF not found**: no surface form was found in the utterance, **X Remaining SFs**: surface forms other than those displayed in the x-axis labels

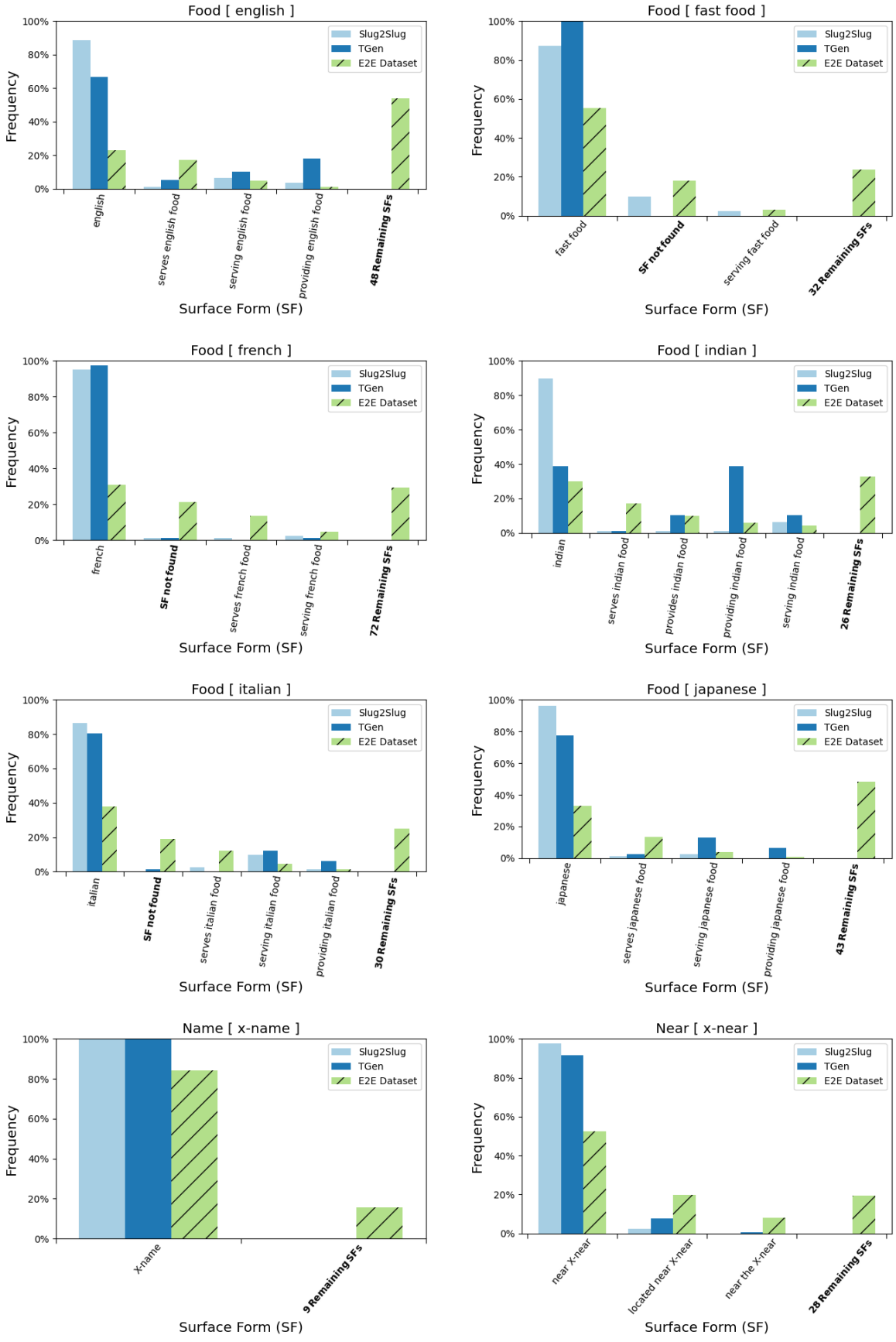

Figure 4: Frequency of surface forms used to express attribute-value pairs. **SF not found**: no surface form was found in the utterance, **X Remaining SFs**: surface forms other than those displayed in the x-axis labels

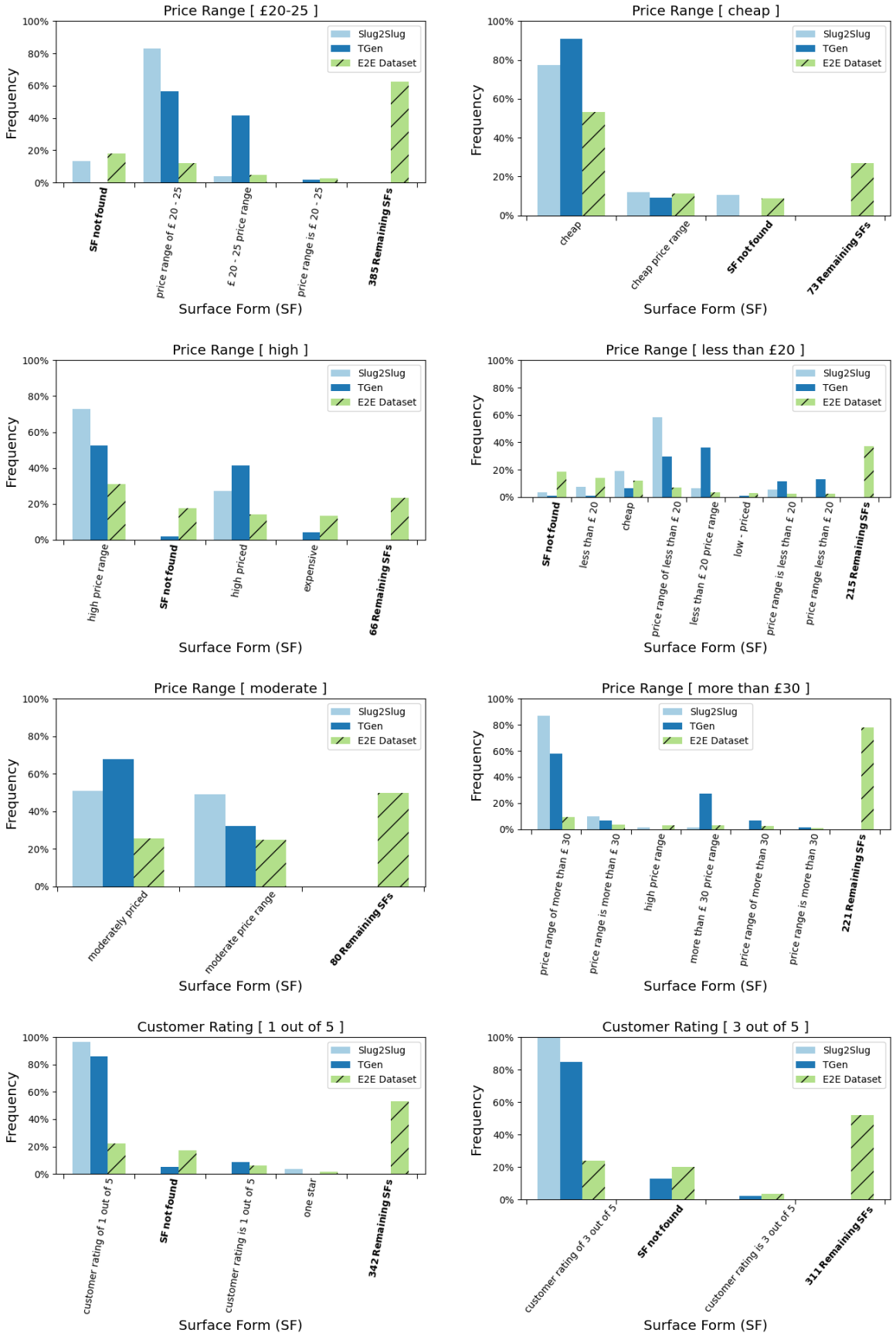

Figure 5: Frequency of surface forms used to express attribute-value pairs. **SF not found**: no surface form was found in the utterance, **X Remaining SFs**: surface forms other than those displayed in the x-axis labels

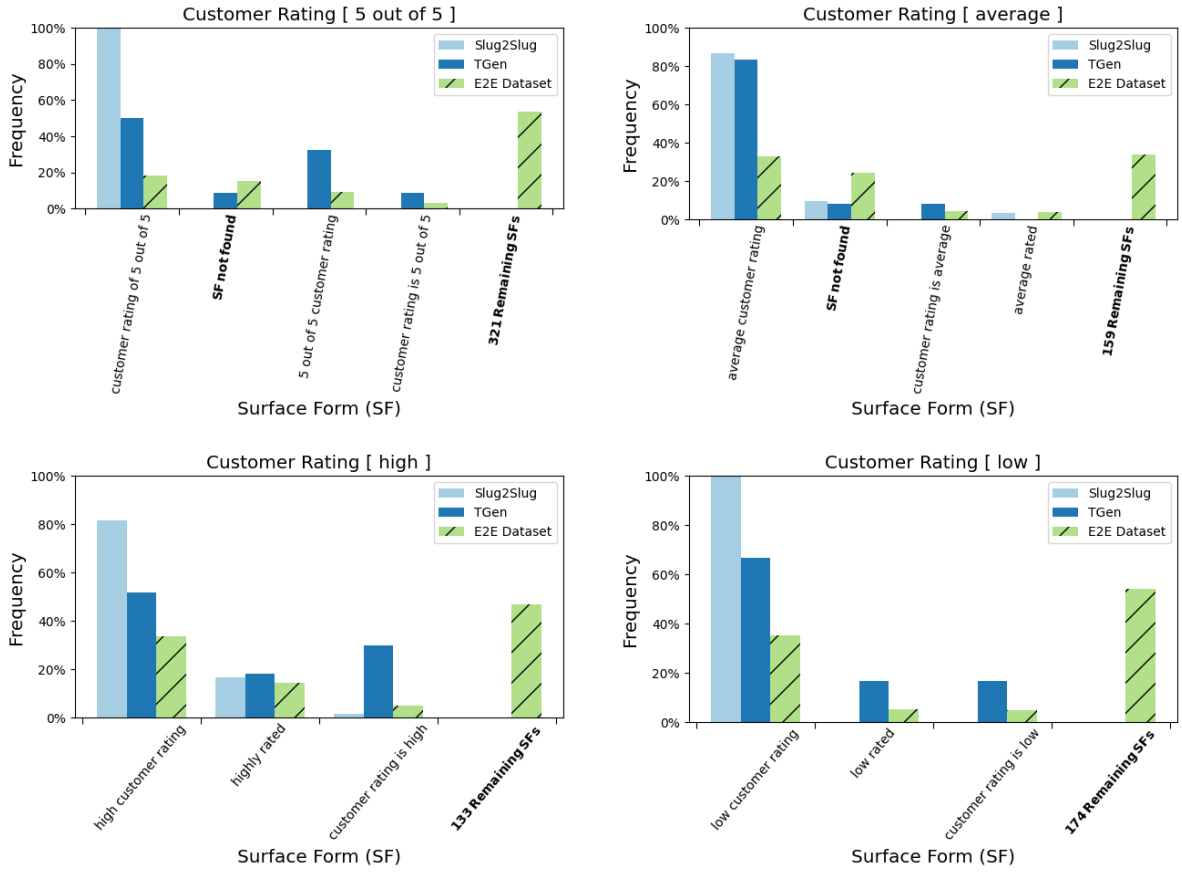

Figure 6: Frequency of surface forms used to express attribute-value pairs. **SF not found**: no surface form was found in the utterance, **X Remaining SFs**: surface forms other than those displayed in the x-axis labels