

No that’s not what I meant: Handling Third Position Repair in Conversational Question Answering

Vevake Balaraman Arash Eshghi Ioannis Konstas Ioannis Papaioannou

AlanaAI

{vevake, arash, ioannis.k, ioannis}@alanaai.com

Abstract

The ability to handle *miscommunication* is crucial to robust and faithful conversational AI. People usually deal with miscommunication immediately as they detect it, using highly systematic interactional mechanisms called *repair*. One important type of repair is *Third Position Repair* (TPR) whereby a speaker is initially misunderstood but then corrects the misunderstanding as it becomes apparent after the addressee’s erroneous response (see Fig. 1). Here, we collect and publicly release REPAIR-QA¹, the first large dataset of TPRs in a conversational question answering (QA) setting. The data is comprised of the TPR turns, corresponding dialogue contexts, and candidate repairs of the original turn for execution of TPRs. We demonstrate the usefulness of the data by training and evaluating strong baseline models for executing TPRs. For stand-alone TPR execution, we perform both automatic and human evaluations on a fine-tuned T5 model, as well as OpenAI’s GPT-3 LLMs. Additionally, we *extrinsically* evaluate the LLMs’ TPR processing capabilities in the downstream conversational QA task. The results indicate poor out-of-the-box performance on TPR’s by the GPT-3 models, which then significantly improves when exposed to REPAIR-QA.

1 Introduction

Participants in conversation need to work together on a moment by moment basis to achieve shared understanding and coordination (Clark, 1996; Clark and Brennan, 1991; Goodwin, 1981; Healey et al., 2018; Mills, 2007). One of the key interactional mechanisms that enables this is called *repair* (Schegloff et al., 1977; Schegloff, 1992) – see Fig. 1: a set of universal, highly systematised (Dingemanse et al., 2015), local methods for dealing with *miscommunication* as it is detected.

¹The dataset, models and code for all experiments are available at <https://github.com/alanaai/Repair-QA>

Figure 1. TPR Example from REPAIR-QA

(T1) U: What is the name of **the princess in Frozen?** (Trouble Source)
(T2) S: The name of the princess who eventually becomes queen is Elsa
(T3) U: **no I mean the name of the younger sister** (Third Position Repair)
(T4) S: The name of the younger sister is Anna

Miscommunication likewise arises in human-machine conversation. Therefore, the ability to interpret and generate effective repair sequences is crucial to *robust* Conversational AI technology, and to ensuring that Natural Language Understanding (NLU) output and/or subsequent system responses remain *faithful* to what the user intended.

Considerable attention has been paid to computational models for the interpretation and generation of *self-repair* (see (Hough and Schlangen, 2015; Hough, 2015; Shalyminov et al., 2017; Skantze and Hjalmarrsson, 2010; Buß and Schlangen, 2011; Hough and Purver, 2012) among others): a class of repairs whereby the speaker corrects themselves on the fly within the same conversational turn (e.g. “User: I want to go to London uhm sorry Paris”). Similarly, the crucial role of generating and responding to *Clarification Requests* (e.g. “Pardon/what/who?”) in conversational models has long been recognised (see (San-Segundo et al., 2001; Purver, 2004; Purver and Ginzburg, 2004; Rieser and Moore, 2005; Rodríguez and Schlangen, 2004; Rieser and Lemon, 2006) among others), but existing systems either remain limited (e.g. Curry et al. (2018)) or do not support this at all – see Purver et al. (2018) for an overview of existing models of repair.

In this paper, we focus on an important class of repairs that has, to our knowledge, been neglected in the NLP community, likely due to the unavail-

ability of data: *Third Position Repair* (TPR; (Scheffloff, 1992); aka repair after next turn). These occur when the addressee initially misunderstands the speaker (Fig. 1 at T1, the *trouble source* turn), responds based on this misunderstanding (at T2), which in turn reveals the misunderstanding to the addressee who then goes on to correct the misunderstanding (at T3). Our **contributions** are: (1) We collect, analyse and release REPAIR-QA, the first large dataset of Third Position Repairs (TPR) in a conversational QA setting together with candidate repair outcomes (rewrites) for training *repair execution* models; and (2) We then use REPAIR-QA to: (a) train and intrinsically evaluate strong baseline models for the execution of TPRs; and (b) systematically probe the TPR processing capabilities of GPT-3-Curie and GPT-3-Davinci with and without exposing them to examples from REPAIR-QA.

2 The REPAIR-QA dataset

In this section, we describe our method for eliciting Third Position Repairs (TPR) from AMT crowd workers (henceforth annotators). Overall, we set this up as a dialogue completion task whereby the annotators are given a dialogue snippet in which a miscommunication has occurred: they are given T1 (Fig. 1; the *Trouble Source*) and T2 (the erroneous system response). They are then asked to provide a (Third Position) correction at T3 to resolve the miscommunication.

Method: Eliciting TPRs We built our dialogue completion tasks on Amazon Mechanical Turk (AMT). Annotators were paid \$0.29 per annotation for their work (estimated at \$11 per hour). To generate the dialogue completion tasks in order to elicit TPRs, we start from the AmbigQA dataset (Min et al., 2020) since it contains ambiguous questions (i.e. questions that have multiple interpretations and answers) and their corresponding unambiguous questions along with their answers. For each ambiguous question, Q , and the corresponding pair of unambiguous questions with their answers, (Q_1, A_1) and (Q_2, A_2) , we build a dialogue snippet to be completed by the annotator with a TPR as follows: (1) We build an informative *context*, C , that differentiates between questions Q_1 and Q_2 ; (2) The answers in AmbigQA are mostly short, Noun Phrase answers, which do not reveal how the ambiguous question was interpreted or reveal the apparent miscommunication to the annotator. To remedy this, we transform these short

answers to full sentential form using the rule-based approach of Demszky et al. (2018). This allows us to derive sentential forms for A_1 , call it A'_1 ; (3) We build the dialogue snippet with two turns, T1 and T2 – see Fig. 1 – where $T1 = Q$ and $T2 = A'_1$. Annotators are told that their goal was to get a response to Q_2 (indicated by context C); then, given the dialogue snippet which erroneously provides an answer to Q_1 , they are asked to provide *two* alternative TPRs at T3 to get a response to Q_2 instead. For example, in Fig. 1: Q is T1; Q_1 is “What is the name of the princess in Frozen who eventually becomes queen?”; A_1 is “Elsa”; A'_1 is T2; and C is “who eventually becomes queen vs. the younger sister”. The context C is built by identifying the difference between Q_1 and Q_2 . We employ this approach as the AmbigQA unambiguous questions have the same syntactic form as the ambiguous question. Another big advantage of using the AmbigQA dataset is that Q_2 can be seen as the contextually resolved meaning of the TPR which we call the gold ‘rewrite’ following (Anantha et al., 2021). This gold rewrite is used below in our *repair execution* models. See Appendix B for more details.

Statistics and Quality Control The REPAIR-QA dataset consists of **3305** examples (training: 2657, test: 648) which are chosen and annotated from the 4749 examples from the AmbigQA dataset. Each conversation in REPAIR-QA consists of two different TPRs yielding a total 6610 TPR annotations. Table 6 in Appendix shows some examples of the collected data. For quality control, we randomly select 100 TPR annotations from the testset to perform a qualitative inspection of the collected data. We annotate them for (i) Quality: Does the TPR convey the information needed to convey the necessary correction?; (ii) Context-Dependence: Does the TPR contain any context-dependent phenomena (e.g. fragments, ellipsis, pronominals); and (iii) Corrective: Is the TPR formulated explicitly as a correction? (e.g. The TPR in Fig. 1 could have been: “what about the name of the younger sister?” which does not explicitly signal a correction). We find that only 16% of the data contains some noise; that 93% of TPRs contain some form of context-dependency; and that 80% of the TPRs formulate the TPR explicitly as a correction. To further measure the degree to which the interpretation of the TPRs relies on the dialogue context, we measure the unigram overlap between the TPR and the refer-

	BERT Score	BLEU	EM
T5-REPAIR-QA	97.48	72.06	30.40
GPT-3-Davinci	97.22	64.18	25.68
GPT-3-Curie	93.19	52.43	7.60

Table 1: Model performance on the testset of the REPAIR-QA dataset.

	BERTScore	BLEU
T5-REPAIR-QA	1.48	20.12
GPT-3-Davinci	1.76	19.94
GPT-3-Curie	(0.11)	1.85

Table 2: Model ability to generate corrective tokens computed based on the difference in performance of the prediction against the rewrite and the trouble source.

ence rewrite (viz. Q_2 above). We find 28% overlap between them, suggesting that the TPRs are highly context-dependent.

Limitations As such, REPAIR-QA has two important limitations: (1) TPRs can in general sometimes – but rarely – occur at a distance of more than two turns from the *trouble-source* turn (Schegloff, 1992). But the TPRs we collected are always in the third turn following the trouble source: this is an artefact not just of our data collection design as a unilateral dialogue completion task, but also of the architecture of most Conversational QA models that REPAIR-QA is designed to be useful for; and (2) overall we’d have preferred a more ecologically valid setup where TPRs are elicited within a more dynamic, interactive setting rather than as a dialogue completion task. Nevertheless, we believe that this trade-off between difficulty of collecting human-human dialogues, and the breadth of the types of TPR sequences collected is justified.

3 TPR execution

We cast the TPR execution task as a sequence to sequence problem, where input to the model is the dialogue history up to and including the TPR turn, and the model is trained to generate a rewrite of the ambiguous, trouble-source question, reflecting the correction in the TPR. We use a pre-trained T5 model (Raffel et al., 2022) for our experiments and compare against OpenAI’s GPT-3 (Brown et al., 2020) when prompted with TPR examples.

3.1 Repair Execution Results

The models are evaluated against metrics of BERTScore (Zhang et al., 2020), BLEU and Exact Match (EM) between the reference rewrite and the generated output ².

Table 1 shows the performance of all models on the REPAIR-QA testset. The T5 model is fine-tuned using the REPAIR-QA and its performance is reported as T5-REPAIR-QA. The fine-tuned T5-REPAIR-QA model achieves the best performance against the gold rewrites on all the 3 metrics considered. The GPT-3 models (Davinci and Curie) are few-shot prompted with 10 random examples, per test instance, pooled from REPAIR-QA followed by the test data; (see Appendix C for details); unlike the T5-REPAIR-QA model which is fine-tuned using the REPAIR-QA training data. We see a slightly lower performance for Davinci compared to the T5-REPAIR-QA on the automatic evaluation; the Curie model shows significantly inferior performance, especially when looking at EM ³.

Generally, the correction that a TPR provides to the *trouble source* question (T1 in Fig. 1) is very specific and small (often just 1 or 2 words, e.g. “the younger sister” in Fig. 1). Thus a higher BLEU score is more likely even when the model prediction is similar to the trouble source. To evaluate the ability of the models to produce specifically the corrective tokens, we evaluate the models’ predictions against both the gold rewrite and the trouble source itself, and compare these across all metrics. We compute the metrics for the models’ prediction against the gold rewrite on the one hand, and, the trouble source separately on the other hand, and compute the difference between them (simple subtraction). This difference in performance against them is therefore attributable to whether the model was able to produce the few corrective tokens. Table 2 shows this differential evaluation: a similar trend is seen on the models for the BLEU metric but GPT-3-Davinci outperforms other models on BERTScore. This result is discussed further below.

²We also tried an NLI-based text-classifier (Yin et al., 2019) for evaluation but the metric was not suited for this task, hence not reported here.

³We also did a zero-shot evaluation of a T5 model trained only on QReCC (Anantha et al., 2021) – a contextual resolution dataset – against the REPAIR-QA testset: it performed very poorly (BLEU = 37.44) indicating that the patterns of context-dependency in the TPRs are very different from the general patterns of context-dependency found in the QReCC dataset. This further demonstrates the usefulness of REPAIR-QA.

	Q1	Q2
T5-REPAIR-QA	3.53	4.01
GPT-3-Davinci	4.56	4.27

Table 3: Human evaluation of TPR execution models

3.2 Human Evaluation

We asked two expert annotators (two co-authors of the paper) to rate the quality of T5-REPAIR-QA and GPT-3-Davinci model’s output rewrites for executing the TPRs. We separately asked them the following questions: **Q1**: “On a scale of 1 to 5, how well does the model prediction avoid the misunderstanding caused by the ambiguity in the original question?”; and **Q2**: “On a scale of 1 to 5, to what degree is the model prediction asking for the same information as the gold?”. While the answer to Q2 depends on the gold rewrites from REPAIR-QA, the answer to Q1 does not. This is because in executing a TPR what we care about is not necessarily the surface form of the output but instead the overall correction on a *semantic level*. The annotators showed very high interannotator agreement on both questions (average Krippendorff’s $\alpha = 0.8$).

As Table 3 shows, the Davinci model’s performance in the human evaluation is superior to the T5-REPAIR-QA model for both Q1 and Q2. At first glance, this would seem to be inconsistent with the word overlap metrics in Table 1 since the fine-tuned T5-REPAIR-QA model outputs show more overall overlap with the gold rewrites. However, a qualitative inspection of the respective outputs of each model shows that the Davinci model manages to produce rewrites which sufficiently capture the meaning of the TPR even as it doesn’t always reproduce exactly the same words. This explanation is further supported by the BERTScore, semantic similarity results in Table 2 which shows slightly superior performance of the Davinci model (see Table 5 in Appendix for an example comparison). We believe that this is due to the fact the Davinci model is only exposed to ten examples in the prompt each time, whereas the T5-REPAIR-QA model is fine-tuned on all the training data from REPAIR-QA.

4 Extrinsic evaluation of GPT-3’s TPR capabilities in conversational QA

In this section, we use REPAIR-QA to evaluate the TPR processing capabilities of OpenAI’s GPT-3 Davinci model extrinsically in an end-to-end, conversational QA setting. We do this by comparing:

Prompting	BLEU	EM	Unknown
w/o TPR examples	11.40	11.71%	230
with TPR examples	16.98	31.90%	57

Table 4: End-to-end, TPR processing capability of GPT-3 Davinci, with and without being exposed to TPR examples from REPAIR-QA

- the model’s response to the reference rewrite (the corrected, unambiguous form of each question); with
- the response returned after the dialogue snippet with the TPR as its last turn.

If (a) and (b) are identical or highly similar, we can infer that the model was able to interpret the TPR correctly; independently of whether the responses are faithful. We compute the automatic evaluation on the model’s response in (b) while treating the model’s response in (a) as the ground truth. This would evaluate if the model was consistent in generating responses for both the rewrite and the TPR dialogue snippet. This evaluation is performed under two *prompting conditions*: **With TPR examples**: where the model is exposed to 10 TPR examples in the prompt; and; **Without TPR examples**: where the model is prompted without any TPR examples. In both conditions, the preamble instructs Davinci to generate *unknown* as the answer if the question is either nonsense, trickery, or Davinci has no clear answer. In addition, in both cases, the model is instructed to provide short form, Noun Phrase answers (for details of all of the preambles used, see Appendix, Sec. C).

There could in general be two reasons for *unknown* predictions after a TPR: (i) the Davinci’s closed-book knowledge is insufficient to answer the (disambiguated, corrected) question; or; (ii) It was unable to interpret the TPR sequence. Since we are interested only in (ii), we *exclude* all cases where the model was not able to answer the unambiguous question (i.e case (a) above), viz. the reference rewrite (the meaning of the TPR). This way we ensure that the model can actually answer the target, rewritten / corrected question. After these are excluded, the ‘Unknown’ column in Table 4 contains the number of *unknown* responses to the TPRs; showing how the model improves when exposed to TPR examples in conversational QA.

For cases where both (a) and (b) above receive answers from GPT3, we perform automatic evaluation to measure the similarity between them: this is

also shown in Table 4. As a surface overlap metric, BLEU is suitable for this evaluation since we compare short answer tokens with many of these being bare Noun Phrases, e.g. names of movies, persons, dates, etc: there are no or few semantically similar paraphrases of these answers.

As is evident in Table 4, the TPR processing capability of Davinci in conversational QA when not exposed to any TPR example is very poor, but this improves significantly with a handful of TPR examples in the prompt. This shows that state-of-the-art LLMs do not handle TPRs well at all out-of-the-box, validating the requirement for datasets addressing specific dialogue phenomena like TPRs.

Even when the model is exposed to TPR sequences in the prompt (the “with TPR examples” condition) the model’s performance still leaves a lot to be desired: the model’s responses to the TPRs matches the expected response only in 31.9% of cases.

To verify the meaningfulness of the 31.9% exact match and the corresponding low BLEU score of 16.98 between model responses in (a) and (b), we went on to do a manual inspection of the data. Fig. 2 shows two examples of these responses:

User: Who plays the leprechaun in the leprechaun movie?
System: Warwick Davis
TPR: I was referring to leprechaun origins
Rewrite: Who plays the leprechaun in the Leprechaun Origins movie?

Response to (a): Dylan Postl
Response to (b): Linden Porco

User: Who created the quote keep calm and carry on?
System: British government
TPR: I wanted to know the name of the ministry though.
Rewrite: Which ministry created the quote keep calm and carry on?

Response to (a): British Ministry of Information
Response to (b): Ministry of Information

Figure 2: Two pairs of example responses provided by Davinci in its responses to (a): the unambiguous, corrected question rewrite; and; (b): the three turn TPR sequence

We can see different answers when prompted with the dialogue including the TPR ((b) above) and when prompted with the rewrite (unambiguous form of the input; (a) above). Such inconsistent answers are frequent from the model even when REPAIR-QA examples are provided in the prompt.

For more certainty, we further computed more focused BLEU scores only in cases where there was

no exact match between the model’s responses in (a) and (b). The BLEU scores on these not exactly matching responses, **with** and **without** exposure to TPR examples were 8.81 and 8.08 respectively. This shows that the model provides different, inconsistent answers for a large part of the REPAIR-QA dataset even when exposed to TPR examples in the prompt; which in turn shows that the model is not able to interpret or integrate the TPR for too large a part of REPAIR-QA. On a very small proportion of cases, Davinci provides responses which are similar (usually a partial match as in the second example above: “British Ministry of Information” vs. “Ministry of Information”), which is captured by the BLEU score metric.

5 Conclusion

The ability to interpret and generate repairs is essential to robust and faithful Conversational AI. In this paper, we focused on Third Position Repair (TPR) that’s been largely neglected in the NLP community. We collect, analyse and release the first large dataset of TPRs and use it to evaluate strong baseline repair execution models, as well as the conversational QA performance of Open AI’s Davinci model when it encounters TPRs. The results show very poor out-of-the-box performance on TPRs which then improves when the model is exposed to REPAIR-QA dataset. But even then, Davinci does not exhibit an acceptable performance on TPRs when evaluated end to end in a Conversational QA setting. This is a symptom of the sparsity of TPRs in the original dialogic data used to pre-train Davinci and LLMs in general; and suggests that LLM researchers should be more selective in how they compile the datasets used for pretraining.

For this paper, we did not have a chance to evaluate later releases of LLMs (e.g. GPT3.5; GPT4) - it would be telling to see how much performance improvement the later models might exhibit on TPRs. Our evaluation methods above in conjunction with the REPAIR-QA dataset can be used easily to perform these evaluations. Finally, we hope that this paper inspires further computational research into miscommunication phenomena in dialogue in the context of recent astonishing successes with LLMs.

References

Raviteja Anantha, Svitlana Vakulenko, Zhucheng Tu, Shayne Longpre, Stephen Pulman, and Srinivas Chappidi. 2021. [Open-domain question an-](#)

- swering goes conversational via question rewriting. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 520–534, Online. Association for Computational Linguistics.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS’20, Red Hook, NY, USA. Curran Associates Inc.
- Okko Buß and David Schlangen. 2011. Dium : An incremental dialogue manager that can produce self-corrections. In Proceedings of SemDial 2011 (Los Angelogue), Los Angeles, CA, pages 47–54.
- H. H. Clark and S. A. Brennan. 1991. Grounding in communication, pages 127–149. Washington: APA Books.
- Herbert H. Clark. 1996. Using Language. Cambridge University Press.
- Amanda Cercas Curry, Ioannis Papaioannou, Alessandro Suglia, Shubham Agarwal, Igor Shalyminov, Xu Xinnuo, Ondrej Dusek, Arash Eshghi, Ioannis Konstas, Verena Rieser, and Oliver Lemon. 2018. Alana v2: Entertaining and informative open-domain social dialogue using ontologies and entity linking. In 1st Proceedings of Alexa Prize (Alexa Prize 2018).
- Dorottya Demszky, Kelvin Guu, and Percy Liang. 2018. Transforming question answering datasets into natural language inference datasets. arXiv preprint arXiv:1809.02922.
- Mark Dingemanse, Seán G. Roberts, Julija Baranova, Joe Blythe, Paul Drew, Simeon Floyd, Rosa S. Gisladottir, Kobin H. Kendrick, Stephen C. Levinson, Elizabeth Manrique, Giovanni Rossi, and N. J. Enfield. 2015. Universal principles in the repair of communication problems. *PLOS ONE*, 10(9):1–15.
- C. Goodwin. 1981. Conversational organization: Interaction between speakers and hearers. Academic Press, New York.
- Patrick G. T. Healey, Gregory J. Mills, Arash Eshghi, and Christine Howes. 2018. Running Repairs: Coordinating Meaning in Dialogue. *Topics in Cognitive Science (topiCS)*, 10(2).
- Julian Hough. 2015. Modelling Incremental Self-Repair Processing in Dialogue. Ph.D. thesis, Queen Mary University of London.
- Julian Hough and Matthew Purver. 2012. Processing self-repairs in an incremental type-theoretic dialogue system. In Proceedings of the 16th SemDial Workshop on the Semantics and Pragmatics of Dialogue (SeineDial), pages 136–144, Paris, France.
- Julian Hough and David Schlangen. 2015. Recurrent neural networks for incremental disfluency detection. In INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015, pages 849–853.
- Gregory J. Mills. 2007. Semantic co-ordination in dialogue: the role of direct interaction. Ph.D. thesis, Queen Mary University of London.
- Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 5783–5797, Online. Association for Computational Linguistics.
- Matthew Purver. 2004. CLARIE: the Clarification Engine. In Proceedings of the 8th Workshop on the Semantics and Pragmatics of Dialogue (SEMDIAL), pages 77–84, Barcelona, Spain.
- Matthew Purver and Jonathan Ginzburg. 2004. Clarifying noun phrase semantics. *Journal of Semantics*, 21(3):283–339.
- Matthew Purver, Julian Hough, and Christine Howes. 2018. Computational models of miscommunication phenomena. In Patrick G. T. Healey, Jan de Ruiter, and Gregory J. Mills, editors, Topics in Cognitive Science (topiCS), volume 10.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2022. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).
- Verena Rieser and Oliver Lemon. 2006. Using machine learning to explore human multimodal clarification strategies. In Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions, pages 659–666, Sydney, Australia. Association for Computational Linguistics.
- Verena Rieser and Johanna Moore. 2005. Implications for generating clarification requests in task-oriented dialogues. In Proceedings of the 43rd Annual Meeting of the ACL, pages 239–246, Ann Arbor. Association for Computational Linguistics.
- Kepa Rodríguez and David Schlangen. 2004. Form, intonation and function of clarification requests in German task-oriented spoken dialogues. In Proceedings of the 8th Workshop on the Semantics and Pragmatics of Dialogue (SEMDIAL), Barcelona, Spain.

- Ruben San-Segundo, Juan M. Montero, J. Ferreiros, R. Córdoba, and José M. Pardo. 2001. Designing confirmation mechanisms and error recover techniques in a railway information system for Spanish. In *Proceedings of the 2nd SIGDial Workshop on Discourse and Dialogue*, pages 136–139, Aalborg, Denmark. Association for Computational Linguistics.
- E.A. Schegloff. 1992. Repair after next turn: The last structurally provided defense of intersubjectivity in conversation. *American Journal of Sociology*, pages 1295–1345.
- E.A. Schegloff, Gail Jefferson, and Harvey Sacks. 1977. The preference for self-correction in the organization of repair in conversation. *Language*, 53(2):361–382.
- Igor Shalyminov, Arash Eshghi, and Oliver Lemon. 2017. Challenging neural dialogue models with natural data: Memory networks fail on incremental phenomena. In *Proceedings of the 21st Workshop on the Semantics and Pragmatics of Dialogue (SemDial 2017 - SaarDial)*, Barcelona.
- Gabriel Skantze and Anna Hjalmarsson. 2010. [Towards incremental speech generation in dialogue systems](#). In *Proceedings of the SIGDIAL 2010 Conference*, pages 1–8, Tokyo, Japan. Association for Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. [Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3914–3923, Hong Kong, China. Association for Computational Linguistics.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. [Bertscore: Evaluating text generation with BERT](#). In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

A Model Training and Inference

The T5 models reported in this paper are implemented in pytorch using HuggingFace (Wolf et al.,

2020) library. The hyperparameter of the models are set as default with the batch size set to 16. The T5 models are trained on a single 16GB GPU and fine-tuned for 5 epochs. The results in Table 1 for T5 models for a single run on the train/test split. For GPT-3 inference, we use OpenAI’s playground⁴ API and get predictions from both Davinci (text-davinci-003) and Curie (text-curie-001) models.

B Data Collection Details

We use Amazon Mechanical Turk⁵ for collecting the human annotations for TPR. The data collection was conducted anonymously.

Crowdworker Quality Control. We conduct a pilot with 4 internal annotators to verify the instructions and revise them before deploying to AMT crowdworkers. To control for the quality of annotations and the language, the crowdworkers are restricted to i) Location is one of Australia, Canada, New Zealand, United Kingdom, United States; ii) HIT approval rate > 80% and; ii) Number of HITs approved > 50. This was done explicitly to control the quality of the annotations collected after examining the annotations from a pilot phase in AMT.

Crowdworker Instructions. Figure 3 shows the instruction provided to the crowdworker and Figure 4 shows the interface, which the crowdworker uses to annotate the provided example. We explicitly instruct the crowdworkers to mark examples in which any of the information is unclear. To better explain the concept of TPR to the crowdworkers, we use the term **late correction instead of TPR** in the annotation instructions.

C GPT-3 prompts

The prompt used to query GPT-3 model to get predictions for both rewrite and QA is presented here. The **text in blue** indicate the tokens that the GPT-3 has to generate.

Rewriter prompts. Prompt used to generate rewrites from GPT-3. We use 5 examples in the prompts (single example is shown here for reference).

"Rewrite the Question Q based on the late correction LC.

⁴<https://beta.openai.com/playground>

⁵www.mturk.com

User: What is the most current episode of Ray Donovan?
System: The title of the most current episode of Ray Donovan is you'll Never Walk Alone.
User (TPR): What number was it in the series?

GPT-3-Davinci: What is the number of the most current episode of Ray Donovan titled "You'll Never Walk Alone"?
T5-QReCC+REPAIR-QA: What number was the most current episode of Ray Donovan?
Reference: What is the number overall of the most current episode of Ray Donovan?

Table 5: Prediction from different models on an example from REPAIR-QA.

User: Where do you hit to test your reflexes?
System: You hit to test your ankle jerk reflexes in Achilles tendon.

TPR-1: No, I meant your biceps, not ankle.
TPR-2: I should have been clearer. I wanted to know about the location to test for biceps reflexes.
Rewrite: Where do you hit to test your biceps reflexes?

User: Who sings i'm telling you i'm not going?
System: Jennifer Holliday sings i'm telling you i'm not going in the musical Dreamgirls.

TPR-1: I should have asked, who sang the song in 1982.
TPR-2: I wanted to know the singer in 1982, not in the musical Dreamgirls.
Rewrite: Who sings i'm telling you i'm not going in 1982?

User: Who is the lead singer of doobie brothers?
System: Johnston is the first lead singer of doobie brothers.

TPR-1: I want to know who was the second lead singer not the first.
TPR-2: I was wanting to know the second lead singer not the first.
Rewrite: Who is the second lead singer of doobie brothers?

User: Who has won the european cup the most?
System: Real Madrid has won the european cup the most.

TPR-1: Instead of club, can you tell me the country with the most.
TPR-2: I am looking for the country instead of the club with them most.
Rewrite: What country has won the european cup the most?

User: How much did titanic make at the box office?
System: Titanic (1953 film) made \$2,250,000 at the box office.

TPR-1: I meant the 1997 version.
TPR-2: I was thinking of the 1997 one.
Rewrite: How much did Titanic (1997 film) make at the box office?

User: Who is winner of womens world cup 2017?
System: New Zealand is the winner of the Women's Rugby World Cup in 2017.

TPR-1: Yeah, but who won the cricket world cup?
TPR-2: What I wanted to know is who won the cricket cup.
Rewrite: Who is the winner of the Women's Cricket World Cup in 2017?

User: Who plays the king of france in the borgias?
System: Michel Muller plays King Charles VIII of France in The Borgias (2011 TV series).

TPR-1: I meant to ask who played louis xii.
TPR-2: Sorry but I was looking for louis xii.
Rewrite: Who plays King Louis XII of France in The Borgias (2011 TV series)?

Table 6: Examples from the REPAIR-QA dataset.

Annotation Instructions (Click to collapse)

Late Correction

Late Correction

A **Late Correction** is one of the ways somebody might deal with or correct a misunderstanding in conversation with another. Suppose a User and a Chatbot are having a conversation, and the Chatbot initially misunderstands the User, maybe because the User was being too vague about what they meant. As a result, the Chatbot might say or do something which the User didn't actually intend/want, and thereby reveal to the User that the Chatbot has misunderstood them. User might then go on to reformulate, or reiterate what he/she had said in order to repair, correct or otherwise deal with the misunderstanding.

For example, consider the following snippet of a conversation:

Example :

The user wants to know the date on which the Harry Potter and the Sorcerer's Stone movie came out in **cinemas**. But he poses a vague question to the system as below and receives an answer.

User said: When did Harry Potter and the Sorcerer's Stone movie come out?
Chatbot reply: Harry Potter and the Sorcerer's Stone movie came out at **the Odeon Leicester square** on 4 November 2001.

From the Chatbot's response, the User can notice that he was interested in the date for (all) cinemas but the Chatbot has provided a response for the Odeon Leicester square (which is usually a different date). This response from the Chatbot is not incorrect, since the user did not provide any specific information on what they were exactly looking for. So the User can then correct the Chatbot (using a "Late Correction"), clarifying what they meant, e.g. by specifying a different event (or) location (or) place, etc.

Annotation examples:

Context: Odeon Leicester square **VS** cinemas
User said: When did Harry Potter and the Sorcerer's Stone come out?
Chatbot reply: Harry Potter and the Sorcerer's Stone movie came out at the Odeon Leicester square on **4 November 2001** .

Possible Answer 1: I mean in **cinemas**.
Possible Answer 2: no, when did it come out in **cinemas**.
Possible Answer 3: Sorry, I was actually asking when it came out in **cinemas**.

Here, the **Context** provide two contrasting pieces of information (separated by **VS**), where the former is what was understood by in the Chatbot and the latter is what the user really intended; This is therefore a misunderstanding on the Chatbot's part which the user needs to repair/correct. The highlighted text **4 November 2001** is the answer to what the system understood the question to be. But this is not what the user wanted, so the user has to correct the system and clarify what they meant using the information provided in **Context**. The **possible Late Corrections** show **some of** the many possible ways the user can perform this correction.

Note: As the example above shows, the "Possible Answers" can be formulated in many different ways and the annotator is expected to be creative in providing these. These examples are for reference only and they are not the only possible correct answers.

Figure 3: Annotation Instructions provided to the crowd annotators.

Correct the Chatbot by specifying which information you are looking for instead of the information provided by the Chatbot. Please use the information in Context to provide this correction.

Note: If any of the information (either of Context, User or Chatbot) in a given task is unclear, please enter "UNCLEAR" in the textboxes.

Context: winter youth **VS** summer

User said: Where are the 2020 olympic games being held?
Chatbot reply: The 2020 Winter Youth Olympics are being held in **Lausanne, Switzerland**.

Enter two different ways the user can correct the Chatbot and clarify what they actually meant to ask (i.e. provide two possible Late Corrections)

Possible answer 1:

Possible answer 2:

Submit

Figure 4: Interface of the annotation page as shown to the annotators.

Q: What is the percentage of agriculture in gdp of india?

A: The percentage of agriculture in gdp of india in 2017 is 15.4.

LC: I am looking for the year 2014 instead.

Rewrite: What is the percentage of agriculture in gdp of india in 2014?

Q: Who sang the song it's the final countdown?

A: Europe was the band that sang the song it's the final countdown, released in 1986.

LC: I was looking for the name of the lead singer.

Rewrite: [Who sang lead vocals for the song it's the final countdown, released in 1986?](#)

"

QA prompts. The prompt used for the conversational QA task is as below. We use 10 examples in the prompts (single example is shown here for reference).

"I am a highly intelligent question answering bot. If you ask me a question that is rooted in truth, I will give you only the answer phrase. If you ask me a question that is nonsense, trickery, or has no clear answer, I will respond with "Unknown".

Q: Who is the lead singer of doobie brothers?

A: Johnston is the first lead singer of doobie brothers.

Q: I want to know who was the second lead singer not the first.

A: Michael McDonald

Q: Who sang dedicated to the one i love?

A: The Shirelles sang Dedicated to the one I love in 1959.

Q: Could you also tell me who sang the 1967 version of dedicated to the one I love?

A: [The Mamas and the Papas](#)

"