Neural Methodius Revisited: Do Discourse Relations Help with Pre-Trained Models Too?

Aleksandre Maskharashvili, Xintong Li, Symon Jory Stevens-Guille and Michael White

Department of Linguistics
The Ohio State University
maskharashvili.1@osu.edu znculee@gmail.com
stevensguille.1@osu.edu mwhite@ling.osu.edu

Abstract

Recent developments in natural language generation (NLG) have bolstered arguments in favor of re-introducing explicit coding of discourse relations in the input to neural models. In the Methodius corpus, a meaning representation (MR) is hierarchically structured and includes discourse relations. Meanwhile pre-trained language models have been shown to implicitly encode rich linguistic knowledge which provides an excellent resource for NLG. By virtue of synthesizing these lines of research, we conduct extensive experiments on the benefits of using pre-trained models and discourse relation information in MRs, focusing on the improvement of discourse coherence and correctness. We redesign the Methodius corpus; we also construct another Methodius corpus in which MRs are not hierarchically structured but flat. We report experiments on different versions of the corpora, which probe when, where, and how pre-trained models benefit from MRs with discourse relation information in them. We conclude that discourse relations significantly improve NLG when data is limited.

1 Introduction

The success of neural methods in numerous subfields of NLP lead to recent development of neural 'end-to-end' (e2e) architectures in natural language generation (NLG) (Dušek et al., 2020), where a direct mapping from meaning representations (MRs) to text is learned. While recent neural approaches mostly map flat inputs to texts without representing discourse level information explicitly within MRs, Balakrishnan et al. (2019) argues that discourse relations should be reintroduced into neural generation, echoing what has been long argued in more traditional approaches to natural language processing where discourse relations play one of the central roles in natural language text understanding and generation (Mann and Thompson, 1988; Reiter and Dale, 2000; Lascarides and Asher, 2007).

To study whether discourse relations are beneficial for neural NLG, Stevens-Guille et al. (2020) proposed the Methodius corpus, which was developed as an experiment in recreating the classic rule-based NLG system Methodius (Isard, 2016) using a neural generator. In their corpus, the meaning representation (MR) of a text is a tree that encodes the overall discourse structure of the texts plus facts related by discourse relations therein. They were concerned with whether explicit encoding of discourse relations improves the quality of generated texts by LSTM recurrent neural networks (Hochreiter and Schmidhuber, 1997). However, they left open the question whether discourse relations are helpful for pre-trained transformer-based (Vaswani et al., 2017) language models (Lewis et al., 2020; Raffel et al., 2019), which have recently shown remarkable performance on NLG tasks. In this work, we address that question using the T5-Large implementation of Wolf et al. (2019).

A particularly attractive quality of pre-trained models is their ability to generalize from limited data. For example, Peng et al. (2020) proposed to fine-tune a model pre-trained on a large NLG corpus using a small amount of labeled data from a specific domain to adapt the model to generate texts in that domain. In a similar vein, when the labeled data is limited, Arun et al. (2020) suggest to use a large pre-trained model with self-training and knowledge distillation to smaller, faster models. Kale and Rastogi (2020) argue that pre-trained language models make it possible to transform a sequence of semantically correct, but (possibly) ungrammatical template-based texts into a natural sounding, felicitous text of English. They find that template-based textual input is beneficial to use with pre-trained language models when the model needs to generalize from relatively few examples.

Proceedings of the 14th International Conference on Natural Language Generation (INLG), pages 12–23,
Aberdeen, Scotland, UK, 20-24 September 2021.

Given these considerations, we cannot answer the question whether it is helpful to include discourse relations in the input to a pre-trained model for NLG without considering the form of the input, the size of the training data, and the extent to which the test data goes beyond what has been seen in training. As such, we conduct experiments using several versions of the Methodius corpus, where these versions possess one or more of the following properties: (a) discourse relations included in the MR; (b) discourse relations excluded from the MR; (c) tree-structured MR (a hierarchically structured representation of the meaning); (d) flat, textual MR (i.e., non hierarchically structured). We are furthermore concerned with how the linguistic knowledge encoded in pre-trained language models interacts with the different versions of the corpus. We want to be able to scrutinize the structure of the outputs, i.e., texts, too since our intention is to check the models' capabilities in realizing particular phenomena. For these purposes, we conduct experiments using the following setup: (1) Use various portions of the labeled data. (2) Train zero-shot models (with respect to certain discourse-related phenomena) together with various few-shot models (with respect to the same phenomena). (3) Test various aspects of generated texts, both with respect to discourse structure congruence and correctness (factual information).

2 Re-lexicalized & Flat Versions of Methodius

The Methodius system (Isard, 2016) uses discrete rules to generate texts containing predefined subtexts, such as descriptions of exhibits and historical facts about certain periods. To avoid data sparsity and long sequences, Stevens-Guille et al. (2020) delexicalize texts as they substitute certain parts of text by tokens, which they dub special terminals. We want to take advantage of pre-trained language models, which are not exposed to these tokens. Tokens should be substituted by text wherever possible to ensure the input is consistent with the texts the pre-trained models were trained on. However, using some predefined morpho-syntactic constructions and lexical items makes it more manageable to check whether a model performs well with respect to automatic checks. Moreover, if the model experiences problems on such data, it suggests the model would have problems with even less homogeneous data.

Instead of training the models directly on the Methodius corpus or texts harvested through crowdsourcing, we modify the Methodius corpus (i.e., MRs paired with texts) by substituting custom homogeneous texts for the Methodius corpus's special terminals. We substitute some predetermined names for named entities in the Methodius corpus to further homogenize the inputs. This procedure deterministically rewrites the texts in the corpus of Stevens-Guille et al. (2020) into pure English texts and thus maintains the homogeneity of the Methodius corpus. The corresponding MRs are also rewritten into their lexicalized versions.[1] Moreover, we transform Rhetorical structure theory (Mann and Thompson, 1988) style hierarchically structured meaning representations of Methodius texts into a flat, textual input by translating every fact and every discourse relation into a sequence of sentences. Figure 1 shows an MR from the Methodius corpus, the corresponding text from the Methodius corpus, and the new MR that we have substituted for the Methodius corpus MR.

3 Models: RSTSTRUCT, FACTSTRUCT, RSTT2T, and FACTT2T

We fine-tune T5-large (Raffel et al., 2019) on the following types of labeled data:

- Input MRs from the Methodius corpus modified by the procedure described in the foregoing (see Figure 1b). It contains discourse relations. We dub the result RSTSTRUCT.
- Input MRs obtained by erasing discourse information from the inputs of RSTSTRUCT. This amounts to deleting discourse relation markers (SIMILARITY and CONTRAST) in the inputs of RSTSTRUCT. We dub the result FACTSTRUCT.
- Input MRs obtained by transforming the MRs of RSTSTRUCT into flat, purely textual representations (see Figure 1c).[2] We dub the result RSTT2T.
- Input MRs obtained by removing discourse information from RSTT2T MRs. This amounts to deleting discourse relation markers ('however' and 'likewise') in the inputs of RSTT2T. We dub the result FACTT2T.

[1] The code can be found at https://github.com/aleksadre/methodiusNeuralINLG2021.

[2] We have defined a set of rules that transform hierarchically structured MRs into texts.

Figure 1: A Methodius MR, the re-lexicalized MR, and its flat, textual version, together with the surface realization

(a) Delexicalized meaning representation from Methodius corpus

```
[__content_plan
 [__rst_elaboration
  [__fact_type [__arg1 entity0 ] [__arg2 statue ] ]
  [__rst_joint [__fact_made_of [__arg1 entity0 ] [__arg2 material_0 ] ]
               [__fact_exhibit_portrays [__arg1 entity0 ] [__arg2 god_0 ] ] ] ]
 [__rst_contrast
  [__fact_creation_period compare_additive [__arg1 entity1 ]
                                           [__arg2 historical_period_0 ] ]
  [__fact_creation_period [__arg1 entity0 ] [__arg2 historical_period_1 ] ] ]
 [__optional_type [__arg1 entity1 ] [__arg2 vessel ] ] ]
```

(b) Lexicalized meaning representation of the foregoing (we treat tokens of the form '$[\_\_xyz$' and '$]$' as indivisible tokens in our experiments)

```
[__content_plan
 [__rst_elaboration
  [__fact_type [__arg1 entity0 ] [__arg2 statue ] ]
  [__rst_joint [__fact_made_of [__arg1 entity0 ] [__arg2 bronze ] ]
               [__fact_exhibit_portrays [__arg1 entity0 ] [__arg2 apollo ] ] ] ]
 [__rst_contrast
  [__fact_creation_period compare_additive [__arg1 entity1 ]
                                           [__arg2 classical period ] ]
  [__fact_creation_period [__arg1 entity0 ] [__arg2 hellenistic period ] ] ]
 [__optional_type [__arg1 entity1 ] [__arg2 vessel ] ] ]
```

(c) Flat, textual meaning representation

```
this statue is a statue. this statue is made of bronze. this statue portrays apollo.
the previously seen vessel was created in the classical period.
however this statue was created in the hellenistic period.
```

Text: This is a statue; it is made of bronze and it portrays Apollo. Unlike the vessel you recently saw, which was created during the classical period, this statue was created during the hellenistic period.

Figure 2: Instances of constructions starting with SIMILARITY and CONTRAST, which are not included in zero-shot data

(a) The Like Construction and the corresponding text

```
[__content_plan
    [__rst_similarity
        [__fact_original_location [__arg1 entity1 ] [__arg2 attica ] ]
        [__fact_original_location [__arg1 entity0 ] [__arg2 attica ] ] ]
    [__fact_exhibit_story [__arg1 entity0 ] [__arg2 it was part of a collection dedicated to athena ] ]
    [__fact_current_location [__arg1 entity0 ] [__arg2 the national archaeological museum ] ]
    [__fact_exhibit_depicts [__arg1 entity0 ] [__arg2 the goddess athena ] ]
    [__optional_type [__arg1 entity0 ] [__arg2 lekythos ] ]
    [__optional_type [__arg1 entity1 ] [__arg2 kylix ] ] ]
```

Text: Like the kylix you recently saw, this lekythos originates from Attica. It was part of a collection dedicated to Athena. This lekythos is located in The National Archaeological Museum. It depicts the goddess Athena.

(b) The Unlike Construction and the corresponding text

```
[__content_plan
    [__rst_contrast [__fact_original_location [__arg1 entityplural ] [__arg2 attica ] ]
    [__fact_original_location [__arg1 entity0 ] [__arg2 macedonia ] ] ]
[__fact_exhibit_story [__arg1 entity0 ] [__arg2 it was part of a collection dedicated to athena ] ]
    [__fact_current_location [__arg1 entity0 ] [__arg2 the national archaeological museum ] ]
    [__fact_exhibit_depicts [__arg1 entity0 ] [__arg2 the goddess athena ] ]
    [__optional_type [__arg1 entityplural ] [__arg2 vessel ] ]
    [__optional_type [__arg1 entity0 ] [__arg2 tetradrachm ] ] ]
```

Text: Unlike the vessels you recently saw, which were originally from Attica, this tetradrachm originates from Macedonia. It was part of a collection dedicated to Athena. Now this tetradrachm is exhibited in The National Archaeological Museum. It shows the goddess Athena.

We refer to models by the name of the data type they are fine-tuned on.

Name	Size 100%	Tok. Av.	SIM.	CONTRA.
Training	4222	180	2892	777
Validation	417	181	290	76
Challenge Test	237	96	80	80
Standard Test	799	134	495	166

Table 1: Size of training, validation and test sets; average tokens per pair (MR,text); numbers of SIMILARITY and CONTRAST relations in data sets.

In addition to using the whole dataset for training, we conduct experiments on (randomly selected) 1%, 3%, 5%, 10%, 20%, and 50% portions of the data. With 100% percent data, we train each model three times, while for the sub portions of the data set, we train the models five times each (each time we select random dataset of that portion). This lets us get an idea of the variance between different runs of the same model.

We distinguish three further subtypes of data, calling them zero-, few- and ten-shot data (which we also denote by prefixes Z-, F-, and D-, respectively). In zero-shot data, none of the MRs beginning with SIMILARITY or CONTRAST, the surface realization of which would start with 'Like' or 'Unlike,' are included in the training data. These constructions are exemplified in Figure 2. When constructing the few-shot data, the foregoing restriction on the form of the MRs is removed. But in each portion of the few-shot training data, we include only three examples of each construction. When constructing the ten-shot data, ten instances of each of the constructions that were introduced in the few-shot data are included. Tying the number of these constructions to the size of the dataset lets us more effectively compare a model behavior with and without these constructions.

4 Evaluation Methods

We adopt the double test set style from Stevens-Guille et al. (2020). One test set is called Standard and the other is called Challenge (see Table 1). There are several differences between them. The Standard test set examples are independently selected, while the Challenge test set examples are not. In the Challenge test, around 12% of the test items have structure not observed in the training set for zero-shot models.

4.1 Types of Errors

Discourse Relation Errors We are interested in observing the performance of the models with respect to generating coherent discourse relations. While there are several discourse relations in the Methodius corpus, we focus on CONTRAST and SIMILARITY for several reasons. First, they are interesting in terms of their meaning—they require identifying whether properties or entities are co-extensive or distinct, but can be inferred from the facts alone. Second, there is a consistent method of expressing them in the Methodius corpus outputs: SIMILARITY is realized by *like* and CONTRAST is realized by *unlike*.

Repetitions, Hallucinations, and Omissions (RHOs) Given the way the revised Methodius corpus is constructed, its texts follow certain predetermined lexical and morpho-syntactic patterns. We use this property of the texts to measure the performance of models with respect to the following errors: hallucination of content; omission of content; and repetitions of content. To be more precise, for every test item we compare the model output and the reference text by determining their difference with respect to the *special terminals*, i.e., the content that is obtained by relexicalizing the delexicalized content).

Lexical Hallucinations Since Methodius is designed purposely to be homogeneous, it is useful to measure how many novel strings pre-trained models come up with when fine-tuned on data that does not contain these strings. For that, we count per test set the lexical hallucinations, i.e., items produced by the model which are not observed in the corpus.

Mistaken Role Identity (mistID) We sometimes observe a mismatch between the exhibit type in the input and its realization in the output. For instance, in Example (1), 'imperial portrait' and 'vessel' are swapped, i.e, their roles are misidentified. We consider this kind of error distinct from the previous error types and refer to it by mistID.

(1) Ref: This is a vessel; it was created between 500 and 480 B.C. Unlike the imperial portraits you recently saw, which were originally from Attica, this vessel was originally from Acropolis.
Gen: This imperial portrait was created between 500 and 480 B.C. Unlike the vessels you recently saw, which originate from Attica, this imperial portrait was originally from Acropolis.

4.2 Statistical Significance: Stratified Approximate Randomization

To compare various models, we use stratified approximate randomization (AR; Noreen 1989), which is a powerful and generic method of establishing significant differences between models. One advantage of AR over more traditional paired tests for NLP tasks is that it does not require independence of samples, which is usually violated when we consider various runs of the same model on the same test set (as the same test item gets tested several times by the same model) (Clark et al., 2011). In the present work, we rely on stratified AR to identify whether differences between the performance of various models over several runs is significant. (The description of the stratified AR algorithm is provided in Section A.1 of Appendix A.)

5 Results

Below, we report results on the data portions 1%, 3%, 5%, 10%, 20%, and 50% of the few-shot models (results on the corresponding zero-shot data models are provided in Appendix A.4). For 100% data usage, we report results of zero-, few-, and ten-shot models.

5.1 Data portions: 1%, 3%, and 5%

Discourse Relations: As Figure 3 indicates, there are fewer errors in discourse relation realization for T2T (RSTT2T and FACTT2T) models compared to structure models (RSTSTRUCT and FACTSTRUCT).

RHOs: Figure 4 shows the number of RHO mistakes the models make. T2T models make less RHO mistakes compared to structure models. Also, each of the models produces a large number of lexical hallucinations, but T2T are less prone to lexical hallucinations compared to structured models as RSTSTRUCT and FACTSTRUCT each produce on average 100 lexical hallucinations at 1% and 50 lexical hallucinations at 3% and 5% data portion, whereas T2T make only third of those lexical hallucinations on each of the data portion (for full details on various runs see Figure 9 in Appendix A). We note that at the 1% portion of the data, the quality of generated texts is unsatisfactory, even by T2T models. This can be seen by automatic metrics, as well as by eyeballing the generated texts. On 3% and 5%, the quality gets slightly better for structured models and we see more rapid improvements for T2T models.

Summary: RST vs. FACT The question whether models with discourse relations (RSTSTRUCT and RSTT2T) perform better than ones without discourse relations (FACTSTRUCT and FACTT2T respectively) can be answered positively. As for T2T models, we declare with high confidence that RSTT2T outperforms FACTT2T in every collected statistics. We are not however able to say that for structured models, even though in more than half of the comparisons RSTSTRUCT is at least as good as FACTSTRUCT.

5.2 Data portions: 10%, 20%, and 50%

By using data portions 10%, 20%, and 50%, we see many improvements in quality of texts compared to 1%, 3%, and 5%. Also in this case (i.e. on the data 10%, 20%, and 50%), T2T models show better performance compared to structure models.

Discourse Relations: Figure 5 illustrates that RSTT2T does better or at least as good as FACTT2T. The same can be said about RSTSTRUCT and FACTSTRUCT, with the only exception of the case of the 10% data on the Challenge set as FACTSTRUCT shows better results than RSTSTRUCT.

RHOs: In terms of RHO errors, RSTT2T together with FACTT2T are winners on either test sets, as it is indicated by the results on Figure 6. In addition, by measuring lexical hallucinations, we conclude that the both RSTT2T and FACTT2T are the least hallucinating models (the detailed statistics is given on Figure 9 in Appendix A).

Summary: RST vs. FACT Again, RSTT2T comes out as the winner among all models (vs. RSTSTRUCT, FACTSTRUCT, and FACTT2T) by all the evaluation metrics involved. It must be noted though that as we reach 50%, we do not see significant differences between RSTT2T and FACTT2T. Also, RSTSTRUCT does better or at least as good as FACTSTRUCT, except for one case.

5.3 100%: Zero, Few, and Ten shot

In 100% data, we see less difference among performance of models as structured models show visible improvement, catching up with T2T models. Below, we compare models trained on zero-, few- and ten-shot data.

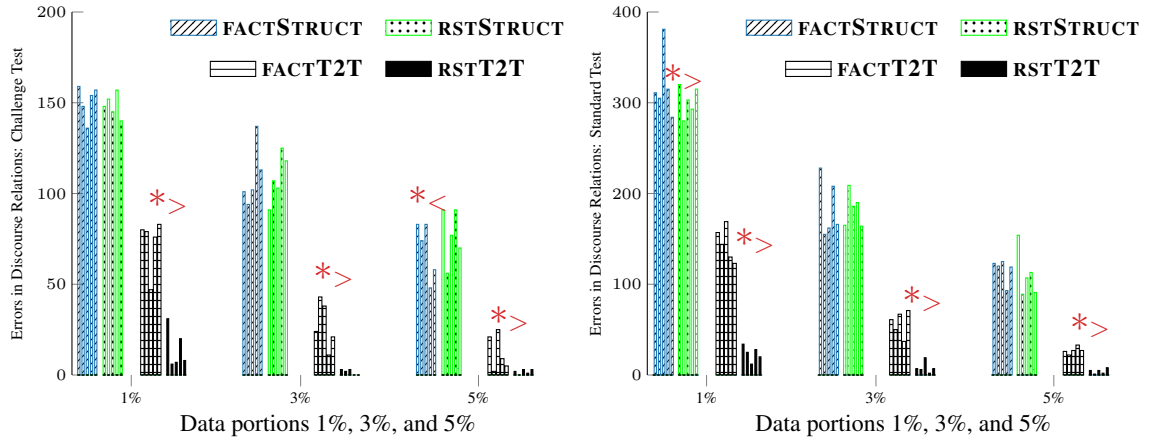

Figure 3: Few Shot Models: Discourse relation realization on the Challenge and Standard tests (A $*_>$ B or B $*_<$ A indicate that the model A has significantly more errors than the model B, where the significance level is set to 0.05; we use this convention in all figures)

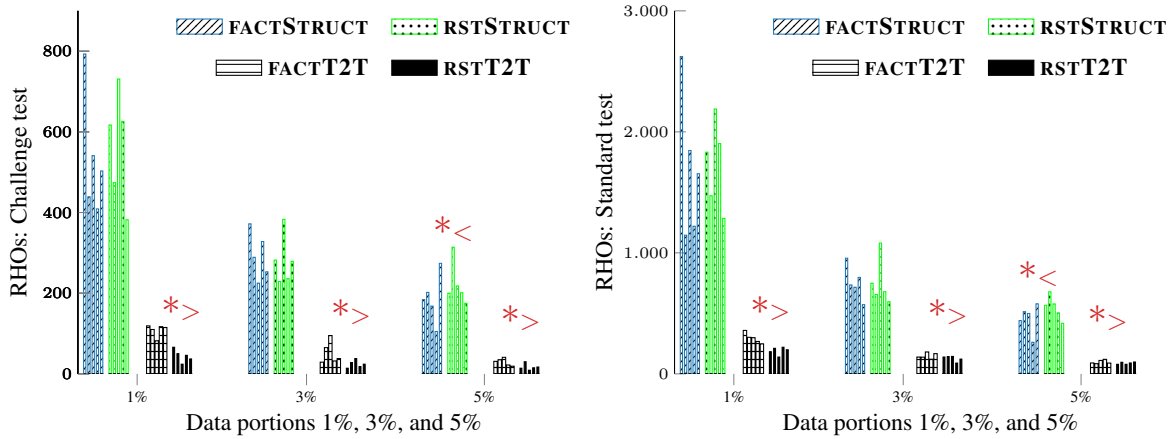

Figure 4: Few Shot Models: RHOs on the Challenge and Standard tests

Discourse Relations: Few-shot models realize discourse connectives on average better than zero-shot models, even when deploying 100% of the training data, which we see in Figure 2. On the Challenge set, which contains the constructions whose lookalikes are not contained at all in the zero-shot training data, it is not unexpected that few-shot models perform better. But, even on the Standard test set, we see that few-shot models are better than zero-shot. That being said, we can see that on one of the runs a zero-shot model achieves one of the best scores. We can say that few-shot models are more consistent than zero-shot models; moreover, they are beneficial when training models on small data sets where models may not have enough data to generalize over every possible phenomenon.

RHOs: In Figure 8, we see that RHOs are lower than in cases of 50% data usage. But nevertheless, they are present. Here as well, the best performing models are T2T models. On the Challenge set, few-shot and ten-shot models make less mistakes than zero-shot models. This indeed is correlated with the fact that few-shot and ten-shot models perform better in terms of discourse relations: By realizing discourse structure correctly, the model needs to repeat or omit less information than by making a mistake and then either repeating the same information again or omitting it because it does not fit into the structure it has been building.

We also found that 100% models make very few lexical hallucinations (usually 0). However, we see mistID errors in 100% models almost as many as

Model Name	Z-100	F-100	D-100	F-50	F-5
FACTSTRUCT	2	2	1	4	12
RSTSTRUCT	**4**	**10**	**3**	**8**	**8**
FACTT2T	3	0	0	7	0
RSTT2T	3	0	0	0	3

Table 2: Maximum of mistID errors of models on the Challenge test set

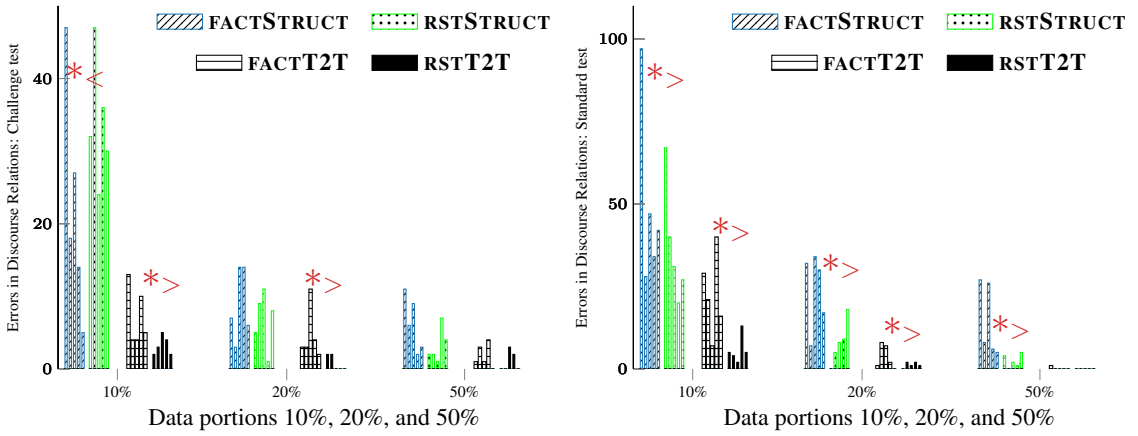

Figure 5: Few Shot Models: Discourse relation realization on the Challenge and Standard tests

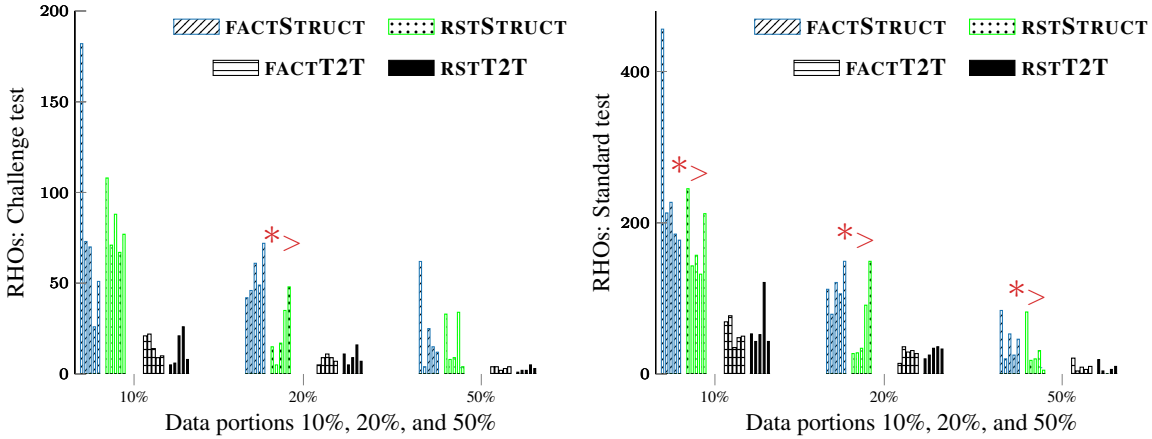

Figure 6: Few Shot Models: RHOs on the Challenge and Standard tests

we see in models trained on small portions of data. Table 2 shows that RSTSTRUCT models trained on Z-100 and F-100 data do not have significant improvements over the models trained on smaller portions of data. It may also seem that a zero-shot RSTSTRUCT model does better than a few-shot one. We have closely examined those cases. The mistID errors arise in those cases whose lookalikes have not been seen by zero-shot models, i.e., the ones similar to the cases shown in Figure 2b and Figure 2a. Zero-shot models either skip some of the comparisons or do it differently—as shown in Example (2), Z.a makes the comparison differently from Ref, whereas Z.b skips it entirely. This is apparently why zero-shot models do not produce as many mistID errors. By contrast, few-shot models are able to recognize those constructions (as they have seen three of each in training) and try to realize them, which they do quite successfully but in so doing they may commit mistID errors, as shown in Example (2), F.

(2) Ref: Like the kylix you recently saw, this lekythos originates from Attica. It was part of a collection dedicated to Athena. Now this lekythos is exhibited in The National Archaeological Museum. It shows the goddess Athena.

F: Like the lekythos you recently saw, this kylix was originally from Attica. It was part of a collection dedicated to Athena. Now this kylix is exhibited in The National Archaeological Museum. It shows the goddess Athena.

Z.a: This lekythos originates from Attica. Like the kylix, this lekythos was originally from Attica. It was part of a collection dedicated to Athena. This lekythos is currently in The National Archaeological Museum. It depicts the goddess Athena.

Z.b: This is a lekythos and it was originally from Attica. It was part of a collection dedicated to Athena. This lekythos is currently in The National Archaeological Museum. It depicts the goddess Athena.

This hypothesis can be checked by looking at ten-shot model performance: They have fewer mistID errors than few-shot models, presumably because they are more comfortable with those constructions, as they see them more than few-shot models. If we look at F-50% models (see Table 2), they do not do worse on mistIDs than F-100% models. That

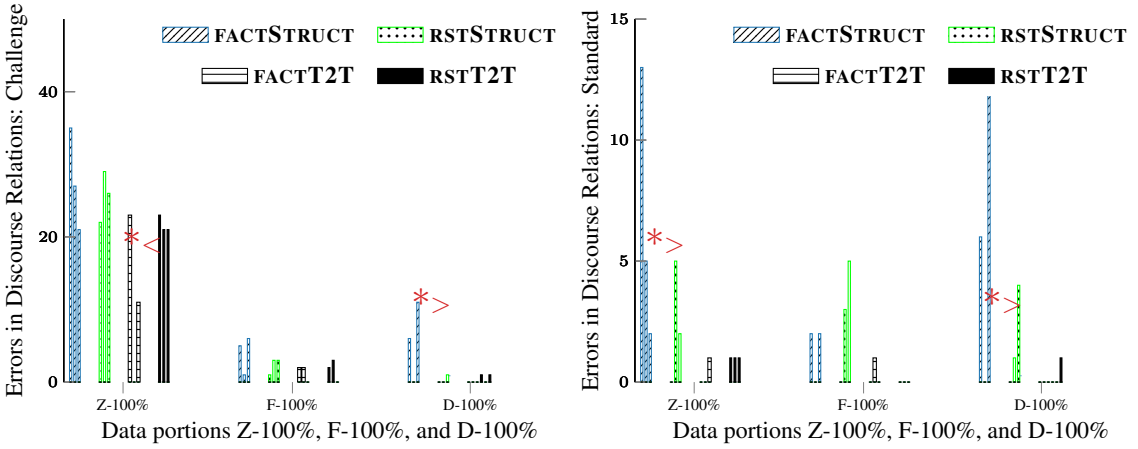

Figure 7: 100% models, Zero-, Few-, and 10-Shot: Discourse relation realization on the Challenge and Standard tests

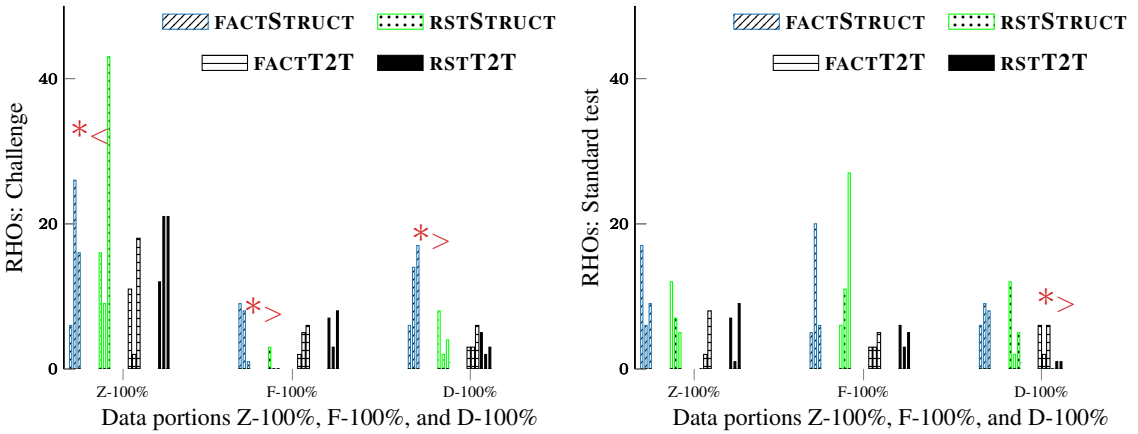

Figure 8: 100% models, Zero-, Few-, and 10-Shot: RHOs on the Challenge and Standard tests

could perhaps be explained by the fact that with F-50% models and D-100% models, both have the same relative number of constructions of interest, which means that 50% models have twice as high concentration of those examples compared to the corresponding 100% few-shot models.

Summary: RST vs. FACT At 100% data, all models show more or less the same performance according to the metrics we use. In terms of hallucinations, we detected that only on Z-100 data, FACTSTRUCT model was prone to hallucinating 'large vessel.' We also found that in D-100 data, FACTSTRUCT has high variance, both in lexical hallucinations and RHOs.

6 Discussion and Conclusion

The development of neural NLG led to an understandable focus on simpler phenomena; the networks in currency at the time seemed to perform best on short, entity-focused texts. While new methods frequently make progress by working on simple domains, we echo the conclusions of Stevens-Guille et al. (2020) that neural methods can and should address more complex, rhetorically structured text, which they must if they are to produce genuinely coherent discourses (Prasad et al., 2008). Our results here bolster those conclusions and provide further evidence for the usefulness of explicit discourse coding in the input to neural systems, especially when data is limited in size. In line with the contemporary wisdom concerning pre-trained models, our results suggest that fine-tuning such models when labeled data for specific domains is limited improves the felicity of generated texts. While increases in available data do always improve the quality of generated texts in terms of grammatically and correctness, we see fast and dramatic improvements when using text inputs, with only more gradual increases in quality in the case of structured input. But we stress that dis-

course relations are enormously helpful when the dataset for the domain is limited: at lower levels of data usage, RSTT2T consistently significantly outperforms FACTT2T on every metric we use. Given the benefits of explicitly encoding discourse relations in the input to the models reported here, we conclude by recommending the continued development of NLG corpora in which discourse relations are present in the meaning representations.

For today, even though various corpora have been designed for natural language generation purposes, corpora with discourse structure information are not available. Given our results showing the benefits of having discourse information in the input, we hope that more corpora will be designed where discourse information is provided with the help of discourse relations.

Acknowledgments

We thank Amy Isard for helping us with Methodius. We are thankful to three anonymous reviewers for their helpful comments. We also want to thank The Ohio Super Computer Center (Center, 1987) for their support as they provided us with needed computational power. This research was supported by a collaborative open science research agreement between Facebook and The Ohio State University. The last author is a paid consultant for Facebook.

References

Ankit Arun, Soumya Batra, Vikas Bhardwaj, Ashwini Challa, Pinar Donmez, Peyman Heidari, Hakan Inan, Shashank Jain, Anuj Kumar, Shawn Mei, Karthik Mohan, and Michael White. 2020. Best practices for data-efficient modeling in NLG:how to train production-ready neural models with less data. In *Proceedings of the 28th International Conference on Computational Linguistics: Industry Track*, pages 64–77, Online. International Committee on Computational Linguistics.

Anusha Balakrishnan, Vera Demberg, Chandra Khatri, Abhinav Rastogi, Donia Scott, Marilyn Walker, and Michael White. 2019. Proceedings of the 1st workshop on discourse structure in neural nlg. In *Proceedings of the 1st Workshop on Discourse Structure in Neural NLG*.

Ohio Supercomputer Center. 1987. Ohio supercomputer center.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 176–181, Portland, Oregon, USA. Association for Computational Linguistics.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2020. Evaluating the state-of-the-art of end-to-end natural language generation: The e2e nlg challenge. *Computer Speech & Language*, 59:123–156.

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

Amy Isard. 2016. The methodius corpus of rhetorical discourse structures and generated texts. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 1732–1736, Portorož, Slovenia. European Language Resources Association (ELRA).

Mihir Kale and Abhinav Rastogi. 2020. Template guided text generation for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6505–6520, Online. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Alex Lascarides and Nicholas Asher. 2007. Segmented discourse representation theory: Dynamic semantics with discourse structure. In H. Bunt and R. Muskens, editors, *Computing Meaning: Volume 3*, pages 87–124. Kluwer Academic Publishers.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

William C. Mann and Sandra A. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text & Talk*, 8(3):243 – 281.

Eric W. Noreen. 1989. *Computer-intensive methods for testing hypotheses : an introduction*. Wiley, New York.

Baolin Peng, Chenguang Zhu, Chunyuan Li, Xiujun Li, Jinchao Li, Michael Zeng, and Jianfeng Gao. 2020. Few-shot natural language generation for task-oriented dialog.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind K Joshi, and Bonnie L Webber. 2008. The penn discourse treebank 2.0. In *LREC*. Citeseer.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Ehud Reiter and Robert Dale. 2000. *Building natural language generation systems*. Cambridge university press.

Symon Stevens-Guille, Aleksandre Maskharashvili, Amy Isard, Xintong Li, and Michael White. 2020. Neural NLG for methodius: From RST meaning representations to texts. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 306–315, Dublin, Ireland. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

A Appendix 0

A.1 Stratified Approximated Randomization (AR)

The principle behind of Stratified Approximated Randomization (AR) test can be explained as follows: Given that model A output on strata of size $n > 0$ (e.g. a test item can be a stratum) are $a_1 \dots a_n$ and model B outputs on the same n strata are $b_1 \dots b_n$, the performance of the models A and B can be considered significantly different if by swapping a_i with b_i with probability 0.5 would result in a sequence $a_1^{'} \dots a_n^{'}$ (i.e., for every $i \in \{1..n\}$, $a_i^{'}$ is a_i with probability 0.5 or b_i with probability 0.5) and a sequence $b_1^{'} \dots b_n^{'}$ (where, for every $i \in \{1..n\}$, $b_i^{'}$ is a_i with probability 0.5 or b_i with probability 0.5) usually differ less from each other than the original sequences $a_1 \dots a_n$ and $b_1 \dots b_n$ differ from each other.

One may take in the role of a_i (where $i \in \{1..n\}$), not just single output of a model, but a set of outputs obtained by several different runs of the same model. That is, we can have $a_i = \{r_1^1, r_i^2, \dots, r_i^k\}$ where $k \geq 2$ and r_i^l is the output of the l-th run of the model A on the stratum i. Below, we assume that each r_i^l has a numerical value. This allows us to compare two models A and B, each run k times with their respective outputs.

We first compute the expectation (mean) of the sample $a_1 \dots a_n$ by taking mean of each set $a_i = \{r_1^1, r_i^2, \dots, r_i^k\}$ and then calculating their mean. We denote it by m_A. We do the same for the sample $b_1 \dots b_n$ and denote their mean by m_B. Let $d_m = |m_A - m_B|$.

Now we define the following procedure: Construct $a_1^{'} \dots a_n^{'}$ and $b_1^{'} \dots b_n^{'}$ by swapping $a_i = \{r_1^1, r_i^2, \dots, r_i^k\}$ with $b_i = \{r_1^1, f_i^2, \dots, f_i^k\}$. Calculate the mean of $a_1^{'} \dots a_n^{'}$ and the mean of $b_1^{'} \dots b_n^{'}$, denote them by $m_A^{'}$ and $m_B^{'}$ respectively. Compute $d_m^{'} = |m_A^{'} - m_B^{'}|$. We perform this procedure multiple times, say N. If out of N cases, for p-percent (usually p is 5) or less cases we find that $d_m^{'} \geq d_m$, we say that model A and B are significantly different with significance at $p\%$. (Below, in our experiments we take $N = 1000$ and $p = 5$, which is usually considered to be a sufficient margin of significance.)

A.2 Lexical Hallucinations

Figure 9 shows numbers of lexical hallucinations various models make.

A.3 mistID Statistics

Figure 10, Figure 11, and Figure 12 show mistID errors on various runs and models trained on various data portions.

A.4 Zero-shot Results on Discourse Relation Realization

We report performance of the zero-shot models in terms of generating discourse relations relations on Figures 13, Figures 14, Figures 15, and Figures 16.

B Reproducibility Details

We use the pretrained T5-Large HuggingFace transformer model (Wolf et al., 2019). There are total 737683456 trainable parameters in this model. The T5 models are fine-tuned using cross entropy loss without label smoothing. The learning rate is constantly 2×10^{-5} and the batch size is 8 samples. The optimizer is Adam (Kingma and Ba, 2014) where $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \times 10^{-8}$, and the weight decay is 0. The best checkpoint is selected by validation with patience of 10 training epochs. For every experiment, the computing infrastructure we used is an NVIDIA V100 GPU and an Intel(R) Xeon(R) Platinum 8268 CPU @ 2.90GHz CPU.

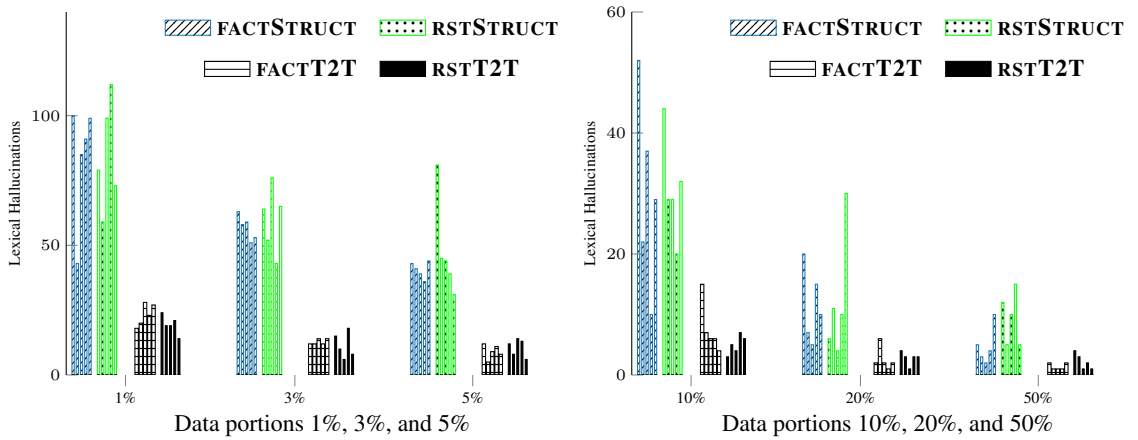

Figure 9: Few Shot Models: Lexical hallucinations combined on the Challenge and Standard tests (no significance tests were performed on lexical hallucinations as they were counted per test set, not per example)

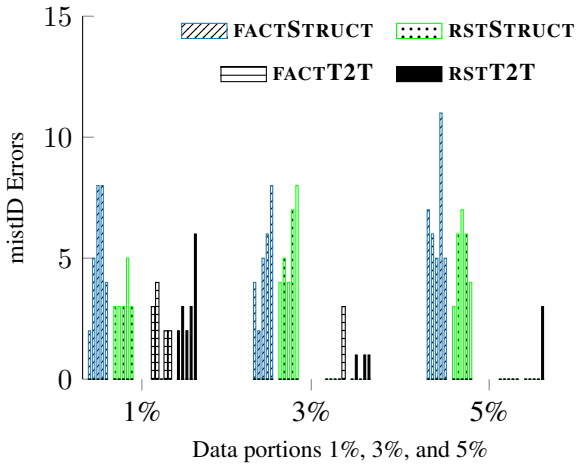

Figure 10: Few Shot Models: mistID errors on the Challange set

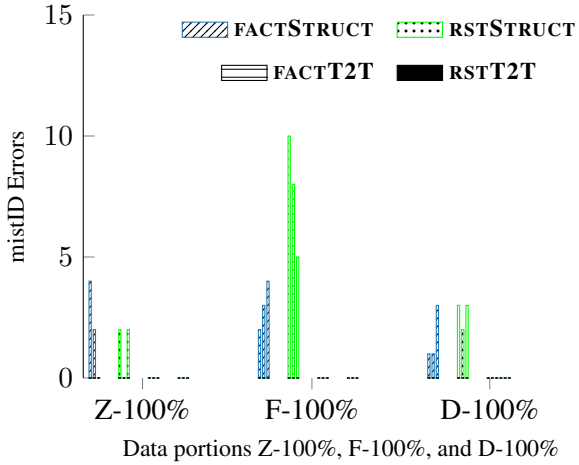

Figure 12: 100% Models: mistID errors on the Challenge test

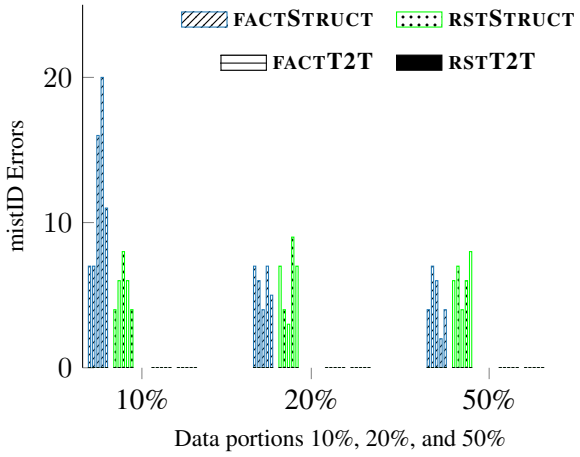

Figure 11: Few Shot Models: mistID errors on the Challenge set

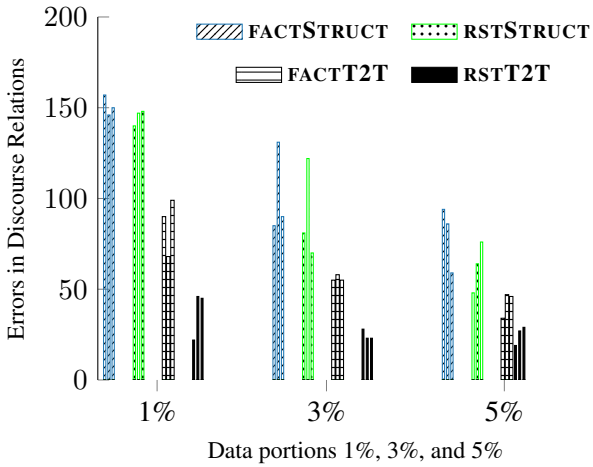

Figure 13: Zero-shot Models: Discourse relation realization on the Challenge test set

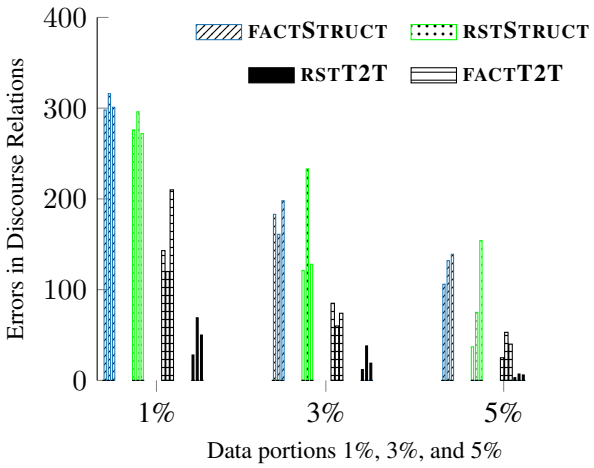

Figure 14: Zero-shot Models: Discourse relation realization on the Standard test set

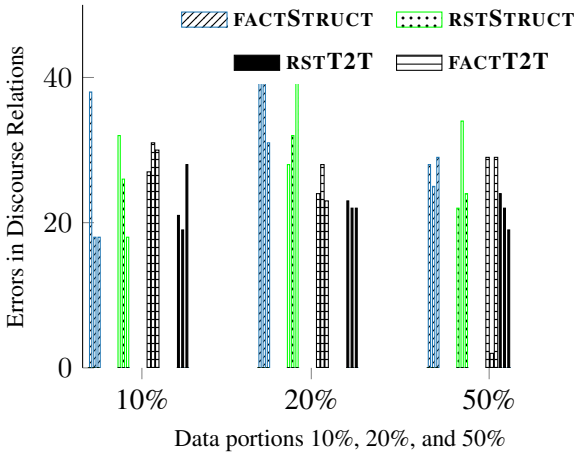

Figure 15: Zero-shot Models: Discourse relation realization on the Challenge test set

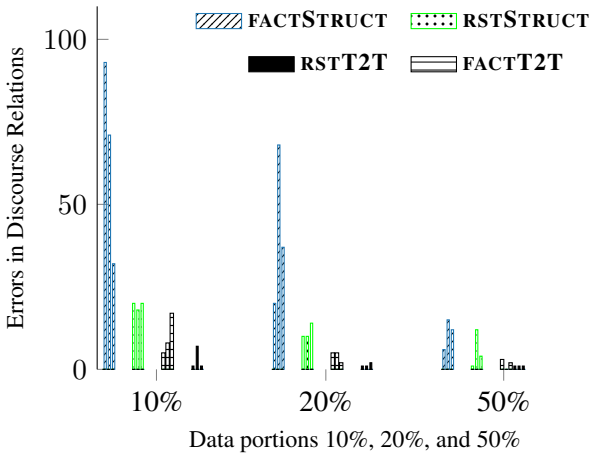

Figure 16: Zero-shot Models: Discourse relation realization on the Standard test set