Pragmatically Informative Text Generation

Sheng Shen[†] **Daniel Fried**[†] **Jacob Andreas**[‡] **Dan Klein**[†]
[†]Computer Science Division, UC Berkeley
[‡]Computer Science and Artificial Intelligence Laboratory, MIT
{sheng.s,dfried,klein}@berkeley.edu, jda@mit.edu

Abstract

We improve the informativeness of models for conditional text generation using techniques from computational pragmatics. These techniques formulate language production as a game between speakers and listeners, in which a speaker should generate output text that a listener can use to correctly identify the original input that the text describes. While such approaches are widely used in cognitive science and grounded language learning, they have received less attention for more standard language generation tasks. We consider two pragmatic modeling methods for text generation: one where pragmatics is imposed by information preservation, and another where pragmatics is imposed by explicit modeling of distractors. We find that these methods improve the performance of strong existing systems for abstractive summarization and generation from structured meaning representations.

Input meaning representation (i)**:**
Name[Fitzbillies], EatType[coffee shop], Food[English], PriceRange[cheap], CustomerRating[5 out of 5], Area[Riverside], FamilyFriendly[yes]

Human-written
A cheap coffee shop in riverside with a 5 out of 5 customer rating is Fitzbillies. Fitzbillies is family friendly and serves English food.

Base sequence-to-sequence model (S_0)
Fitzbillies is a family friendly coffee shop located near the river.

Distractor-based pragmatic system (S_1^D)
Fitzbillies is a family friendly coffee shop that serves English food. It is located in riverside area. It has a customer rating of 5 out of 5 and is cheap.

Reconstructor-based pragmatic system (S_1^R)
Fitzbillies is a family friendly coffee shop that serves cheap English food in the riverside area. It has a customer rating of 5 out of 5.

Figure 1: Example outputs of our systems on the E2E generation task. While a base sequence-to-sequence model (S_0, Sec. 2) fails to describe all attributes in the input meaning representation, both of our pragmatic systems (S_1^R, Sec. 3.1 and S_1^D, Sec. 3.2) and the human-written reference do.

1 Introduction

Computational approaches to pragmatics cast language generation and interpretation as game-theoretic or Bayesian inference procedures (Golland et al., 2010; Frank and Goodman, 2012). While such approaches are capable of modeling a variety of pragmatic phenomena, their main application in natural language processing has been to improve the *informativeness* of generated text in grounded language learning problems (Monroe et al., 2018). In this paper, we show that pragmatic reasoning can be similarly used to improve performance in more traditional language generation tasks like generation from structured meaning representations (Figure 1) and summarization.

Our work builds on a line of learned Rational Speech Acts (RSA) models (Monroe and Potts, 2015; Andreas and Klein, 2016), in which generated strings are selected to optimize the behavior of an embedded listener model. The canonical presentation of the RSA framework (Frank and Goodman, 2012) is grounded in reference resolution: models of speakers attempt to describe referents in the presence of distractors, and models of listeners attempt to resolve descriptors to referents. Recent work has extended these models to more complex groundings, including images (Mao et al., 2015) and trajectories (Fried et al., 2018). The techniques used in these settings are similar, and the primary intuition of the RSA framework is preserved: from the speaker's perspective, a good description is one that picks out, as discriminatively as possible, the content the speaker intends for the listener to identify.

Outside of grounding, cognitive modeling (Frank et al., 2009), and targeted analysis of linguistic phenomena (Orita et al., 2015), rational speech acts models have seen limited application in the natural language processing literature. In this work we show that they can be extended

Proceedings of NAACL-HLT 2019, pages 4060–4067
Minneapolis, Minnesota, June 2 - June 7, 2019. ©2019 Association for Computational Linguistics

to a distinct class of language generation problems that use as referents structured descriptions of lingustic content, or other natural language texts. In accordance with the maxim of quantity (Grice, 1970) or the Q-principle (Horn, 1984), pragmatic approaches naturally correct underinformativeness problems observed in state-of-the-art language generation systems (S_0 in Figure 1).

We present experiments on two language generation tasks: generation from meaning representations (Novikova et al., 2017) and summarization. For each task, we evaluate two models of pragmatics: the reconstructor-based model of Fried et al. (2018) and the distractor-based model of Cohn-Gordon et al. (2018). Both models improve performance on both tasks, increasing ROUGE scores by 0.2–0.5 points on the CNN/Daily Mail abstractive summarization dataset and BLEU scores by 2 points on the End-to-End (E2E) generation dataset, obtaining new state-of-the-art results.

2 Tasks

We formulate a conditional generation task as taking an input i from a space of possible inputs $\mathcal{I}$ (e.g., input sentences for abstractive summarization; meaning representations for structured generation) and producing an output o as a sequence of tokens $(o_1, \ldots, o_T)$. We build our pragmatic approaches on top of learned *base speaker* models S_0, which produce a probability distribution $S_0(o \mid i)$ over output text for a given input. We focus on two conditional generation tasks where the information in the input context should largely be preserved in the output text, and apply the pragmatic procedures outlined in Sec. 3 to each task. For these S_0 models we use systems from past work that are strong, but may still be underinformative relative to human reference outputs (e.g., Figure 1).

Meaning Representations Our first task is generation from structured meaning representations (MRs) containing attribute-value pairs (Novikova et al., 2017). An example is shown in Figure 1, where systems must generate a description of the restaurant with the specified attributes. We apply pragmatics to encourage output strings from which the input MR can be identified. For our S_0 model, we use a publicly-released neural generation system (Puzikov and Gurevych, 2018) that achieves comparable performance to the best published results in Dušek et al. (2018).

Abstractive Summarization Our second task is multi-sentence document summarization. There is a vast amount of past work on summarization (Nenkova and McKeown, 2011); recent neural models have used large datasets (e.g., Hermann et al. (2015)) to train models in both the extractive (Cheng and Lapata, 2016; Nallapati et al., 2017) and abstractive (Rush et al., 2015; See et al., 2017) settings. Among these works, we build on the recent abstractive neural summarization system of Chen and Bansal (2018). First, this system uses a sentence-level extractive model RNN-EXT to identify a sequence of salient sentences $i^{(1)}, \ldots i^{(P)}$ in each source document. Second, the system uses an abstractive model ABS to rewrite each $i^{(p)}$ into an output $o^{(p)}$, which are then concatenated to produce the final summary. We rely on the fixed RNN-EXT model to extract sentences as inputs in our pragmatic procedure, using ABS as our S_0 model and applying pragmatics to the $i^{(p)} \rightarrow o^{(p)}$ abstractive step.

3 Pragmatic Models

To produce informative outputs, we consider pragmatic methods that extend the *base speaker* models, S_0, using *listener* models, L, which produce a distribution $L(i \mid o)$ over possible inputs given an output. Listener models are used to derive *pragmatic speakers*, $S_1(o \mid i)$, which produce output that has a high probability of making a listener model L identify the correct input. There are a large space of possible choices for designing L and deriving S_1; we follow two lines of past work which we categorize as *reconstructor-based* and *distractor-based*. We tailor each of these pragmatic methods to both our two tasks by developing reconstructor models and methods of choosing distractors.

3.1 Reconstructor-Based Pragmatics

Pragmatic approaches in this category (Dušek and Jurčíček, 2016; Fried et al., 2018) rely on a *reconstructor listener* model L^R defined independently of the speaker. This listener model produces a distribution $L^R(i \mid o)$ over all possible input contexts $i \in \mathcal{I}$, given an output description o. We use sequence-to-sequence or structured classification models for L^R (described below), and train these models on the same data used to supervise the S_0 models.

The listener model and the base speaker model together define a *pragmatic speaker*, with output score given by:

$$S_1^R(o \mid i) = L^R(i \mid o)^\lambda \cdot S_0(o \mid i)^{1-\lambda} \quad (1)$$

where λ is a *rationality parameter* that controls how much the model optimizes for discriminative outputs (see Monroe et al. (2017) and Fried et al. (2018) for a discussion). We select an output text sequence o for a given input i by choosing the highest scoring output under Eq. 1 from a set of candidates obtained by beam search in $S_0(\cdot \mid i)$.

Meaning Representations We construct L^R for the meaning representation generation task as a multi-task, multi-class classifier, defining a distribution over possible values for each attribute. Each MR attribute has its own prediction layer and attention-based aggregation layer, which conditions on a basic encoding of o shared across all attributes. See Appendix A.1 for architecture details. We then define $L^R(i \mid o)$ as the joint probability of predicting all input MR attributes in i from o.

Summarization To construct L^R for summarization, we train an ABS model (of the type we use for S_0, Chen and Bansal (2018)) but in reverse, i.e., taking as input a sentence in the summary and producing a sentence in the source document. We train L^R on the same heuristically-extracted and aligned source document sentences used to train S_0 (Chen and Bansal, 2018).

3.2 Distractor-Based Pragmatics

Pragmatic approaches in this category (Frank and Goodman, 2012; Andreas and Klein, 2016; Vedantam et al., 2017; Cohn-Gordon et al., 2018) derive pragmatic behavior by producing outputs that distinguish the input i from an alternate *distractor* input (or inputs). We construct a distractor $\widetilde{\imath}$ for a given input i in a task-dependent way.[1]

We follow the approach of Cohn-Gordon et al. (2018), outlined briefly here. The base speakers we build on produce outputs incrementally, where the probability of o_t, the word output at time t, is conditioned on the input and the previously generated words: $S_0(o_t \mid i, o_{<t})$. Since the output is generated incrementally and there is no separate listener model that needs to condition on entire output decisions, the distractor-based approach is able to make pragmatic decisions at each word rather than choosing between entire output candidates (as in the reconstructor approaches).

The listener L^D and pragmatic speaker S_1^D are derived from the base speaker S_0 and a belief distribution $p_t(\cdot)$ maintained at each timestep t over the possible inputs $\mathcal{I}^D$:

$$L^D(i \mid o_{<t}) \propto S_0(o_{<t} \mid i) \cdot p_{t-1}(i) \quad (2)$$

$$S_1^D(o_t \mid i, o_{<t}) \propto L^D(i \mid o_{<t})^\alpha \cdot S_0(o_t \mid i, o_{<t}) \quad (3)$$

$$p_t(i) \propto S_0(o_t \mid i, o_{<t}) \cdot L^D(i \mid o_{<t}) \quad (4)$$

where α is again a rationality parameter, and the initial belief distribution $p_0(\cdot)$ is uniform, i.e., $p_0(i) = p_0(\widetilde{\imath}) = 0.5$. Eqs. 2 and 4 are normalized over the true input i and distractor $\widetilde{\imath}$; Eq. 3 is normalized over the output vocabulary. We construct an output text sequence for the pragmatic speaker S_1^D incrementally using beam search to approximately maximize Eq. 3.

Meaning Representations A distractor MR is automatically constructed for each input to be the most distinctive possible against the input. We construct this distractor by masking each present input attribute and replacing the value of each non-present attribute with the value that is most frequent for that attribute in the training data. For example, for the input MR in Figure 1, the distractor is NEAR[BURGER KING].

Summarization For each extracted input sentence $i^{(p)}$, we use the previous extracted sentence $i^{(p-1)}$ from the same document as the distractor input $\widetilde{\imath}$ (for the first sentence we do not use a distractor). This is intended to encourage outputs $o^{(p)}$ to contain distinctive information against other summaries produced within the same document.

4 Experiments

For each of our two conditional generation tasks we evaluate on a standard benchmark dataset, following past work by using automatic evaluation against human-produced reference text. We choose hyperparameters for our models (beam size, and parameters α and λ) to maximize task metrics on each dataset's development set; see Appendix A.2 for the settings used.[2]

[1] In tasks such as contrastive captioning or referring expression generation, these distractors are given; for the conditional generation task, we will show that pragmatic behavior can be obtained by constructing or selecting a single distractor that contrasts with the input i.

[2] Our code is publicly available at `https://github.com/sIncerass/prag_generation`.

System	BLEU	NIST	METEOR	R-L	CIDEr
T-Gen	65.93	8.61	44.83	68.50	2.23
Best Prev.	66.19†	8.61†	**45.29**‡	**70.83**◇	2.27•
S_0	66.52	8.55	44.45	69.34	2.23
S_0 ×2	65.93	8.31	43.52	69.58	2.12
S_1^R	**68.60**	**8.73**	**45.25**	**70.82**	**2.37**
S_1^D	67.76	8.72	44.59	69.41	2.27

Table 1: Test results for the E2E generation task, in comparison to the T-Gen baseline (Dušek and Jurčíček, 2016) and the best results from the E2E challenge, reported by Dušek et al. (2018): †Juraska et al. (2018), ‡Puzikov and Gurevych (2018), ◇Zhang et al. (2018), and •Gong (2018). We bold our highest performing model on each metric, as well as previous work if it outperforms all of our models.

4.1 Meaning Representations

We evaluate on the E2E task of generation from meaning representations containing restaurant attributes (Novikova et al., 2017). We report the task's five automatic metrics: BLEU (Papineni et al., 2002), NIST (Doddington, 2002), METEOR (Lavie and Agarwal, 2007), ROUGE-L (Lin, 2004) and CIDEr (Vedantam et al., 2015).

Table 1 compares the performance of our base S_0 and pragmatic models to the baseline T-Gen system (Dušek and Jurčíček, 2016) and the *best previous* result from the 20 primary systems evaluated in the E2E challenge (Dušek et al., 2018). The systems obtaining these results encompass a range of approaches: a template system (Puzikov and Gurevych, 2018), a neural model (Zhang et al., 2018), models trained with reinforcement learning (Gong, 2018), and systems using ensembling and reranking (Juraska et al., 2018). To ensure that the benefit of the reconstructor-based pragmatic approach, which uses two models, is not due solely to a model combination effect, we also compare to an ensemble of two base models (S_0 ×2). This ensemble uses a weighted combination of scores of two independently-trained S_0 models, following Eq. 1 (with weights tuned on the development data).

Both of our pragmatic systems improve over the strong baseline S_0 system on all five metrics, with the largest improvements (2.1 BLEU, 0.2 NIST, 0.8 METEOR, 1.5 ROUGE-L, and 0.1 CIDEr) from the S_1^R model. This S_1^R model outperforms the previous best results obtained by any system in the E2E challenge on BLEU, NIST, and CIDEr, with comparable performance on METEOR and ROUGE-L.

System	R-1	R-2	R-L	METEOR
Extractive				
Lead-3	40.34	17.70	36.57	**22.21**
Inputs	38.93	18.23	35.90	**24.66**
Abstractive				
Best Previous	**41.69**†	**19.47**†	**39.08**‡	21.00◇
S_0	40.88	17.80	38.54	20.38
S_0 ×2	40.76	17.88	38.46	19.88
S_1^R	41.23	18.07	38.76	20.57
S_1^D	**41.39**	**18.30**	**38.78**	**21.70**

Table 2: Test results for the non-anonymized CNN/Daily Mail summarization task. We compare to extractive baselines, and the best previous abstractive results of †Celikyilmaz et al. (2018), ‡Paulus et al. (2018) and ◇Chen and Bansal (2018). We bold our highest performing model on each metric, as well as previous work if it outperforms all of our models.

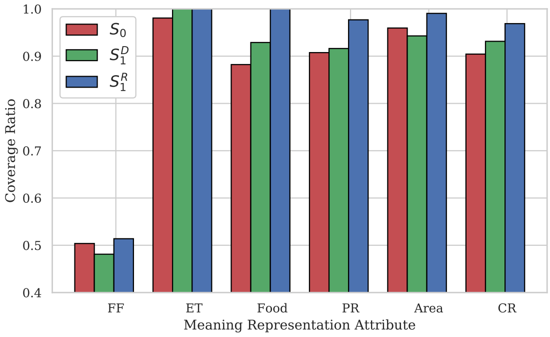
4.2 Abstractive Summarization

We evaluate on the CNN/Daily Mail summarization dataset (Hermann et al., 2015; Nallapati et al., 2016), using See et al.'s (2017) non-anonymized preprocessing. As in previous work (Chen and Bansal, 2018), we evaluate using ROUGE and METEOR.

Table 2 compares our pragmatic systems to the base S_0 model (with scores taken from Chen and Bansal (2018); we obtained comparable performance in our reproduction[3]), an ensemble of two of these base models, and the *best previous* abstractive summarization result for each metric on this dataset (Celikyilmaz et al., 2018; Paulus et al., 2018; Chen and Bansal, 2018). We also report two extractive baselines: *Lead-3*, which uses the first three sentences of the document as the summary (See et al., 2017), and *Inputs*, the concatenation of the extracted sentences used as inputs to our models (i.e., $i^{(1)}, \ldots, i^{(P)}$).

The pragmatic methods obtain improvements of 0.2–0.5 in ROUGE scores and 0.2–1.8 METEOR over the base S_0 model, with the distractor-based approach S_1^D outperforming the reconstructor-based approach S_1^R. S_1^D is strong across all metrics, obtaining results competitive to the best previous abstractive systems.

[3] We use retrained versions of Chen and Bansal (2018)'s sentence extractor and abstractive S_0 models in all our experiments, as well as their n-gram reranking-based inference procedure, replacing scores from the base model S_0 with scores from S_1^R or S_1^D in the respective pragmatic procedures.



(a) Coverage ratios by attribute type for the base model S_0 and pragmatic models S_1^R and S_1^D. The pragmatic models typically improve coverage ratios across attribute types when compared to the base model.

		Coverage Ratio for Attribute					
		FF	ET	Food	PR	Area	CR
	S_0	0.50	0.98	0.88	0.91	0.96	0.90
Distractor Attr.	S_1^D-FF	0.57	1.00	0.92	0.90	0.95	0.95
	S_1^D-ET	0.47	1.00	0.96	0.92	0.96	0.95
	S_1^D-Food	0.45	1.00	1.00	0.93	0.95	0.94
	S_1^D-PR	0.51	1.00	0.90	0.98	0.93	0.92
	S_1^D-Area	0.47	1.00	0.93	0.91	0.98	0.93
	S_1^D-CR	0.45	1.00	0.91	0.90	0.91	0.95

(b) Coverage ratios by attribute type (columns) for the base model S_0, and for the pragmatic system S_1^D when constructing the distractor by masking the specified attribute (rows). Cell colors are the degree the coverage ratio increases (green) or decreases (red) relative to S_0.

Figure 2: Coverage ratios for the E2E task by attribute type, estimating how frequently the values for each attribute from the input meaning representations are mentioned in the output text.

5 Analysis

The base speaker S_0 model is often underinformative, e.g., for the E2E task failing to mention certain attributes of a MR, even though almost all the training examples incorporate all of them. To better understand the performance improvements from the pragmatic models for E2E, we compute a *coverage ratio* as a proxy measure of how well content in the input is preserved in the generated outputs. The coverage ratio for each attribute is the fraction of times there is an exact match between the text in the generated output and the attribute's value in the source MR (for instances where the attribute is specified).[4]

Figure 2(a) shows coverage ratio by attribute category for all models. The S_1^R model increases the coverage ratio when compared to S_0 across all attributes, showing that using the reconstruction model score to select outputs does lead to an increase in mentions for each attribute. Coverage ratios increase for S_1^D as well in four out of six categories, but the increase is typically less than that produced by S_1^R.

While S_1^D optimizes less explicitly for attribute mentions than S_1^R, it still provides a potential method to control generated outputs by choosing alternate distractors. Figure 2(b) shows coverage ratios for S_1^D when masking only a single attribute in the distractor. The highest coverage ratio for each attribute is usually obtained when masking that attribute in the distractor MR (entries on the main diagonal, underlined), in particular for FAMILYFRIENDLY (FF), FOOD, PRICERANGE (PR), and AREA. However, masking a single attribute sometimes results in decreasing the coverage ratio, and we also observe substantial increases from masking other attributes: e.g., masking either FAMILYFRIENDLY or CUSTOMERRATING (CR) produces an equal increase in coverage ratio for the CUSTOMERRATING attribute. This may reflect underlying correlations in the training data, as these two attributes have a small number of possible values (3 and 7, respectively).

6 Conclusion

Our results show that S_0 models from previous work, while strong, still imperfectly capture the behavior that people exhibit when generating text; and an explicit pragmatic modeling procedure can improve results. Both pragmatic methods evaluated in this paper encourage prediction of outputs that can be used to identify their inputs, either by reconstructing inputs in their entirety or distinguishing true inputs from distractors, so it is perhaps unsurprising that both methods produce similar improvements in performance. Future work might allow finer-grained modeling of the tradeoff between *under*- and *over*-informativity within the sequence generation pipeline (e.g., with a learned communication cost model) or explore applications of pragmatics for content selection earlier in the generation pipeline.

Acknowledgments

Thanks to Reuben Cohn-Gordon for many helpful discussions and suggestions. This work was supported by DARPA through the XAI program. DF is supported by a Tencent AI Lab Fellowship.

[4] Note that this measure roughly provides a lower bound on the model's actual informativeness for each attribute, since the measure does not assign credit for paraphrases.

References

Jacob Andreas and Dan Klein. 2016. Reasoning about pragmatics with neural listeners and speakers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. In *Proceedings of the Annual Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 1662–1675, New Orleans, Louisiana. Association for Computational Linguistics.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 484–494, Berlin, Germany. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Reuben Cohn-Gordon, Noah Goodman, and Chris Potts. 2018. Pragmatically informative image captioning with character-level reference. In *Proceedings of the Annual Meeting of the North American Chapter of the Association for Computational Linguistics*.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram cooccurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pages 138–145. Morgan Kaufmann Publishers Inc.

Ondrej Dušek and Filip Jurčíček. 2016. Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. In *ACL*.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2018. Findings of the E2E NLG challenge. In *Proceedings of the International Conference on Natural Language Generation*.

Michael C Frank and Noah D Goodman. 2012. Predicting pragmatic reasoning in language games. *Science*, 336(6084):998–998.

Michael C Frank, Noah D Goodman, Peter Lai, and Joshua B Tenenbaum. 2009. Informative communication in word production and word learning. In *Proceedings of the 31st annual conference of the cognitive science society*, pages 1228–1233.

Daniel Fried, Jacob Andreas, and Dan Klein. 2018. Unified pragmatic models for generating and following instructions. In *Proceedings of the Annual Meeting of the North American Chapter of the Association for Computational Linguistics*.

Dave Golland, Percy Liang, and Dan Klein. 2010. A game-theoretic approach to generating spatial descriptions. In *Proceedings of the 2010 conference on Empirical Methods in Natural Language Processing*, pages 410–419. Association for Computational Linguistics.

Heng Gong. 2018. Technical report for E2E NLG challenge. In *E2E NLG Challenge System Descriptions*.

Herbert P Grice. 1970. Logic and conversation.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*.

Laurence Horn. 1984. Toward a new taxonomy for pragmatic inference: Q-based and r-based implicature. *Meaning, form, and use in context: Linguistic applications*, 11:42.

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 Annual Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 152–162, New Orleans, Louisiana. 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*, pages 228–231, Prague, Czech Republic. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan Yuille, and Kevin Murphy. 2015. Generation and comprehension of unambiguous object descriptions. *arXiv preprint arXiv:1511.02283*.

Will Monroe, Robert XD Hawkins, Noah D Goodman, and Christopher Potts. 2017. Colors in context: A pragmatic neural model for grounded language understanding. *Transactions of the Association for Computational Linguistics*.

Will Monroe, Jennifer Hu, Andrew Jong, and Christopher Potts. 2018. Generating bilingual pragmatic color references. In *Proceedings of the Annual Meeting of the North American Chapter of the Association for Computational Linguistics*.

Will Monroe and Christopher Potts. 2015. Learning in the Rational Speech Acts model. In *Proceedings of 20th Amsterdam Colloquium*, Amsterdam. ILLC.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the Meeting of the Association for the Advancement of Artificial Intelligence*.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Ani Nenkova and Kathleen McKeown. 2011. Automatic summarization. *Foundations and Trends in Information Retrieval*, 5(2–3):103–233.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the Annual SIGdial Meeting on Discourse and Dialogue*.

Naho Orita, Eliana Vornov, Naomi Feldman, and Hal Daumé III. 2015. Why discourse affects speakers' choice of referring expressions. 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)*, volume 1, pages 1639–1649.

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

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *Proceedings of the International Conference on Learning Representations*, volume abs/1705.04304.

Yevgeniy Puzikov and Iryna Gurevych. 2018. E2E NLG challenge: Neural models vs. templates. In *Proceedings of the International Conference on Natural Language Generation*.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Ramakrishna Vedantam, Samy Bengio, Kevin Murphy, Devi Parikh, and Gal Chechik. 2017. Context-aware captions from context-agnostic supervision.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. CIDEr: Consensus-based image description evaluation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*.

Biao Zhang, Jing Yang, Qian Lin, and Jinsong Su. 2018. Attention regularized sequence-to-sequence learning for E2E NLG challenge. In *E2E NLG Challenge System Descriptions*.

A Supplemental Material

A.1 Reconstructor Model Details

For the reconstructor-based speaker in the E2E task, we first follow the same data preprocessing steps as Puzikov and Gurevych (2018), which includes a delexicalization module that deals with sparsely occurring MR attributes (NAME, NEAR) by mapping such values to placeholder tokens.

MRs have only a few possible values for most attributes: six out of eight attributes have fewer than seven unique values, and the remaining two attributes (NAME, NEAR) are handled by our S_0 and S_1^D using delexicalized placeholders, following Puzikov and Gurevych (2018). In this way, the reconstructor only needs to predict the presence of these two attributes with a boolean variable, and other attributes with the corresponding categorical variable. We use a one layer bi-directional GRU (Cho et al., 2014) for the shared sentence encoder. We concatenate the latent vectors from both directions to construct a bi-directional encoded vector h_i for every single word vector d_i as:

$$\overrightarrow{h_i} = \overrightarrow{GRU}(d_i, h_{i-1}), \overleftarrow{h_i} = \overleftarrow{GRU}(d_i, h_{i+1})$$

$$h_i = [\overrightarrow{h_i}, \overleftarrow{h_i}], i \in [1, L]$$

Since not all words contribute equally to predicting each MR attribute, we thus use an attention mechanism (Bahdanau et al., 2014) to determine the importance of every single word. The aggregated sentence vector for task k is calculated by

$$a_i^{(k)} = \frac{exp(W_a^{(k)} h_i)}{\sum_{j=1}^{L} exp(W_a^{(k)} h_j)}, v^{(k)} = \sum_{i=1}^{L} a_i^{(k)} h_i,$$

The task-specific sentence representation is then used as input to k layers with softmax outputs, returning a probability vector $Y^{(k)}$ for each of the k MR attributes.

A.2 Hyperparameters

For structured generation, we use beam size 10, $\lambda = 0.4$, and $\alpha = 0.2$, tuned to maximize the normalized average of all five metrics on the development set.

For abstractive summarization, we use beam size 20, $\lambda = 0.9$, and $\alpha = 1.0$, tuned to maximize ROUGE-L on the development set.