INLG 2022

International Natural Language Generation Conference (INLG 2022)

Proceedings of the 15th International Conference on Natural Language Generation: Generation Challenges

July 17-22, 2022

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 978-1-955917-60-5

Preface to the Generation Challenges

The Generation Challenges (GenChal) are an umbrella event designed to bring together a variety of shared-task efforts that involve the generation of natural language. This year GenChal features two new challenge proposals and reports of organisers and participants from three completed challenges. The generation challenges were coordinated by Anastasia Shimorina.
We thank all challenge organisers and participants for their efforts.

New Challenge Proposals

- The Second Automatic Minuting (AutoMin) Challenge: Generating and Evaluating Minutes from Multi-Party Meetings — Tirthankar Ghosal, Marie Hledíková, Muskaan Singh, Anna Nedoluzhko, Ondrej Bojar

- The Cross-lingual Conversation Summarization Challenge — Yulong Chen, Ming Zhong, Xuefeng Bai, Naihao Deng, Jing Li, Xianchao Zhu, Yue Zhang

Completed Challenges

- https://cylnlp.github.io/dialogsum-challenge/ — Organisers: Yulong Chen, Yang Liu, Yue Zhang

- https://sites.google.com/view/hinglisheval — Organisers: Vivek Srivastava, Mayank Singh, Pritam Kadasi; https://codalab.lisn.upsaclay.fr/competitions/1688

- https://reprogen.github.io/ — Organisers: Anya Belz, Maja Popovic, Anastasia Shimorina, Ehud Reiter

Further information

More information about previous challenges at GenChal can be found through `https://sites.google.com/site/genchalrepository/`

Organizing Committee

Program Chair

Samira Shaikh
Thiago Castro Ferreira

Local Chair

Amanda Stent

Invited Speakers

Dimitra Gkatzia, Edinburgh Napier University, UK
Emiel Krahmer, Tilburg University
Margaret Mitchell, HuggingFace
Mohit Bansal, University of North Carolina (UNC) Chapel Hill

SIGGEN Representatives

Ehud Reiter
Emiel Van Miltenburg

Tutorial and Hackathon Chair

Joshua Maynez

Publication Chair

Miruna Clinciu

Sponsor Chair

Dave Howcroft

Social Media Chair

Luou (Lilly) Wen

Area Chairs

Albert Gatt
Chris van der Lee
Claire Gardent
Dimitra Gkatzia
Fei Liu
Malihe Alikhani
Michael White
Saad Mahamood

Tirthankar Ghosal
Yufang Hou

Local Organizing Team

Amy Poulin
Charlotte Buswick
Jake Rogers

Program Committee

Program Committee

Thiago Castro Ferreira, Universidade Federal de Minas Gerais
Miruna Clinciu
Ehud Reiter, University of Aberdeen
Samira Shaikh, University of North Carolina, Charlotte
Amanda Stent, Colby College
Emiel Van Miltenburg, Tilburg University

Table of Contents

The Second Automatic Minuting (*AutoMin*) Challenge: Generating and Evaluating Minutes from Multi-Party Meetings

Tirthankar Ghosal[*], Marie Hledíková[*], Muskaan Singh[$], Anna Nedoluzhko[*], Ondřej Bojar[*]
[*]Charles University, Faculty of Mathematics and Physics, ÚFAL, Czech Republic
[$]Speech and Audio Processing Group, IDIAP Research Institute, Martigny, Switzerland
(ghosal,hledikova,nedoluzhko,bojar)@ufal.mff.cuni.cz, msingh@idiap.ch

Abstract

We would host the *AutoMin* generation challenge at INLG 2023 as a follow-up of the first AutoMin shared task at Interspeech 2021. Our shared task primarily concerns the automated generation of meeting minutes from multi-party meeting transcripts. In our first venture, we observed the difficulty of the task and highlighted a number of open problems for the community to discuss, attempt, and solve. Hence, we invite the Natural Language Generation (NLG) community to take part in the second iteration of AutoMin. Like the first, the second AutoMin will feature both English and Czech meetings and the core task of summarizing the manually-revised transcripts into bulleted minutes. A *new* challenge we are introducing this year is to devise efficient metrics for evaluating the *quality* of minutes. We will also host an optional track to generate minutes for European parliamentary sessions.

We carefully curated the datasets for the above tasks. Our ELITR Minuting Corpus has been recently accepted to LREC 2022 and publicly released.[1] We are already preparing a new test set for evaluating the new shared tasks. We hope to carry forward the learning from the first AutoMin and instigate more community attention and interest in this timely yet challenging problem. INLG, the premier forum for the NLG community, would be an appropriate venue to discuss the challenges and future of *Automatic Minuting*. The main objective of the AutoMin GenChal at INLG 2023 would be to come up with efficient methods to automatically generate meeting minutes and design evaluation metrics to measure the quality of the minutes.

1 Introduction

Ever since most of our interactions went virtual, the need for automatic support to run online meetings became essential. Due to frequent meetings and the resulting context switching, people are experiencing an information overload (Fauville et al., 2021) of epic proportions. Hence a tool to automatically summarize a meeting proceeding would be a valuable addition to the virtual workplace. Automatic minuting (Nedoluzhko and Bojar, 2019) is close to summarization; however, there are subtle differences. While summarization is motivated towards generating a concise and coherent summary of the text, minuting is more inclined towards adequately capturing the contents of the meeting (*where coverage is probably more significant than coherence and conciseness*). Summarizing spoken multi-party dialogues (Bhattacharjee et al., 2022) comes with its own challenges: incorrect/noisy automated speech recognition (ASR) outputs, long discourse, topical shifts, the dialogue turns, redundancies and small talk, etc. Hence we deem automatic minuting to be more difficult than text summarization (Figure 2 in Appendix A shows an envisaged demonstration of the task).

With the *AutoMin* challenge, we want to explore the various problems associated with the task and their potential solutions from the perspective of a multi-year joint community initiative. Apart from the main task of summarizing meeting transcripts into concise, bulleted minute items, another crucial task is to develop efficient evaluation measures to judge the quality of the automatically generated minutes. It is a known fact that the current popular methods of automatic summarization evaluation (e.g., ROUGE (Lin, 2004)) do not guarantee critical quality parameters like *fluency, adequacy, grammatical correctness, etc.* (Ghosal et al., 2021a,b), which is why we have to primarily rely on human evaluation metrics in our shared task. The proposed instance of the AutoMin challenge will venture into developing automatic/semi-automatic evaluation metrics to measure the *quality* of generated minutes. Summarizing the participants' ideas for this challenge and the anticipated follow-up dis-

[1] http://hdl.handle.net/11234/1-4692

Proceedings of the 15th International Conference on Natural Language Generation: Generation Challenges, pages 1 - 11
July 17-22, 2022 ©2022 Association for Computational Linguistics

cussions, we will try to define an ideal meeting summary. Since the task suffers from resource scarcity, we would launch an initiative where interested parties could donate their meetings to prepare a public multimodal, multilingual dataset of real meetings.

2 First AutoMin @ Interspeech 2021

The AutoMin[2] shared task at Interspeech 2021 (Ghosal et al., 2021a) was a first of its kind with this problem. It generated considerable interest in the speech and NLP community. Twenty-seven teams from diverse geographical regions registered, and finally, ten teams (both from academia and industry) actively participated in the challenge. Almost 70 people attended the shared task virtual event. The first AutoMin consisted of one main task (Task A) and two supporting tasks (Task B and Task C), relying on a dataset of transcripts and minutes from primarily technical meetings in English and Czech (Nedoluzhko et al., 2022).

Considering the current non-availability of large-scale domain datasets on multiparty meeting summarization, the best recipe for automatic minuting that evolved out from the first AutoMin is roughly the following: training a deep neural model on available dialogue summarization datasets (SAMSum (Gliwa et al., 2019); DialSum (Chen et al., 2021); etc.) and further fine-tuning it on the minuting or meeting summarization datasets (AMI (Mccowan et al., 2005); ICSI (Janin et al., 2003); AutoMin (Ghosal et al., 2021a)), accompanied by some intelligent pre- and post-processing steps.

3 Task Overview

We would continue with the tasks in the previous AutoMin challenge in the current iteration. However, the new additions would be: (1) *automatically generating the meeting minutes of parliamentary sessions* as part of Task A, and (2) *designing appropriate evaluation schema/metrics to evaluate the generated minutes* as a new Task D.

3.1 Task A

*The **main task** consists of automatically generating minutes from multiparty meeting conversations* provided in the form of transcripts. The objective is to generate minutes as bulleted lists, summarizing the main contents of the meeting, as opposed to usual paragraph-like text summaries. This task would run for the meetings in the ELITR Minuting Corpora (Nedoluzhko et al., 2022) and the *new data we curated from the European parliamentary sessions*.[3] Note that the nature of meetings as well as the corresponding minutes are very different in the two datasets (technical project meetings vs. parliamentary sessions).

3.2 Task B

Given a pair of a meeting transcript and a manually-created minute, the task is to identify whether the minute belongs to the transcript.

During our data preparation from meetings on similar topics, we found that this task could be challenging due to the similarity of the discussed content and anchor points like named entities, e.g., in recurring meetings of the same project on the one hand, and the differences in the style of minuting, on the other hand. Another reason is that some minutes do not capture the central points in the meeting because the external scribes did not understand the context correctly and created minutes that miss significant issues discussed in the meeting or are simply too short.

3.3 Task C

Task C is a variation of Task B. *Given a pair of minutes, the task is to identify whether the two minutes belong to the same meeting or to two different ones.* This task is important as we want to uncover how minutes created by two different persons for the same meeting may differ in content and coverage.

3.4 Task D (New Task)

Given a meeting transcript, a candidate minute, and a set of one or more reference minutes, assign a score indicating the quality of the candidate minute.

The participating evaluation methods can focus on diverse aspects of minutes quality, such as the coverage of content discussed, the adequacy of the description, the readability, etc. We will evaluate the submitted scores with respect to correlation with human judgements in terms of *adequacy*, *fluency* and *grammatical correctness* from AutoMin 2021 human evaluations, and possibly in terms of additional criteria.

In other words, there is not a single evaluation criterion for submissions to Task D. Task D should

[2] https://elitr.github.io/automatic-minuting/index.html

[3] https://emeeting.europarl.europa.eu/emeeting/committee/en/archives

	English		Czech	
Meeting Minuted	#meetings	#hours	#meetings	#hours
Once	30	22	8	2
Twice	65	65	20	20
More than twice	25	22	31	31
Total meetings	120	109	59	53

Table 1: Basic transcript and minutes statistics for ELITR Minuting Corpus.

be treated as a joint exploration rather than an optimization exercise.

4 Dataset Description

We provide the AutoMin 2023 participants with ELITR Minuting Corpus (Nedoluzhko et al., 2022); however, we would allow them to use any external datasets if they explicitly describe them in their system reports.

4.1 ELITR Minuting Corpus for Task A

In our ELITR Minuting Corpus (Nedoluzhko et al., 2022), a meeting usually contains one manually corrected transcript, one original minute (created by a meeting participant; in some cases, these minutes are a detailed agenda which got further updated during or after the meeting), and one or more generated minutes (by annotators).

Table 1 presents our dataset's statistics regarding the number of meetings and hours. We separately count meetings for which we have only one, two, and more than two (up to 11) minutes. For English meetings, either (i) our annotators created both minutes or (ii) one minute was written by one of the participants before or after the meeting and another by our annotator. In contrast, most meetings in the Czech portion of the dataset are minuted at least twice, and more than half of the Czech portion of ELITR Minuting Corpus is minuted 3-5 times.

To address GDPR issues (privacy of participants), we de-identify any information concerning Person, Organisation, Project and Location (in specific cases) names were replaced with the lexical substitute strings [PERSON*number*], [ORGANIZATION*number*], [PROJECT*number*] and [LOCATION*number*] respectively. Additionally, we replaced the names of annotators mentioned in minutes with [ANNOTATOR*number*].

Table 2 reports summary statistics of the texts in ELITR Minuting Corpus and Figure 4 in Appendix B shows a sample minute from the corpus.

Language	English	Czech
# of Meetings	120	59
avg. words per transcript	7,066	8,534
avg. words per summary	373	236
avg. turns per transcript	727	1,205
avg. number of speakers	5.9	7.6

Table 2: Text statistics of ELITR Minuting Corpus.

4.2 EuroParlMin for Task A

We curate this dataset from the publicly available European parliamentary sessions by using the transcripts in the EuroParl dataset (Koehn, 2005) and crawling the corresponding minutes from the EU parliament website.[4]

We automatically create a set of transcript–minute pairs (∼2000). This dataset is new, and we would make this available to the shared task participants.

4.3 Test Data for Task D

There is no training data for Task D (except training data available for Tasks A–C anyway).

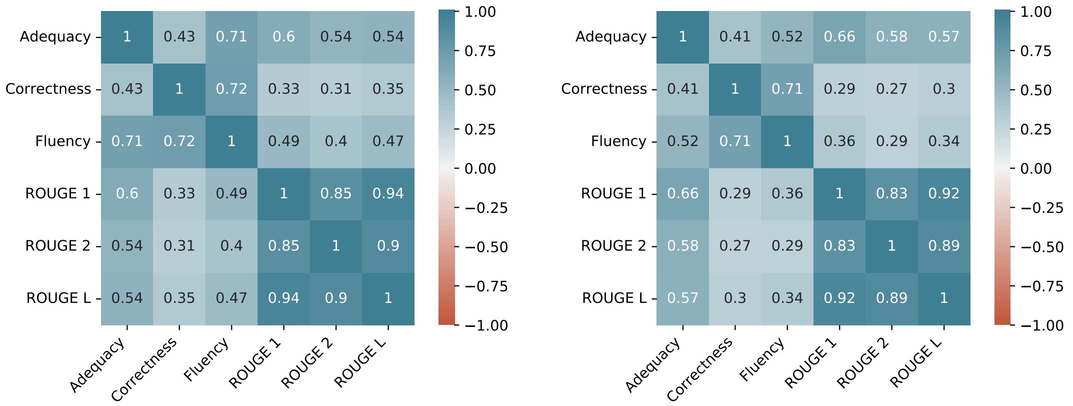
The test data for Task D consists of participants' submissions to AutoMin 2021. Our human evaluators rated each submitted minute by the ten different participating teams (some had multiple submission runs) in three criteria: *adequacy*, *fluency*, and *grammatical correctness* on the test set. Additionally, we plan to design some methods of minute scoring based on the (manual) alignments between the transcript and the minute Polák et al. (2022). These alignments are included in ELITR Minuting Corpus for many of the meetings and their manually created minutes, which can be used as training data. We will also prepare these alignments for AutoMin 2021 submissions, i.e., automatic minutes.

We will use these annotations (*adequacy*, *fluency*, *grammatical correctness* and different scores based on the alignments) as different possible ground truth values for participants in Task D.

It is up to the participants of Task D to propose which type of criterion their metric will focus on. We will evaluate each submission against all available ground truths.

We prepared data for Task B and Task C from ELITR Minuting Corpus (leaving the meetings we selected to run AutoMin 2021).

[4] https://www.europarl.europa.eu/committees/en/meetings/minutes



(a) Average across multiple references.

(b) Maximum across multiple references.

Figure 1: Correlations of metrics (human and automatic) used in AutoMin 2021 across all participants. Each cell represents the Pearson correlation between the two types of measurements of a given meeting. With multiple reference minutes, the automatic scores are aggregated with (a) average and (b) maximum. Two independent judges assigned manual scores, and to arrive at a single score per minute, we again aggregated them with average or maximum.

5 Evaluation Campaign

5.1 Human Evaluation

We will perform human evaluation on the submissions in Task A (both English and Czech) with the usual metrics: *adequacy, fluency, relevance, and grammatical correctness* (Kryscinski et al., 2020) on a Likert scale of 1-5.

1. **Adequacy** assesses if the minute adequately captures the major topics discussed in the meeting, also considering coverage (all such topics reflected).
2. **Fluency** reflects if the minute consists of fluent, coherent texts and is readable to the evaluator.
3. **Grammatical Correctness** checks the level to which the minute is grammatically consistent.
4. **Relevance** signifies whether the important content from the source transcript appear in the candidate minutes.

Along with that, we will launch a pilot evaluation of the submitted minutes via our ALIGNMEET tool (Polák et al., 2022). An alignment maps each turn of the transcript to either one line of the minute's file in which it is summarized, a "problem" label, both or neither. The alignments are done in such a way that whole discussions are aligned to the minutes lines (e.g., speaker A agreeing to a statement by speaker B is aligned to the same minutes line as speaker B's original statement; see Figure 3 for an example of an alignment in Appendix A).

There will be no manual evaluation for Tasks B, C, and D.

5.2 Automatic Evaluation

For our automatic evaluation of Task A, we will still rely on the widely popular text summarization metric ROUGE (Lin, 2004) in its three variants: ROUGE-1, ROUGE-2, ROUGE-L. Additionally, we will use BERTScore (Zhang et al., 2019) and/or BARTScore (Yuan et al., 2021) which are currently being used to evaluate generation tasks.

For Tasks B and C, which are actually classification tasks, we will use accuracy and class-wise F1 scores.

Task D will not be evaluated by a single criterion. As mentioned above, all submissions to Task D will be evaluated in terms of Pearson correlation against all manual and all other automatic evaluation scores.

Figure 1 plots the heatmaps of Pearson correlations between various types of evaluations of minutes. For automatic scores (ROUGE variants), we utilize multiple reference minutes, where available, and average or maximize over them. For manual scores (Adequacy, Correctness, and Fluency), we average or maximize the score assigned by two annotators to get a single score for a given minute.

We see that all ROUGE score types are significantly correlated with each other but not much related to the manual scores. The highest correlation is between ROUGE 1 and Adequacy in the (b) plot in Figure 1, reaching 0.66, which is approximately the same level of correlation as between Correctness and Fluency. Any variants of the automatic score do not reflect Correctness and Fluency. Figure 5 in Appendix C shows one of the good minutes generated by a participating team in the First AutoMin shared task.

6 Baseline Evaluations

We provide our participants with the baseline codes for automatic minuting (Task A) here.[5] The details of the experiments are described in Singh et al. (2021). It includes initial exploration using *off-the-shelf* text summarization models for future investigations. For generating abstractive meeting minutes we use BART (Lewis et al., 2019), BERTSUM (Liu and Lapata, 2019), BERT2BERT (Rothe et al., 2020), LED (Beltagy et al., 2020), Pegasus (Zhang et al., 2020), Roberta2Roberta (Liu et al., 2019), and T5 (Raffel et al., 2019) models. For extractive meeting summaries we use *TF-IDF*-based summarizer (Christian et al., 2016), an unsupervised extractive summarizer, TextRank (Mihalcea and Tarau, 2004), LexRank (Erkan and Radev, 2004), Luhn Algorithm (Luhn, 1958), and LSA (Gong and Liu, 2001) based summarizer. These off-the-shelf text summarization models are not the best candidates for generating minutes which calls for further research on this challenging task for meeting-specific summarization or minuting models.

7 AutoMin 2023 Procedure and Timeline

Table 3 summarizes the tentative timeline for AutoMin 2023. We would create and host a server to handle the shared task system submissions. We would use START or EasyChair for paper submissions and reviewing. We would also set up a program committee to review the system submissions and help the authors improve their reports.

July 2022	Announcement at INLG 2022
August 2022	Call for Participation
September 2022	Training Data Release
December 2023	Test Data Release
February 2023	System Submission
March 2023	Evaluation Notification
April 2023	System Report Submission
May 2023	System Report Review Notification
May 2023	Camera-ready Submission
June 2023	Proceedings appear in ACL Anthology
July 2023	Second AutoMin at INLG 2023

Table 3: Tentative Timeline for second AutoMin at INLG 2023 (may change depending on INLG 2023 schedule)

8 Diversity and Inclusion

As our commitment to diversity and inclusion, like the previous iteration, we would like to make our event *open-to-all* (and possibly hybrid) in consultation with INLG 2023 chairs. We would especially reach out to organizations like Widening NLP[6] (where our first organizer is also a chair) to help us reach the underrepresented groups and communities and encourage them to participate. We would also look for funding from industries/labs interested in the application of this research to sponsor resources (especially compute) and/or travel/registration of our participants in need of those logistics.

9 Conclusion

AutoMin is a very timely yet complex task for the speech and natural language processing community. Given the array of problems this task had to offer, we are very excited to continue this iteration of the generation challenge at INLG 2023. We look forward to uncovering the several linguistic phenomena and insights that should go into action while a machine writes a minute and see how much we have progressed towards an acceptable automated minuting output. In that essence, Task A and Task D are of more interest to the NLG and summarization community than Task B and Task C.

10 Ethical Considerations

For our ELITR Minuting Corpus, all meeting participants consented to make the data publicly available. Please refer to Nedoluzhko et al. (2022); Ghosal et al. (2021a) for a detailed description of our de-identification and participant-consent process. We would follow the same conditions to prepare the hidden test set. The EuroParl (Koehn, 2005) data, as well as the minutes for those parliamentary sessions, is publicly available (on the EuroParl website). Hence, there should not be any privacy or ethical issues.

Acknowledgement

The shared task would receive funding from the grant 19-26934X (NEUREM3) of the Czech Science Foundation.

[5] https://github.com/ELITR/Minuting_Baseline_Experiments

[6] http://www.winlp.org

References

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Saprativa Bhattacharjee, Kartik Shinde, Tirthankar Ghosal, and Asif Ekbal. 2022. A multi-task learning approach for summarization of dialogues. In *Proceedings of the 15th International Conference on Natural Language Generation*, page TBA, Maine, US. Association for Computational Linguistics.

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021. DialogSum: A real-life scenario dialogue summarization dataset. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5062–5074, Online. Association for Computational Linguistics.

Hans Christian, Mikhael Pramodana Agus, and Derwin Suhartono. 2016. Single document automatic text summarization using term frequency-inverse document frequency (tf-idf). *ComTech: Computer, Mathematics and Engineering Applications*, 7(4):285–294.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

G. Fauville, M. Luo, A.C.M. Queiroz, J.N. Bailenson, and J. Hancock. 2021. Zoom exhaustion & fatigue scale. *Computers in Human Behavior Reports*, 4:100119.

Tirthankar Ghosal, Ondřej Bojar, Muskaan Singh, and Anja Nedoluzhko. 2021a. Overview of the first shared task on automatic minuting (automin) at interspeech 2021. In *Proceedings of the First Shared Task on Automatic Minuting at Interspeech 2021*, pages 1–25.

Tirthankar Ghosal, Muskaan Singh, Anna Nedoluzhko, and Ondřej Bojar. 2021b. Report on the sigdial 2021 special session on summarization of dialogues and multi-party meetings (summdial). *ACM SIGIR Forum*, December 2021:1–17.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

Yihong Gong and Xin Liu. 2001. Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–25.

Adam Janin, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke, and Chuck Wooters. 2003. The icsi meeting corpus. pages 364–367.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MTSUMMIT*.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. *arXiv preprint arXiv:1908.08345*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165.

I. Mccowan, G. Lathoud, M. Lincoln, A. Lisowska, W. Post, D. Reidsma, and P. Wellner. 2005. The ami meeting corpus. In *In: Proceedings Measuring Behavior 2005, 5th International Conference on Methods and Techniques in Behavioral Research. L.P.J.J. Noldus, F. Grieco, L.W.S. Loijens and P.H. Zimmerman (Eds.), Wageningen: Noldus Information Technology*.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Anna Nedoluzhko and Ondrej Bojar. 2019. Towards automatic minuting of the meetings. In *ITAT*.

Anna Nedoluzhko, Muskaan Singh, Marie Hledíková, Tirthankar Ghosal, and Ondřej Bojar. 2022. Elitr minuting corpus: A novel dataset for automatic minuting from multi-party meetings in english and czech. In *Proceedings of the Language Resources and Evaluation Conference*, pages 3174–3182, Marseille, France. European Language Resources Association.

Peter Polák, Muskaan Singh, Anna Nedoluzhko, and Ondřej Bojar. 2022. Alignmeet: A comprehensive tool for meeting annotation, alignment, and evaluation. In *Proceedings of The 13th Language Resources and Evaluation Conference*, page To Appear.

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. *arXiv preprint arXiv:1910.10683*.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*, 8:264–280.

Kartik Shinde, Nidhir Bhavsar, Aakash Bhatnagar, and Tirthankar Ghosal. 2021. Team ABC @ AutoMin 2021: Generating Readable Minutes with a BART-based Automatic Minuting Approach. In *Proc. First Shared Task on Automatic Minuting at Interspeech 2021*, pages 26–33.

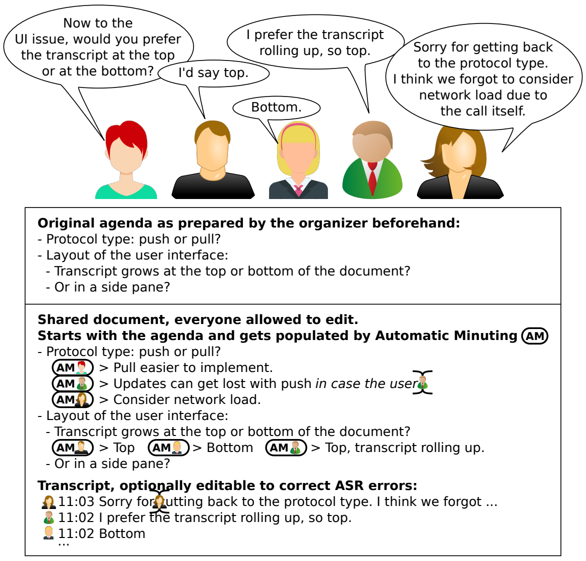
Muskaan Singh, Tirthankar Ghosal, and Ondrej Bojar. 2021. An empirical performance analysis of state-of-the-art summarization models for automatic minuting. In *Proceedings of the 35th Pacific Asia Conference on Language, Information and Computation*, pages 50–60, Shanghai, China. Association for Computational Linguistics.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems*, volume 34, pages 27263–27277. Curran Associates, Inc.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

A Appendix

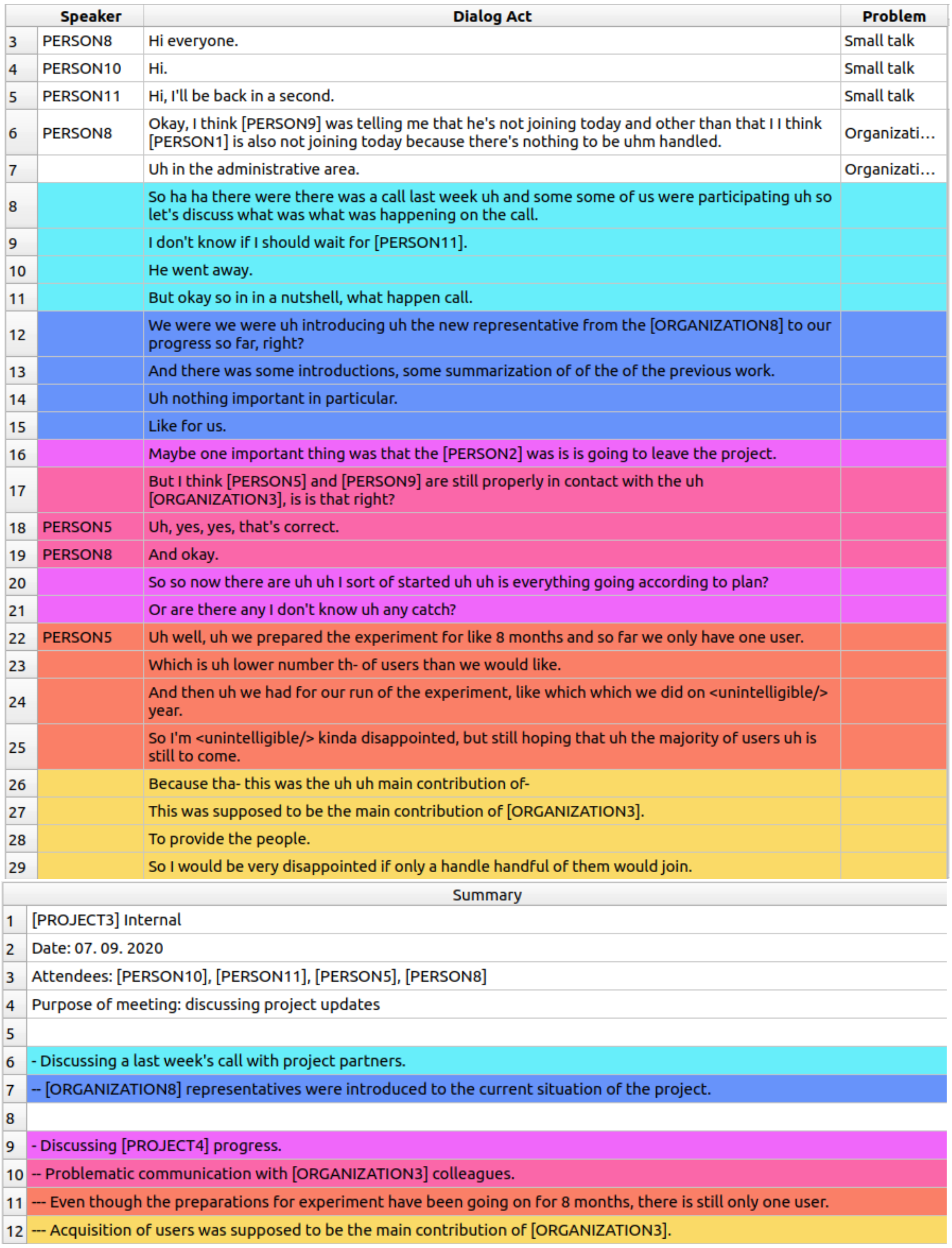

Now to the UI issue, would you prefer the transcript at the top or at the bottom?
I'd say top.
Bottom.
I prefer the transcript rolling up, so top.
Sorry for getting back to the protocol type. I think we forgot to consider network load due to the call itself.

Original agenda as prepared by the organizer beforehand:
- Protocol type: push or pull?
- Layout of the user interface:
  - Transcript grows at the top or bottom of the document?
  - Or in a side pane?

Shared document, everyone allowed to edit.
Starts with the agenda and gets populated by Automatic Minuting AM
- Protocol type: push or pull?
  AM > Pull easier to implement.
  AM > Updates can get lost with push *in case the user*
  AM > Consider network load.
- Layout of the user interface:
  - Transcript grows at the top or bottom of the document?
  AM > Top   AM > Bottom   AM > Top, transcript rolling up.
  - Or in a side pane?

Transcript, optionally editable to correct ASR errors:
11:03 Sorry for utting back to the protocol type. I think we forgot ...
11:02 I prefer the transcript rolling up, so top.
11:02 Bottom
...

Figure 2: Envisioning Automatic Minuting



	Speaker	Dialog Act	Problem
3	PERSON8	Hi everyone.	Small talk
4	PERSON10	Hi.	Small talk
5	PERSON11	Hi, I'll be back in a second.	Small talk
6	PERSON8	Okay, I think [PERSON9] was telling me that he's not joining today and other than that I I think [PERSON1] is also not joining today because there's nothing to be uhm handled.	Organizati…
7		Uh in the administrative area.	Organizati…
8		So ha ha there were there was a call last week uh and some some of us were participating uh so let's discuss what was what was happening on the call.	
9		I don't know if I should wait for [PERSON11].	
10		He went away.	
11		But okay so in in a nutshell, what happen call.	
12		We were we were uh introducing uh the new representative from the [ORGANIZATION8] to our progress so far, right?	
13		And there was some introductions, some summarization of of the of the previous work.	
14		Uh nothing important in particular.	
15		Like for us.	
16		Maybe one important thing was that the [PERSON2] was is is going to leave the project.	
17		But I think [PERSON5] and [PERSON9] are still properly in contact with the uh [ORGANIZATION3], is is that right?	
18	PERSON5	Uh, yes, yes, that's correct.	
19	PERSON8	And okay.	
20		So so now there are uh uh I sort of started uh uh is everything going according to plan?	
21		Or are there any I don't know uh any catch?	
22	PERSON5	Uh well, uh we prepared the experiment for like 8 months and so far we only have one user.	
23		Which is uh lower number th- of users than we would like.	
24		And then uh we had for our run of the experiment, like which which we did on <unintelligible/> year.	
25		So I'm <unintelligible/> kinda disappointed, but still hoping that uh the majority of users uh is still to come.	
26		Because tha- this was the uh uh main contribution of-	
27		This was supposed to be the main contribution of [ORGANIZATION3].	
28		To provide the people.	
29		So I would be very disappointed if only a handle handful of them would join.	

	Summary
1	[PROJECT3] Internal
2	Date: 07. 09. 2020
3	Attendees: [PERSON10], [PERSON11], [PERSON5], [PERSON8]
4	Purpose of meeting: discussing project updates
5	
6	- Discussing a last week's call with project partners.
7	-- [ORGANIZATION8] representatives were introduced to the current situation of the project.
8	
9	- Discussing [PROJECT4] progress.
10	-- Problematic communication with [ORGANIZATION3] colleagues.
11	--- Even though the preparations for experiment have been going on for 8 months, there is still only one user.
12	--- Acquisition of users was supposed to be the main contribution of [ORGANIZATION3].

Figure 3: Example of an alignment viewed in ALIGNMEET. Dialogue Acts with white background are not aligned to minutes, other colors indicate alignment to minutes line of the same color. Problems are shown in the right column of the transcript view.

B Sample Reference Minutes in ELITR Minuting Corpus

```
Date: 2019/04/01
Attendees: [PERSON10], [PERSON2], [PERSON3], [PERSON7], [PERSON11], [PERSON8], [
    PERSON1]
Purpose of meeting: Technical prepare for [ORGANIZATION6] congress

Agenda:
- Start recording.
- Date for [PROJECT1] call.
- Collecting photos and videos from Trade Fair.
- Confirmation of proposed scheme of wiring for [ORGANIZATION6] Congress.
- Digital interface to audio mix pult.
- Microphones.
- Get a contact for someone from [ORGANIZATION4], who will handle the presentation
    platform.
- Will [ORGANIZATION4] also try get their ASR.
- When will the python version of [ORGANIZATION4] platform sample connector.

Summary of meeting:

[PERSON3], [PERSON7]:
- After reminder missing vote for [PROJECT1] call date was chosen the April 16th.

[PERSON3], [PERSON7]:
- Ask for photos from the trade fair. Will be sent to e-mail immediately.

[PERSON3], [PERSON7], [PERSON11]:
- It is needed to specify the settings for workshop in June and [ORGANIZATION6]
    congress.
The hardware will provide outside company.
It is supposed to translating and transcribing the main session.
There will be rented tablets and is supposed that everyone will have their cell
    phones.
It is needed to connect the microphones to the mean audio mixer and then to have
    digital output to the booth for listening and ASR.
Any of the separate notebooks after the ASR can provide input to the multilingual
    translation system.
Proposal that every input language has uhm have to have its own ehm session with the
    mediator, this will be implemented by [PERSON2].
It is needed original sound from the microphones as possible from booth main
    microphone of the plenary session, ideally the digital signal captured at
    microphone.
Languages: English, German, Czech, French, Italian, Spanish, Russian.
There is experience only with Dante, but it is very expensive and doesn't simplify
    setting.
It is needed one PC for each language, one PC per input channel.
It is recomended to keep audio data and network traffic separated.
Will be demand one direct microphone output from the main microphone.
And one direct microphone output from each of the booths and for these booth
    microphones we demand that only the predefined languages is spoken at that
    channel.
Proposal to say get booth analog output as a call back and digital interface scholar
    choice.
[ORGANIZATION4] will let know what digital audio should be specify in the
    documentation until Tuesday.

[PERSON3], [PERSON11], [PERSON7]:
- It is needed to demand also Microphones.
Ask for definition all the individual microphones that the speakers will use.
After discussion they agreed that there will be preferred wired microphone for main
    stage.
Until Tuesday [PERSON7] will provide specification for main stage wired microphones
    and interpreters booths large microphones and also for wireless.

[PERSON3], [PERSON7], [PERSON11]:
- Presentation platform will have to be different for the workshop in June and for
    the [ORGANIZATION6] congress, because the setting is different.
Explain idea.
[PERSON2] will be coding this thing.
```

```
[PERSON3], [PERSON7]:
- [ORGANIZATION4] won't try their own ASR.

[PERSON3], [PERSON7]:
- Ask when the python connector to the [ORGANIZATION4] platform would be ready.
People using python at the [ORGANIZATION8] will help with this point.
It will be published at public website.

Minutes submitted by: [ANNOTATOR1]
```

Figure 4: A sample minute taken by our external annotators

C Sample Candidate Minutes by one participating team in AutoMin 2021

```
DATE : 2021-07-16
ATTENDEES : PERSON5, PERSON15, PERSON1, PERSON13, PERSON9, PERSON6, PERSON16


SUMMARY-

  The Czech Republic government has lifted the rules.
 -People can go out even if they don't need to, but they have to wait until the 4th
    of June for the free circulation of people.
  They can go to the forest, but if you are in PERSON6, PERSON5, PERSON1, PERSON3,
     PERSON15, PERSON16 and PERSON12 are going to do the summarization and three-
     point-one review.
 -PERSON6, PERSON5, PERSON8, PERSON2, PERSON1 and ORGANIZATION6 are writing a
    project management guide for a party.
  There is no description of the deliverable and there are no project management
     guides.
 -PERSON5, PERSON1, PERSON6 and PERSON4 are working on the EU projects.
  They need to finish the internal reviews by mid June at the latest.
  They have two weeks to finish it and then they have a week to fix it.
  There is one more milestone, the PERSON6 wants to have the PROJECT1 test sets
     populated and described by August so they can be ready to submit as a
     deliverable.
 -PERSON10 is not feeding the annotators with the prepared files.
  The annotators are searching for poll documents and in many of the languages.
  They need more people to be added to the language map.
 -PERSON6, PERSON1 and PERSON9 agree that the public use of the test sets should be
    limited to few of them.
  They also agree that there should be only 3 file lists for the general public.
 -PERSON1, PERSON9, PERSON6, PERSON16 and PERSON9 are discussing the implementation
    of the SLTF.
  According to PERSON6, the only reliable way to do the comparison is to run the
     models or a serve the model.
 -People can misinterpret the time stamps and the forced alignment is not reliable
    for them.
 -PERSON6 and PERSON1 are doing both finding and curating the translations and
    translating them into Czech.
  They made progress in getting translations out of the auditing websites.
 -PERSON1, PERSON15, PERSON6, PERSON7, PERSON5, PERSON11 and PERSON16 are working on
     a project.
  The project was started when the EU still existed.
  There are ten tens of thousands of sentences.
  Irish is equally important to the project as other languages.
 -PERSON1, PERSON9 and PERSON6 are discussing ASR's retranslation policy.
  They discuss the pros and cons of retranslating.
  There is no internal SLT in the endtoend ASR.
  The MT only translate will be get from ASR hypothesis.
  There is research going on how to integrate the ASR and MT.
 -PERSON6 is trying to run GPT tool to predict the tail of the sentence.
  The interpreters can guess up to 90% of the time, but sometimes they get it wrong.
  There is no way to touch up on these topics before the PERSON16 will create a
     Doodle, send it to both partners and ask them what they would like to demo.
  The demo should include both the ORGANIZATION1 representation and the sub-
     representation with subtitles.
 -PERSON1, PERSON6, PERSON13 and PERSON9 discuss screenshare and how to improve the
    quality of the machine translation.
 -PERSON1 thinks the idea screenshare is a good one, but it takes away one indicate.
 -PERSON6 is sorry for not managing the half an hour for the demo in the coming days
    .


Minuted by: Team ABC
```

Figure 5: A sample minute from Team ABC (Shinde et al., 2021) in AutoMin 2021 (Ghosal et al., 2021a)

The Cross-lingual Conversation Summarization Challenge

Yulong Chen[*♠♡], **Ming Zhong**[*♣], **Xuefeng Bai**[*♠♡], **Naihao Deng**[‡],
Jing Li[◇], **Xianchao Zhu**[◇], **Yue Zhang**[♡]
[♠] Zhejiang University
[♡] School of Engineering, Westlake University
[♣] University of Illinois at Urbana-Champaign
[‡] University of Michigan, Ann Arbor
[◇] Sichuan Lan-bridge Information Technology Co., Ltd.
yulongchen1010@gmail.com *yue.zhang@wias.org.cn*

Abstract

We propose the shared task of cross-lingual conversation summarization, *ConvSumX Challenge*, opening new avenues for researchers to investigate solutions that integrate conversation summarization and machine translation. This task can be particularly useful due to the emergence of online meetings and conferences. We use a new benchmark, covering 2 real-world scenarios and 3 language directions, including a low-resource language, for evaluation. We hope that *ConvSumX* can motivate research to go beyond English and break the barrier for non-English speakers to benefit from recent advances of conversation summarization.

1 Task Overview

The cross-lingual conversation summarization (*ConvSumX*) task [1] asks models to output a salient, concise and coherent summary in target languages (*e.g.*, Chinese), given a conversation in a source language (*e.g.*, English). In particular, *ConvSumX* contains 2 tracks: daily dialogue summarization and query-based meeting minute. Each covers 3 language directions: `English-to-Chinese` (`En2Zh`), `English-to-French` (`En2Fr`) and `English-to-Ukrainian` (`En2Uk`). Figure 1 gives examples in *ConvSumX*, where we show summaries in 4 languages (including English). Both automatic and manual evaluations are used to measure the model performance, while the evaluation is highly inclined to human evaluation (Section 3.5).

2 Motivation

Thanks to the availability of large-scale corpora (Gliwa et al., 2019; Chen et al., 2021a; Zhong et al., 2021b), research on conversation summarization has made great progress (Zhong et al., 2021a; Ni et al., 2021; Ghazvininejad et al., 2021; Lin et al., 2022). However, existing corpora in this area focus on English while ignoring other languages (Feng et al., 2021a). Such English-dominated corpora lead to a barrier for non-English speakers to benefit from conversation summarization research, which becomes more severe in the era of epidemic, where international meetings are held online and participants communicate in English.

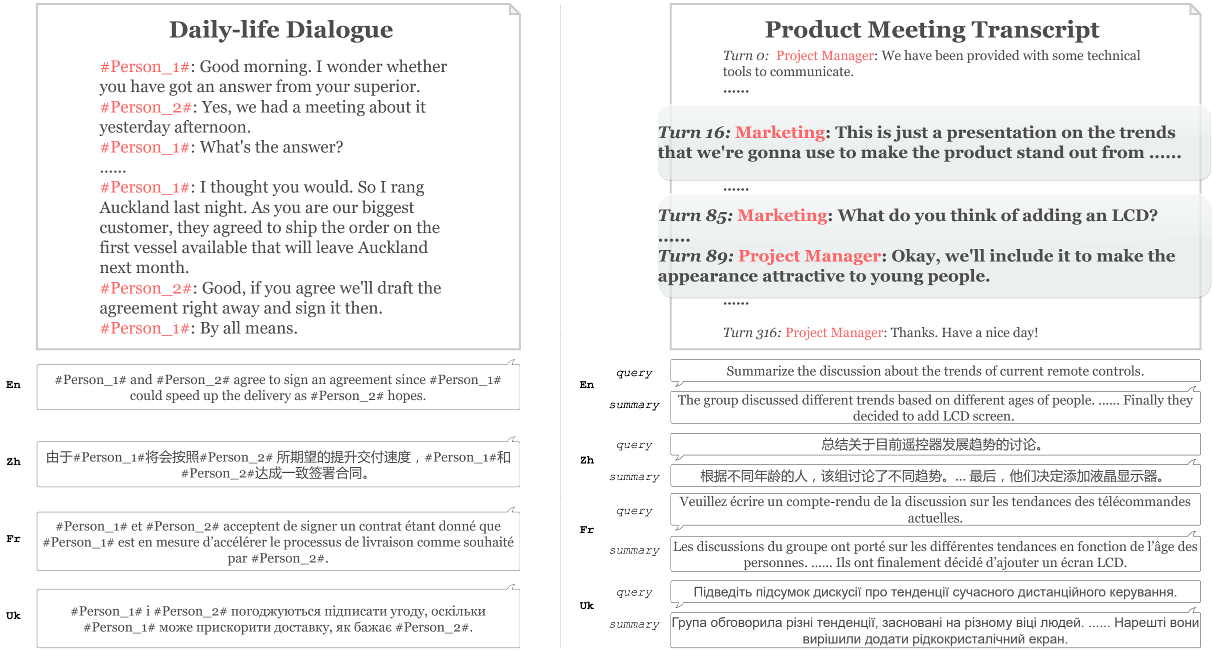
ConvSumX integrates conversation summarization and machine translation, involving the language shift from one to another and stylistic shift from long spoken conversations to concise written monologues. Ideally, using the *first translate, then summarize* and *vice versa* pipelines can solve the task. However, besides the difficulties in monolingual conversation summarization (Chen et al., 2021b; Feng et al., 2021b), pipeline methods suffer from problems caused by machine translation systems. For *translation-first* systems, Zhu et al. (2019) find that machine translation introduces errors for summarizers on news text. In addition, existing machine translation systems show poor performance on conversation text (Wang et al., 2021). For *summarization-first* systems, translating summaries without conversation context can lead to inconsistent translation, in particular for polysemous words. Take $\texttt{C}^{\texttt{En}}\texttt{2S}^{\texttt{Zh}}$ [2] for example. The summary "*Bob is going to the bank.*", where "*bank*" can be translated into "岸边" (river bank) or "银行" (financial bank), requires models to determine the proper translation by considering conversation context. Such issues can be also found in end-to-end systems developed for cross-lingual news summarization and directly using those methods can lead to error propagation (Zhu et al., 2019; Xu et al., 2020; Liang et al., 2022). Thus, more sophisticated designs that take care of conversation natures or data selection strategies that can make better use of silver data are in need.

[*] Equal Contribution.

[1] https://cylnlp.github.io/convsumx-challenge/

[2] The setting means the input conversation text is in English, and the output summary is in Chinese.

Proceedings of the 15th International Conference on Natural Language Generation: Generation Challenges, pages 12 - 18
July 17-22, 2022 ©2022 Association for Computational Linguistics



#Person_1#: Good morning. I wonder whether you have got an answer from your superior.
#Person_2#: Yes, we had a meeting about it yesterday afternoon.
#Person_1#: What's the answer?
......
#Person_1#: I thought you would. So I rang Auckland last night. As you are our biggest customer, they agreed to ship the order on the first vessel available that will leave Auckland next month.
#Person_2#: Good, if you agree we'll draft the agreement right away and sign it then.
#Person_1#: By all means.

#Person_1# and #Person_2# agree to sign an agreement since #Person_1# could speed up the delivery as #Person_2# hopes.

由于#Person_1#将会按照#Person_2# 所期望的提升交付速度，#Person_1#和#Person_2#达成一致签署合同。

#Person_1# et #Person_2# acceptent de signer un contrat étant donné que #Person_1# est en mesure d’accélérer le processus de livraison comme souhaité par #Person_2#.

#Person_1# i #Person_2# погоджуються підписати угоду, оскільки #Person_1# може прискорити доставку, як бажає #Person_2#.

(a)

Turn 0: Project Manager: We have been provided with some technical tools to communicate.
......
***Turn 16:* Marketing: This is just a presentation on the trends that we're gonna use to make the product stand out from ......**
......
***Turn 85:* Marketing: What do you think of adding an LCD?**
......
***Turn 89:* Project Manager: Okay, we'll include it to make the appearance attractive to young people.**
......
Turn 316: Project Manager: Thanks. Have a nice day!

Summarize the discussion about the trends of current remote controls.

The group discussed different trends based on different ages of people. ...... Finally they decided to add LCD screen.

总结关于目前遥控器发展趋势的讨论。

根据不同年龄的人，该组讨论了不同趋势。... 最后，他们决定添加液晶显示器。

Veuillez écrire un compte-rendu de la discussion sur les tendances des télécommandes actuelles.

Les discussions du groupe ont porté sur les différentes tendances en fonction de l’âge des personnes. ...... Ils ont finalement décidé d’ajouter un écran LCD.

Підведіть підсумок дискусії про тенденції сучасного дистанційного керування.

Група обговорила різні тенденції, засновані на різному віці людей. ...... Нарешті вони вирішили додати рідкокристалічний екран.

(b)

Figure 1: Examples of *ConvSumX*: (a) daily-life dialogue summarization and; (b) query-based meeting minute. From top to bottom, the languages are English (`En`), Chinese (`Zh`), French (`Fr`) and Ukrainian (`Uk`). The English conversations and summaries (and queries) are from DIALOGSUM and QMSum, respectively.

To this end, *ConvSumX Challenge* encourages researchers to investigate different solutions to cross-lingual conversation summarization. First, from the perspective of downstream applications, *ConvSumX* is useful for both business and personal uses. Second, from the perspective of research, *ConvSumX Challenge* looks for a general method that can deal with cross-lingual conversation summarization. Although only 3 typical target languages are presented in this shared task, we hope that *ConvSumX* can motivate research to a broader range of target languages. Third, from the perspective of social good, *ConvSumX* aims to break the barrier of accessing information for non-English speakers and to make them benefit from the advance of conventional English-dominated conversation summarization technologies.

We hope that *ConvSumX Challenge* can gain interest from both communities of text summarization and machine translation and also push the progress on related fields for other languages, including more low-resource languages.

3 Task Description

Formally, the task of *ConvSumX Challenge* asks participants to provide a system that can output a summary in a target language given the input conversation text in a source language.

3.1 Setting

The *ConvSumX Challenge* focuses on the low-resource/few-shot setting and cross-lingual/domain transfer technologies. The low-resource/few-shot here is stated from the perspective of *lacking large gold training data*. The term *gold data* refers to cross-lingual ⟨`conversation-summary`⟩ pairs that are annotated by translators who are expert at both source and target languages.

The reasons are: (1) gold data are limited as the annotation is very costly, in particular when conversations involve domain expert knowledge (*e.g.*, academic meeting). In contrast, machine translation and monolingual summarization data are abundant and useful (Perez-Beltrachini and Lapata, 2021); (2) we seek for a general solution that can be applied to not only the target languages in this paper, but also other languages. However, for practical consideration, we provide large silver data (Section 3.3.2). We also encourage participants to make use of other external resources to solve the task.

The above setting is widely adopted by existing cross-lingual summarization datasets in other domains, such as the first large-scale cross-lingual summarization corpus, NCLS dataset (Zhu et al., 2019) and its succeeding works (Xu et al., 2020; Bai et al., 2021; Liang et al., 2022).

Track	Data Source	Domain	Query	# Conv.	# Summ.	Train/Dev/Test
Track 1	DIALOGSUM	Daily-life Dialogue	✗	131.0	13.8	12,460/500/500
Track 2	QMSum	Product Meeting	✓	6,007.7	70.5	690/145/151
		Academic Meeting	✓	13,317.3	52.7	259/54/56

Table 1: Statistics of *ConvSumX*. # Conv. and # Summ. are averaged token lengths of conversations and summaries.

3.2 Tracks

The *ConvSumX Challenge* consists of 2 tracks, focusing on different scenarios, respectively.

- **Track 1** focuses on cross-lingual summarization for real-life dialogues. This track is in line with the INLG 2021 *DialogSum Challenge* (Chen et al., 2021b) while we extend *DialogSum* into a cross-lingual setting. *ConvSumX* can be particularly useful in scenarios such as travelling abroad where summarizers can serve as personal assistants.

- **Track 2** focuses on cross-lingual meeting minutes. Compared with daily conversations, meetings are much longer and contain richer topic switches and more professional knowledge. Generating cross-lingual meeting minutes can help non-English speakers to quickly access information of their interest, especially in cases where conferences are mostly held in English. In particular, Track 2 asks a system to generate a summary in the target language, given an input meeting text in the source language and a query in the target language.

3.3 Data

3.3.1 Data Selection

The data of *ConvSumX* are derived from two public English datasets, namely DIALOGSUM (Chen et al., 2021a) and QMSum (Zhong et al., 2021b). Table 1 shows the statistics.

DIALOGSUM is a large-scale dialogue summarization dataset, consisting of face-to-face spoken dialogues that focus on real-life scenarios. In particular, DIALOGSUM provides multi-references for each test dialogue. We ask annotators to first choose the best reference summary and then annotate it into the target languages.

QMSum is a query-based meeting minute dataset, covering 3 domains, namely academic, product and committee. We choose academic meeting and product discussion meeting for annotation as they are more in line with our motivation.

3.3.2 Annotation

Each summary in the dev and test sets of DIALOGSUM and QMSum is annotated into 3 target languages by expert translators [3]. Note that the annotation is not the simple translation of summaries, instead, each annotation needs to take care of original English conversations to ensure that the annotated summary is consistent with the input (Section 2).

In addition to manually annotated dev and test sets, following Zhu et al. (2019), we construct silver training data using machine translation. In particular, we translate summaries in target languages using multiple engines, including Google translate [4], NiuTrans [5] and LanMT [6]. Besides, to provide resources for pipeline methods, we translate the conversation texts using the same methods. Note that we do not filter these silver data. Instead, we leave this issue as an open question for the participants.

3.4 Protocol

We propose the following schedule:

- **Phase 1** (from Jul, 2022): The shared task is announced at the INLG 2022 conference, and the data are available on the shared task website; participants can register to the task.

- **Phase 2** (from Dec, 2022): The leaderboard is open; participants can submit their systems to the organizers and the online leaderboard keeps updating the best performance on each track using automatic evaluation metrics.

- **Phase 3** (from Mar, 2023): The submission is closed; organizers conduct manual evaluation.

- **Phase 4** (Jun, 2023): The *ConvSumX Challenge* shared task is fully completed. Organizers submit participant reports and challenge reports to INLG 2023 and present at the conference. The hidden test set is made public.

[3] More information can be found in Appendix A.

[4] https://translate.google.com

[5] https://niutrans.com

[6] https://www.dtranx.com

In particular, in Phase 1 participants can train and validate summarization systems on their hardwares using data provided by the organizers. Participants are encouraged to use external resources to train their systems. Such resources include, but are not limited to: monologue summarization data, machine translation data, and other public or additional cross-lingual summarization data that are manually/automatically created by the participants. However, for fairness and reproducibility, participants should specify what and how external resources are used in their system reports. In Phase 3, after the submission deadline, the organizers will start to evaluate summaries generated by final submitted models with the help from linguistic experts. For fairness, the test set will not be publicly available during the shared task.

Please note that the above schedule can be modified accordingly when the schedule of INLG 2023 is released. The leaderboard and the detailed schedule will be announced on the shared task website at `https://cylnlp.github.io/convsumx-challenge/`.

3.5 Evaluation

The evaluation of the *ConvSumX Challenge* considers both automatic and manual evaluation metrics.

3.5.1 Automatic Evaluation

Following previous cross-lingual summarization work (Zhu et al., 2019), we use ROUGE scores (Lin, 2004) for automatic evaluation. ROUGE scores evaluate the model performance by considering the overlap of n-grams in the system-generated summary against the reference summary. Although recent works claim that ROUGE fails to measure important information regarding factual consistency (Zhang et al., 2020; Fabbri et al., 2021), we choose ROUGE because: (1) it directly reflects model's ability of obtaining salient information and; (2) it can be easily applied to multiple languages including low-resource languages.

3.5.2 Manual Evaluation

As neural summarizers mostly contain factual errors that cannot be easily detected by automatic metrics (Zhu et al., 2019; Fabbri et al., 2021) and translated words can be various (Freitag et al., 2021), automatic evaluation such as ROUGE can be less accurate. Thus, our evaluation highly relies on manual evaluation. Given that the *ConvSumX* integrates conversation summarization and machine translation, we adopt multiple human evaluation metrics from both tasks to better measure model performance.

In particular, standard summarization metrics include: Fluency, Consistency, Relevance and Coherence (Kryscinski et al., 2019); standard machine translation metrics include: Omission, Untranslation, Mistranslation, Addition and Terminology (Mariana, 2014). However, except for Fluency, summarization metrics evaluate generated summaries from the perspective of input documents in the same language while machine translation metrics evaluate translation from the perspective of source sentences (the English summary in our case). There can be an evaluation inconsistency between these two tasks. In addition, there is an overlap between these two groups of metrics. For example, a mistranslated summary can be regarded as containing consistency errors.

To unify the aforementioned evaluation metrics and obtain fine-grained evaluations, we propose to evaluate system-generated summaries from the following aspects against source conversation texts.

Fluency and Use of Language evaluates the quality of generated sentences, including the grammar and word order. Moreover, it evaluates whether the language in generated summaries is natural and conventional, e.g., the syntactic structure is not normal or the summary contains untranslated words.

Relevance evaluates the importance of information in the generated summary.

Factual and Translation Consistency evaluates the factual alignment of the generated summaries (target languages) against the source conversation (source languages), including information that is not presented in the conversation, wrong causal relation, etc. Moreover, for pipeline methods, if the final summary contains mis-translated words, we consider it inconsistency.

Terminology evaluates the use of language. For example, the generated word can be a right translation but is improper in certain domains (*e.g.,* academic meeting).

Overall score measures the overall quality for each summary.

For each metric above, we randomly extract 10% generated summaries and ask annotators who are native in the target languages to give scores from 1 to 5. The higher, the better.

4 Related Work

Zhu et al. (2019) propose the first large scale cross-lingual news summarization dataset, facilitating the study in this filed using neural network models. Bai et al. (2021) construct an `English-to-German` news summarization dataset using the automatic method of Zhu et al. (2019). Perez-Beltrachini and Lapata (2021) construct a cross-lingual dataset based on Wikipedia, focusing on European languages. In particular, Perez-Beltrachini and Lapata (2021) use the document and the lead paragraph in other languages aligned by Wikipedia inter-language links to construct cross-lingual ⟨`document-summary`⟩ pairs. Similarly, Ladhak et al. (2020) construct the WikiLingua dataset based on multi-lingual WikiHow.

Very recently, Wang et al. (2022) and Feng et al. (2022) construct cross-lingual dialogue summarization datasets. In particular, Wang et al. (2022) manually translate summaries from SAMSum (Gliwa et al., 2019), an online written dialogue summarization dataset, and 40k data in MediaSum (Zhu et al., 2021) into German and Chinese. Feng et al. (2022) construct MSAMSum by automatically translating SAMSum into Chinese, French and Russian. Compared with them, our work focuses on spoken conversation in multiple scenrios, and covers low-resource language (`Ukrainian`). In addition, we also focus on query-based meeting scenarios, which can be more useful in real-world applications.

Similarly to *ConvSumX Challenge*, Ghosal et al. (2021) propose a shared task, AutoMin, at Interspeech 2021. AutoMin focuses on monolingual meeting minutes in English and Czech. In contrast, we focus on the cross-lingual setting and consider more scenarios, domains and languages.

5 Conclusion

We propose the *ConvSumX Challenge* to address the task of cross-lingual conversation summarization, with the hope that *ConvSumX* can encourage researchers to investigate various methods for conversation summarization beyond English, in particular for low and mid-resource languages, and the frontier of cross-lingual conversation summarization can be pushed further.

Copyright and License of Datasets

The *ConvSumX Challenge* uses cross-lingual ⟨`conversation-summary`⟩ pairs that are annotated on the top of two English conversation summarization datasets, namely DIALOGSUM and QMSum, to evaluate models. Both DIALOGSUM and QMSum are free for academic use with the MIT license, which contains no limitation to use, modification or distribution. We will also make our annotated data available for the academia.

Acknowledgement

Yue Zhang is the corresponding author. We thank all reviewers for their insightful comments and Yang Liu, Da Yin, Qian Cao and Jianyu Wang for discussion and proofreading. This project receives a support from the Sichuan Lan-bridge Information Technology Co., Ltd. and China's Language Service Center (Sichuan Province). Our sincere appreciation goes to professional translators from Lan-bridge.

References

Yu Bai, Yang Gao, and Heyan Huang. 2021. Cross-lingual abstractive summarization with limited parallel resources. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6910–6924, Online. Association for Computational Linguistics.

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021a. DialogSum: A real-life scenario dialogue summarization dataset. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5062–5074, Online. Association for Computational Linguistics.

Yulong Chen, Yang Liu, and Yue Zhang. 2021b. DialogSum challenge: Summarizing real-life scenario dialogues. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 308–313, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating Summarization Evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Xiachong Feng, Xiaocheng Feng, and Bing Qin. 2021a. A survey on dialogue summarization: Recent advances and new frontiers. *arXiv preprint arXiv:2107.03175*.

Xiachong Feng, Xiaocheng Feng, and Bing Qin. 2022. Msamsum: Towards benchmarking multi-lingual dialogue summarization. In *Proceedings of the 2st Workshop on Document-grounded Dialogue and*

Conversational Question Answering (DialDoc 2022), Online. Association for Computational Linguistics.

Xiachong Feng, Xiaocheng Feng, Libo Qin, Bing Qin, and Ting Liu. 2021b. Language model as an annotator: Exploring dialogpt for dialogue summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 1479–1491. Association for Computational Linguistics.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, Errors, and Context: A Large-Scale Study of Human Evaluation for Machine Translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Marjan Ghazvininejad, Vladimir Karpukhin, and Asli Celikyilmaz. 2021. Discourse-aware prompt design for text generation. *arXiv preprint arXiv:2112.05717*.

Tirthankar Ghosal, Muskaan Singh, Ondřej Bojar, and Anja Nedoluzhko. 2021. Overview of the first shared task on automatic minuting (automin) at interspeech 2021. In *Proceedings of the First Shared Task on Automatic Minuting at Interspeech 2021*, page to appear.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. 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 540–551, Hong Kong, China. Association for Computational Linguistics.

Faisal Ladhak, Esin Durmus, Claire Cardie, and Kathleen McKeown. 2020. WikiLingua: A new benchmark dataset for cross-lingual abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4034–4048, Online. Association for Computational Linguistics.

Yunlong Liang, Fandong Meng, Chulun Zhou, Jinan Xu, Yufeng Chen, Jinsong Su, and Jie Zhou. 2022. A variational hierarchical model for neural cross-lingual summarization. *arXiv preprint arXiv:2203.03820*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Haitao Lin, Junnan Zhu, Lu Xiang, Yu Zhou, Jiajun Zhang, and Chengqing Zong. 2022. Other roles matter! enhancing role-oriented dialogue summarization via role interactions.

Valerie R Mariana. 2014. *The Multidimensional Quality Metric (MQM) framework: A new framework for translation quality assessment*. Brigham Young University.

Ansong Ni, Zhangir Azerbayev, Mutethia Mutuma, Troy Feng, Yusen Zhang, Tao Yu, Ahmed Hassan Awadallah, and Dragomir Radev. 2021. SummerTime: Text summarization toolkit for non-experts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 329–338, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Laura Perez-Beltrachini and Mirella Lapata. 2021. Models and datasets for cross-lingual summarisation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9408–9423, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jiaan Wang, Fandong Meng, Ziyao Lu, Duo Zheng, Zhixu Li, Jianfeng Qu, and Jie Zhou. 2022. Clidsum: A benchmark dataset for cross-lingual dialogue summarization. *arXiv preprint arXiv:2202.05599*.

Tao Wang, Chengqi Zhao, Mingxuan Wang, Lei Li, and Deyi Xiong. 2021. Autocorrect in the process of translation — multi-task learning improves dialogue machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*, pages 105–112, Online. Association for Computational Linguistics.

Ruochen Xu, Chenguang Zhu, Yu Shi, Michael Zeng, and Xuedong Huang. 2020. Mixed-lingual pre-training for cross-lingual summarization. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 536–541, Suzhou, 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.

Ming Zhong, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2021a. Dialoglm: Pre-trained model for long dialogue understanding and summarization. *arXiv preprint arXiv:2109.02492*.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021b. QMSum: A new benchmark for query-based multi-domain meeting summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5905–5921, Online. Association for Computational Linguistics.

Chenguang Zhu, Yang Liu, Jie Mei, and Michael Zeng. 2021. MediaSum: A large-scale media interview dataset for dialogue summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5927–5934, Online. Association for Computational Linguistics.

Junnan Zhu, Qian Wang, Yining Wang, Yu Zhou, Jiajun Zhang, Shaonan Wang, and Chengqing Zong. 2019. NCLS: Neural cross-lingual summarization. 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 3054–3064, Hong Kong, China. Association for Computational Linguistics.

A More Information about Annotation

The cross-lingual ⟨`conversation-summary`⟩ pairs used for *ConvsumX Challenge* are constructed by expert translators from the Sichuan Lan-bridge Information Technology which is recognized as a qualified institution for translation service [7] by the ISO [8]. The entire construction process involves 9 annotators, 3 editors and 1 project manager.

For each language direction (*e.g.,* `En2Zh`), we have 3 annotators and 1 editor. All summaries and queries are first annotated by annotators and then reviewed by an editor. If bad summaries are found by editors (*e.g.,* grammar and inconsistency errors or unnatural language), the annotator would re-annotate the batch until they are qualified.

All annotators/editors are native in the target language (*i.e.*, Chinese, French or Ukrainian), and professional in English. Annotators/editors have following competences:

- translation competence and;
- linguistic and textual competence in the source language and the target language and;
- competence in research, information acquisition and processing and;
- culture competence and;
- technical competence and;
- domain competence.

In addition, annotators/editors shall meet at least one of the following criteria:

- a recognized graduate qualification in translation from an institution of higher education or;
- a recognized graduate quallfication in any other field from an institution of higher education plus two years of full-time professional experience in translating.

To monitor the whole annotation process and conduct quality control, we invite a senior translator as the project manager. The manager, who also satisfies the above requirements, has more than 5-year experience in multi-lingual translation projects that cover the language directions as described in this paper.

[7] Requirements for translation services: https://www.iso.org/standard/59149.html

[8] International Organization for Standardization: https://www.iso.org/home.html.

HinglishEval Generation Challenge on Quality Estimation of Synthetic Code-Mixed Text: Overview and Results

Vivek Srivastava
TCS Research
Pune, Maharashtra, India
srivastava.vivek2@tcs.com

Mayank Singh
IIT Gandhinagar
Gandhinagar, Gujarat, India
singh.mayank@iitgn.ac.in

Abstract

We hosted a shared task to investigate the factors influencing the quality of the code-mixed text generation systems. The teams experimented with two systems that generate synthetic code-mixed Hinglish sentences. They also experimented with human ratings that evaluate the generation quality of the two systems. The first-of-its-kind, proposed subtasks, *(i) quality rating prediction* and *(ii) annotators' disagreement prediction* of the synthetic Hinglish dataset made the shared task quite popular among the multilingual research community. A total of 46 participants comprising 23 teams from 18 institutions registered for this shared task. The detailed description of the task and the leaderboard is available at https://codalab.lisn.upsaclay.fr/competitions/1688.

1 Introduction

Code-mixing is the phenomenon of mixing words and phrases from multiple languages in a single utterance of a text or speech. Figure 1 shows the example code-mixed Hinglish sentences generated from the corresponding parallel Hindi and English sentences. Code-mixed languages are prevalent in multilingual communities such as Spain, India, and China. With the inflation of social-media platforms in these communities, the availability of code-mixed data is seeking a boom. It has lead to several interesting research avenues for problems in computational linguistics such as language identification (Singh et al., 2018; Shekhar et al., 2020), machine translation (Dhar et al., 2018; Srivastava and Singh, 2020), language modeling (Pratapa et al., 2018), etc.

Over the years, we have observed various computational linguistic conferences and workshops organizing the shared tasks involving code-mixed languages. Diverse set of problems have been hosted such as sentiment analysis (Chakravarthi et al., 2021; Patwa et al., 2020), offensive language identification (Chakravarthi et al., 2021), word-level language identification (Solorio et al., 2014; Molina et al., 2016), information retrieval (Banerjee et al., 2016), etc.

Example I

HINGLISH: ye ek code mixed sentence ka example hai
ENGLISH : this is an example code-mixed sentence

Example II

HINGLISH : kal me movie dekhne ja raha hu. How are the reviews?
ENGLISH: I am going to watch the movie tomorrow. How are the reviews?

Figure 1: Example parallel Hinglish and English sentences. The code-mixed Hinglish sentences contain words from Hindi and English languages.

Despite these overwhelming attempts, the natural language generation (NLG) and evaluation of the code-mixed data remain understudied. The noisy and informal nature of the code-mixed text adds to the complexity of solving and evaluating the various NLG tasks such as summarization and machine translation. These inherent challenges (Srivastava and Singh, 2020) with the code-mixed data makes the widely popular evaluation metrics like BLEU and WER obsolete. Various metrics (e.g., CMI (Das and Gambäck, 2014; Gambäck and Das, 2016), M-index (Barnett et al., 2000), I-index (Guzmán et al., 2017), Burstiness (Goh and Barabási, 2008), Memory (Goh and Barabási, 2008), etc.) have been proposed to measure the complexity of code-mixed data, but they fail to capture the linguistic diversity which leads to poorly estimating the quality of code-mixed text (Srivastava and Singh, 2021a).

In this shared task[1] (see Section 2) for the de-

[1] https://sites.google.com/view/hinglisheval

Proceedings of the 15th International Conference on Natural Language Generation: Generation Challenges, pages 19 - 25
July 17-22, 2022 ©2022 Association for Computational Linguistics

tailed description), we look forward to the new strategies that cater to the broad requirement of the quality evaluation of the generated code-mixed text. These methods will entail various linguistic features encompassing syntax and semantics and the perspectives of human cognition such as writing style, emotion, sentiment, language, and preference. We also put forward a subtask to understand the factors influencing the human disagreement on the quality rating of the generated code-mixed text. This could help design a more robust quality evaluation system for the code-mixed data.

A total of 46 participants comprising 23 teams from 18 institutions registered for this shared task. Out of which four teams have submitted their final reports. Section 3 presents the overview of the participants and submission methodology. Section 4 compares the four submissions and presents discussions around the similarity and differences of the approaches. We conclude and present future directions in Section 5.

2 The HinglishEval Shared Task

In this shared task, we propose two subtasks to evaluate the quality of the code-mixed Hinglish text. First, we propose to predict the quality of Hinglish text on a scale of 1–10. We aim to identify the factors influencing the text's quality, which will help build high-quality code-mixed text generation systems. We synthetically generate the Hinglish sentences using two different approaches (see Section 2.1) leveraging popular English-Hindi parallel corpus. Besides, we also have at least two human-generated Hinglish sentences corresponding to each parallel sentence. The second subtask aims to predict the disagreement on a scale of 0–9 between the two annotators who have annotated the synthetically generated Hinglish sentences. Various factors influence this human disagreement, and we seek to investigate the reasoning behind this behavior.

2.1 Dataset

As outlined in Section 1, the code-mixed NLG task observes a scarcity of high-quality datasets. Consequently, the quality evaluation of the generated code-mixed text remains unexplored. To address this challenge, we propose a new dataset with Hinglish sentences generated synthetically and rated by human annotators. We create the dataset in two phases.

Phase 1: Human-generated Hinglish sentences: We select the English-Hindi parallel sentences from the IIT-B parallel corpus (Kunchukuttan et al., 2018) to generate the Hinglish sentences. The parallel corpus has 1,561,840 sentence pairs. We randomly select 5,000 sentence pairs in which the number of tokens in both the sentences is more than five. We employ five human annotators and assign each 1,000 sentence pairs. Table 1 shows the annotation guidelines to generate the Hinglish sentences. Post annotation, we obtain 1,976 sentence pairs for which the annotators have generated at least two Hinglish sentences.

Phase 2: Synthetic Hinglish sentence generation and quality evaluation: We synthetically generate the Hinglish sentence corresponding to each of the parallel 1,976 English-Hindi sentence pairs. We employ two different code-mixed text generation (CMTG) techniques:

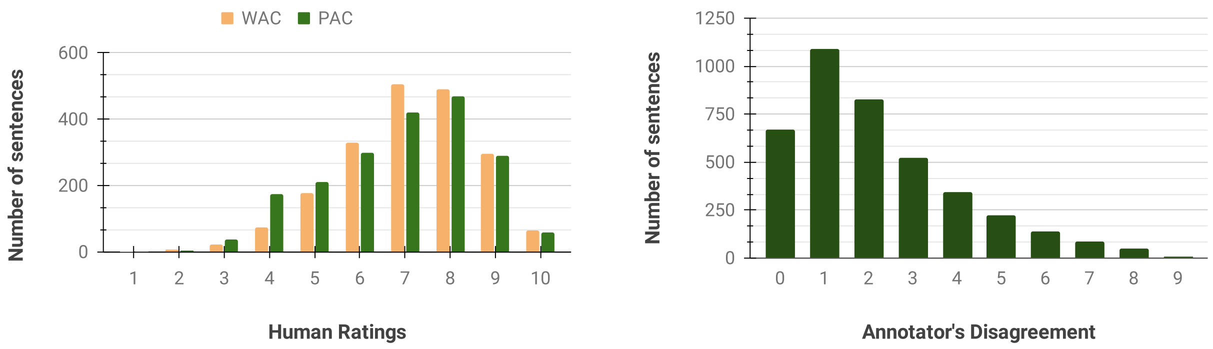
- Word-aligned CMTG (WAC): Here, we align the noun and adjective tokens between the parallel sentences. We replace the aligned Hindi token with the corresponding English token and transliterate the Hindi sentence to the Roman script.
- Phrase-aligned CMTG (PAC): Here, we align the key-phrases of length up to three tokens between the parallel sentences. We replace the aligned Hindi phrase with the corresponding English phrase and transliterate the Hindi sentence to the Roman script.

For the token alignment between parallel sentences, we use the online curated dictionaries, GIZA++ (Och and Ney, 2003) trained on the remaining IIT-B corpus, and cross-lingual word embedding trained on English and Hindi word vectors from FastText (Bojanowski et al., 2017). We employ eight human annotators[2] to provide a rating between 1 (low quality) to 10 (high quality) to the generated Hinglish sentences. Table 1 shows the annotation guidelines to rate the sentences. Figure 2a and 2b shows the distribution of the annotators' rating and their disagreement, respectively.

Data format: Table 2 shows an instance from the dataset. In total, we have 3,952 instances[3] in the dataset where each data instance i for subtask-1 (see Section 2.2.1) is represented as $\mathbf{X1}_i$={Eng$_i$, Hin$_i$, Synthetic_Hing$_i$} and $\mathbf{Y1}_i$={Average_rating$_i$}.

[2] Different from the annotators in Phase 1. Each annotator gets 247 sentences generated by PAC and WAC, each corresponding to the same set of parallel sentences.

[3] Two synthetic Hinglish sentences are generated for each parallel sentence pair.



(a) (b)

Figure 2: Distribution of (a) human evaluation scores and (b) disagreement in human scores in the synthetically generated Hinglish sentences.

Task	Guidelines
Hinglish text generation	1. The Hinglish sentence should be written in Roman script. 2. The Hinglish sentence should have words from both the source languages. 3. Avoid using new words, wherever possible, that are not present in both sentences. 4. If the source sentences are not the translation of each other, mark the sentence pair as "#".
Quality rating	The rating depends on the following three factors: 1. The similarity between the generated Hinglish sentence and the source sentences. 2. The readability of the generated sentence. 3. The grammatical correctness of the generated sentence.

Table 1: Annotation guidelines to the annotators for the two different tasks.

For subtask-2 (see Section 2.2.2), the instance j is represented as $\mathbf{X2}_j$={Eng$_j$, Hin$_j$, Synthetic_Hing$_j$} and $\mathbf{Y2}_j$={Annotator_disagreement$_j$}. In addition, we provide at least two human generated Hinglish sentences corresponding to each data instance for both the subtasks. We shuffle and split the dataset in the ratio 70:10:20 with 2766, 395, and 791 data instances in train, validation, and test respectively. The more detailed description of the dataset is available in (Srivastava and Singh, 2021b).

2.2 The Two Sub-tasks

2.2.1 Subtask 1: Quality rating prediction

The first subtask is predicting the quality rating of the code-mixed text. The participating teams can use the English, Hindi, and human-generated Hinglish sentences to predict the average rating[4] as provided by the human annotators to the synthetic Hinglish sentences.

2.2.2 Subtask 2: Annotators' disagreement prediction

The next subtask is predicting the disagreement between the ratings provided by the human annotators to the synthetic Hinglish sentences. We calculate the disagreement between the ratings as the absolute difference between the two rating scores.

2.3 Baseline Experiments

We created a baseline with SOTA multilingual contextual language model M-BERT (Devlin et al., 2019). We finetune the pre-trained M-BERT model by adding one hidden-layer neural network on the top. We use the Relu activation function, AdamW optimizer with 0.03 learning rate, cross-entropy loss, and a batch size of 32. We use the contextual word-embedding corresponding to the synthetic Hinglish sentences in the dataset as an input to the model. The architecture remains the same for both subtasks.

2.4 Evaluation Metrics

We use the following three evaluation metrics:

- **F1-score (FS)**: We use the weighted F1-score to evaluate the system performance. The score

[4] We take the greatest integer i ≤ average of the two rating scores.

English	Hindi	Human-generated Hinglish	Synthetic Hinglish 1	Synthetic Hinglish 2
The reward of goodness shall be nothing but goodness.	अच्छाई का बदला अच्छाई के सिवा और क्या हो सकता है?	The reward of achai shall be nothing but achai.	reward ka badla reward ke nothing aur kya ho sakta hai **Rating1**: 7 **Rating2**: 4	reward of goodness goodness ke siva aur kya ho sakta hai **Rating1**: 9 **Rating2**: 7
		Goodness ka badla goodness ke siva aur kya ho sakta hai.		
		Achai ka badla shall be nothing but achai.		

Table 2: Example human-generated and synthetic Hinglish sentences from the dataset along with the source English and Hindi sentences. Two different human annotators rate the synthetic Hinglish sentences on the scale 1-10 (low-high quality).

ranges from 0 (worst) to 1 (best).

- **Cohen's Kappa (CK)**: We use the Cohen's Kappa score to measure the agreement between the predicted and the actual rating. The score ranges from ≤ 0 (high disagreement) to 1 (high agreement).
- **Mean Squared Error (MSE)**: MSE suggests the difference between the actual and the predicted scores. A low MSE score is preferred, with zero being the lowest possible score.

We use all three metrics for the first subtask, whereas we use FS and MSE to evaluate the second subtask.

3 Overview of Participants and Submissions

In total, 46 participants grouped in **18** teams have registered for the shared task. This includes teams from top US universities like Stanford University and Carnegie Mellon University, companies like Tencent QQ, and top Indian universities like IISc, IITK, and IITBHU. Out of **18** teams, **nine** and **eight** teams have submitted at least once during train/validation and test phase, respectively.

We requested all teams that submitted the test scores to submit the paper illustrating the methodology. Out of eight teams, we received papers from four teams listed below:

1. **IIIT Hyderabad, India** (Kodali et al., 2022): This team comprises seven researchers. The team ranked 2nd against FS and CK metrics and 1st against MSE metric in Subtask 1 and 3rd against FS metric and 1st against MSE metric in Subtask 2. Hereafter, we refer to this team as **'IIITH'**.

2. **Manipal University, India** (Singh, 2022): This team comprises one researcher. The team ranked 3rd against FS and CK metrics and 1st against MSE metric in Subtask 1, and 1st in Subtask 2 against both FS and MSE metrics. Hereafter, we refer to this team as **'MU'**.

3. **BITS Pilani, India** (Furniturewala et al., 2022): This team comprises five researchers. The team ranked 5th against FS and CK metrics and 2nd against MSE metric in Subtask 1, and 2nd in Subtask 2 against both FS and MSE metrics. Hereafter, we refer to this team as **'BITS'**.

4. **Jadavpur University, India** (Guha et al., 2022): This team comprises three researchers. The team ranked 9th against FS, 8th against CK and 3rd against the MSE metric in Subtask 1 and 6th against FS, and 3rd against the MSE metric in Subtask 2. Hereafter, we refer to this team as **'JU'**.

Next, we discuss each of the submissions in detail:

3.1 IIITH

IIITH team leveraged two Multilingual Large Language Models (MLLMs), XLM-R (Conneau et al., 2020) and LABSE (Feng et al., 2022) to generate embeddings for Hindi, English, synthetic, and Human-generated code-mixed Hinglish sentences. In addition to the embeddings as a feature, they computed scores from three code-mixing metrics, Code-Mixing Index (CMI, (Gambäck and Das, 2016)), Number of Switch Points, and Burstiness (Goh and Barabási, 2008). All metric scores and embeddings are combined together to generate features. The features are used to train Linear Regression, MLP Regressor, and XGBoost classifiers. Out of these three, MLP Regressor performed best.

3.2 MU

MU team leveraged LABSE (Feng et al., 2022) to create embeddings for Hindi and English sentences and BERT (Devlin et al., 2019) to create embeddings for Hinglish sentences. The obtained vectors

Team Name	Subtask 1			Subtask 2	
	FS	CK	MSE	FS	MSE
Baseline	0.26637 (1)	0.09922 (1)	2.00000 (1)	0.14323 (8)	5.00000 (3)
IIITH	0.25734 (2)	0.09858 (2)	2.00000 (1)	0.23523 (3)	3.00000 (1)
MU	0.25062 (3)	0.08153 (3)	2.00000 (1)	0.26115 (1)	3.00000 (1)
BITS	0.21796 (5)	0.07337 (5)	3.00000 (2)	0.23940 (2)	4.00000 (2)
JU	0.11582 (9)	0.00337 (8)	6.00000 (3)	0.18331 (6)	5.00000 (3)

Table 3: Comparison between the four submissions. Number inside a bracket represent relative rank in respective shared task for a particular metric.

are then concatenated and fed into a catboost-based classifier (Prokhorenkova et al., 2018) to generate final predictions.

3.3 BITS

BITS team, first-of-all, finetuned a Multilingual BERT model (Pires et al., 2019), a language model pretrained on 104 languages. Then, they utilized the deep semantic features extracted from Multilingual BERT for different sentence types to train a fully connected neural network. They used the same two-fold architecture for both tasks.

3.4 JU

JU team leveraged GloVe embeddings (Pennington et al., 2014) to represent English and Hindi sentences and one hot vectors to represent Hinglish sentences. Further, GloVe embeddings were passed through respective Bidirectional LSTMs (Bi-LSTMs). The one-hot vectors are fed to a dense layer. The combined vectors from the Bi-LSTMs and dense layers are further passed through a dense layer for final predictions. They used the same architecture for both tasks.

4 Results and Discussion

In this section, we compare the four submissions for both sub-tasks. Table 3 showcases the results for four systems. Note that the table contains only those entries that submitted the final methodology paper. As illustrated, no team was able to outperform the baseline for Subtask 1. MU performed best for Subtask 2. The other entries and their corresponding rankings are present on the official leaderboard[5] of the shared task.

The four teams have leveraged large-scale language models (XLM-R, LABSE or BERT). The models were either finetuned or used for generating embeddings. The embeddings were passed to a classifier model for final predictions. Subtask 2 showcases significant improvements over baseline scores.

5 Conclusion and Future Directions

In this shared task, the participating teams have created systems to evaluate the quality of the code-mixed text. These systems can help develop futuristic NLP tools that filter out noisy poor, quality code-mixed text from the good quality code-mixed text. We also proposed several research questions that need to be answered implicitly with the experiments. However, none of the team has answered these questions. We plan to explore these questions in our future editions. Overall, this shared task will help the code-mixing research community build efficient and robust code-mixed text generation and evaluation systems.

Acknowledgements

We would like to thank Mr. Pritam Kadasi for helping in setting up the competition page. We would also like to thank several researchers from academia and industry for this shared task's publicity.

References

Somnath Banerjee, Kunal Chakma, Sudip Kumar Naskar, Amitava Das, Paolo Rosso, Sivaji Bandyopadhyay, and Monojit Choudhury. 2016. Overview of the mixed script information retrieval (msir) at fire-2016. In *Forum for Information Retrieval Evaluation*, pages 39–49. Springer.

Ruthanna Barnett, Eva Codó, Eva Eppler, Montse Forcadell, Penelope Gardner-Chloros, Roeland van Hout, Melissa Moyer, Maria Carme Torras, Maria Teresa Turell, Mark Sebba, Marianne Starren, and Sietse

[5] https://codalab.lisn.upsaclay.fr/competitions/1688#results

Wensing. 2000. The lides coding manual: A document for preparing and analyzing language interaction data version 1.1—july, 1999. *International Journal of Bilingualism*, 4(2):131–132.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Navya Jose, Anand Kumar M, Thomas Mandl, Prasanna Kumar Kumaresan, Rahul Ponnusamy, Hariharan R L, John P. McCrae, and Elizabeth Sherly. 2021. Findings of the shared task on offensive language identification in Tamil, Malayalam, and Kannada. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 133–145, Kyiv. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Amitava Das and Björn Gambäck. 2014. Identifying languages at the word level in code-mixed indian social media text. In *Proceedings of the 11th International Conference on Natural Language Processing*, pages 378–387.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Mrinal Dhar, Vaibhav Kumar, and Manish Shrivastava. 2018. Enabling code-mixed translation: Parallel corpus creation and mt augmentation approach. In *Proceedings of the First Workshop on Linguistic Resources for Natural Language Processing*, pages 131–140.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic bert sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891.

Shaz Furniturewala, Vijay Kumari, Amulya Ratna Dash, Hriday Kedia, and Yashvardhan Sharma. 2022. Bits pilani at hinglisheval: Quality evaluation for code-mixed hinglish text using transformers. *arXiv preprint arXiv:2206.08680*.

Björn Gambäck and Amitava Das. 2016. Comparing the level of code-switching in corpora. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1850–1855.

K-I Goh and A-L Barabási. 2008. Burstiness and memory in complex systems. *EPL (Europhysics Letters)*, 81(4):48002.

Prantik Guha, Rudra Dhar, and Dipankar Das. 2022. Ju_nlp at hinglisheval: Quality evaluation of the low-resource code-mixed hinglish text. *arXiv preprint arXiv:2206.08053*.

Gualberto A Guzmán, Joseph Ricard, Jacqueline Serigos, Barbara E Bullock, and Almeida Jacqueline Toribio. 2017. Metrics for modeling code-switching across corpora. In *INTERSPEECH*, pages 67–71.

Prashant Kodali, Tanmay Sachan, Akshay Goindani, Anmol Goel, Naman Ahuja, Manish Shrivastava, and Ponnurangam Kumaraguru. 2022. Precogiiith at hinglisheval: Leveraging code-mixing metrics & language model embeddings to estimate code-mix quality. *arXiv preprint arXiv:2206.07988*.

Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. 2018. The iit bombay english-hindi parallel corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Giovanni Molina, Fahad AlGhamdi, Mahmoud Ghoneim, Abdelati Hawwari, Nicolas Rey-Villamizar, Mona Diab, and Thamar Solorio. 2016. Overview for the second shared task on language identification in code-switched data. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 40–49.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Parth Patwa, Gustavo Aguilar, Sudipta Kar, Suraj Pandey, PYKL Srinivas, Björn Gambäck, Tanmoy Chakraborty, Thamar Solorio, and Amitava Das. 2020. Semeval-2020 task 9: Overview of sentiment analysis of code-mixed tweets. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 774–790.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001.

Adithya Pratapa, Gayatri Bhat, Monojit Choudhury, Sunayana Sitaram, Sandipan Dandapat, and Kalika Bali. 2018. Language modeling for code-mixing: The role of linguistic theory based synthetic data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1543–1553.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2018. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31.

Shashi Shekhar, Dilip Kumar Sharma, and MM Sufyan Beg. 2020. Language identification framework in code-mixed social media text based on quantum lstm—the word belongs to which language? *Modern Physics Letters B*, 34(06):2050086.

Kushagra Singh, Indira Sen, and Ponnurangam Kumaraguru. 2018. Language identification and named entity recognition in hinglish code mixed tweets. In *Proceedings of ACL 2018, Student Research Workshop*, pages 52–58.

Nikhil Singh. 2022. niksss at hinglisheval: Language-agnostic bert-based contextual embeddings with catboost for quality evaluation of the low-resource synthetically generated code-mixed hinglish text. *arXiv preprint arXiv:2206.08910*.

Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Ghoneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, et al. 2014. Overview for the first shared task on language identification in code-switched data. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 62–72.

Vivek Srivastava and Mayank Singh. 2020. Phinc: A parallel hinglish social media code-mixed corpus for machine translation. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 41–49.

Vivek Srivastava and Mayank Singh. 2021a. Challenges and limitations with the metrics measuring the complexity of code-mixed text. In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 6–14.

Vivek Srivastava and Mayank Singh. 2021b. Hinge: A dataset for generation and evaluation of code-mixed hinglish text. *arXiv preprint arXiv:2107.03760*.

PreCogIIITH at HinglishEval : Leveraging Code-Mixing Metrics & Language Model Embeddings To Estimate Code-Mix Quality

Prashant Kodali† **Tanmay Sachan**† **Akshay Goindani**† **Anmol Goel**†
Naman Ahuja† **Manish Shrivastava**† **Ponnurangam Kumaraguru**†
†International Institute of Information Technology Hyderabad
{prashant.kodali, tanmay.sachan, akshay.goindani, anmol.goel}@research.iiit.ac.in
naman.ahuja@students.iiit.ac.in, {m.shrivastava, pk.guru}@iiit.ac.in

Abstract

Code-Mixing is a phenomenon of mixing two or more languages in a speech event and is prevalent in multilingual societies. Given the low-resource nature of Code-Mixing, machine generation of code-mixed text is a prevalent approach for data augmentation. However, evaluating the quality of such machine generated code-mixed text is an open problem. In our submission to HinglishEval, a shared-task collocated with INLG2022, we attempt to build models factors that impact the quality of synthetically generated code-mix text by predicting ratings for code-mix quality. HinglishEval Shared Task consists of two sub-tasks - a) Quality rating prediction); b) Disagreement prediction. We leverage popular code-mixed metrics and embeddings of multilingual large language models (MLLMs) as features, and train task specific MLP regression models. Our approach could not beat the baseline results. However, for Subtask-A our team ranked a close second on F-1 and Cohen's Kappa Score measures and first for Mean Squared Error measure. For Subtask-B our approach ranked third for F1 score, and first for Mean Squared Error measure. Code of our submission can be accessed here.

1 Introduction

Code-mixing[1] is a phenomenon where linguistic units from two or more languages are interspersed in a single utterance or a speech event and is common in multilingual communities. Due to increased penetration of the Internet and social media, code-mixing has become common and, at the same time, has posed challenges to automatic text processing pipelines (Çetinoğlu et al., 2016). One such challenge is the dearth of naturally occurring code-mixed data. Data constraints have been the primary motive for researchers to leverage data augmentation and construct synthetically generated code-mixed corpora using monolingual parallel data as input.

Synthetic code-mixed data generation, using monolingual parallel corpora, is a non-trivial generation task. In the generated code-mixed sentence, one has to be careful about both the adequacy (preserving semantic content of monolingual sentence) and fluency (grammatical correctness). The task is further obscured by the fact that there is no single way of writing a code-mixed sentence.

In this work, we describe our approaches implemented in our submission to HinglishEval, a shared task co-located at INLG2022. HinglishEval is based on the HinGE dataset (Srivastava and Singh, 2021a). HinGE is created in two phases: a) Human-generated Hinglish sentences: at least two Hinglish sentences corresponding to the 1,976 English-Hindi sentence pairs; b) Synthetic Hinglish sentence generation and quality evaluation: generate Hinglish sentences using two rule-based algorithms, with human annotations for quality rating for each synthetically generated sentence. Two annotators rated each sentence on a scale of 1(low-quality) to 10 (high-quality).

HinglishEval (Srivastava and Singh, 2021b), consists of two subtasks - a) Subtask-1 Quality rating prediction: For each synthetically generated sentence, predict the average (rounded-off) quality rating; b) Subtask-2 Disagreement prediction: predict the disagreement score (absolute difference between the two human ratings) for the synthetically generated sentence.

In our approach, we use a combination of code-mixed metrics and language model embeddings as features and train an MLP regressor for both the tasks. Rest of the paper is organised as follows: Section 2 describes the features and models in detail as well as the experimental setup; Section 3 covers the results of our experimentation; and

[1] "Code-switching" also refers to the phenomenon of mixing two or more languages and is often used interchangeably with code-mixing by the research community. Following the same convention, we use both terms interchangeably.

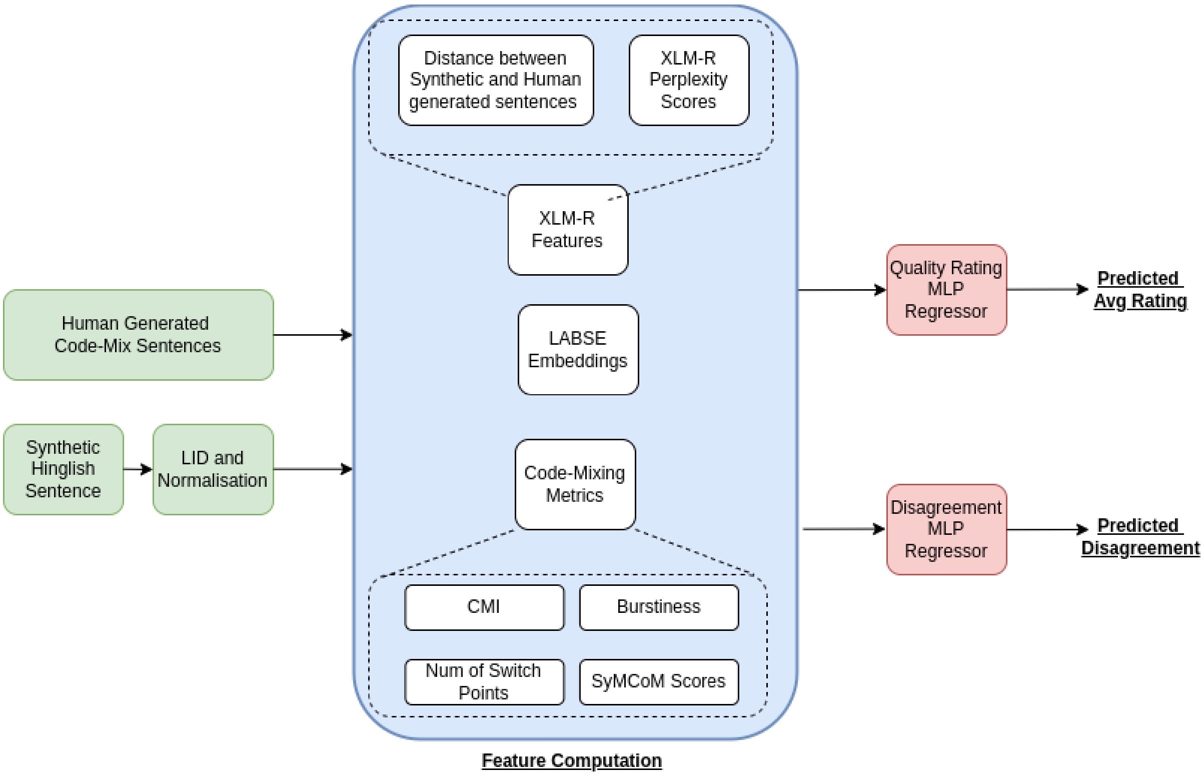


Figure 1: System Architecture for predicting Average Quality Rating and Disagreement Rating of Synthetic Code-Mixed sentences. Human-generated code-mixed sentences are also used as input to our Regression model.

we end with Section 4 discussing the implications, limitations, and future work.

2 System Overview

Figure 1 shows the system architecture of our submission for both the Sub-Tasks. We describe the Pre-Processing involved, methodology for feature computation, and model architecture in subsequent sub-sections.

2.1 Pre-Processing

Before computing features on code-mixed sentences, we pre-process the sentences using CSNLI tool [2]. CSNLI computes the token-wise Language ID (LID) and converts romanised Hindi tokens to native (Devanagari) script, a step that is also known as Normalisation. LID tags are used to compute LID based code-mixing metrics. The normalised code-mixed sentences are useful in computing Multilingual Large Language Model (MLLM) features. MLLMs have been shown to perform better in downstream tasks when input code-mixed text is in normalised form (Pires et al., 2019).

2.2 Features

Features used in our regression model can be broadly categorised as a) Code-Mixing Metrics as features, b) MLLM based Features.

1. Code-Mixing Metrics: (Guzmán et al., 2017; Gambäck and Das, 2016) proposed multiple Language ID based metrics which are used to compare code-mix corpora. However, such measures fail to capture syntactic variety in code-mixing, and to overcome this limitation we utilise SyMCoM measures proposed by (Kodali et al., 2022). We use the en-hi code-mix PoS tagger released by authors to compute PoS tags based on which SyMCoM scores are computed. For syntactic code-mix measures, we use SyMCoM scores for each PoS tag (Eq 3), and sentence level scores (Eq 4). For Eq 3 & 4, SU is a POS tag, and L_1 and L_2 are languages that are mixed. We use the following LID based code-mixing measures:

    - Code-Mixing Index (CMI) as described in Eq. 1, where N is the total number of languages mixed, w_i is the number of words present from i^{th} language, n is

[2] https://github.com/irshadbhat/csnli

	F1 Score	Cohen's Kappa	Mean Squared Error
Sub-Task 1 **Quality rating prediction**	0.25734 (2) $\Delta =\ 0.009$	0.09858 (2) $\Delta =\ 0.00064$	2.00000 (1) $\Delta = 0$
Sub-Task 2 **Disagreement prediction**	0.23523 (3) $\Delta = -0.02592$	-	3.00000 (1) $\Delta = 0$

Table 1: Performance Measures of our system for individual Sub-Tasks. Values in the bracket show position of our system in the task leaderboard. Δ is indicating the difference between the top-performing system for the sub-task and our system.

the total number of tokens, and u is the number of tokens given other tags.

- Number of Switch Points: number of times the language is switched within a sentence
- Burstiness, as described in Eq. 2, where σ_t denotes the standard deviation of the language spans and m_t the mean of the language spans. Burstiness captures the periodicity in the switch patterns, with periodic dispersion of switch points taking on burstiness values closer to -1, and sentences with less predictable patterns of switching take on values closer to 1.

Code-mix metrics are only computed for the synthetic code-mix sentences. Further, we scale normalised code-mixing metric based features.

$$CMI = \frac{\sum_{i=1}^{N}(w_i) - max(w_i)}{n - u} \tag{1}$$

$$Burstiness = \frac{\sigma_t - m_t}{\sigma_t + m_t} \tag{2}$$

2. MLLM Features: In recent years, Multilingual Large Language Models (MLLMs), such as XLM-R (Conneau et al., 2020), have performed well across semantic tasks and cross-lingual transfer, and have been the go-to methods in code-mixed settings as well (Khanuja et al., 2020). We utilise embeddings from two models - XLM-R and LABSE (Feng et al., 2022). We compute the pseudo-log-likelihood(PPL) scores proposed by (Salazar et al., 2020), which are akin to perplexity scores of conventional LMs. In our model, PPL scores are computed for both synthetic code-mixed sentences as well as human generated code-mixed sentences, and delta between the two PPL scores is considered as a feature.

   We use LABSE model to compute sentence embeddings which are used as features. We compute LABSE embeddings for Hindi, English monolingual sentences, and synthetic code-mixed sentence. The intuition behind using features from two different LMs was to improve the discriminative power of the model.

All the aforementioned features are concatenated resulting in a vector of dimension 2,385, and these features are used to train task-specific models.

$$SyMCoM_{SU} = \frac{(Count_{SU_{L1}}) - (Count_{SU_{L2}})}{\sum_{i=1}^{2} Count_{SU_{Li}}} \tag{3}$$

$$SyMCoM_{sent} = \sum_{SU} \frac{Count_{SU}}{len} \times |SyMCoM_{SU}| \tag{4}$$

2.3 Models

We experimented with various models such as - Linear Regression, MLP Regressor, and XGBoost with the combination of the above features. In the Validation phase, we noticed that the MLP outperformed all the other models. For the test phase we used only the MLP Regressor models. We train task-specific MLP Regression models using the same features, and rely on them to learn the complex function to predict the task-specific values in the same feature space.

We use the Sklearn library (Pedregosa et al., 2011) to implement the MLP Regressor models. We implement the MLP with three hidden layers - consisting of 1000, 100, and 10 neurons, respectively, paired with ReLU activation functions, an Adam optimizer, adaptive learning rate, and a default batch size and number of epochs. We could do only a limited hyper-parameter search, and a more structured and comprehensive hyperparameter search could lead to further improvement in the model's performance. Our hyperparameter search space for learning rate was 0.01, 0.001, 0.0001, and for hidden layer dimensions search space was {10,100,1000}.

3 Results

The scores for the Sub-Tasks achieved by our model are given in Table 1. For Sub-Task 1, we achieved rank 2 on the leaderboard for F1-Score and Cohen's Kappa, while a rank of 1 for Mean Squared Error (tied with the baseline model).

For Sub-Task 2, we beat the baseline model and achieved rank three on the leaderboard for F1-Score and one for Mean Squared Error (tied with the models achieving ranks 1 and 2).

For Sub-Task 1, our system is closest to the baseline model, as none of the competing models would beat the baseline model's performance. We hypothesize that the low-performance scores can be attributed to the task's hardness and the data's size.

As noted earlier, a comprehensive and structured hyper-parameter search will likely improve the results. Because of the very low delta between our system and the best performing system, hyperparameter tuning could be crucial to surpassing the baseline models.

4 Discussion

In this work, we propose a system to predict the Quality and Disagreement scores given code-mixed sentences and their monolingual counterparts. We leverage the combination of code-mixing metrics and MLLMs embeddings as features and train MLP regressor models. While our approach fails to beat the baseline/best performing system, the performance of our system is a close second or third and ranks first on MSE for both Sub-Tasks. Further hyper-parameter tuning can further improve the results.

Even the best-performing systems/baselines have very low scores across performance measures, which can be attributed to the difficulty of the task at hand, and the subjectivity of annotators while rating a sentence on a scale of 1-10. The size of the dataset could also be a limitation for solving the task at hand.

While MLP-based regressors are black boxes, having an explainable/interpretable model could help rank the features that impact the scores. In our system, an ablation study could help prune the feature space and identify the kind of features that are useful in rating prediction, and such features could be augmented. We leave these pursuits as part of our future work.

References

Özlem Çetinoğlu, Sarah Schulz, and Ngoc Thang Vu. 2016. Challenges of computational processing of code-switching. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 1–11, Austin, Texas. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Björn Gambäck and Amitava Das. 2016. Comparing the level of code-switching in corpora. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1850–1855, Portorož, Slovenia. European Language Resources Association (ELRA).

Gualberto A. Guzmán, Joseph Ricard, Jacqueline Serigos, Barbara E. Bullock, and Almeida Jacqueline Toribio. 2017. Metrics for modeling code-switching across corpora. In *INTERSPEECH*.

Simran Khanuja, Sandipan Dandapat, Anirudh Srinivasan, Sunayana Sitaram, and Monojit Choudhury. 2020. GLUECoS: An evaluation benchmark for code-switched NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3575–3585, Online. Association for Computational Linguistics.

Prashant Kodali, Anmol Goel, Monojit Choudhury, Manish Shrivastava, and Ponnurangam Kumaraguru. 2022. SyMCoM - syntactic measure of code mixing a study of English-Hindi code-mixing. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 472–480, Dublin, Ireland. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer,

R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.

Vivek Srivastava and Mayank Singh. 2021a. HinGE: A dataset for generation and evaluation of code-mixed Hinglish text. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 200–208, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Vivek Srivastava and Mayank Singh. 2021b. Quality evaluation of the low-resource synthetically generated code-mixed Hinglish text. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 314–319, Aberdeen, Scotland, UK. Association for Computational Linguistics.

niksss at HinglishEval: Language-agnostic BERT-based Contextual Embeddings with Catboost for Quality Evaluation of the Low-Resource Synthetically Generated Code-Mixed Hinglish Text

Nikhil Singh
Manipal University Jaipur
nikhil3198@gmail.com

Abstract

This paper describes the system description for the HinglishEval challenge at INLG 2022. The goal of this task was to investigate the factors influencing the quality of the code-mixed text generation system. The task was divided into two subtasks, quality rating prediction and annotators' disagreement prediction of the synthetic Hinglish dataset. We attempted to solve these tasks using sentence-level embeddings, which are obtained from mean pooling the contextualized word embeddings for all input tokens in our text. We experimented with various classifiers on top of the embeddings produced for respective tasks. Our best-performing system ranked 1st on subtask B and 3rd on subtask A. We make our code available here: https://github.com/nikhilbyte/Hinglish-qEval

1 Introduction

With the increase in popularity of social media platforms like blogs, Facebook, and Twitter in India, the amount of spoken and written Hinglish data has been on the rise. Hinglish is a blend of English and Hindi, involving code-switching between the above-mentioned languages. Due to the increasing number of users, the analysis of this new hybrid language using computational techniques has gotten important in a number of natural language processing applications like machine translation (MT) and speech-to-speech translation. (Bali et al., 2014),(Das and Gambäck, 2013).

Classical NLP problems such as language modeling (Pratapa et al., 2018), sentiment analysis (Singh and Lefever, 2020), (Chakravarthi et al., 2021),Hate-Speech Identification (Sreelakshmi et al., 2020) and language identification (Molina et al., 2016) are covered for Code-Mixed textual data. However, the generation and evaluation aspect of CM data hasn't been explored a lot.

This shared task aims to further the research of quality evaluation of the generated code-mixed text in a new way, proposing two tasks that will help quantify the quality of the synthetically generated CM text. Moreover, the organizers put forward another task that will help estimate the disagreement between the different human annotators, which further strengthens and reduce the noisiness of the ground-truth *quality* labels of the generated CM text sequence.

2 Related Work

There has been an increased interest in Code-Mixed data for various NLG tasks.(Yang et al., 2020) proposed a new pre-training strategy to tackle the complexities in CM text sequences in a non-traditional way. (Gautam et al., 2021) talks about generating low-resource Code-Mixed language from a high resource language such as English using various Seq2Seq models such as mBART (Liu et al., 2020). Other than this, various augmentation techniques were also proposed to improve the quality of generated Hinglish text sequences (Gupta et al., 2021). Due to its high linguistic diversity and lack of standardization, the basic Natural Language generation needs to be tackled and evaluated differently as shown in (Garg et al., 2021) where they propose different metrics to evaluate the quality of generated CM data and show why traditional translation metrics such as BLUE (Papineni et al., 2002) etc. cannot capture the quality evaluation properly.

3 Task Overview and Dataset

The task (Srivastava and Singh, 2021b) was divided into two subtasks. Subtask-A comprised of predicting the quality of the generated Hinglish sentences text on a scale of 1–10. 1 is low quality and 10 is the highest quality, considering the semantics and meaningfulness of the generated text sequence. However, the code-mixed language is seldom used in a formal setting, leading the popular evaluation

Proceedings of the 15th International Conference on Natural Language Generation: Generation Challenges, pages 31 - 34
July 17-22, 2022 ©2022 Association for Computational Linguistics

techniques such as BLUE and WER being inappropriate. The organizers tried to tackle this using another way of evaluation to curb the noisiness of labels occurring in subtask-A by proposing another subtask-B. This subtask tests the capacity of the proposed models for estimating the disagreement between individual annotators, which often occurs when trying to evaluate the quality of informal text sequences.

The data for this task introduced in (Srivastava and Singh, 2021a) is called the HinGE dataset. Its dataset comprises 3,952 instances. Where a particular instance i comprises a text sequence triplet in English, Hindi, and hinglish language and *Average rating* as the label for subtask-A and *Annotator disagreement* as the label for subtask-B. These instances were shuffled and divided into three parts in a ratio of 70:10:20, leading to 2766, 395, and 791 data instances in train, validation, and test respectively. An instance of the dataset can be found in Figure 1.

4 Methodology

We attempted these tasks as a text triplet classification problem, wherein we have three text sequences side-by-side and a label attached to them. We analyzed the text sequences and found them to be clean and without any redundant information, hence we didn't perform any traditional pre-processing step. The following steps were taken to build the submitend system:

- Out of the three text sequences in a particular data instance, we feed the English and Hindi input sentences or texts into a transformer network named Language-agnostic BERT sentence embedding model (LaBSE) (Feng et al., 2020). The model produces contextualized word embeddings for all input tokens in our text into a shared latent space that produces similar vector/embeddings for similar sentences in a language-agnostic way. As we want a fixed-sized output representation (vector u), we need a pooling layer. Different pooling options are available, the most basic one is mean-pooling: We simply average all contextualized word embeddings the model is giving us. This gives us a fixed 768-dimensional output vector independent of how long our input text was.
- The hinglish sequence was embedded using a BERT (Devlin et al., 2018) based model for hinglish text sequences available here [1] after using the same strategy as done for the English and Hindi counterparts.
- By this point, we have the three sentences/texts mapped to a fixed sized dense vector.
- The obtained vectors are then concatenated and fed into a catboost (Prokhorenkova et al., 2018) based classifier.
- The model was trained in a supervised manner using the default catboost classifiers with a logloss objective.A seed value of 42 was used to keep the model deterministic.
- The model took approximately 1.75 hours to train on CPU with a memory of 12Gb.
- The complete experiment was done on Google Colab Pro.
- The model architecture can be seen in Figure 2.

SubTask	FS	CK	MSE
SubTask A	0.25062	0.08153	2.00000
SubTask B	0.26115	-	3.00000
Baseline A	0.26637	0.09922	2.00000
Baseline B	0.14323	-	5.00000

Table 1: Results on for Test Set

All our experiments were performed using SBERT [2]

5 Results

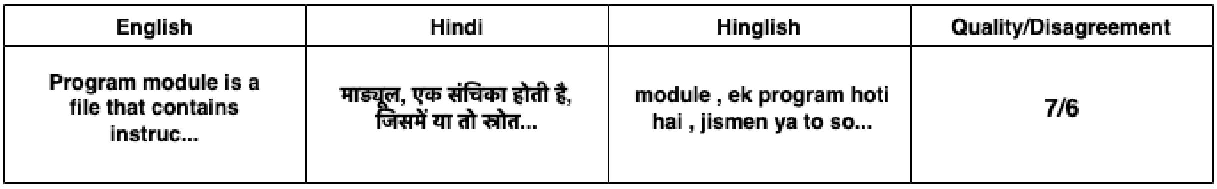
Three evaluation metrics F1-score (FS),Cohen's Kappa (CK),Mean Squared Error (MSE) were used to measure the performance of the submitted systems. We present the results obtained on test set along with the baselines in Table 1 .

6 Conclusion

We developed a system to evaluate the quality of machine-generated text sequences using a combination of deep learning feature vectors and machine

[1] https://huggingface.co/niksss/Hinglish-HATEBERT

[2] https://www.sbert.net/index.html



English	Hindi	Hinglish	Quality/Disagreement
Program module is a file that contains instruc...	माड्यूल, एक संचिका होती है, जिसमें या तो स्रोत...	module , ek program hoti hai , jismen ya to so...	7/6

Figure 1: A Single Instance from the Dataset

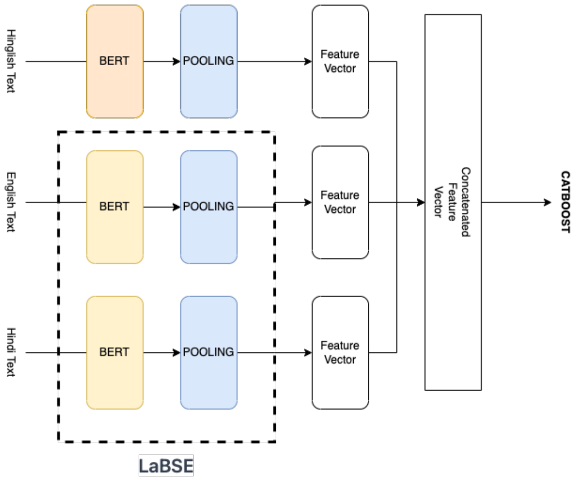


Figure 2: System Design

learning models. The results are nowhere near what would actually be used to evaluate the quality of the generated sequence. However, this is the first installment of the shared task and it sets off the baselines for future research on the same subject.

References

Kalika Bali, Jatin Sharma, Monojit Choudhury, and Yogarshi Vyas. 2014. "I am borrowing ya mixing ?" an analysis of English-Hindi code mixing in Facebook. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 116–126, Doha, Qatar. Association for Computational Linguistics.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Navya Jose, Anand Kumar M, Thomas Mandl, Prasanna Kumar Kumaresan, Rahul Ponnusamy, Hariharan R L, John P. McCrae, and Elizabeth Sherly. 2021. Findings of the shared task on offensive language identification in Tamil, Malayalam, and Kannada. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 133–145, Kyiv. Association for Computational Linguistics.

Amitava Das and Björn Gambäck. 2013. Code-mixing in social media text. the last language identification frontier? *Trait. Autom. des Langues*, 54:41–64.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*.

Ayush Garg, Sammed Kagi, Vivek Srivastava, and Mayank Singh. 2021. MIPE: A metric independent pipeline for effective code-mixed NLG evaluation. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 123–132, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Devansh Gautam, Prashant Kodali, Kshitij Gupta, Anmol Goel, Manish Shrivastava, and Ponnurangam Kumaraguru. 2021. Comet: Towards code-mixed translation using parallel monolingual sentences. In *CALCS*.

Abhirut Gupta, Aditya Vavre, and Sunita Sarawagi. 2021. Training data augmentation for code-mixed translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5760–5766, Online.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Giovanni Molina, Fahad AlGhamdi, Mahmoud Ghoneim, Abdelati Hawwari, Nicolas Rey-Villamizar, Mona Diab, and Thamar Solorio. 2016. Overview for the second shared task on language identification in code-switched data. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 40–49, Austin, Texas. Association for Computational Linguistics.

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

Adithya Pratapa, Gayatri Bhat, Monojit Choudhury, Sunayana Sitaram, Sandipan Dandapat, and Kalika Bali. 2018. Language modeling for code-mixing: The role of linguistic theory based synthetic data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1543–1553, Melbourne, Australia. Association for Computational Linguistics.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2018. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31.

Pranaydeep Singh and Els Lefever. 2020. Sentiment analysis for Hinglish code-mixed tweets by means of cross-lingual word embeddings. In *Proceedings of the The 4th Workshop on Computational Approaches to Code Switching*, pages 45–51, Marseille, France. European Language Resources Association.

K Sreelakshmi, B Premjith, and K.P. Soman. 2020. Detection of hate speech text in hindi-english code-mixed data. *Procedia Computer Science*, 171:737–744. Third International Conference on Computing and Network Communications (CoCoNet'19).

Vivek Srivastava and Mayank Singh. 2021a. Hinge: A dataset for generation and evaluation of code-mixed hinglish text. *arXiv preprint arXiv:2107.03760*.

Vivek Srivastava and Mayank Singh. 2021b. Quality evaluation of the low-resource synthetically generated code-mixed Hinglish text. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 314–319, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Zhen Yang, Bojie Hu, Ambyera Han, Shen Huang, and Qi Ju. 2020. Code-switching pre-training for neural machine translation. *arXiv preprint arXiv:2009.08088*.

BITS Pilani at HinglishEval: Quality Evaluation for Code-Mixed Hinglish Text Using Transformers

Shaz Furniturewala, Vijay Kumari, Amulya Ratna Dash, Hriday Kedia, Yashvardhan Sharma
BITS Pilani,Pilani,Rajasthan
(f20200025,p20190065)@pilani.bits-pilani.ac.in
(p20200105,f20190964,yash)@pilani.bits-pilani.ac.in

Abstract

Code-Mixed text data consists of sentences having words or phrases from more than one language. Most multi-lingual communities worldwide communicate using multiple languages, with English usually one of them. Hinglish is a Code-Mixed text composed of Hindi and English but written in Roman script. This paper aims to determine the factors influencing the quality of Code-Mixed text data generated by the system. For the HinglishEval task, the proposed model uses multilingual BERT to find the similarity between synthetically generated and human-generated sentences to predict the quality of synthetically generated Hinglish sentences.

1 Introduction

The term "Code-Mixing" refers to mixing words or phrases from different languages into a single text or speech utterance. It embeds linguistic units from one language, such as phrases, words, and morphemes, into an utterance from another language. An example of the Code-Mixed data can be seen in the Figure 1 (Srivastava and Singh, 2021b). In countries where bilingualism is a common practice, we often see people naturally switching between the two languages. A significant challenge to research is that there are no formal sources like books or news articles in Code-Mixed languages, and studies have to rely on sources like Twitter or messaging platforms. Generating and evaluating the available or produced data without a baseline is primarily reliant on people who are fluent in both languages.

Furthermore, present language models are ineffective in Code-Mixed situations, where morphemes, words, and phrases from one language are embedded in the other. As Code-Mixing has long been a way of communication in a multi-cultural, multi-lingual society, the next generation of AI bots should be able to understand Code-Mixed text.

The inherent challenges with the code-mixed data make the widely popular evaluation metrics like BLEU (Bilingual Evaluation Understudy Score) and WER (Word Error Rate) less effective. With the given task, the main objective is to propose and develop new strategies that address the overall need for quality evaluation of the generated Code-Mixed text.

This paper aims to assess the quality of the generated Hinglish text. The proposed model uses the transformer-based multi-lingual BERT (Pires et al., 2019) to obtain the embeddings of the Hindi, English, and Hinglish text. A similarity score is computed between synthetically generated Hindi and English text and human-generated Hindi and English sentences. This score will serve as the sentence's agreement or disagreement factor.

Example I

Hinglish: Image ko 90 degrees par right ghumaen.
English: Rotate the image 90 degrees to the right.

Example II

Hinglish: jis day sky chakkar khane lagega
English: The day the sky will tremble

Figure 1: Examples of Hinglish and English sentences

2 Related Work

Developing methodologies and resources for diverse Natural Language Processing applications incorporating multi-lingual and Code-Mixed languages has recently become popular. Few of them are word-embedding (Chen and Cardie, 2018), question answering system (Raghavi et al., 2015; Kumari et al., 2022), Code-Mixed text generation

Proceedings of the 15th International Conference on Natural Language Generation: Generation Challenges, pages 35 - 38
July 17-22, 2022 ©2022 Association for Computational Linguistics

(Pratapa et al., 2018) and Code-Mixed language modeling (Winata et al., 2018).

Various techniques were used to generate the Code-Mixed data, including matrix frame language theory, semi-supervised approach (Gupta et al., 2020), using dependency parsing (Jain et al., 2021), equivalent constraint theory (Pratapa et al., 2018) and Generative Adversarial Networks (Gao et al., 2019). A Metric Independent Evaluation Pipeline (MIPE) (Garg et al., 2021) considerably enhances the correlation among evaluation measures and human assessments of the generated Code-Mixed data. In the candidate's Hinglish phrase, MIPE minimises spelling differences and language switches for evaluation. Based on the significance of the terms missing from the candidate Hinglish sentence, deduct credit from the evaluation score. By arranging the candidates as well as the reference sentences into the phrases and using the paraphrasing ability, it also deals with the issue of having a restricted amount of reference sentences.

3 Dataset

HinGE(Srivastava and Singh, 2021a), a recently proposed dataset, is used for the HinglishEval task. The dataset contains 1976 English-Hindi sentence pairs and corresponding synthetic and human-generated Hinglish sentences.

Human Generated Sentences: Each English-Hindi sentence pair has at least two Hinglish sentences, with a total of 6694 such sentences.

Synthetically Generated Sentences: Using two different algorithms(WAC and PAC), synthetic Hinglish sentences are generated for each English-Hindi sentence pair. Each sentence is then given a quality rating by two human annotators. There are 2766 such sentences. For each sentence, the rounded off average of the two quality ratings is provided under the label of 'Average Rating' and the absolute difference of their scores is provided as 'Disagreement'. An instance of the HinGE dataset (Srivastava and Singh, 2021a) can be seen in Table 1.

4 Proposed Model

The proposed approach used a two-step procedure for both the rating and disagreement prediction tasks. The first step is to fine-tune multi-lingual BERT, a language model that has been pre-trained on 104 languages and is used to classify and evaluate disagreements further on. The second step used the deep semantic features obtained from multi-lingual BERT for various phrase categories to train a classifier neural network.

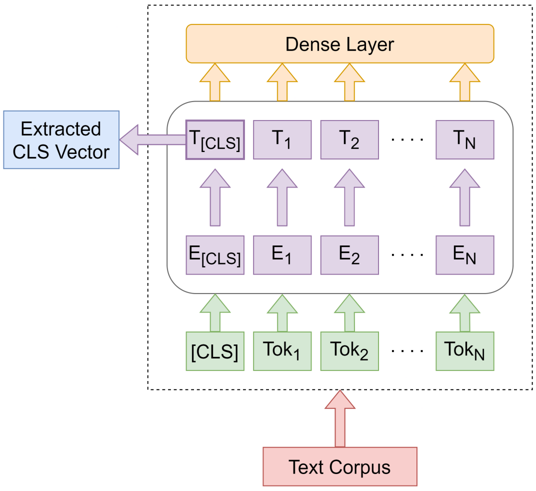


Figure 2: Extraction of CLS Vectors

4.1 Multi-lingual BERT

We employed the BERT-base-multi-lingual-cased model, a modified pre-trained BERT encoder that is pretrained in a self-supervised manner using the largest Wikipedias with the goal of masked language modeling. Multi-lingual BERT allows us to provide it with two sentences as input in the form:

[CLS] Sentence A [SEP] Sentence B [SEP]

Here, the CLS and SEP tokens are special tokens that allow BERT to recognize the beginning of an input and a separation between two different input sections. The central SEP token ensures that BERT knows there are two different sentences in the input.

Based on the dataset, we had four models for different types of sentences—English, Hindi, Synthetic Hinglish, and Human Hinglish. We ran the language model four times for each sub-tasks with various sorts of sentence pairs each time. The first two models trained had Synthetic Hinglish sentences as Sentence B, with Sentence A being the corresponding English sentence for model 1 and Hindi sentence for model 2. The next two models did the same, with Human Hinglish sentences paired with English sentences for model 3 and Hindi sentences for model 4. This process is repeated twice for rating classification and again for disagreement classification. The proposed model employs the BERT AutoTokenizer to tokenize the inputs and the Adam optimizer to train at a learning

English	Hindi	Human-Generated Hinglish	WAC	PAC
The reward of goodness shall be nothing but goodness	अचछाई का बदला अचछाई कए अलावा ओर कया हो सकता ह?	The reward of achai shall be nothing but achai. Goodness ka badla goodness ke siva aur kya ho sakta hai. Achai ka badla shall be nothing but achai.	reward ka badla reward ke nothing aur kya ho sakta hai **Rating1: 7** **Rating2: 4**	reward of goodness goodness ke siva aur kya ho sakta hai **Rating1: 9** **Rating2: 7**

Table 1: An Instance of the HinGE dataset

rate of 1e-6 for five epochs.

We extracted the deep semantic text features from each model using the BERT source code. The CLS token's feature vector is extracted from the output results of the last hidden layer, which is a 768-dimensional deep semantic feature of the legal language. We chose the CLS token, also known as the Classification token, as it has the fixed embedding that appears at the beginning of every sentence. Since all words infer the output of this token in the phrase, this CLS vector provides BERT's understanding of the sentence, which is particularly beneficial for a sentence classification task. The extraction process can be seen in Figure 2, where $T_{[CLS]}$ is the CLS vector output of the last hidden layer of multi-lingual BERT extracted from the model.

4.2 Classifier Neural Network

We obtained four sets of feature vectors from multilingual BERT for each sub-task. Two of them had dimensions of (2766, 768) and used synthetic Hinglish words as input. The other two used human-generated sentences and had dimensions of (6694, 768). To reduce the dimensionality of the latter, we averaged the vectors corresponding to each English-Hindi sentence pair resulting in a set of dimensions (1976, 768).

For each synthetic sentence, we concatenated the first two sets of vectors, and corresponding to each vector in this set, we appended its respective human vector. After combining the four-vector sets, we had an input of size (2766, 3072) for each sub-task.

The proposed model is trained with two fully connected neural networks, one for each sub-task, using these concatenated vector sets as their respective inputs. Both neural networks had two Linear hidden layers of dimensions (3072, 1536) and (1536, 768) with a final layer of size (768, 10). After each layer, a Rectified Linear activation function is being used. Binary Cross-Entropy loss is the chosen loss function. Adam optimizer were used along with a learning rate of 5e-6. Ten training epochs were used to train the disagreement classifier compared to just three for the ratings classifier. The entire procedure is illustrated in Figure 3, where the four concatenated CLS vectors are passed as input to a classifier neural network.

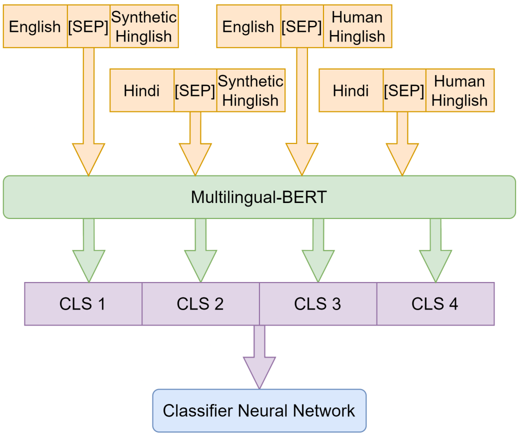


Figure 3: Proposed Model Architecture

		Sub-Task 1(Average Rating)			Sub-Task 2(Disagreement)	
	Model	F1	CK	MSE	F1	MSE
Validation	Baseline [1]	0.09504	-0.01530	15.000	0.15541	17.000
	Proposed Model	0.23493	0.06515	3.000	0.19400	4.000
Test	Baseline [1]	0.26637	0.09922	2.000	0.14323	5.000
	Proposed Model	0.21796	0.07337	3.000	0.24252	4.000

Table 2: Validation and Test results on the HinGE dataset

5 Results and Evaluation

5.1 Results

The evaluation metrics used are F1 Score, Cohen's Kappa (CK), and Mean Square Error (MSE). The submissions created by the model achieved rank 4 in rating prediction and rank 2 in disagreement prediction, not accounting for baseline scores, based on F1 Score, Cohen's Kappa, and mean squared error. The proposed approach results can be seen in Table 2, compared with the baseline scores.

We attained F1 scores of 0.218 on Rating classification and 0.242 on Disagreement classification after training over 2766 synthetic sentences and testing over 791 synthetic sentences.

6 Conclusion and Future Work

This paper used a two-step approach to solve the text classification problem. For each pair of phrase types, deep semantic text features were initially extracted using multilingual BERT as CLS vectors. These vectors were then properly combined, and processed by a fully-connected classifier neural network. The results suggest the proposed model is useful and that the obtained results can be greatly improved by fine-tuning and training with larger data, which could be a future research direction.

References

Xilun Chen and Claire Cardie. 2018. Unsupervised multilingual word embeddings. *arXiv preprint arXiv:1808.08933*.

Yingying Gao, Junlan Feng, Ying Liu, Leijing Hou, Xin Pan, and Yong Ma. 2019. Code-switching sentence generation by bert and generative adversarial networks. In *INTERSPEECH*, pages 3525–3529.

Ayush Garg, Sammed S Kagi, Vivek Srivastava, and Mayank Singh. 2021. Mipe: A metric independent pipeline for effective code-mixed nlg evaluation. *arXiv preprint arXiv:2107.11534*.

Deepak Gupta, Asif Ekbal, and Pushpak Bhattacharyya. 2020. A semi-supervised approach to generate the code-mixed text using pre-trained encoder and transfer learning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2267–2280.

Dhruval Jain, Arun D Prabhu, Shubham Vatsal, Gopi Ramena, and Naresh Purre. 2021. Codeswitched sentence creation using dependency parsing. In *2021 IEEE 15th International Conference on Semantic Computing (ICSC)*, pages 124–129. IEEE.

Vijay Kumari, Srishti Keshari, Yashvardhan Sharma, and Lavika Goel. 2022. Context-based question answering system with suggested questions. In *2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pages 368–373. IEEE.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? *arXiv preprint arXiv:1906.01502*.

Adithya Pratapa, Gayatri Bhat, Monojit Choudhury, Sunayana Sitaram, Sandipan Dandapat, and Kalika Bali. 2018. Language modeling for code-mixing: The role of linguistic theory based synthetic data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1543–1553.

Khyathi Chandu Raghavi, Manoj Kumar Chinnakotla, and Manish Shrivastava. 2015. " answer ka type kya he?" learning to classify questions in code-mixed language. In *Proceedings of the 24th International Conference on World Wide Web*, pages 853–858.

Vivek Srivastava and Mayank Singh. 2021a. Hinge: A dataset for generation and evaluation of code-mixed hinglish text. *arXiv preprint arXiv:2107.03760*.

Vivek Srivastava and Mayank Singh. 2021b. Quality evaluation of the low-resource synthetically generated code-mixed Hinglish text. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 314–319, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Genta Indra Winata, Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Code-switching language modeling using syntax-aware multi-task learning. *arXiv preprint arXiv:1805.12070*.

[1] https://codalab.lisn.upsaclay.fr/competitions/1688#results

JU_NLP at HinglishEval: Quality Evaluation of the Low-Resource Code-Mixed Hinglish Text

Prantik Guha[1] and **Rudra Dhar**[2] and **Dipankar Das**[3]
Department of Computer Science & Engineering, Jadavpur University, Kolkata, India
{[1]prantikguha706, [2]rudradharrd, [3]dipankar.dipnil2005}@gmail.com

Abstract

In this paper we describe a system submitted to the INLG 2022 Generation Challenge (GenChal) on Quality Evaluation of the Low-Resource Synthetically Generated Code-Mixed Hinglish Text. We implement a Bi-LSTM-based neural network model to predict the Average rating score and Disagreement score of the synthetic Hinglish dataset. In our models, we used word embeddings for English and Hindi data, and one hot encodings for Hinglish data. We achieved a F1 score of 0.11, and mean squared error of 6.0 in the average rating score prediction task. In the task of Disagreement score prediction, we achieve a F1 score of 0.18, and mean squared error of 5.0.

1 Introduction

In India, social media's enduring popularity has resulted in massive amounts of user-generated textual content. During a conversation, multilingual speakers frequently flip between languages. Speakers frequently talk in multiple languages, and often transliterate. Listeners may not always be able to keep up with the multilingual speakers. That's why we need automated systems for transliterated translations.

But we don't have a significant amount of transliterated translation data to train our models. So we might use synthetic data for this purpose. Synthetic data has become a common resource for a variety of applications. It may be required because of data unavailability, cost savings, security, or privacy concerns. Because synthetic data matches the statistical properties of production data, it can be used to train models, validate models, and evaluate performance. Machine learning models have now made it possible to create incredibly fast natural language generating systems by building and training a model.

Now the next challenge is to evaluate the data which is synthetically generated. In this paper we have introduced an algorithm to check the quality of the generated data. We have proposed a supervised learning model using multiple Bi-LSTM and dense layers to predict two types of scores (Average Rating score and Disagreement score). In this paper we are using the data from Srivastava and Singh (2021a).

This is a transliterated translation verification problem which essentially boils down to a task of document similarity evaluation. Document similarity evaluation is a well researched task in NLP. As Merlo et al. (2003) suggests, various Machine learning techniques, and Natural Language Processing tools can be used for this purpose. Linhares Pontes et al. (2018) shows us how hybrid models of LSTM's can be used for document similarity prediction. Some work has also been done in the multilingual senario, as in Wang et al. (2018). However not much work has been done in transliterated translation verification, and certainly none has been done in the Indian domain. Srivastava and Singh (2020) explains the challenges in both generating transliterated translations and evaluating it.

2 Dataset

The phenomena of code-mixing are the mingling of words and phrases from various languages in a single text or spoken utterance. Examples of code-mixed Hinglish sentences created from parallel Hindi and English utterances are shown in Fig-1.

In this shared task, there are two subtasks for evaluating the quality of the code-mixed Hinglish text in this common task (Srivastava and Singh, 2021b). In the first sub-task, they proposed using a scale of 110 to determine the quality of Hinglish content. They want to figure out what elements influence text quality, so high-quality code-mixed text generating systems can be created. The second sub-task is to predict how much the two annotators

Proceedings of the 15th International Conference on Natural Language Generation: Generation Challenges, pages 39 - 42
July 17-22, 2022 ©2022 Association for Computational Linguistics

Example I
English : Program module is a file that contains instructions which are either in the form of source code or machine language.

Hinglish : module , ek program hoti hai , jismen ya to source code ya machine language ke form men instructions nihit hote hain.

Example II
English : In France, the news of one deed spreads like a flash and brings some pride to a disillusioned people.

Hinglish : france men is ek deed ki news bijli ki tarah phail jati hai aur people bhram se mut hokar pride mahsoos karte hain.

Figure 1: Example from (Srivastava and Singh, 2021a) data

who annotated the synthetically generated Hinglish sentences differ on a scale of 09. Various factors influence human disagreement.

The dataset consists of five columns (English, Hindi, Hinglish, Average Rating, Disagreement). Highlish sentences are generated using two rule-based algorithms (i.e., WAC and PAC). For the two rating columns (Average Rating & Disagreement) each sentence is rated on a scale of 1(low-quality) to 10 (high-quality) by two annotators. The quality of the synthetically generated sentences is calculated by rounding off the average of the two human ratings and using this score (in the range of 1-10) in the Average rating column. And the Disagreement score is calculated by the absolute difference of the two human ratings as the disagreement score (in the range of 0-9).

3 System Description

We used a sequence of Glove embeddings as input for English and Hindi sentences. However, for Hinglish sentences we used one hot vector as inputs. We fed the English and Hindi embeddings to separate Bi-lstm's[l-e, l-h], and retrieved sequence output from them. To capture the word sequences of different Hindi and English sentences we have used two different LSTMs. Then we concatenated these 2 outputs and passed it through another Lstm layer to get a fixed (not sequence) vector output [l-h-e].

No. of data	F1-Score	Cohen's Kappa	Mean Squared Error
395	0.09899	-0.01521	6.00

Table 1: This result is obtained from 395 validation data for Sub-task 1(Average rating score)

No. of data	F1-Score	Mean Squared Error
395	0.21622	5.00

Table 2: This result is obtained from 395 validation data for Sub-task 2(Disagreement score)

We fed the one hot vector from the Hinglish data to a dense layer and received a vector output [d-he]. Since one hot vector does not capture the sequential information, we have used a dense layer. We then concatenated these two [l-h-e and d-he] vectors, and passed it through a dense layer to get a final class (score between 1 to 10). We used the same model for both the tasks. Please refer to Fig-2 for complete system architecture.

4 Training

On a total of 2766 training data points, we train the LSTM model using the Adam optimizer with a batch size of 32. Started with loss of 0.1810 & accuracy of 0.9658. In the final epoch loss was 0.0300 & accuracy was 0.9864.

In this phase, we validated the input using our developed model. For this phase the total available data was 395. We have validated our model for both Average Rating as well as Disagreement.On 395 data we validated our system to predict Average rating for corresponding inputs. Please refer to Table: 1 for detailed results related to this validation. On 395 data we validated our system to predict Disagreement score for corresponding inputs. Please refer to Table: 2 for detailed results related to this validation.

5 Test

In this phase, our developed model gets tested on test data. For this phase the total available data was 791. Model was tested for both Average Rating as well as Disagreement.

On 791 data, our system is able to predict Average rating for corresponding inputs. Please refer to

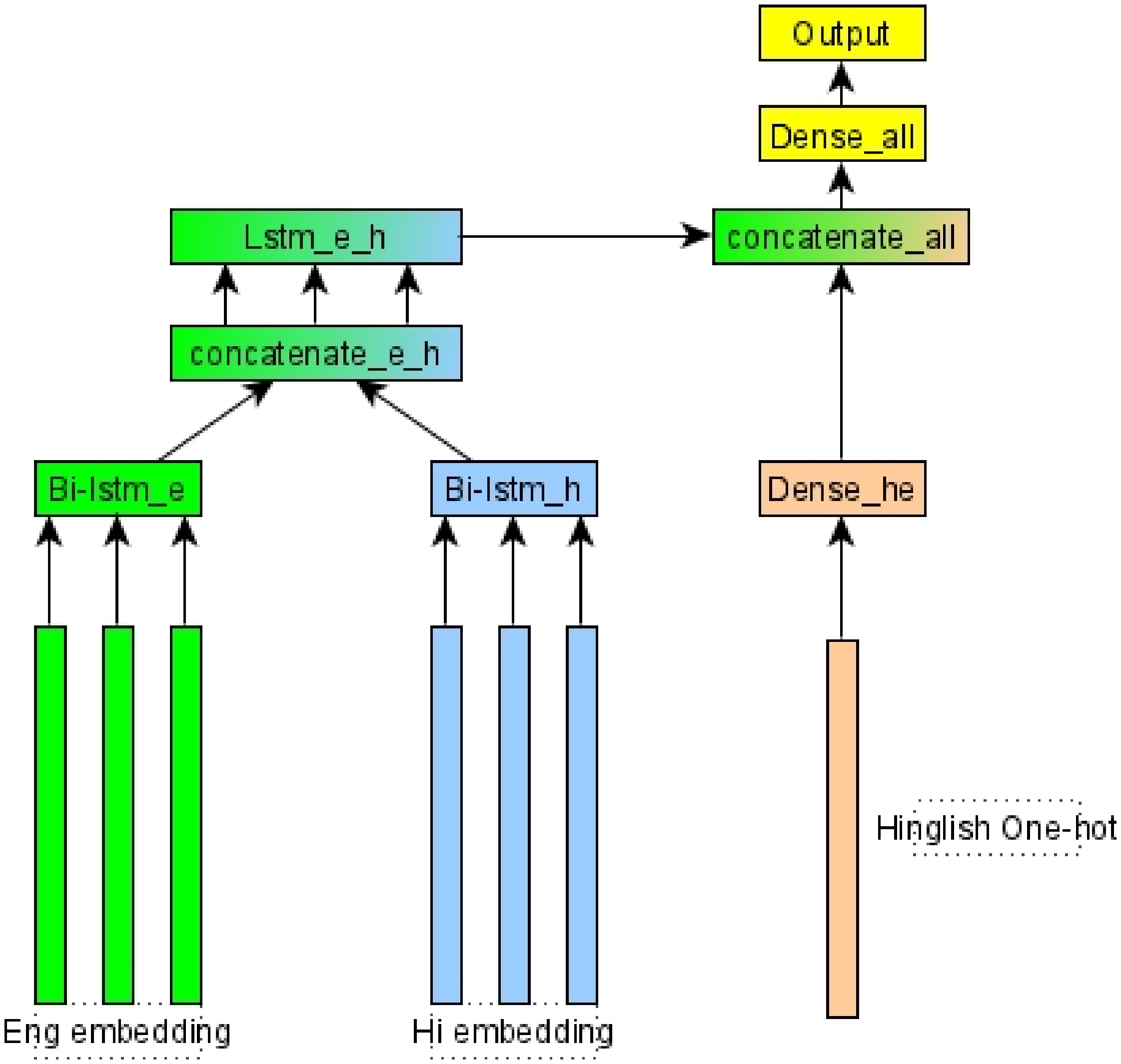


Figure 2: system architecture

No. of data	F1-Score	Cohen's Kappa	Mean Squared Error
791	0.11582	0.00337	6.00

Table 3: This result is obtained from 791 test data for Sub-task 1(Average rating score)

No. of data	F1-Score	Mean Squared Error
791	0.18331	5.00

Table 4: This result is obtained from 791 test data for Sub-task 2(Disagreement score)

Table: 3 for detailed results related to this validation. On 791 data, our system is able to predict Disagreement score for corresponding inputs. Please refer to Table: 4 for detailed results related to this validation.

6 Conclusion

For INLG 2022, we created a system to predict the Average Rating of synthetically generated Hinglish sentences (Sub-Task 1) & Disagreement score for the same (Sub-Task 2). We didn't use any outside information. We have used GLOVE embedding for English and Hindi sentences. And for Hinglish sentences we have used multi label vectors.

References

Amitava Das and Björn Gambäck. 2014. Identifying languages at the word level in code-mixed indian social media text.

Jinyu Li, Sibel Yaman, Chin-hui Lee, Bin Ma, Rong Tong, Donglai Zhu, and Haizhou Li. 2006. Language recognition based on score distribution feature vectors and discriminative classifier fusion. In *2006 IEEE Odyssey - The Speaker and Language Recognition Workshop*, pages 1–5.

Elvys Linhares Pontes, Stéphane Huet, Andréa Linhares, and Juan-Manuel Torres-Moreno. 2018. Predicting the semantic textual similarity with siamese cnn and lstm.

Ruibo Liu, Jason Wei, and Soroush Vosoughi. 2021. Language model augmented relevance score. *arXiv preprint arXiv:2108.08485*.

Paola Merlo, James Henderson, Gerold Schneider, and Eric Wehrli. 2003. Learning document similarity using natural language processing. *Linguistik online*, 17.

Braja Gopal Patra, Dipankar Das, and Amitava Das. 2018. Sentiment analysis of code-mixed indian languages: An overview of sail_code-mixed shared task@ icon-2017. *arXiv preprint arXiv:1803.06745*.

Adithya Pratapa, Monojit Choudhury, and Sunayana Sitaram. 2018. Word embeddings for code-mixed language processing. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 3067–3072.

Vivek Srivastava and Mayank Singh. 2020. Phinc: A parallel hinglish social media code-mixed corpus for machine translation. *arXiv preprint arXiv:2004.09447*.

Vivek Srivastava and Mayank Singh. 2021a. HinGE: A dataset for generation and evaluation of code-mixed Hinglish text. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 200–208, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Vivek Srivastava and Mayank Singh. 2021b. Quality evaluation of the low-resource synthetically generated code-mixed Hinglish text. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 314–319, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Vivek Srivastava and Mayank Singh. 2022. Code-mixed nlg: Resources, metrics, and challenges. In *5th Joint International Conference on Data Science & Management of Data (9th ACM IKDD CODS and 27th COMAD)*, pages 328–332.

Zhouhao Wang, Enda Liu, Hiroki Sakaji, Tomoki Ito, Kiyoshi Izumi, Kota Tsubouchi, and Tatsuo Yamashita. 2018. Estimation of cross-lingual news similarities using text-mining methods. *Journal of Risk and Financial Management*, 11(1).

Siddharth Yadav and Tanmoy Chakraborty. 2020. Unsupervised sentiment analysis for code-mixed data. *arXiv preprint arXiv:2001.11384*.

The 2022 ReproGen Shared Task on Reproducibility of Evaluations in NLG: Overview and Results

Anya Belz
ADAPT/DCU, Ireland and Univ. of Aberdeen
anya.belz@adaptcentre.ie

Anastasia Shimorina
Orange, Lannion, France
anastasia.shimorina@orange.com

Maja Popović
ADAPT/DCU, Ireland
maja.popovic@adaptcentre.ie

Ehud Reiter
University of Aberdeen, UK
e.reiter@abdn.ac.uk

Abstract

Against a background of growing interest in reproducibility in NLP and ML, and as part of an ongoing research programme designed to develop theory and practice of reproducibility assessment in NLP, we organised the second shared task on reproducibility of evaluations in NLG, ReproGen 2022. This paper describes the shared task, summarises results from the reproduction studies submitted, and provides further comparative analysis of the results. Out of six initial team registrations, we received submissions from five teams. Meta-analysis of the five reproduction studies revealed varying degrees of reproducibility, and allowed further tentative conclusions about what types of evaluation tend to have better reproducibility.

1 Introduction

Interest in reproducibility continues to grow across Natural Language Processing (NLP).[1] However, we still do not understand well enough what makes evaluations easier or harder to reproduce, and reproduction studies often reveal alarmingly low degrees of reproducibility not only for human evaluations but also for automatically computed metrics.

With the ReproGen shared task on Reproducibility of Evaluations in NLG, our aim is to add to the body of reproduction studies in order to increase the data points available for investigating reproducibility, and to begin to identify properties of evaluations that are associated with better reproducibility.

We start in Section 2 by describing the organisation and structure of the shared task, followed by an overview of the participating teams (Section 3). Next, we present high-level degree-of-reproducibility results for each reproduction study, and in the case of the more complex studies, also for subsets of results (Section 4). We look at the properties of the ReproGen evaluation studies in standardised terms as facilitated by the HEDS sheets completed by participants, and explore if any properties appear to have an effect on degree of reproducibility (Section 5). We conclude with some discussion (Section 6) and a look to future work (Section 7).

2 ReproGen 2022

Like its predecessor, ReproGen 2022[2] had two tracks, one a shared task in which teams try to reproduce the same previous evaluation results, the other an 'unshared task' in which teams attempt to reproduce their own previous evaluation results:

A ***Main Reproducibility Track:*** For a shared set of selected evaluation studies, participants repeat one or more studies, and attempt to reproduce the results, using published information plus additional information and resources provided by the authors, and making commonsense assumptions where information is still incomplete.

B ***RYO Track:*** *R*eproduce *Y*our *O*wn previous evaluation results, and report what happened. Unshared task.

For the main track (A above), we used the same papers as in ReproGen 2021, with the addition of one paper (Nisioi et al., 2017) previously used

[1] See our systematic review of reproducibility research in NLP carried out in part as background research for ReproGen (Belz et al., 2021).

[2] All information and resources relating to ReproGen are available at https://reprogen.github.io/.

Proceedings of the 15th International Conference on Natural Language Generation: Generation Challenges, pages 43 - 51
July 17-22, 2022 ©2022 Association for Computational Linguistics

Track	Team	Original paper	Reproduction paper	Metrics
A	Tilburg University	Santhanam and Shaikh (2019)	Braggaar et al. (2022)	automatic/human
	ADAPT Centre @ DCU	Nisioi et al. (2017)	Popović et al. (2022)	human
	University of Illinois at Chicago	Nisioi et al. (2017)	Arvan et al. (2022)	automatic
B	University of Aberdeen	Thomson and Reiter (2021)	Thomson and Reiter (2022)	human
	ADAPT, Charles Univ. Prague, Fed. Univ. of Minas Gerais	Dušek and Kasner (2020)	Huidrom et al. (2022)	human

Table 1: Overview of ReproGen submissions (tracks, teams, original papers, reproduction reports and types of reproduced evaluation measures).

in the REPROLANG 2020 shared task (Branco et al., 2020), all with consent and confirmation of willingness to support from the authors:

1. van der Lee et al. (2017): *PASS: A Dutch data-to-text system for soccer, targeted towards specific audiences*: 1 evaluation study; Dutch; 20 evaluators; 3 quality criteria; reproduction target: primary scores.
2. Dušek et al. (2018): *Findings of the E2E NLG Challenge*: 1 evaluation study; English; MTurk; 2 quality criteria; reproduction target: primary scores.
3. Qader et al. (2018): *Generation of Company descriptions using concept-to-text and text-to-text deep models: dataset collection and systems evaluation*: 1 evaluation study; English; 19 evaluators; 4 quality criteria; reproduction target: primary scores.
4. Santhanam and Shaikh (2019): *Towards Best Experiment Design for Evaluating Dialogue System Output*: 4 evaluation studies differing in experimental design; English; 40 evaluators; 2 quality criteria; reproduction target: intraclass correlation between studies.
5. Nisioi et al. (2017): *Exploring Neural Text Simplification Models*: 1 evaluation study; English; 3 evaluators; 2 metric scores; 4 human-evaluated quality criteria; reproduction target: primary scores.

Authors of original papers in Track A were asked (i) to complete a HEDS datasheet[3] (Shimorina and Belz, 2022) for their paper, (ii) to make available all code and other resources needed for the study, and (iii) to be available to answer questions and provide other help during the ReproGen participation period. Authors of reproduction papers were also asked to complete a HEDS datasheet.

We issued a call for participation in one or both tracks. Six teams registered for ReproGen, of which five teams submitted reproduction studies (for an overview, see Table 1).

We made available broad guidelines[4] to participating teams about how to report reproduction results, and provided light-touch review with comments and feedback on papers.

3 Participants and Submissions

Five submissions were received by the deadline on June 6, 2022. One submission was from the Netherlands, one from the UK, one from the US, one from Ireland, and one was a collaboration between groups in Czechia, Brazil and Ireland. Three of the teams participated in Track A (Braggaar et al., 2022; Popović et al., 2022; Arvan et al., 2022); the other two in Track B (Thomson and Reiter, 2022; Huidrom et al., 2022).

Two of the submissions reported a reproduction study of Nisioi et al. (2017), one of Santhanam and Shaikh (2019), and two reproduced own earlier work. All of the evaluated systems produced outputs in English. Popović et al. and Arvan et al. reproduced the human and metric-based evaluations of Nisioi et al. (2017)'s simplification systems, respectively, with Arvan et al. additionally exploring variations in the system code. Braggaar et al. reproduced inter-rater agreement and consistency measures for human evaluations of a dialogue system involving different rating scales studied by Santhanam and Shaikh (2019). Thomson and Reiter looked at reproducing an evaluation by error annotation from their own work on data-to-text generation, using different evaluation data samples, and Huidrom et al. reproduced human evaluations of Dušek and Kasner (2020)'s semantic error detection system for data-to-text generation. An overview of all submissions is provided in Table 1, and the properties of participating systems and studies are discussed in more detail in Section 5.

[3] https://forms.gle/MgWiKVu7i5UHeMNQ9

[4] https://reprogen.github.io/2022/submission/

Measurand(s)	Pearson's r	Spearman's ρ	mean % change		mean CV*
			+/-	abs	
Original study = Nisioi et al. (2017); reproduction study = Arvan et al. (2022), Repro 1:					
All Scores (2 systems × 2 metrics)	1	1	0	0	0
Original study = Nisioi et al. (2017); reproduction study = Arvan et al. (2022), Repro 2:					
All Scores (2 systems × 2 metrics)	1	0.8	-1.02	3.30	3.34
Original study = Nisioi et al. (2017); reproduction study = Arvan et al. (2022), Repro 3:					
All Scores (2 systems × 2 metrics)	1	0.8	0.63	3.19	3.16
Original study = Nisioi et al. (2017); reproduction study = Popović et al. (2022):					
All Scores (9 systems × 1 quality criterion)	0.766^{**}	0.787^{*}	40.16	85.82	8.98
Original study = Santhanam and Shaikh (2019); reproduction study = Braggaar et al. (2022):					
Likert (2 corr coeffs × 2 quality criteria × 1 scale)	0.95^{*}	0.81	25.37	25.37	21.88
RME (2 corr coeffs × 2 quality criteria × 1 scale)	-0.57	-0.54	-6.895	6.895	7.25
BME (2 corr coeffs × 2 quality criteria × 1 scale)	0	-0.07	8.55	8.55	8.15
BWS (2 corr coeffs × 2 quality criteria × 1 scale)	0.99^{**}	0.88	10.02	10.02	9.52
Readability (2 corr coeffs × 1 quality criterion × 4 scales)	-0.08	0.13	10.28	15.54	14.1
Coherence (2 corr coeffs × 1 quality criterion × 4 scales)	-0.16	0.1	8.25	9.88	9.1
ICC-C (1 corr coeffs × 2 quality criterion × 4 scales)	0.33	0.5	8.12	10.24	9.67
ICC-A (1 corr coeffs × 2 quality criterion × 4 scales)	-0.27	-0.22	10.41	15.18	13.73
All Scores (2 corr coeffs × 2 quality criteria × 4 scales)	0.01	0.16	9.26	12.71	11.699
Original study = Dušek and Kasner (2020); reproduction study = Huidrom et al. (2022), Repro 1:					
E2E (9 label counts × 1 system × 1 dataset)	0.98^{**}	0.91^{**}	1.15	18.9	19.62
WebNLG (8 label counts × 1 system × 1 dataset)	0.8^{**}	0.76^{*}	41.46	70.12	50.89
All Scores (8/9 label counts × 1 system × 2 datasets)	0.81^{**}	0.87^{**}	20.12	43.00	34.34
Original study = Dušek and Kasner (2020); reproduction study = Huidrom et al. (2022), Repro 2:					
E2E (9 label counts × 1 system × 1 dataset)	0.87^{**}	0.8^{*}	18.57	40.45	32.32
WebNLG (8 label counts × 1 system × 1 dataset)	0.82^{**}	0.54	18.97	58.17	46.86
All Scores (8/9 label counts × 1 system × 2 datasets)	0.84^{**}	0.66^{**}	18.76	48.79	39.16
Original study = Thomson and Reiter (2021); reproduction study = Thomson and Reiter (2022), Repro 1:					
Cond-copy (6 label counts × 1 system)	0.995	0.98	31.14	46.64	33.297
Doc-plan (6 label counts × 1 system)	0.91	0.90	-7.92	16.50	48.88
Hier-enc (6 label counts × 1 system)	0.85	0.70	70.67	109.9	76.07
All Scores (6 label counts × 3 systems)	0.89	0.88	33.6	60.10	52.75
Original study = Thomson and Reiter (2021); reproduction study = Thomson and Reiter (2022), Repro 2:					
Cond-copy (6 label counts × 1 system)	0.99^{**}	0.94^{*}	31.79	57.37	46.73
Doc-plan (6 label counts × 1 system)	0.92^{**}	0.82	-24.35	29.18	68.57
Hier-enc (6 label counts × 1 system)	0.83^{*}	0.72	73.86	136.64	88.70
All Scores (6 label counts × 3 system)	0.896^{**}	0.84^{**}	30.12	77.06	68.00

Table 2: Pearson's and Spearman's correlation coefficients, mean percentage change, and mean coefficients of variation (CV*), for the ReproGen'22 reproduction studies. For the correlation coefficients, $**$ = statistically significant at $\alpha = .01$, $*$ = statistically significant at $\alpha = .05$.

4 Results: Degree of Reproducibility

Table 2 shows summarising results for all submissions, or rather for every reproduction in every submission, i.e. nine original/reproduction study pairs, in terms of Pearson's r, Spearman's rho, mean percentage in/decrease, mean absolute percentage in/decrease, and the de-biased coefficient of variation, CV* (last column), following Belz (2022)'s Quantified Reproducibility Assessment (QRA) approach. The coefficient of variation (CV) is a standard measure of precision used in metrological studies to quantify reproducibility of measurements. Unlike mean and standard deviation, CV is not in the unit of the measurements, and captures the amount of variation there is in a set of n scores in a general way, providing a quantification of precision (degree of reproducibility) that is comparable across studies (Ahmed, 1995, p. 57). Note that all evaluation scales need to be shifted to start at zero, to ensure fair comparison across evaluations, because both percentage change and CV in general underestimate variation for scales with a lower end greater than 0. Rather than standard CV, QRA uses CV*, a de-biased version of CV (Belz, 2022), because sample size (number of repeat measures) tends to be very small in NLP.[5]

For the simpler reproductions in Table 2, where there were one or more systems and one or more conventional evaluation measures and the reproduction target was the overall scores in terms of the measure(s), Table 2 reports a single CV* figure in the last column, namely mean CV* over all systems and measures. For example, the fourth study in the table, Popović et al. (2022)'s reproduction of Nisioi et al. (2017), has an overall mean CV* of 8.98, computed from 9 individual CV* figures (9 systems $\times$ 1 quality criterion).

For the five remaining studies, we also show mean CV* for constituent subsets of individual CV* figures, grouped by rating scale, quality criterion and correlation coefficient for Braggaar et al. (2022)'s reproductions, by dataset for Huidrom et al. (2022)'s reproductions, and by system for Thomson and Reiter (2022)'s reproductions.

Columns 2 and 3 in Table 2 show Pearson's r and Spearman's rho, respectively, for the corresponding (sub)sets of original/reproduction score pairs, while Columns 4 and 5 show average percentage in/decrease from original to reproduction score pairs for each of the same (sub)sets.

We have ordered the studies by study-level mean CV* (lowest, i.e. best, first). Study-level mean CV* ranges from the perfectly reproduced metric scores in Arvan et al. (2022)'s first reproduction, to the particularly high CV* of Thomson and Reiter (2022)'s second reproduction of an error annotation. In the case of the former, the authors managed to obtain the exact same SARI and BLEU scores, by running the scripts for these metrics provided by the original authors on the system outputs also provided by the original authors. Thomson and Reiter (2022)'s reproductions involve error-type labelling of system outputs which appear to be a particularly difficult to reproduce form of evaluation: this was the reproduction target in the four studies in the lower half of Table 2 which have substantially higher (>34) resulting overall mean CV* than the other studies (<12).

Interpreting the mean CV* figures for subsets of results for Braggaar et al. (2022)'s reproduction is not simple. The original authors collected evaluations of a set of dialogue turns in context for 2 quality criteria (Readability and Coherence), repeated this for 4 different rating scales, and computed two measures of inter-rater similarity for each rating scale. The two measures of inter-rater similarity were the consistency intraclass correlation (ICC-C) and the agreement intraclass correlation (ICC-A). The mean CV* figures for Braggaar et al. (2022)'s reproduction in Table 2 thus measure the similarity between the ICC scores (automatically computed on the human ratings) in the original study and the ICC scores in the reproduction study, with the ICC scores themselves computed for each set of ratings (where each set corresponds to one of the scales combined with one of the quality criteria).

Under these circumstances, CV* expresses how reproducible (stable) the inter-rater consistency/agreement is from one experiment to a repetition of it, in other words whether inter-rater consistency/agreement is similarly high, or similarly low, across multiple repeats of the same evaluation. Because Braggaar et al. (2022) repeated the evaluations for four different rating instruments, the mean CV* figures can tell us whether this differs for different rating instruments (as well as for different evaluation criteria and inter-rater consistency/agreement measures). The answer is that it does differ substantially for different rating scales, is equally low for both evaluation criteria, and does differ for the two inter-rater measures.

Taking a slightly closer look, the inter-rater measures (ICCs) for the Likert scale have remarkably higher (worse) mean CV* than the other three scales, while nevertheless achieving strong Pearson's and Spearman's between individual ICC scores in the original and reproduction studies. While the ICCs for the other three scales have similarly good CV*, only the BWS scale also has strong Pearson's and Spearman's, with BME having no

[5] For full details of, and rationale for, using CV*, even for sets of just two scores, see Belz et al. (2022); Belz (2022).

correlation and RME having medium-strength *negative* correlation. This shows that CV* and Pearson's and Spearman's correlation coefficients provide complementary information in assessments of the similarity of original vs. reproduction scores. Looking at these in combination, it would seem that the BWS scale (best-to-worst ranking) achieves the most similar levels of inter-rater agreement and consistency across repeat studies.

From the results for Huidrom et al. (2022)'s reproductions, we can see that the error annotations produced for outputs for WebNLG data have worse CV* figures than for E2E (the difference is not just the data but also a subset of the error categories which are tailored to the data). Here, better CV* is aligned with better correlations.

Finally, the results for Thomson and Reiter (2022)'s reproductions show that the hierarchical encoder based data-to-text system produced outputs for which both mean CV* and correlations were worse on average than for the other two systems. However, this latter observation ought to be read with the proviso that each reproduction used a different sample from the three systems.

5 Comparison of Properties of Original vs. Reproduction Studies

Overall, all teams tried to follow the original studies as closely as possible (see also Discussion section below), but cohorts of human evaluators involved were different across all pairs of original and reproduction studies, except for the two reproductions by Thomson and Reiter (2022), and one of the two by Huidrom et al. (2022).

In this section, we summarise differences in each pair of studies and highlight the possible factors that might have led to different results in reproduction results. In the case of Track A contributions, our notes are based on the HEDS datasheets completed by both the original study authors and the shared task participants. For Track B, we describe differences as reported by the authors themselves in their original and reproduction reports, also consulting the HEDS sheets completed by them. See also Table 3 which lists some of the more fine-grained information for each study from the HEDS sheets.

5.1 Track A

Popović et al. (2022) reproduced the human evaluation reported by Nisioi et al. (2017), and point out the following differences that might have influenced the reproduction: evaluator background (native language, profession, experience with text simplification evaluation), evaluator assignments to texts, and experimental setup (e.g. whether evaluators were allowed to ask questions about guidelines), all of which were not reported for the original study, and not obtainable from the original authors.

Arvan et al. (2022) also reproduced Nisioi et al. (2017)'s work, but just the metric scores. They focused on exploring different ways of obtaining the outputs to be evaluated (having discovered several substantial issues with the original code): (a) using the same outputs, (b) regenerating outputs with the same code, and (c) regenerating outputs with corrected code. They found an "extreme level of resilience [to such differences that] is, in fact, quite alarming," which is reflected in the low mean CV* figures which as it happens also reflect variation from different versions of SacreBLEU.

Braggaar et al. (2022)'s reproduction of Santhanam and Shaikh (2019) used crowdsourced human evaluation like the original study, but on a different platform: Qualtrics and Prolific in the reproduction study, and MTurk in the original. Due to platform feature restrictions, questionnaire layouts were not exactly the same across the two studies. As for the inter-rater measures, Braggaar et al. wrote their own code to compute ICC scores, since it was not provided by the original authors.

5.2 Track B

Huidrom et al. (2022) carried out two reproduction studies of Dušek and Kasner (2020): the first one with the same two evaluators and the second one with two new evaluators. The main difference between the original and reproduction studies lies in error annotation guidelines and output assignments to evaluators. While the original study did not formalise the annotation guidelines and performed evaluation based on common understanding developed between the two evaluators, for the reproduction studies, instructions for applying the error annotation scheme were created and used. The original study also did not record which texts were evaluated by which annotator, so the reproduction studies randomly assigned annotators to evaluated texts.

The main difference between the two reproductions and the original work addressed by Thomson and Reiter (2022) was the use of different samples of outputs albeit from the same larger test set. This did result in substantial differences between results, as we shall see below.

5.3 Study properties and reproducibility

Table 3 provides an overview of the five ReproGen'22 submissions in terms of the quality criteria assessed in the evaluations and the properties of the evaluation design. The first column identifies the study and criteria, the last column shows the corresponding mean study-level and mean criterion-level CV*. The remaining columns show seven properties of each study/criterion, as per the HEDS datasheets; column headings identify HEDS question number (for explanation of each see table caption). The lower half of the table shows the corresponding overview of study/criterion properties from ReproGen'21, for ease of comparison.

In the ReproGen'22 studies, annotation-based evaluation (4.3.8=Anno) is clearly associated with lower reproducibility. Evaluations which involve assessment of content alone (4.1.2=Cont) also tend to have worse reproducibility. Assessing evaluation items relative to a system input (4.1.3=RtI) is also associated with lower reproducibility for the bottom three studies (where comparison of outputs to inputs is far more complex than a straightforward is-it-simpler decision as in *Nisioi et al/Popovic et al*). Finally, correctness assessment (4.1.1=Corr) is also associated with lower reproducibility. For those of these properties that were present in ReproGen'21, the tendencies are the same.

6 Discussion

In metrological terms, a *repeatability* assessment keeps all conditions under which a measurement was taken the same, whereas a *reproducibility* assessment varies some of them. Strictly speaking, only the first reproduction by Arvan et al. (2022) can be considered a repeatability assessment, as it keeps all conditions exactly the same. All other ReproGen'22 reproductions were human evaluations, and for these, conditions can only be the same if the same evaluators are used again. One of the studies (the first reproduction by Huidrom et al. (2022)) did use the same evaluators, but instructions were written down and used for the first time instead of evaluators conferring.

Nevertheless, all studies tried to keep things as much as possible the same. One study which looked at automatic metrics (only) (Arvan et al., 2022) went beyond reusing system outputs provided by original authors, (a) regenerating outputs with unchanged author-provided code, and (b) regenerating outputs with a retrained system, including with a substantial correction to the code. Interestingly, evaluation results were very similar in all versions where outputs were regenerated, including switching word2vec embeddings on/off.

For the studies looking at human evaluations, new cohorts of evaluators were rarely able to achieve low CV* scores, generally only in very simple assessments. Pearson and Spearman correlations were generally better, with some exceptions where comparison was between inter-rater similarity measures, rather than evaluation scores (*Santhanam & Shaikh/Braggaar et al*).

We saw that correlation coefficients and mean CV* often but not always give the same indication of similarity between a set of original and reproduction scores. For example, the results in Table 2 for Braggaar et al. (2022)'s reproduction of Santhanam and Shaikh (2019) show that for Likert we have high correlation but poor CV*, for RME and BME, correlation is inverse or absent, but CV* is good, and for BWS both are good. For all other studies, better CV* always means better correlations.

This year we had a few (new) firsts at ReproGen, in addition to automatic metrics being reproduced for the first time: e.g. Thomson and Reiter (2022) investigated the effect of swapping out the data sample (from the same superset), while keeping all other conditions the same including annotators. As the sample size is fairly small, and differed in size between original study and the two reproductions, it's perhaps not surprising that error label counts varied substantially between studies.

Some of the ReproGen'22 participants' reports mention less than ideal support from original authors during reproductions, despite the fact that all original authors had agreed to support and help with ReproGen'22 reproductions. Of course, such help is essential to testing reproducibility, and in future shared tasks, we will consider the option of obtaining more of the resources and information prior to the start of the shared task.

7 Conclusions

We first proposed the ReproGen shared task at Generation Challenges 2020[6] (Belz et al., 2020) and, taking into account feedback received, developed it

[6] INLG'20, Dublin.

ReproGen 2022									
Studies, **measurands**	**3.1.1**	**3.2.1**	**4.3.4**	**4.3.8**	**4.1.1**	**4.1.2**	**4.1.3**	**scores /item**	**mean CV***
Nisioi et al / Arvan et al 1–3							EFoR		2.17
SARI	∼50	NA/NA	[0..1]	DQE	Good	Form	+RtI	NA	2.34
BLEU	∼50	NA/NA	[0..1]	DQE	Good	Form	EFoR	NA	1.99
Nisioi et al / Popovic et al, **Simplicity**	70	3/3	-2, -1, 0, 1, 2	DQE	Feature	Both	RtI	2	8.98
Santhanam & Shaikh / Braggaar et al									11.7
ICC for Readability									14.1
Likert scale	50	160/163	1–6	DQE	Good	Both	iiOR	1	28.19
Magnitude est. (stdval=100)	50	160/163	100	DQE	Good	Both	iiOR	1	11.18
Magnitude est. (stdval=var)	50	160/163	100	DQE	Good	Both	iiOR	1	6.93
Best-to-worst ranking	50	160/163	4! rankings	RQE	Good	Both	iiOR	1	10.1
ICC for Coherence									9.3
Likert scale	50	160/163	1–6	DQE	Good	Cont	iiOR	1	15.58
Magnitude est. (stdval=100)	50	160/163	100	DQE	Good	Cont	iiOR	1	3.31
Magnitude est. (stdval=var)	50	160/163	100	DQE	Good	Cont	iiOR	1	9.38
Best-to-worst ranking	50	160/163	4! rankings	RQE	Good	Cont	iiOR	1	8.93
Dusek & Kasner / Huidrom et al 1&2									36.75
Label counts from correctness annotations	200	2/2	3 labels	Anno	Corr	Cont	RtI	1	18.11
Label counts from error type annotations	200	2/2	6/5 labels	Anno	Corr	Cont	RtI	1	46.92
Thomson & Reiter / Thomson & Reiter 1 & 2, **Label counts from error type annotations**	13, 10	3/3	6 labels	Anno	Corr	Cont	RtI	3	68
ReproGen 2021									
Lee et al./Mille et al.									11.89
Stance ID Acc	10	20/20	stance A, stance B	output classif	Feature	Both	EFoR	20	6.11
Clarity S3 ('Understandability')	20	20/20	1–7	DQE	Good	Both	iiOR	20	12.03
Clarity S4 ('Clarity')	20	20/20	1–7	DQE	Good	Both	iiOR	20	14.61
Fluency S1 ('Grammaticality')	20	20/20	1–7	DQE	Corr	Form	iiOR	20	18.3
Fluency S2 ('Readability')	20	20/20	1–7	DQE	Good	Both	iiOR	20	13.71
Popović/Popović & Belz									29.22
Comprehension Minor	557, 279, 467	7/7	2 labels	Anno	Good	Both	iiOR	2	22.14
Comprehension Major		7/7		Anno	Good	Both	iiOR	2	38.23
Adequacy Minor		7/7	3 labels	Anno	Corr	Cont	RtI	2	17.83
Adequacy Major		7/7		Anno	Corr	Cont	RtI	2	38.67
Qader et al./Richter et al.									22.16
Information Coverage	30	19/19	1–5	DQE	Corr	Cont	RtI	1	34.04
Information Non-redundancy	30	19/19	1–5	DQE	Good	Cont	iiOR	1	19.11
Semantic Adequacy	30	19/19	1–5	DQE	Corr	Cont	iiOR	1	20.4
Grammatical Correctness	30	19/19	1–5	DQE	Corr	Form	iiOR	1	15.09
Mahamood et al./Mahamood, Binary Preference Strength	2†	25‡/11	-3..+3	RQE	Good	Both	EFoR	25/11	72.34

Table 3: Summary of some properties from HEDS datasheets provided by ReproGen participants (in some cases corrected by organisers. 3.1.1 = number of items assessed per system; 3.2.1 = number of evaluators in original/reproduction experiment; 4.3.4 = List/range of possible responses; 4.3.8 = Form of response elicitation (DQE: direct quality estimation, RQE: relative quality estimation, Anno: evaluation through annotation); 4.1.1 = Correctness/Goodness/Features; 4.1.2 = Form/Content/Both; 4.1.3 = each output assessed in its own right (iiOR) / relative to inputs (RtI) / relative to external reference (EFoR); scores/item = number of evaluators who evaluate each evaluation item; mean CV*. † considering texts with and without hedges to be the two systems being compared. ‡ subset of 32 evaluators from original studies: 14 native + 11 fluent speakers.

into the two iterations of ReproGen, 2021 and 2022, the latter reported in the present paper. ReproGen was intended as a testbed for an NLP-wide shared task on reproduction, and in 2023 we intend to run an expanded version, the ReproHum Shared Task on Reproducibility of Evaluation Results in NLP, initially for just human evaluations.

We have gained some important insights from ReproGen, in particular with regard to what kind of properties of evaluations tend to increase or decrease degree of reproducibility. Perhaps not surprisingly, it is very clear that the lower the cognitive load on evaluators while making individual assessments, the better reproducibility.

In a research culture that prizes leaderboard success, it was always going to be difficult to incentivise people to carry out tasks that are basically just good scientific hygiene, but we hope we have made a contribution to raising awareness of the importance of having reproducible evaluations, and of testing our methods for reproducibility. After all, how else are we going to know for sure that one approach is better than another.

Acknowledgments

We thank the authors of the five original papers that were up for reproduction in Track A. And of course the authors of the reproduction papers, without whom there would be no shared task.

Our work was carried out as part of the ReproHum project on Investigating Reproducibility of Human Evaluations in Natural Language Processing, funded by EPSRC (UK) under grant number EP/V05645X/1.

Popović's work is directly funded by the ADAPT SFI Centre for Digital Media Technology which is funded by Science Foundation Ireland through the SFI Research Centres Programme and is co-funded under the European Regional Development Fund (ERDF) through Grant 13/RC/2106. Both Popović and Belz also benefit in other ways from being members of ADAPT.

References

S. E. Ahmed. 1995. A pooling methodology for coefficient of variation. *Sankhyā: The Indian Journal of Statistics, Series B*, pages 57–75.

Mohammad Arvan, Luís Pina, and Natalie Parde. 2022. Reproducibility of exploring neural text simplification models: A review. In *Proceedings of the 15th International Conference on Natural Language Generation: Generation Challenges*, Waterville, Maine, USA. Association for Computational Linguistics.

Anya Belz. 2022. A metrological perspective on reproducibility in nlp. *Computational Linguistics*, 48.

Anya Belz, Shubham Agarwal, Anastasia Shimorina, and Ehud Reiter. 2020. ReproGen: Proposal for a shared task on reproducibility of human evaluations in NLG. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 232–236.

Anya Belz, Shubham Agarwal, Anastasia Shimorina, and Ehud Reiter. 2021. A systematic review of reproducibility research in natural language processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 381–393, Online. Association for Computational Linguistics.

Anya Belz, Maja Popović, and Simon Mille. 2022. Quantified reproducibility assessment of nlp results. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16–28.

Anouck Braggaar, Frédéric Tomas, Peter Blomsma, Saar Hommes, Nadine Braun, Emiel van Miltenburg, Chris van der Lee, Martijn Goudbeek, and Emiel Krahmer. 2022. A reproduction study of methods for evaluating dialogue system output: Replicating Santhanam and Shaikh (2019). In *Proceedings of the 15th International Conference on Natural Language Generation: Generation Challenges*, Waterville, Maine, USA. Association for Computational Linguistics.

António Branco, Nicoletta Calzolari, Piek Vossen, Gertjan Van Noord, Dieter van Uytvanck, João Silva, Luís Gomes, André Moreira, and Willem Elbers. 2020. A shared task of a new, collaborative type to foster reproducibility: A first exercise in the area of language science and technology with REPROLANG2020. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5539–5545, Marseille, France. European Language Resources Association.

Ondřej Dušek and Zdeněk Kasner. 2020. Evaluating semantic accuracy of data-to-text generation with natural language inference. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 131–137, Dublin, Ireland. Association for Computational Linguistics.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2018. Findings of the E2E NLG challenge. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 322–328, Tilburg University, The Netherlands. Association for Computational Linguistics.

Rudali Huidrom, Ondřej Dušek, Zdeněk Kasner, Thiago Castro Ferreira, and Anya Belz. 2022. Two reproductions of a human-assessed comparative evaluation of a semantic error detection system. In *Proceedings of the 15th International Conference on Natural Language Generation: Generation Challenges*, Waterville, Maine, USA. Association for Computational Linguistics.

Chris van der Lee, Emiel Krahmer, and Sander Wubben. 2017. PASS: A Dutch data-to-text system for soccer, targeted towards specific audiences. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 95–104, Santiago de Compostela, Spain. Association for Computational Linguistics.

Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P. Dinu. 2017. Exploring neural text simplification models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 85–91, Vancouver, Canada. Association for Computational Linguistics.

Maja Popović, Sheila Castilho, Rudali Huidrom, and Anya Belz. 2022. Reproducing a manual evaluation of simplicity in text simplification system outputs. In *Proceedings of the 15th International Conference on Natural Language Generation: Generation Challenges*, Waterville, Maine, USA. Association for Computational Linguistics.

Raheel Qader, Khoder Jneid, François Portet, and Cyril Labbé. 2018. Generation of company descriptions using concept-to-text and text-to-text deep models: dataset collection and systems evaluation. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 254–263, Tilburg University, The Netherlands. Association for Computational Linguistics.

Sashank Santhanam and Samira Shaikh. 2019. Towards best experiment design for evaluating dialogue system output. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 88–94, Tokyo, Japan. Association for Computational Linguistics.

Anastasia Shimorina and Anya Belz. 2022. The human evaluation datasheet: A template for recording details of human evaluation experiments in NLP. In *Proceedings of the 2nd Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 54–75, Dublin, Ireland. Association for Computational Linguistics.

Craig Thomson and Ehud Reiter. 2021. Generation challenges: Results of the accuracy evaluation shared task. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 240–248, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Craig Thomson and Ehud Reiter. 2022. The accuracy evaluation shared task as a retrospective reproduction study. In *Proceedings of the 15th International Conference on Natural Language Generation: Generation Challenges*, Waterville, Maine, USA. Association for Computational Linguistics.

Two Reproductions of a Human-Assessed Comparative Evaluation of a Semantic Error Detection System

Rudali Huidrom
ADAPT/DCU, Dublin, Ireland
rudali.huidrom@adaptcentre.ie

Ondřej Dušek and **Zdeněk Kasner**
Charles University, Prague, Czechia
{odusek,kasner}@ufal.mff.cuni.cz

Thiago Castro Ferreira
Federal University of Minas Gerais, Brazil
Thiagocf05@ufmg.br

Anya Belz
ADAPT/DCU and University of Aberdeen
anya.belz@adaptcentre.ie

Abstract

In this paper, we present the results of two reproduction studies[1] for the human evaluation originally reported by Dušek and Kasner (2020) in which the authors comparatively evaluated outputs produced by a semantic error detection system for data-to-text generation against reference outputs. In the first reproduction, the original evaluators repeat the evaluation, in a test of the *repeatability* of the original evaluation. In the second study, two new evaluators carry out the evaluation task, in a test of the *reproducibility* of the original evaluation under otherwise identical conditions.[2] We describe our approach to reproduction, and present and analyse results, finding different degrees of reproducibility depending on result type, data and labelling task. Our resources are available and open-sourced[3].

1 Introduction

Reproduction studies are garnering growing interest in natural language processing (NLP), most recently as the subject of shared tasks (Branco et al., 2020; Belz et al., 2021). The importance of ensuring good levels of reproducibility in NLP work is increasingly recognised, and approaches to defining and assessing reproducibility are emerging (Cohen et al., 2018; Belz et al., 2022). With this paper, we add to the growing body of reproduction studies by tackling a particularly hard case for reproducibility assessment, namely error analysis that involves identifying which of two disagreeing systems is making error(s), and further classifying the types of errors being made.

We perform two reproductions, one involving the same evaluators as in the original study, one involving new evaluators. The former can be seen as a test of the repeatability of the original study, where nothing is changed except the point in time, and the latter as a test of its reproducibility where the reproduction differs from the original study in some specfied respect(s), here the evaluator cohort.[4]

Below we start by describing the original study and outlining our approach to reproduction. Next we describe our two reproductions and present an analysis which examines three types of results from the evaluations, applying different tools for measuring similarity in each case. We finish with a discussion of the reasons behind and possible mitigation strategies for what is, on the face of it, a mostly poor set of reproducibility results.

2 Original Evaluation

2.1 Semantic error detection method

Dušek and Kasner (2020) presented an automatic method for semantic error detection (SED) in data-to-text generation (see Figure 1 for example data/text pairs) based on textual entailment checking. The basic idea is to trivially (and automatically) map each triple in the input meaning representation (top part of each example in Figure 1) to a text representation using simple generation templates, and then to check whether input and output entail each other. If the input does not entail the output, a hallucination error is diagnosed (some content in the output is not present in the input); if the output does not entail the input, it is taken to

[1] Carried out as part of the ReproGen 2022 shared task.

[2] With the proviso that instructions had to be created for the reproductions.

[3] https://github.com/RHuidrom/reprogen22_dusek_and_kasner_2020.git

[4] See also Section 4 re new instructions.

Proceedings of the 15th International Conference on Natural Language Generation: Generation Challenges, pages 52 - 61
July 17-22, 2022 ©2022 Association for Computational Linguistics

MR: Atlantic City, New Jersey | country | United States
United States | capital | Washington, D.C.

NLG system output: atlantic city, new jersey comes from the united states where the capital is washington, d.c.

SED label: Reference label (derived from human rating): *not OK*; NLI-SED label: *OK*

Error analysis annotation: *other* (both system and reference are incorrect), *bad sentence*

MR: FC Dinamo Batumi | manager | Levan Khomeriki
Aleksandre Guruli | club | FC Dinamo Batumi

NLG system output: fc dinamo batumi was at levan khomeriki and manages aleksandre guruli.

SED label: Reference label (derived from ruman rating): *not OK*; NLI-SED: *OK*

Error analysis annotation: *reference correct*, *unjustified OK*

Figure 1: Two examples each consisting of a meaning representation (MR); an NLG system output (from WebNLG 2017); two SED labels (the reference error label derived from the WebNLG 2017 human ratings, and the output from the NLI-SED system); and correctness and error label annotations as produced in one of our reproductions.

mean an omission error (some content in the input is not present in the output). If input and output do entail each other, then the output is taken to be error-free.

For the entailment checking, the method used a pretrained RoBERTa model (Liu et al., 2019) from the Transformers library (Wolf et al., 2020) finetuned on the MultiNLI dataset (Williams et al., 2018). The model (referred to as the NLI-SED system below) produces probability estimates for the three possible outputs: contradiction, neutral and entailment. To pass an entailment check, the entailment probability simply has to be higher than the neutral and contradiction probabilities.

When checking whether the output entails the input, Dušek and Kasner paired the simple text representation of each triple with all of the output text and performed the entailment check on each pair individually. When checking whether the input entails the output, the simple text representations of all input triples were concatenated and paired with the output text in a single entailment check.

Ultimately, the output from Dušek and Kasner's NLI-SED system is one of the following: *OK, omission, hallucination, hallucination+omission.*

2.2 Manual evaluation of the SED method

The original study that is the subject of reproduction in this paper is a manual evaluation in which Dušek and Kasner compared the SED labels obtained from their NLI-SED system for data from the E2E (Dušek et al., 2020) and the WebNLG (Gardent et al., 2017) shared tasks with reference labels. They performed an error analysis on a sample of 100 cases where NLI-SED system generated label and reference label disagreed. In each case, they decided which was right and which was wrong, and additionally selected labels indicating the likely source(s) of any error(s), from among six different error labels for E2E, and five for WebNLG (labels as described for each dataset below).[5] Finally, in each case, the authors also provided unstructured notes which explain their annotations.

E2E		
	Slot-error	
Counts of	Script	NLI-SED
OK	33	54
omission	42	32
hallucination	17	7
omiss+halluc	8	7

WebNLG		
	Human	
Counts of	Ratings	NLI-SED
OK	45	54
not OK	55	46

Table 1: Counts for different SED labels as per the reference labels (produced by the slot-error script in the case of E2E, and derived from human ratings in the case of WebNLG), and the NLI-SED system.

2.2.1 E2E

For E2E, reference labels (*OK, omission, hallucination, hallucination+omission*) were available from the E2E shared task where they were generated by the organisers with what they termed a slot-error script based on regular expression matching, with patterns informed by a subset of the E2E development set.

In the sample Dušek and Kasner annotated in their error analysis, the counts for reference labels produced by the slot-error script and for the NLI-SED system generated labels look as shown in Table 1. In addition, there was partial agreement between the reference labels from the slot-error script

[5] The error classes and raw counts from the annotations we use in this paper were not reported in the original publication, but were instead mapped to less fine-grained findings.

Counts of	E2E					Counts of	WebNLG				
	Dušek & Kasner 2020	Repeat. Test (A1+A2)	CV^*	Reprod. Test (A3+A4)	CV^*		Dušek & Kasner 2020	Repeat. Test (A1+A2)	CV^*	Reprod. Test (A3+A4)	CV^*
ref correct	34	36	5.697	41	18.611	ref correct	51	38	29.126	59	14.502
SED correct	45	48	6.432	44	2.240	SED correct	42	40	4.863	35	18.127
other	18	16	11.730	15	18.127	other	7	15	72.510	6	15.339
[eatType]	5	6	18.127	6	18.127	[bias-templ]	22	16	31.484	5	125.549
[priceRange]	30	33	9.495	28	6.876	[val-format]	7	3	79.760	10	35.188
[famFriend]	10	13	26.019	8	22.156	[bad-sent]	14	27	63.225	10	33.234
[f-halluc]	8	5	46.016	22	93.054	[unj-OK]	8	25	102.722	28	110.778
[f+omiss]	16	11	36.926	24	39.880	[unj-notOK]	15	19	23.460	12	22.156
[f+halluc]	17	20	16.168	8	71.784						

Table 2: QRA assessment of correctness and error label counts (type *i* results), on the ***combined*** annotations (in Repeatability Test, half randomly taken from each original annotator; in Reproducibility Test, half randomly taken from each of the new annotators).

and NLI-SED system in 12 cases, where both detected an omission (and one additionally detected a hallucination). There was no partial agreement on hallucinations.

In Dušek and Kasner's annotations, the script-generated labels were deemed to be correct (and the NLI-SED system's prediction wrong) in 34 out of 100 cases, and the NLI-SED system's predictions were deemed correct (and the script wrong) in 45 cases. In 18 cases, either both the script-generated labels and the NLI-SED system's prediction were wrong or the evaluators were unable to decide. These numbers are also included in the upper part of the first Dusek & Kasner column in Table 2.[6]

The six error class types for the E2E error annotations were as listed below. Note that the descriptions and definitions given here were created as part of our reproductions. The implications of creating new instructions for a reproduction are discussed in Section 5.

Each error class represents a different possible source of an error made by Dušek and Kasner's NLI-SED system or the slot-error script, and as many error classes were selected as applied in each case, in some cases none were selected (frequencies are shown in the lower part of the first Dusek & Kasner column in Table 2). These error classes tend to apply predominantly to either the NLI-SED system or the slot-error script, indicated by underlines below. The short labels in square brackets are used to refer to each class in the results tables below.

1. **Error related to *eatType=restaurant* slot value [eatType]:** The incorrect SED label (produced either by the NLI-SED system or the slot error script) is likely caused by something involving the slot/value pair eatType=restaurant, e.g. not detecting a hallucination when the eatType slot is not in the input, but the output mentions a restaurant.

2. **Error related to *priceRange* slot [priceRange]**: The incorrect SED label (produced by either the NLI-SED system or the slot error script) makes an error related to the priceRange slot, e.g. incorrectly identifying a hallucination in the priceRange slot, when the price range information has in fact been correctly verbalised.

3. **Error related to *familyFriendly attribute* [famFriend]**: The incorrect SED label (produced by either the NLI-SED system or the slot error script) makes an error related to the familyFriendly slot, e.g. incorrectly identifying an omission when the information has in fact been correctly verbalised.

4. **Other false negative hallucination ('off-topic blabber') [f-halluc]**: The incorrect SED label (produced by either the NLI-SED system or the slot error script) fails to detect a hallucination (unrelated to E2E slots) present in the verbalisation.

5. **Other false positive omission ('unjustified omission') [f+omiss]:** The incorrect SED label (produced by either the NLI-SED system or the slot error script) wrongly detects an omission in the verbalisation.

6. **Other false positive hallucination ('unjustified hallucination') [f+halluc]:** The in-

[6]Numbers don't add up to 100 because of missing annotations.

correct SED label (produced by either the NLI-SED system or the slot error script) wrongly detects a hallucination in the verbalisation.

2.2.2 WebNLG

For the WebNLG sample, Dušek and Kasner (2020) created reference SED labels by mapping human quality judgements on a 1–3 scale (crowdsourced for WebNLG 2017) to *OK* (>=2.5) and *not OK* (<2.5). The crowdscourced quality judgements exist for a subset of 223 inputs from the WebNLG 2017 test set each paired with 10 different NLG outputs from participating systems. SED labels were taken to differ unless they were both *OK*, or one was *not OK* and the other was one of *omission, hallucination, omission+hallucination*).[7]

In the sample Dušek and Kasner annotated in their error analysis, the counts for reference labels derived from human ratings and for the NLI-SED system generated labels look as shown in the lower half of Table 1. The *not OK* label count of 46 shown for the NLI-SED system breaks down into 29 cases of omission, 13 cases of hallucination, and 4 cases of combined omission+hallucination).

In Dušek and Kasner's annotations, the reference label (the mapped human rating) was deemed to be correct (and the NLI-SED system prediction wrong) in 51 out of 100 cases, and the NLI-SED system prediction was deemed correct (and the reference label wrong) in 42 cases. In 7 cases either both reference label and NLI-SED system prediction were deemed wrong or the evaluators were unable to decide. These numbers are also shown in the top part of the second Dusek & Kasner column in Table 2)

The five error class labels for the WebNLG error annotations were as shown below. Each item may have more than one or none of these. The first three classes indicate, where possible, the likely source of the error in the SED label that was deemed wrong (produced by either the NLI-SED system or the mapped human ratings). Otherwise one of the last two will apply. Label frequencies are shown in lower part of the second Dusek & Kasner column in Table 2. NB: each error class predominantly applies to the underlined method.

1. **NLI-SED system error due to poor triple-to-text input mapping ('biased template') [bias-templ]**: Incorrect NLI-SED system label due to an inappropriate template being used in mapping the input triples to text (templates tend to work better for certain subject/object values, but the same template is used for all cases with a given predicate), resulting in ungrammatical sentences or even shift in meaning.

2. **NLI-SED system failure to recognise subject or object semantic equivalence ('value format') [val-format]**: In the verbalisation, the formatting of a subject or object differs from the input to the extent where the NLI check in the NLI-SED system failed to recognise them as equivalent in meaning (e.g. metres vs. kilometres).

3. **Incorrect reference SED label due to disfluent verbalisation ('bad sentence') [bad-sent]**: The human reference label, mapped to *not OK*, is incorrect, and this is likely because the human rating was affected by the disfluency/ungrammaticality of the verbalisation.

4. **Other cases of incorrect *OK* label ('unjustified OK') [unj-OK]**: The incorrect label (from either the human references or the NLI-SED system) is OK, and none of the above apply.

5. **Other cases of incorrectly identifying a semantic error ('unjustified not OK') [unj-notOK]**: the incorrect label (from either the human references or the NLI-SED system) either literally a *not OK* label, or one of *omission, hallucination, omission+hallucination*, *not OK*, and none of the above apply.

2.3 Reproduction targets

In the present context, there are four types of results that are candidates for reproduction: (i) *single numeric values* for the same measure (e.g. the overall number of times the SED label produced by the NLI-SED system was correct); (ii) *sets of numeric values for a set of related measures* (e.g. the numbers of input/output pairs annotated with each error label); (iii) *sets of discrete labels* from the same task (e.g. the correct/incorrect labels assigned to the NLI-SED system labels and the reference SED labels); and (iv) *unstructured textual comments* from the same task (here, the evaluator notes for each of the SED-label error annotations).

In order to draw conclusions regarding repeatability and reproducibility, results from original and

[7] One case of agreement, where the mapped human label was *Not OK* and the NLI-SED system produced *omission*, was included by mistake.

reproduction studies need to be compared, and how they're compared depends on which type (*i*, *ii*, *iii*, or *iv* above) a result is. We pick this up again in Section 3; here we list the results of types (i)–(iii) from Dušek and Kasner that we attempt to reproduce in our two reproduction studies (the free textual comments (type *iv*) were too disparate for us to try to compare):

i. Single numeric values (overall counts):
   a. Count of reference correct;
   b. Count of NLI-SED system correct;
   c. Count of both reference and NLI-SED system incorrect or evaluators couldn't decide;
   d. Count of individual error labels, six different labels for E2E, five for WebNLG (see Tables 2 and 3 for short-form labels).

ii. Sets of related numeric values:
   a. Set of counts of *Correctness* labels (i.a–i.c above);
   b. Set of counts of SED *Error-class* labels (i.d above).

iii. Sets of categorical values:
   a. Set of *Correctness* labels (one of {*NLI-SED, reference, neither*}; exactly one label per evaluation item);
   b. Set of SED *Error-class* labels; multiple labels per evaluation item).

3 Approach to Reproduction

For results of type *i* above (where we have single measured quantity values to compare), we follow the quantified reproducibility assessment (QRA) approach (Belz et al., 2022) which means (a) identifying and documenting (as we do in the attached HEDS sheet) the properties of evaluation experiments as standardised attribute-value pairs (*conditions of measurement* in QRA terms); and (b) computing the small-sample coefficient of variation (CV*) over compared quantity values, as the measure of degree of reproducibility. QRA assessment results are shown in Tables 2 and 3 and discussed in Section 4.1.

For type *ii* results (Table 4, Section 4.2) we compute Pearson's r for pairwise correlation.

For results of type *iii*, we compute Fleiss's kappa (the multi-evaluator generalisation of Scott's pi) on aligned sets of categorical values where we have exactly one label per item (which is the case for the correctness labels), and Krippendorff's alpha where we have multiple labels per item (which is the case for the error labels). Results are shown in Table 5 and discussed in Section 4.3.

4 Two Reproductions

Our two reproduction studies repeated the Dušek and Kasner evaluations as closely as possible, the first using the same evaluators, the second using different evaluators. There were two complicating factors, necessitating the use of (i) new evaluator instructions, and (ii) a different way of allocating evaluators to evaluation items.

The reason for the difference in evaluator instructions is that in the original work, instructions were not written down, a shared understanding being evolved in the course of the work instead. In order to repeat the evaluations with new evaluators less familiar with the work, instructions had to be written down and shared which were then used in all reproductions. The instructions are included verbatim in the appendix.

Regarding evaluator allocation, in the original work, the work was shared between the two authors who each did about half of E2E and half of WebNLG, but it was not recorded who did which ones. For that reason, we decided to get the evaluators in the reproduction studies (the original two authors, and authors 4 and 5 of this paper) to each annotate all 100 E2E items and all 100 WebNLG items, and then we randomly selected half from each evaluator pair for a like-for-like comparison (in the tables below we call this the *combined* set of annotations). Assessing the similarity between these combined results and the original results forms the main body of our reproduction study: type *i* results are shown in Table 2, type *ii* in Table 4, and type *iii* in Table 5.

Additionally, we compare the four complete sets of annotations with the original annotations individually, for the single numeric values (type *i* results) from E2E and WebNLG only (Table 3).

Each evaluator worked on a separate Google spreadsheet in the exact same format as in the original study,[8] except that in the repeatability test which involved the original annotators, we shuffled the order of evaluation items to avoid inadvertent

[8] A blank copy of the evaluation sheet can be found here: https://docs.google.com/spreadsheets/d/1_4DZVu6Ow-9kZOjQJCjg2qZCLUt435Og

recall of original annotations.

4.1 Comparison of type *i* results

The results from the QRA test on label counts (type *i* results, i.e. single numeric values) for the *combined* annotations are shown in Table 2. The counts from the original study are in the Dusek & Kasner column in the left half of the table for E2E,[9] and in the right half for WebNLG. Counts from the repeatability (original annotators) and reproducibility (new annotators) tests and the corresponding CV* scores are shown in columns labeled as such in each half.

Looking at correctness label counts for E2E (rows 1–3, left half), the original annotators (A1+A2) are on the whole better able to reproduce their own results than the new annotators (A3+A4), which is as expected. However, if we look at the corresponding figures for WebNLG (right half) it turns out that here, the *new* annotators reproduce the original counts more closely. In terms of differences between correctness labels, the 'SED correct' counts are overall easiest to reproduce.

Moving on to error class counts, for E2E, CV* is broadly the same for original/new annotators for error classes relating to specific slots (eatType, priceRange, famFriend), but considerably worse for the new annotators for the remaining, more generic, error classes. For WebNLG, it is a more mixed picture: the new annotators reproduce the original counts better than the original annotators for error classes val-format and bad-sent, worse for error classes bias-templ and unj-OK, and equally well for error class unj-notOK.

Table 3 sheds additional light on the reproducibility of the individual category counts, by looking at the larger sets of 400 new annotations compared to the original 100, for each of E2E and WebNLG, thus providing a larger sample for, and more reliable estimates from, CV*. The two halves of the table are structured as in Table 2.

The results in Table 3 provide overall estimates across all five sets of annotations of the degree of reproducibility of the individual types of counts. For both E2E and WebNLG, correctness label counts are far easier to reproduce than error class counts which is as expected. Beyond that, again the 'SED correct' count is the most reproducible for both E2E and WebNLG. For E2E, counts for errors related to the priceRange slot (priceRange) are easiest to reproduce, whereas false negative hallucinations (f-halluc) are by far the hardest. For WebNLG, counts for bad-sent (bad grammar/fluency likely leading to 'not OK' label) are easiest to reproduce, and counts for val-format (phrases that are semantically equivalent not being recognised as such) are by far the hardest.

To put these CV* numbers into perspective, in the first ReproGen Shared Task, all except one (an outlier above 70) of the CV* scores for human evaluations were below 39 (Belz et al., 2020).

4.2 Comparison of type *ii* results

The results in the preceding section showed how reproducible correctness and error label *counts* were, for each count type independently, and regardless of whether labels were attached to the same items. In this section, we look at sets of counts in conjunction, and in the next section we look at labels as attached to evaluation items. Table 4 presents results from correlation tests on the set of all three correctness labels (top half), and on the set of all five (WebNLG) or six (E2E) error labels (sets of related numeric values, i.e. type *ii* results). Here too we are using the *combined* annotations.

We can see from the Pearson's r values that for both E2E and WebNLG all correlations are strong for the sets of correctness label counts. For the error class label count sets, on the other hand, while the original annotators achieve high correlation with their own earlier label counts for E2E, they do not for WebNLG, where the correlation is weak. The correlation between the A1+A2 and A3+A4 error label counts is weak to medium for both E2E and WebNLG. The new annotators do a reasonable job reproducing the original labels for E2E (r=0.62), but worst by far is the pronounced negative correlation for the new annotators for the WebNLG error labels.

4.3 Comparison of type *iii* results

The results from the agreement tests with Fleiss's kappa and Krippendorff's alpha on both label types as attached to evaluation items (type *iii* results, i.e. related sets of categorical values), again on the *combined* annotations, are shown in Table 5. For E2E and correctness labels, a similar picture emerges as previously in that agreement is similarly good across all comparisons, reflected also in the '%=' column which shows the percentage of times there was perfect agreement across all labels and all an-

[9] Counts for *ref correct*, *SED correct*, and *other* do not add up to 100 because of 3 missing annotations.

Counts of	E2E						Counts of	WebNLG					
	D&K	A1	A2	A3	A4	CV^*		D&K	A1	A2	A3	A4	CV^*
ref correct	34	41	31	37	50	21.325	ref correct	51	43	34	55	48	19.598
SED correct	45	45	53	41	47	10.594	SED correct	42	44	30	37	48	19.291
other	18	14	15	22	3	55.016	other	7	12	13	8	4	46.984
[eatType]	5	10	5	2	8	57.382	[bias-templ]	22	18	16	7	2	70.856
[priceRange]	30	31	39	42	9	47.756	[val-format]	7	1	3	26	0	162.088
[famFriend]	10	11	10	8	1	56.718	[bad-sent]	14	27	15	9	6	63.275
[f-halluc]	8	8	3	38	0	149.505	[unj-OK]	8	31	17	48	0	102.418
[f+omiss]	16	10	14	42	6	89.937	[unj-notOK]	15	16	25	26	1	67.727
[f+halluc]	17	15	24	19	4	52.288							

Table 3: QRA assessment of individual numeric results (type *i*), using the 4 sets of ***individual*** annotations.

	Pearson's r	E2E	Web-NLG
Correctness	Orig vs. A1+A2	0.999	0.965
	Orig vs. A3+A4	0.948	0.963
	A1+A2 vs. A3+A4	0.959	0.857
Error classes	Orig vs. A1+A2	0.947	0.209
	Orig vs. A3+A4	0.620	-0.630
	A1+A2 vs. A3+A4	0.373	0.414

Table 4: Pearson's r for counts of correctness and error-class labels (type *ii*), using the ***combined*** annotations (see Table 2 caption and Section 4).

			E2E	% =	Web-NLG	% =
Correctness	Fleiss's κ	All	0.674	71%	0.269	40%
		Orig vs. A1+A2	0.676	81%	0.140	50%
		Orig vs. A3+A4	0.677	81%	0.527	73%
		A1+A2 vs. A3+A4	0.643	78%	0.112	48%
Error classes	Kripp.'s α	All	0.467	12%	0.165	3%
		Orig vs. A1+A2	0.735	60%	0.207	21%
		Orig vs. A3+A4	0.347	15%	0.114	7%
		A1+A2 vs. A3+A4	0.330	18%	0.166	12%

Table 5: Fleiss's kappa for correctness and Krippendorff's alpha for error-class labels (type *iii*), using the ***combined*** annotations (see Table 2 caption and in text). '% =' = percentage of items with identical labels.

notators in a given comparison. For E2E and error class labels, the original annotators have strong agreement with their own original annotations, and the rest of the comparisons show medium agreement.

Again the picture is more mixed for WebNLG, where the new annotators have medium label-level agreement with the original labels for correctness, but for the other seven comparisons, label-level agreement is quite startlingly low (0 being chance).

5 Discussion

The error-analysis based evaluation method in this paper compares system outputs with reference outputs, but rather than just counting it against the system if there is disagreement between the two, it examines which is actually right in each case, also identifying the types of errors made by each. For E2E, 4 out of 5 sets of annotations (Table 3) agreed that the NLI-SED system was more often correct than the (automatically generated) references; for WebNLG the balance was slightly tipped in favour of the references (here derived from human ratings). These were important findings in the original paper, and are confirmed in all reproductions.

Other broad-strokes findings that are confirmed in all reproductions are that errors to do with priceRange predicate are the most common, and errors connected to eatType and famFriend are the least common, of the errors considered in E2E. For the error labels in WebNLG no findings are supported by all sets of annotations.

The degree to which the different types of results were reproducible varied. The more high-level correctness labels saw far better agreement than the more fine-grained error labels which also involve greater cognitive load. Moreover, the different backgrounds of the annotators and their degree of familiarity with the system and data may also have contributed to variation.

It is likely that if our instructions had been more precise, and more training/discussions of annotators in interpreting the instructions had taken place, the variation between studies would have been lower, and we can see room for improvement in this respect which we plan to explore in future work, where we will aim to:

- Ensure that annotators are given all relevant information for fully informed assessment of all error categories.

- Follow the iterative cycle in designing a linguistic annotation scheme (Pustejovsky et al., 2017): start with a preliminary annotation scheme and iteratively improve it using empirical observations (Howcroft et al., 2020).

- After a good fit between annotation scheme and task has been achieved and annotators reach a shared understanding, explicitly write down the annotation guidelines including any conclusions from informal discussions.

The iterative annotation design and written guidelines would have been useful even for repeating the study with the original annotators, as even their annotations differed in the repeat. We also noted some ideas for improving the error classes, which probably would have been already implemented with an iterative approach.

6 Conclusion

In this paper, we described two reproductions of a manual error analysis of the outputs from a semantic error detection (SED) system based on two-way entailment detection by an NLI model. We selected three types of results for reproduction, namely single numeric values, sets of numeric values, and sets of discrete labels, each of which requires different methods of comparison. All three types of results have broadly similar degrees of reproducibility: higher-level findings are mostly confirmed but lower-level agreement measures show a more differentiated picture, and are particularly low for WebNLG and error classes. Results for E2E are generally better reproduced than WebNLG, and correctness labels are easier to reproduce than the more fine-grained error classes.

In terms of conclusions to be drawn from the reproduction studies reported here, as with many other reproductions we found that the details of design and execution of the original study had not been recorded at the level of detail required for a reproduction. As a field, NLP is not currently in the habit of recording design/execution details of human evaluations very comprehensively or testing reproducibility during method development, for which time and other resources are often cited as reasons. The latter would be mitigated by the use of standard methods and tools for recording details of experiments and for assessing reproducibility.

Acknowledgements

Huidrom's work on this project was supported by the Faculty of Engineering and Computing, DCU, via a PhD grant. Belz's work was in part funded by EPSRC Grant No. EP/V05645X/1 for the ReproHum project. Dušek and Kasner's work was supported by ERC Grant No. 101039303 NG-NLG. Dušek's work is additionally supported by the Czech Ministry of Education project No. LM2018101 LINDAT/CLARIAH-CZ. Kasner's work is additionally supported by Charles University projects GAUK 140320 and SVV 260575.

References

Anya Belz, Shubham Agarwal, Anastasia Shimorina, and Ehud Reiter. 2020. ReproGen: Proposal for a shared task on reproducibility of human evaluations in NLG. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 232–236, Dublin, Ireland. Association for Computational Linguistics.

Anya Belz, Maja Popovic, and Simon Mille. 2022. Quantified reproducibility assessment of nlp results. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL'22)*.

Anya Belz, Anastasia Shimorina, Shubham Agarwal, and Ehud Reiter. 2021. The reprogen shared task on reproducibility of human evaluations in nlg: Overview and results. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 249–258.

António Branco, Nicoletta Calzolari, Piek Vossen, Gertjan Van Noord, Dieter van Uytvanck, João Silva, Luís Gomes, André Moreira, and Willem Elbers. 2020. A shared task of a new, collaborative type to foster reproducibility: A first exercise in the area of language science and technology with REPROLANG2020. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 5539–5545, Marseille, France. European Language Resources Association.

K. Bretonnel Cohen, Jingbo Xia, Pierre Zweigenbaum, Tiffany Callahan, Orin Hargraves, Foster Goss, Nancy Ide, Aurélie Névéol, Cyril Grouin, and Lawrence E. Hunter. 2018. Three dimensions of reproducibility in natural language processing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Ondřej Dušek and Zdeněk Kasner. 2020. Evaluating semantic accuracy of data-to-text generation with natural language inference. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 131–137.

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. *Comput. Speech Lang.*, 59(C):123–156.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The WebNLG challenge: Generating text from RDF data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133, Santiago de Compostela, Spain. Association for Computational Linguistics.

David M. Howcroft, Anya Belz, Miruna-Adriana Clinciu, Dimitra Gkatzia, Sadid A. Hasan, Saad Mahamood, Simon Mille, Emiel van Miltenburg, Sashank Santhanam, and Verena Rieser. 2020. Twenty years of confusion in human evaluation: NLG needs evaluation sheets and standardised definitions. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 169–182, Dublin, Ireland. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

James Pustejovsky, Harry Bunt, and Annie Zaenen. 2017. Designing annotation schemes: From theory to model. In *Handbook of Linguistic Annotation*, pages 21–72. Springer.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. 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.

A Appendix: Annotator Instructions

A.1 Terms and Abbreviations

- **Semantic Error Detection (SED)**: in data-to-text generation, the task of deciding which errors if any are present in the output relative to the input.

- **SED label**: a label produced by an SED method indicating the semantic error class; typical label set e.g. Ok, omission, hallucination, omission+hallucination.

- **SED method**: here, one of D&K NLI-SED system, slot-error script, the human reference labels from WebNLG.

- **E2E slot**: an attribute in an E2E input, e.g. eatType=?.

- **Template**: a short template for converting each triple to text, used for the NLI checks (links for the lists of templates can be found here: E2E, WebNLG).

A.2 Instructions

A.2.1 E2E and WebNLG

First examine the input/output pair and make a note in the ‘Other’ column indicating the likely source of the error made by the incorrect SED method(s). Then, choose one or more of the error classes below that match the note. If none match, leave empty.

A.2.2 E2E

Each class indicates the likely source of the error made by the SED method that was deemed wrong (here, either the NLI-SED system or the slot-error script), and as many of the labels should be selected as apply to each item, in some cases none. NB: each error class predominantly applies to the underlined method.

1. *Error related to eatType=restaurant slot value*: The incorrect SED label (produced either by the NLI-SED system or the slot error script) is likely caused by something involving the slot/value pair eatType=restaurant, e.g. not detecting a hallucination when the eatType slot is not in the input, but the output mentions a restaurant.

2. *Error related to priceRange slot*: The incorrect SED label (produced by either the NLI-SED system or the slot error script) makes

an error related to the priceRange slot, e.g. incorrectly identifying a hallucination in the priceRange slot, when the price range information has in fact been correctly verbalised.

3. *Error related to familyFriendly attribute*: The incorrect SED label (produced by either the NLI-SED system or the slot error script) makes an error related to the familyFriendly slot, e.g. incorrectly identifying an omission when the information has in fact been correctly verbalised.

4. *Other false negative hallucination ('off-topic blabber')*: The incorrect SED label (produced by either the NLI-SED system or the slot error script) fails to detect a hallucination (unrelated to E2E slots) present in the verbalisation.

5. *Other false positive omission ('unjustified omission')*: the incorrect SED label (produced by either the NLI-SED system or the slot error script) wrongly detects an omission in the verbalisation.

6. *Other false positive hallucination ('unjustified hallucination')*: the incorrect SED label (produced by either the NLI-SED system or the slot error script) wrongly detects a hallucination in the verbalisation.

A.2.3 WebNLG

Each item may have more than one or none of these. The first three classes indicate, where possible, the likely source of the error in the SED label that was deemed wrong (produced by either the NLI-SED system or the reference SED label mapped from the human scores). Otherwise one of the last two will apply. Label frequencies are shown in the second Dušek & Kasner column in Table 1).

1. *SED system error due to poor triple-to-text input mapping ('biased template')*: incorrect NLI-SED system label due to an inappropriate template being used in mapping the input triples to text (templates tend to work better for certain subject/object values, but the same template is used for all cases with a given predicate), resulting in ungrammatical sentences or even shift in meaning. NB: please refer to the WebNLG templates as necessary.

2. *NLI-SED system failure to recognise subject or object semantic equivalence ('value format')*: in the verbalisation the formatting of a subject or object differs from the input to the extent where the NLI check in the NLI-SED system failed to recognise them as equivalent in meaning (e.g. metres vs. kilometres).

3. *Incorrect reference SED label due to disfluent verbalisation ('bad sentence')*: the incorrect human reference is not OK, and this is likely because the human rating was affected by the disfluency/ungrammaticality of the verbalisation.

4. *Other cases of incorrect OK label ('unjustified OK')*: the incorrect label (from either the human references or the NLI-SED system) is OK, and none of the above apply.

5. *Other cases of incorrect not OK label ('unjustified not OK')*: the incorrect label (from either the human references or the NLI-SED system) is not OK, and none of the above apply.

B Appendix: HEDS-Light Datasheet

Link to our HEDS Datasheet.

Reproducibility of *Exploring Neural Text Simplification Models*: A Review

Mohammad Arvan, Luís Pina, and **Natalie Parde**
Department of Computer Science
University of Illinois Chicago
{marvan3,luispina,parde}@uic.edu

Abstract

The reproducibility of NLP research has drawn increased attention over the last few years. Several tools, guidelines, and metrics have been introduced to address concerns in regard to this problem; however, much work still remains to ensure widespread adoption of effective reproducibility standards. In this work, we review the reproducibility of *Exploring Neural Text Simplification Models* by Nisioi et al. (2017), evaluating it from three main aspects: data, software artifacts, and automatic evaluations. We discuss the challenges and issues we faced during this process. Furthermore, we explore the adequacy of current reproducibility standards. Our code, trained models, and a docker container of the environment used for training and evaluation are made publicly available.

1 Introduction

In a survey conducted among 1,576 scientific researchers by Nature (Baker, 2016), 90% believed that there is at least a slight crisis when it comes to the reproducibility of research. Although there are no concrete statistics, the quantity and the growth of machine learning publications that rely on empirical evidence have recently raised alarms. To improve reproducibility, this community has designed checklists (AAAI, 2022; ACL, 2022; Deutsch et al., 2022), guidelines (ACM, 2022), and challenges (Sinha et al., 2022; Belz et al., 2021), which highlight the importance of reproducibility, encourage best practices, and create a platform for conducting reproducibility studies.

Still, measures of reproducibility "in the wild" (that is, pertaining to real, widely cited machine learning and natural language processing studies) are limited. In this work, we set out to reproduce one such study as a case example to provide a concrete measure of reproducibility for a specific work. We select *Exploring Neural Text Simplification Models* by Nisioi et al. (2017). This paper poses an intriguing case: the research artifacts released by the authors are of high quality, the details they have provided match or exceed the current reproducibility recommendations, and two other reproducibility studies (Cooper and Shardlow, 2020; Belz et al., 2022) have successfully reproduced the results with high precision. Reviewing a high-quality scientific publication that has been the focus of multiple reproducibility studies enables us to build and expand upon those works.

Our primary objective is to investigate the ease with and extent to which the selected paper can be reproduced. We limit introducing new configurations, adding them only to cases necessary for further understanding the reproducibility results and not for competing scenarios. In Section 2, we present the background, the task itself, the model, and its variants. In Section 3, we describe our methodology and the steps we take to review the reproducibility of Nisioi et al. (2017)'s work. In particular, we look at associated data, software artifacts, and automatic evaluations. We present our results in Section 4, before concluding by discussing our findings and recommendations for addressing the shortcomings of current checklists (Section 5). We release our reproducibility artifacts to facilitate and promote future reproducibility studies (Arvan et al., 2022). These artifacts include the updated source code, trained model, and complete runtime environment in a self-contained docker container.

2 Neural Text Simplification

Nisioi et al. (2017)'s work explores the task of *neural text simplification*. In this task, the goal is to transform a given text into a simpler version while retaining its meaning. What constitutes simplicity itself raises complicated questions since simplicity could be observed in the form of lexical simplification, content reduction, and grammatical or structural modification. Data-driven techniques attempt to achieve simplicity through automated metrics

Proceedings of the 15th International Conference on Natural Language Generation: Generation Challenges, pages 62 - 70
July 17-22, 2022 ©2022 Association for Computational Linguistics

and human evaluation. The task holds many parallels with machine translation (MT), and this framing allows models studied in the context of neural MT (e.g., neural sequence to sequence models) to be adapted and deployed for neural text simplification.

Nisioi et al. (2017)'s work is one of the first investigations of neural sequence to sequence models for automatic text simplification. In particular, they use Long Short-Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1996, 1997) in an encoder-decoder architecture that has demonstrated success in similar sequence to sequence problems (Luong et al., 2015). The encoder LSTM computes a representation for each source sentence, and the decoder LSTM generates an output given the encoded representation and previously generated tokens. Nisioi et al. (2017) also employ a global attention mechanism that provides a more dynamic information flow and increases the representation bandwidth. To avoid overfitting, they use dropout (Srivastava et al., 2014), a technique that injects noise into the input during training by masking out certain features.

Nisioi et al. (2017) experiment with two variants of networks: one with random embedding weight initialization (*NTS*), and another with pre-trained embeddings. The latter is built by concatenating pre-trained word2vec embeddings from the Google News corpus (Mikolov et al., 2013a) with a locally trained skip-gram model (Mikolov et al., 2013b) with hierarchical softmax and a window size of 10. The concatenation process involves utilizing a unique dictionary associated with the source and target embeddings. The authors refer to this variant as *NTS w2v*.

3 Methodology

There is no standard protocol or set of guidelines for conducting a reproducibility study, so we rely on the best practices suggested by others to provide a subjective and objective evaluation of the reproducibility of Nisioi et al. (2017)'s work. These best practices originate from checklists (Pineau et al., 2021), research focused on reporting and evaluation (Dodge et al., 2019; Patterson et al., 2021; Schwartz et al., 2020), and reproducibility tracks at conferences and workshops (Deutsch et al., 2022). Although we do not fill out any checklists, as they are not created for the purpose of third-party evaluation, we cover nearly all the concerns they attempt to address.

Split	**Sentences**
train (EW-SEW)	284,677
validation (TurkCorpus)	2,000
test (TurkCorpus)	359

Table 1: Distribution of sentence pairs across different data splits, with data sources in parentheses.

In the following subsections, we first examine the data used for this work and then shift our attention to the released software artifacts. We perform these preliminary steps to identify potential obstacles to reproducibility and to test the adequacy of the existing standards. Later, we assess the reproducibility of the reported automatic evaluations.

3.1 Data

Data quality and composition are primary factors that can significantly impact reported results. Unsurprisingly, all reproducibility checklists emphasize the importance of data transparency. Nisioi et al. (2017) used a corpus of parallel English Wikipedia and Simple English Wikipedia (EW-SEW) articles (Hwang et al., 2015) when developing and evaluating their text simplification model. EW-SEW includes both manually and automatically aligned sentence pairs and was one of the largest publicly available datasets for text simplification at the time Nisioi et al. (2017)'s paper was published. Sentences in EW-SEW were filtered based on Wiktionary-based word-level semantic similarity scores included in the dataset, with a retention threshold set at 0.45. This resulted in a final set of 280K+ aligned sentences. EW-SEW does not have standard validation and test splits; thus, although Nisioi et al. (2017) used EW-SEW for training, they used TurkCorpus (Xu et al., 2016) for validation and testing. TurkCorpus is considerably smaller than EW-SEW and consists of 2000 validation and 359 test sentences.

The final distribution of training, validation, and test data is shown in Table 1. To preprocess the data, Nisioi et al. (2017) used the Stanford named entity recognition (NER) system (Finkel et al., 2005) to automatically tag the locations, persons, organizations, and miscellaneous entities in the dataset. We check the reproducibility of the preprocessing steps in Subsection 4.1 by reviewing the original dataset, as well as steps taken to filter and process the data.

3.2 Software Artifacts

Authors may omit purportedly trivial details from research publications due to strict length limits. Such details may be crucial for later successful replication. Fortunately, released software artifacts often provide these details and other necessary engineering steps. The ML Completeness Checklist (Stojnic, 2022) underlines the inclusion of five items in software artifacts that facilitate reproducibility and are expected to result in easier adaptability for future researchers: (1) specification of dependencies, (2) training code, (3) evaluation code, (4) pre-trained models, and (5) a README file including a table of results accompanied by precise commands to run and produce those results. Given that Nisioi et al. (2017) provided all of these items, we investigate the quality and the functionality of the released artifacts within this context by reviewing the aforementioned checklist items, testing out the provided commands, and rebuilding the environment using provided materials.

3.3 Automatic Evaluation

Reproducibility and reporting quality are complementary to one another, and improvements to one often accompany improvements to the other. We include task-agnostic metrics and details commonly used in training and evaluations of neural networks (Dodge et al., 2019; ACL, 2022) in our assessment of the reproducibility of Nisioi et al. (2017)'s automated evaluations. Namely, we check the number of parameters in the model, the computing infrastructure used to achieve results, and the total GPU hours required to train the model. We also report the model's total floating-point operations (fpo) (Schwartz et al., 2020), providing an estimate of the amount of computational work performed irrespective of the hardware setting. In neural networks, the dominant floating-point operations are ADD and MUL operations performed by a GPU.

Nisioi et al. (2017) evaluated the performance of their neural text simplification approach using two automated metrics as well as a human performance assessment. Their automated metrics included BLEU (Papineni et al., 2002; Wagner, 2010), a precision-based metric commonly used for machine translation and text simplification; and SARI (Xu et al., 2016), a metric designed specifically for text simplification that compares the system output against reference output and the input sentence. The evaluation scripts for these metrics are included in the source code released by the Nisioi et al. (2017). In addition to calculating the BLEU score using the script provided by Nisioi et al. (2017), we also calculate it using sacreBLEU v2.1,[1] a Python library that aims to unify standards for calculating the BLEU score (Post, 2018).

Ultimately, these metrics are used on various output files generated by different variants. At first, we evaluate the original outputs provided by Nisioi et al. (2017). Then, we use the trained model released by the authors to generate a new output and evaluate it using the mentioned metrics. Lastly, we use the code and the configuration provided by the authors in their publication and in their source code to train new models. Using these newly trained models, we generate yet another set of outputs. During this process, we are reducing the set of controlled conditions affecting the final results. We expected variation to increase as fewer conditions are controlled.

We used these metrics to evaluate the performance of our reproduced model, facilitating a direct comparison with the originally reported performance. Instead of viewing the reproducibility of the automatic evaluations as a binary state, e.g., reproducible or not reproducible, we use Quantified Reproducibility Assessment (QRA), a framework proposed by Belz et al. (2022). This framework defines reproducibility as a condition of measurement, out of a set of conditions that includes different locations, operators, and measuring systems, among other variables. As a result, identifying and reporting such conditions is an important part of this framework. Belz et al. (2022) quantify reproducibility as a measurement of precision. Given a different set of empirical results in this paper and previously conducted reproducibility studies, we present the coefficient of variation with small sample correction (CV^*) associated with each variant.

We take two additional steps to verify the claims based on the empirical results. At first, we use paired bootstrap resampling (Koehn, 2004) with 1000 samples to compare the performance of the two main variants on the output files released by Nisioi et al. (2017). Lastly, considering the relatively small size of the datasets used for training, validation, and testing, we suspected the random seed may greatly impact results. Therefore, we designed an experiment to quantify its impact. We

[1] https://github.com/mjpost/sacrebleu

trained 36 models [2] with the same configuration but different unique random seeds. A small variation in the final results of this experiment suggests that the effect of the random seed is negligible.

4 Results

In this section, we describe the outcomes of our reproducibility study in terms of data, software artifacts, and automatic evaluation.

4.1 Data

We were unable to analyze the original unfiltered version of the EW-SEW dataset (Hwang et al., 2015) as planned because the webpage containing the dataset no longer exists,[3] nor could earlier versions be retrieved using web archival tools (e.g., the Wayback Machine[4]). The released code repository for the selected paper also does not include scripts for filtering the dataset. As such, we could not review or reproduce the authors' preprocessing steps. However, the code repository does contain preprocessed dataset files, which allowed us to perform all other steps of our reproducibility analysis.

4.2 Software Artifacts

As mentioned earlier, the authors released a five star repository according to the ML Completeness Checklist. The authors listed the required external libraries, as well as Python- and Lua-specific dependencies. Moreover, the authors included a dockerfile containing the computing environment used for the experiments. Unfortunately, since a self-contained docker container was not included, it is not possible to rebuild the dockerfile, and most dependencies have been deprecated for years. These dependencies include Ubuntu 14.04 with an end of life (EOL) of 2019, Python 2.7 with EOL of 2020, Torch7 with last active development of 2017, and OpenNMT made obsolete in 2018 due to lack of support for Torch7, among others. Ultimately, we switched to another docker image based on Nvidia's CUDA 10.1 images that comes with Torch7 installed. This introduced further complications as recently released GPUs (e.g., those in the RTX 3000 series) require CUDA 11 or higher. We avoided this problem for now, but fixing this problem (which is beyond the scope of our present work) requires porting Torch7 and rebuilding it using the appropriate CUDA toolkit.[5]

Aside from the initial hurdle to get the repository to a running state, we did not face any major issues in using the software artifacts. Nisioi et al. (2017) provided the training code, evaluation code, and pre-trained models.[6] The README file contains instructions and required commands to produce the reported results. There were a few minor discrepancies between the provided instructions and real-world use, but we managed to resolve these issues. We note that the repository does not contain all configuration files used for each model variant. Hence, we use the information provided in the paper to recreate those.

In reviewing the source code, we found three issues affecting *NTS w2v* variants. We contacted the authors regarding these issues, and they graciously confirmed the first two. At the time of writing this paper, we still have not heard back regarding the third reported issue. We intend to investigate the impact of the first two issues on the results. These issues are described below.

4.2.1 Issue 1: Data Contamination

The *NTS w2v* models use a multi-step process to concatenate the pre-trained Google News word2vec embeddings and another embedding trained by the authors using the skip-gram technique. We found that during the skip-gram training process, this embedding utilized all datasets (including the development and test set), introducing data contamination that may call into question those models' results. The models affected by this issue are expected to have an advantage over other models. However, the validation and test sets are many times smaller than the training set, so performance gains may be negligible to non-existent.

4.2.2 Issue 2: Mismatched Embedding

This issue occurs during the concatenation process itself. This process uses two dictionaries, one for the encoder and one for the decoder, to generate the embedding matrix. However, we found that these embeddings were mismatched: the encoder used the decoder's dictionary, and the decoder used the encoder's dictionary. We expect fixing this issue will improve the performance of affected models.

[2] 14 training jobs failed after running out of storage.

[3] https://crow.ece.uw.edu/tial/projects/simplification/

[4] https://archive.org/web/

[5] Issue is reported here: https://github.com/nagadomi/distro/issues/11.

[6] https://github.com/senisioi/NeuralTextSimplification

System	BLEU (μ ± 95% CI)
Baseline: NTS w2v	87.9 (87.9 ± 2.0)
NTS	84.6 (84.6 ± 2.9)

Table 2: Statistical significance analysis performed on Nisioi et al. (2017)'s released output. With $p = 0.0079$, the difference in reported results between the two variants is statistically significant.

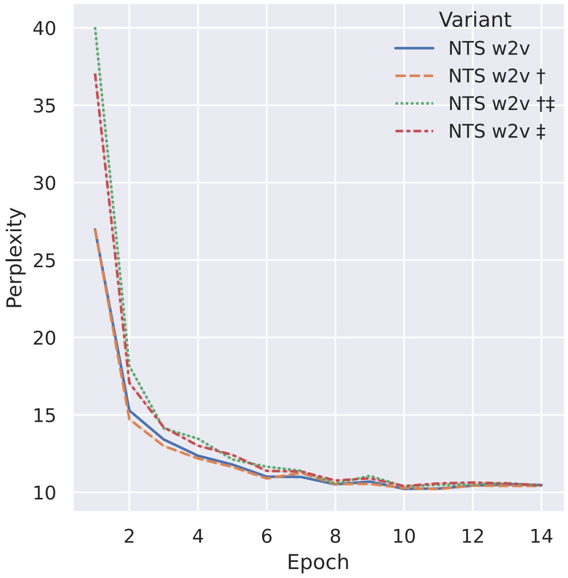


Figure 1: Validation perplexity of *NTS w2v* variants during training (lower is better). † indicates contaminated conditions, and ‡ indicates mismatched conditions.

4.2.3 Issue 3: Zero Embedding Weight

Lastly, we found that the final embedding matrix is missing the concatenation step, which results in zero vectors for all the words. Using a zero embedding weight nullifies the embedding pre-training altogether.

4.3 Automatic Evaluation

We follow the exact training setup provided by Nisioi et al. (2017), training models for 15 epochs with early stopping applied. Unlike the original paper, we did not tune the model using SARI or BLEU, and used the validation perplexity (lower is better) for model selection and early stopping. The translation is performed using beam search. Beam search generates the first *k* hypotheses at each step sorted by log-likelihood of the target sentence given the input sentence. While the authors experimented with using beam sizes of 5 and 12 and various hypotheses, we limit the scope of our experiments to 5 beams and 1 hypothesis. The hardware used for the experiments in the original paper is not explicitly specified. In our case, we use an *RTX 2080 ti* GPU to train the models. Training took approximately 3 hours. The model had 84 million parameters, of which 50 million belong to the embedding layer. With a maximum sequence length of 80 and a batch size of 1, this model used roughly 3G fpo (3×10^9) in a forward pass.

Table 3 contains the results of the original work (Nisioi et al., 2017) referred to as $t1$, reproducibility studies of Cooper and Shardlow (2020) ($t2$) and Belz et al. (2022) ($t3$), and the results calculated by this paper ($t4$) with their associated conditions. To ease the analysis, the results of every variant, measure, and output are grouped together. With the two evaluation scripts for calculating BLEU, we have added six values for BLEU and three for SARI. To be more specific, we have added automatic evaluation results for the output generated by Nisioi et al. (2017) or $o1$, our own output generated by running the trained model provided by Nisioi et al. (2017) or $o4$, and our own output generated by running our own version of the model or $o5$. We note that the model that we trained uses a source with all the fixes applied; however, to the best of our knowledge, all the other *NTS w2v* variants are trained with the mentioned issues. We present the precision results of the QRA framework in Table 4.

The *NTS* variant has CV^* values of 1.92 and 1.94 for SARI and BLEU, respectively. With 3.28 and 2.85 for SARI and BLEU, CV^* values for *NTS w2v* are slightly worse. However, the BLEU score of the *NTS w2v* variant reported by Cooper and Shardlow (2020) seems to be an outlier. By excluding their score (80.75), CV^* reduces to 1.22. There are two other interesting observations in Table 3. First, our reported results for $o1$ exactly match the reported results by the original paper; this suggests that we successfully recreated the environment they used for their evaluation. Second, the difference between the reported BLEU for $o1$ using the sacreBLEU evaluation script (Belz et al., 2022) and that found by our study implies there are still several unaccounted factors. We believe the version of sacreBLEU and the process of running this evaluation script are possible causes for this variation.

Table 2 shows results from the paired bootstrap resampling statistical significance test, with an objective of determining whether the performance of *NTS w2v* in terms of BLEU score is better than the

Object	Measurand	Output	Trained by	Comp. by	Eval. Script by	Performed by	Measured Value
NTS	BLEU	o1	t1	t1	t1	t1	84.51
		o1	t1	t1	t1	t2	84.50
		o1	t1	t1	≈t1	t3	85.60
		o1	t1	t1	sb	t3	84.20
		o1	t1	t1	t1	t4	84.51
		o1	t1	t1	sb2.1	t4	84.60
		o2	t2	t2	t1	t2	87.46
		o3	t1	t3	≈t1	t3	86.61
		o3	t1	t3	sb	t3	86.20
		o4	t1	t4	t1	t4	86.53
		o4	t1	t4	sb2.1	t4	86.60
		o5	t4	t4	t1	t4	88.81
		o5	t4	t4	sb2.1	t4	88.80
	SARI	o1	t1	t1	t1	t1	30.65
		o1	t1	t1	t1	t2	30.65
		o1	t1	t1	t1	t3	30.65
		o1	t1	t1	t1	t4	30.65
		o2	t2	t2	t1	t2	29.13
		o3	t1	t3	t1	t3	29.96
		o4	t1	t4	t1	t4	29.96
		o5	t4	t4	t1	t4	30.23
NTS w2v	BLEU	o1	t1	t1	t1	t1	87.50
		o1	t1	t1	≈t1	t3	89.36
		o1	t1	t1	sb	t3	88.10
		o1	t1	t1	t1	t4	87.50
		o1	t1	t1	sb2.1	t4	87.90
		o2	t2	t2	t1	t2	80.75
		o3	t1	t3	≈t1	t3	89.64
		o3	t1	t3	sb	t3	88.80
		o4	t1	t4	t1	t4	89.40
		o4	t1	t4	sb2.1	t4	89.40
		o5	t4	t4	t1	t4	87.04
		o5	t4	t4	sb2.1	t4	87.10
	SARI	o1	t1	t1	t1	t1	31.11
		o1	t1	t1	t1	t3	31.11
		o1	t1	t1	t1	t4	31.11
		o2	t2	t2	t1	t2	30.28
		o3	t1	t3	t1	t3	29.12
		o4	t1	t4	t1	t4	29.12
		o5	t4	t4	t1	t4	29.70

Table 3: Detailed overview of the results of *NTS* and *NTS-w2v*. All of the results utilize the source code released by Nisioi et al. (2017). Outputs $o1$ to $o5$ are generated based on the conditions provided in their respected row: $t1$=Nisioi et al. (2017), $t2$=Cooper and Shardlow (2020), $t3$=Belz et al. (2022), and $t4$= this paper; and sacreBLEU versions are represented as sb=unknown version, and $sb2.1$=version 2.1.

Object	Measurand	Sample Size	Mean	Unbiased STDEV	STDEV 95% CI	CV^*
NTS	SARI	8	30.23	0.56	[0.23, 0.89]	1.92
NTS	BLEU	13	86.07	1.64	[0.94, 2.34]	1.94
NTS w2v	SARI	7	30.22	0.96	[0.34, 1.58]	3.28
NTS w2v	BLEU	12	87.71	2.45	[1.35, 3.54]	2.85

Table 4: Precision (CV^*) and component measures (mean, standard deviation, standard deviation confidence intervals) for measured quantity values obtained in multiple measurements of the two *NTS* systems.

NTS variant. With $p = 0.0079$, the difference is indeed statistically significant. Since the output of *NTS w2v* is generated using a model affected by the zero weight embedding issue (Issue 3 described in Subsection 4.2.3), these two variants are essentially the same. Thus, understanding what is at play here requires assessing the results in Table 5. Even with a small sample size of 36, we observe values ranging from 84.47 to 89.59. This suggests that the performance difference between the two main

Measurand	Mean	Min	Max
SARI	29.24 ± 0.31	28.62	29.89
BLEU	87.9 ± 1.18	84.47	89.59

Table 5: Results of the random seed experiments on the TurkCorpus (Xu et al., 2016) test set, with a sample size of 36. Models are trained with the same configuration, but have unique random seeds. The evaluation script by Nisioi et al. (2017) was used.

variants may have originated from having different random seeds, even at statistically significant levels.

Finally, we investigate the issues reported for the *NTS-w2v* variant. We exclude Issue 4.2.3, as it simply converts *NTS-w2v* to *NTS* with zero embedding weight. We introduce three new variants:

- *NTS w2v*†: *NTS-w2v* only affected by data contamination (Issue 4.2.1).
- *NTS w2v*‡: *NTS-w2v* only affected by mismatched embeddings (Issue 4.2.2).
- *NTS w2v*†‡: *NTS-w2v* affected by data contamination and mismatched embeddings.

The results are shown in Table 6. Overall, the results are extremely close. We found that the variant with the data contaminated outperform others while *NTS-w2v*, the variant without any issues performed worse than the rest. We expected to observe a noticeable performance difference for the models affected by the mismatched embedding issue, but the performance gap was ultimately marginal and inconsistent. We report the validation performance during training to analyze whether there are any differences between these four variants. As shown in Figure 1, the models with mismatched embeddings had a worst start, by a perplexity gap of almost 15; however, as training progressed, they closed the gap and ended with perplexity differences of less than 1.

5 Discussion

Taking all our experiments into account, we cannot claim that the performance difference between different variants comes from the design decisions made during their development. The random seed and its cascading impact on weight initialization, data order, and sampling during text generation could be the primary cause of the observed variations. Similar to our work, Dodge et al. (2020) have observed that changes to random seeds can result in substantially different results.

Object	Measurand	Eval. Script by	Measured Value
NTS w2v	BLEU	t1	87.04
NTS w2v	BLEU	sb2.1	87.10
NTS w2v	SARI	t1	29.70
NTS w2v †	BLEU	t1	89.43
NTS w2v †	BLEU	sb2.1	89.40
NTS w2v †	SARI	t1	29.80
NTS w2v †‡	BLEU	t1	89.12
NTS w2v †‡	BLEU	sb2.1	89.10
NTS w2v †‡	SARI	t1	29.58
NTS w2v ‡	BLEU	t1	88.01
NTS w2v ‡	BLEU	sb2.1	88.00
NTS w2v ‡	SARI	t1	29.18

Table 6: Results of the experiments tracking performance impacts for identified issues, computed for this paper using our version of the model, our output, and the evaluation script provided by Nisioi et al. (2017) and sacreBLEU. † indicates contaminated conditions, and ‡ indicates mismatched conditions.

Perhaps our most surprising finding is that the *NTS-w2v* variants affected by mismatched embeddings performed on par with the other variants once training was complete. This extreme level of resilience is, in fact, quite alarming. Nearly all publications utilizing neural networks report top-performing empirical results; yet, aside from manual code review and deep analysis of the final results, there are no other clear signs or warnings that may suggest a bug is impacting the model. In this case, we found that our findings from the random seed experiments and validation performance during the training process were the only indicators that something was amiss. We recommend that future studies include random seed analysis demonstrating the range of the results that can be achieved with varied seeds, although we recognize that this search is the most expensive in terms of computation and may not be feasible in every case.

Perhaps due to the age of this repository, getting the project to a running state consumed the most time. We suspect that the situation will deteriorate as most dependencies are no longer being actively maintained. Researchers should be hesitant with introducing new dependencies into their projects. Additionally, we believe it would be fruitful to redirect the time and effort used for identifying and reporting dependencies toward exporting self-contained environments. This is an inadequacy that we found in nearly all of the checklists; in the case of this project, even though we knew all the requirements, we spent hours debugging different errors.

The reproducibility of a reproducibility study is equally important, if not more than the reported findings. While the contribution of the original paper includes a novel idea, our goal was to provide a final artifact having the highest possible degree of reproducibility, and to assess the ease with and extent to which the selected paper could be reproduced. It would be interesting to return and perform a meta-analysis of this work in a few years to see how much the claims hold over time. While it is impossible to stop hardware and software from changing constantly, there are steps that can be taken in order to prolong the lifespan of a research artifact. We have made changes and fixes publicly available in a forked repository of the original paper.[7] Additionally, we exported and released a self-contained docker container capable of training and running the model without any internet access (Arvan et al., 2022). Lastly, all the trained models are available for download.[8] Despite all these attempts, it is hard to predict future problems that might occur. Even the docker container depends upon the host environment (particularly, the kernel, GPU driver, and the docker itself). We have released our full runtime environment through Zendo (Arvan et al., 2022).

6 Conclusions

In this paper, we reviewed the reproducibility of *Exploring Neural Text Simplification Models* by Nisioi et al. (2017). In our three step process, we analyzed the reproducibility of the data, the software artifacts, and the automatic evaluations. We would have liked to analyze the reproducibility of human evaluations given additional time. We hope that our released artifacts offer other researchers a head start for future reproducibility studies.

7 Acknowledgments

We would like to thank Sergiu Nisioi, Maja Popovic, and Anya Belz for their excellent communication and collaboration. They provided further context regarding their experiments that cleared any lingering doubts we had regarding our results. This work would have not been the same without their help. We also thank the anonymous reviewers for their insightful feedback, which was incorporated into the final version of this paper. This work was partially supported by equipment purchased using the University of Illinois Chicago's Provost's Graduate Research Award.

References

AAAI. 2022. AAAI reproducibility checklist.

ACL. 2022. ACL responsible nlp research.

ACM. 2022. ACM artifact review and badging.

Mohammad Arvan, Luís Pina, and Natalie Parde. 2022. Artifacts of Reproducibility of Exploring Neural Text Simplification Models: A Review.

Monya Baker. 2016. 1,500 scientists lift the lid on reproducibility. *Nature*, 533(7604).

Anya Belz, Maja Popovic, and Simon Mille. 2022. Quantified reproducibility assessment of NLP results. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16–28, Dublin, Ireland. Association for Computational Linguistics.

Anya Belz, Anastasia Shimorina, Shubham Agarwal, and Ehud Reiter. 2021. The reprogen shared task on reproducibility of human evaluations in NLG: overview and results. In *Proceedings of the 14th International Conference on Natural Language Generation, INLG 2021, Aberdeen, Scotland, UK, 20-24 September, 2021*, pages 249–258. Association for Computational Linguistics.

Michael Cooper and Matthew Shardlow. 2020. Combinmt: An exploration into neural text simplification models. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 5588–5594. European Language Resources Association.

Daniel Deutsch, Yash Kumar Lal, Annie Louis, Pete Walsh, Jesse Dodge, and Niranjan Balasubramanian. 2022. 2022 north american chapter of the association for computational linguistics reproducibility track.

Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. 2019. Show your work: Improved reporting of experimental results. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 2185–2194. Association for Computational Linguistics.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah A. Smith.

[7] https://github.com/mo-arvan/NeuralTextSimplification

[8] https://drive.google.com/drive/folders/1cDQL08xQjuttu_jxrp1WnXKWZvEaRpSf

2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *CoRR*, abs/2002.06305.

Jenny Rose Finkel, Trond Grenager, and Christopher D. Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*, pages 363–370. The Association for Computer Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1996. LSTM can solve hard long time lag problems. In *Advances in Neural Information Processing Systems 9, NIPS, Denver, CO, USA, December 2-5, 1996*, pages 473–479. MIT Press.

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

William Hwang, Hannaneh Hajishirzi, Mari Ostendorf, and Wei Wu. 2015. Aligning sentences from standard wikipedia to simple wikipedia. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 211–217. The Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing , EMNLP 2004, A meeting of SIGDAT, a Special Interest Group of the ACL, held in conjunction with ACL 2004, 25-26 July 2004, Barcelona, Spain*, pages 388–395. ACL.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1412–1421. The Association for Computational Linguistics.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119.

Sergiu Nisioi, Sanja Stajner, Simone Paolo Ponzetto, and Liviu P. Dinu. 2017. Exploring neural text simplification models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2: Short Papers*, pages 85–91. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.

David A. Patterson, Joseph Gonzalez, Quoc V. Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David R. So, Maud Texier, and Jeff Dean. 2021. Carbon emissions and large neural network training. *CoRR*, abs/2104.10350.

Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Hugo Larochelle. 2021. Improving reproducibility in machine learning research(a report from the neurips 2019 reproducibility program). *J. Mach. Learn. Res.*, 22:164:1–164:20.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pages 186–191. Association for Computational Linguistics.

Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. 2020. Green AI. *Commun. ACM*, 63(12):54–63.

Koustuv Sinha, Jesse Dodge, Sasha Luccioni, Jessica Forde, Sharath Chandra Raparthy, François Mercier, Joelle Pineau, and Robert Stojnic. 2022. ML reproducibility challenge 2021.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Robert Stojnic. 2022. ML code completeness checklist.

Wiebke Wagner. 2010. Steven bird, ewan klein and edward loper: Natural language processing with python, analyzing text with the natural language toolkit - o'reilly media, beijing, 2009, ISBN 978-0-596-51649-9. *Lang. Resour. Evaluation*, 44(4):421–424.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Trans. Assoc. Comput. Linguistics*, 4:401–415.

The Accuracy Evaluation Shared Task as a Retrospective Reproduction Study

Craig Thomson
Dept of Computing Science
University of Aberdeen
Aberdeen, UK
c.thomson@abdn.ac.uk

Ehud Reiter
Dept of Computing Science
University of Aberdeen
Aberdeen, UK
e.reiter@abdn.ac.uk

Abstract

We investigate the data collected for the Accuracy Evaluation Shared Task as a retrospective reproduction study. The shared task was based upon errors found by human annotation of computer generated summaries of basketball games. Annotation was performed in three separate stages, with texts taken from the same three systems and checked for errors by the same three annotators. We show that the mean count of errors was consistent at the highest level for each experiment, with increased variance when looking at per-system and/or per-error-type breakdowns.

1 Introduction

To address issues of factual accuracy in data-to-text systems, we developed a protocol for annotating mistakes in NLG texts (Thomson and Reiter, 2020). This protocol was used to create a corpus of errors found in generated basketball game summaries. The corpus was then used in the Accuracy Evaluation Shared Task (Thomson and Reiter, 2021), where participants submitted automatic metrics or alternative human evaluations that were compared to the gold standard. The corpus was created in three stages under experimental conditions which were largely the same. The same user interface and platform (Amazon Mechanical Turk) were used, along with the same three annotators who each checked every game summary in all experiments. The only changed conditions were the game summaries that were checked, and slight clarification of the instructions based on annotator queries and feedback. Therefore, it can be retrospectively considered a reproduction attempt.

The original goal of these human evaluations was to design and develop a reliable protocol, then create gold list of errors (for training metrics, etc). In Thomson and Reiter (2020) we performed an initial run of the protocol, with two subsequent runs in Thomson and Reiter (2021) to extend the first run to form a training set, and then create a test set in the third run for use in a shared task. Generated texts were annotated in equal proportions, within each experiment, from three different systems (Wiseman et al., 2017; Puduppully et al., 2019; Rebuffel et al., 2020) which at the time were representative state of the art systems on the RotoWire dataset (Wiseman et al., 2017)[1] of English language basketball summaries paired with box score data tables.

In this paper we examine whether a similar number of errors were obtained in each experiment. We do this for all systems combined, the ensemble of representative errors that we intended to collect, and also at the per-system level, where we look at the errors for each system in isolation. We also discuss the issues that might be encountered when trying to reproduce or otherwise verify results obtained using the gold standard protocol for factual accuracy.

2 Related work

It is crucial that our evaluation protocols are reliable, something that can be demonstrated by reproducing experimental results under similar conditions. Such reproduction work is seldom carried out within the field of NLP (Belz et al., 2021a). When it is, researchers experience difficulties obtaining the same results or finding the information required to run the experiment at all (Mieskes et al., 2019). Problems with reproduction are not limited to NLP. In a large scale survey of over 1,500 researchers, Baker (2016) found that 90% felt there was a reproducibility crisis, with over 50% deeming it ‘significant’. The ReproGen shared task, for which this paper is a submission, aims to document reproduction attempts in NLP and provide an improvement in levels of reproducibility over time.

[1] https://github.com/harvardnlp/boxscore-data

Proceedings of the 15th International Conference on Natural Language Generation: Generation Challenges, pages 71 - 79
July 17-22, 2022 ©2022 Association for Computational Linguistics

2.1 Evaluation by annotation

Whilst evaluation of text generation systems is usually done by rating or ranking (van der Lee et al., 2019; Gehrmann et al., 2022), approaches for evaluation by annotation have also been proposed. Popović (2020) asked participants to highlight problematic spans within machine translated text, which were then categorised by severity. This allowed for the count of errors to be used to rank systems, but with the benefit of the individually reported errors being amenable to error analysis, something that is important for MT and NLG (van Miltenburg et al., 2021). With the SCARECROW framework, Dou et al. (2022) asked annotators to highlight problematic spans of text in prompted generation, these were then categorised. The categories are diverse, covering grammatical issues as well as issues readers might have, such as needing an external resource to check a fact. There is also the task-dependant category of 'off prompt'. Agreement for many categories was low, with errors for all categories except 'off-prompt' being reported by two or more annotators (out of ten who annotated each paragraph) in less than 50% of cases. Freitag et al. (2021) instructed annotators to highlight errors within machine translated texts, then categorise each error with one type from a hierarchy of error types. Errors were also assigned a severity level by annotators. For text simplification, Devaraj et al. (2022) used an approach whereby annotators highlighted spans of text then labelled them using the task specific label of whether information was inserted, deleted, or substituted, as well as how severe the error was.

3 The Gold standard protocol for factual accuracy

The gold standard protocol detailed in Thomson and Reiter (2020) uses human annotators to check the factual accuracy of generated texts. As part of this work, basketball games summaries were annotated for factual errors. These summaries are based on complex data, often including information from outwith the game being summarised, such as aggregated statistics or upcoming game schedules. This presents problems of error detection that are not found in simpler tasks. Fact checking is performed against a comprehensive external data source rather than system input data, i.e., annotators check whether the text truthfully reflects what actually happened in the basketball games. Full details can be found in Thomson and Reiter (2020) and Thomson and Reiter (2021), although a brief overview is included here.

3.1 Annotation

Annotators are asked to highlight non-overlapping spans of text that are factually inaccurate, then mark each span with an error type, as well as a correction or comment explaining why the text is inaccurate. The types are:

NAME$^{\text{N}}$: Incorrect named entity - Including people, places, organisations, and days of the week.

NUMBER$^{\text{U}}$: Including both numbers which are spelled out, and those expressed as digits.

WORD$^{\text{W}}$: A word which is not one of the above and is incorrect.

CONTEXT$^{\text{C}}$: A phrase which causes an incorrect inference because of context or discourse.

NOT CHECKABLE$^{\text{X}}$: A statement which can not be checked because the information is not available, or it would be too time-consuming.

OTHER$^{\text{O}}$: Any other mistakes, a last-resort category for when the text is nonsensical.

The colours and superscript for these types are explained in Figure 1.

3.2 Curation and complex annotation

When multiple annotators check each text, a curation process is used to resolve disagreement between annotators. This is done by a researcher, although it could be performed by separate, suitably trained annotator. All errors that are found by the majority of annotators (2/3 in the shared task) are taken to form the Gold Standard Mistake List (GSML). In cases where the spans or categories differ slightly, but it is clear the annotators found the same fundamental problem in the text, the curator can include the error in the GSML, noting how many annotators found the underlying problem.

To highlight errors in text using our annotation scheme we use an accessible colour palette (https://davidmathlogic.com/colorblind, https://personal.sron.nl/~pault) with the addition of superscript letters such that annotations can be read even in black and white. Our annotation categories with their styles are: NAME^{N} NUMBER^{U} WORD^{W} CONTEXT^{C} NOT CHECKABLE^{X} and OTHER^{O}

Figure 1: Annotation key for error types (used throughout)

For example, consider the two following annotated sentences:

> Steph Curry scored 30^{U} points to go with 9 rebounds.
>
> Steph Curry^{N} scored 30 points to go with 9 rebounds.

If in the game Curry had 9 rebounds, but only 25 points, then the sentence can be annotated as per the first example. However, if another player, Kevin Durant, had 30 points and 9 rebounds, then an annotator could instead mark the name as an error (second example). We refer to such cases as complex annotations, where there might be multiple valid ways to indicate an error in the text. To help mitigate this problem, annotators are asked to use as few annotations as possible to express the underlying error. They are also asked to prioritise error categories: NAME^{N} > NUMBER^{U} > WORD^{W} > CONTEXT^{C} > NOT CHECKABLE^{X} > OTHER^{O}, e.g., Steph Curry^{N} would be the preferred annotation in the above example. Errors in neural generated texts are not always this simple. Generally speaking, the more errors that are in a sentence, the more difficult it becomes to find the preferred annotation.

4 Experiment setup

Generated basketball summaries from the same three systems were used in each experiment. The systems were the conditional copy system of Wiseman et al. (2017), the document plan system of Puduppully et al. (2019), and the hierarchical encoder of Rebuffel et al. (2020). These systems were chosen because each was considered state-of-the-art (by one or more metrics) at the time of publication. Generated game summaries were provided by the authors of each paper, with the original RotoWire dataset and partitions having been used. The set of distinct games from the Rotowire test set was taken then randomly converted to a list. Selection of games from within this random list was arbitrary, with games for the training GSML taken from the start of the list, and those for the test GSML taken working backwards from the end.

Each input game record was processed by only one system, therefore there was no comparison between systems of generated texts for the same game data. This was because the original goal was the development of annotation techniques and a list of gold errors, and not the comparison of different systems. When comparing systems retrospectively as we are in this paper, we do so with this caveat.

The experiments we performed to collect data for the shared task were were:

> **Experiment A**: 21 texts, 7 per system (training set pt. 1). Collected in July 2020.
>
> **Experiment B**: 39 texts, 13 per system (training set pt. 2). Collected in January 2021.
>
> **Experiment C**: 30 texts, 10 per system (test set). Collected in March 2021.

where each text is a complete summary (approx. 300 words) of a basketball game, generated by one of the three neural systems.

4.1 Rotowire dataset partitions

The standard partitions of the RotoWire dataset have problems of training, validation, and test partition contamination, whereby the same game record exists within multiple partitions but with a different reference text (Iso et al., 2019; Thomson et al., 2020). Neural systems will memorise the text seen for a game in training, meaning that texts generated for such games in the test set will exhibit human-like levels of factual accuracy. For this reason, games in the standard RotoWire test set that had been seen during training or validation were excluded from selection for our experiment.

4.2 Annotator recruitment, instruction, and fair treatment

Annotators were recruited on the Amazon Mechanical Turk platform. We limited applicants to those from the United States (where basketball is a popular sport), who held U.S. Bachelor degrees and were MTurk Masters[2]. We also screened participants with a qualifying task whereby they had to find 14 of 20 known errors in a text we had already annotated ourselves. Some errors such as whether a team could be said to 'dominate' might be subjective. It is for this reason that the qualifying bar was not set higher. Recruitment was performed only once, before any experiments. Four workers passed the qualification task and three chose to undertake the annotation work; these same three annotators each examined all 90 texts over the 3 experiments.

Workers were paid $8US per tasks, with each task taking 20-25 minutes. The aim was to pay $20US per hour, well above the minimum wage in the UK or any U.S. state. Based on feedback from the workers, we met or exceeded this rate. All workers were paid for all tasks, even those who failed qualification (with the exception of workers who submitted forms with zero errors).

In addition to paying workers fairly and promptly, we considered the impact that doing the work may have on their well-being, and made efforts to provide a positive working environment. Annotators can find repetitive tasks stressful Strassel et al. (2000). This stress could be compounded on crowd-source platforms where workers might have prior experience of being treated unfairly (Shmueli et al., 2021).

We maintained good communication by responding to queries they had and reassuring them that we were interested in their opinion, and they would not be punished for a "wrong" answer. In cases where annotators made procedural mistakes, we still paid them for the work and simply asked that they supply a correction. Feedback from annotators was highly positive, on both the level of communication and how much they enjoyed the work (it was less repetitive than other tasks they had done). Our approach was borne out of common courtesy, there was no complex process and it did not slow down the project. It also hopefully resulted in higher quality of annotation.

4.3 User Interface

We considered creating a custom annotation interface, although due to the relatively small number of annotated texts we instead opted for having annotators highlight errors in an MS Word document, then list the error type and correction in a list below the text. A researcher[3] then transcribed verbatim the annotations to an annotation tool, WebAnno[4]. The transcription process increases the time taken to process each annotated summary, which might be prohibitive in larger studies. It may also introduce a small amount of human error, which could be checked by repeating the transcription. However, given the volume of errors, we believe that mistakes in transcription will have a negligible effect on error counts in this study. This may change as models approach or exceed human levels of factual accuracy. In our case, we believe that the manual transcription work did not take more time than development and deployment of an interface would have. As a low-tech approach, it also reduced the possibility of failure. In the worst case scenario where a document failed to upload (did not happen) a worker could simply send us the document again. A failure on the interface could have resulted in data loss, so software testing would have been required.

Each MS Word document included our 4 pages of instructions and an annotated example that workers had been shown during qualification. Workers were told these were for their reference, and only the text to be annotated changed in each document. These instructions did change slightly between experiments, with difficult examples that annotators had queried being included as examples. The **NOT CHECKABLE**^{X} was also clarified.

4.4 Mean error count

Whilst the purpose of the original study was to find a list of representative errors for analysis and comparison with alternative approaches, we define here the mean error count (MEC) as a measure. The mean is calculated as the total number of errors by the number of summaries.

We consider pairwise combinations of system and error type granularity. System groupings are:

Ensemble: Errors from all systems. This is what we originally set out to collect; a set of errors that is representative of the types of mistakes found in neural system output.

System: Errors for each individual system.

[2] Reliable workers as determined by an Amazon internal metric

[3] The first author of this paper.

[4] https://webanno.github.io

Error type groupings are:

> **Overall Errors**: The count of all errors, of any type.
>
> **Per-type Errors**: The breakdown of errors by type.

For this study we consider the MEC at the level of reported errors, i.e., we count annotated token spans within each basketball summary that were provided by the annotators then combined by the curation process. This was the simplest option. We considered normalizing by token count but decided against it because annotator reported errors can span anything from a single token, to five or more. This does not necessarily mean the 5-token error is equivalent by any measure of severity to 5 single-token errors. Consider the two[5] annotated and tokenized sentences below:

> Steph Curry scored 28 points (**9**$^{\text{U}}$ - **15**$^{\text{U}}$ - FG ; **4**$^{\text{U}}$ - **10**$^{\text{U}}$ 3Pt ; **2**$^{\text{U}}$ - 3 FT) .
>
> The Warriors were the dominant team in this second half of a **back - to - back**$^{\text{W}}$

These sentence may seem equally erroneous if normalized at the token level; both sentences have 5 annotated tokens. However, the annotations in the first sentence represent 5 separately reported **NUMBER**$^{\text{U}}$ errors, whereas in the second there is a single **WORD**$^{\text{W}}$ error spanning 5 tokens. The numbers in the first sentence are part of a shot breakdown, a terse domain specific syntax which shows the made and attempted shots at different ranges. A back-to-back means the team will play games on consecutive days. The problem described here may be compounded by the numbers within the shot breakdown always being included in pairs, they are the numerator and denominator of a fraction and each pair could be considered as a single error. Since we had asked to annotators to report **NUMBER**$^{\text{U}}$ errors individually in our instructions, we performed the analysis at this level.

5 Results

We calculated the mean error counts (total errors by documents in experiment), as well as the coefficient of variation, CV*[6] (Belz et al., 2022). See our repository[7] for complete code and data, including the calculation of mean errors from the GSML. All values are calculated then rounded to two decimal places for inclusion in tables here.

Table 1: Mean Error Count (MEC) for Ensemble

experiment MEC			
A	**B**	**C**	**CV***
19.62	20.56	20.73	3.61

Table 2: Mean Error Count (MEC) for each type within the Ensemble

	experiment MEC			
error type	**A**	**B**	**C**	**CV***
NAME	5.33	5.26	7.07	21.26
NUMBER	8.86	7.38	7.47	12.80
WORD	4.43	6.18	4.67	22.80
CONTEXT	0.76	0.90	0.27	63.22
N-CHECK	0.19	0.85	1.27	86.35
OTHER	0.05	0.00	0.00	211.73

Ensemble overall errors: We can see from Table 1 than the mean error count (MEC) had low variance between experiments, with a coefficient of variation of 3.61. This is what the experiments had originally set out to do; acquire representative samples of errors from neural systems. That similar quantities were found from the same ensemble of systems within each experiment is reassuring.

Ensemble per-type errors: When looking at the per-type breakdown for Ensemble errors (Table 2), we can see that each individual variance is higher than for the overall counts. This is not unexpected, given the complex error resolution problem. The greatest variance is seen in error types having lower frequency; of the 1,836 total errors in the GSML, only about 4% were **NOT CHECKABLE**$^{\text{X}}$, 3% were **CONTEXT**$^{\text{C}}$, and a single **OTHER**$^{\text{O}}$ error was reported between all systems and experiments.

Table 3: Mean Error Count (MEC) for each system

	experiment MEC			
system	**A**	**B**	**C**	**CV***
cond-copy	21.57	25.54	26.60	13.19
doc-plan	21.86	17.77	18.90	13.23
h-encoder	15.43	18.38	16.70	10.77

[5] This is an artificial example for clarity of comparison and conciseness, although multiple errors of both types can be found in the GSML.

[6] https://github.com/asbelz/coeff-var

[7] https://github.com/nlgcat/uoa-reprogen-2022

Table 4: Mean Error Count (MEC) for each error type within each system

system	error type	experiment MEC A	experiment MEC B	experiment MEC C	CV*
conditional copy	NAME	5.57	6.00	7.80	22.39
conditional copy	NUMBER	9.29	10.92	11.40	12.87
conditional copy	WORD	5.86	7.15	6.00	13.72
conditional copy	CONTEXT	0.43	0.23	0.10	79.89
conditional copy	NOT CHECKABLE	0.43	1.23	1.30	60.02
conditional copy	OTHER	0.00	0.00	0.00	-
document plan	NAME	5.71	5.08	6.40	14.12
document plan	NUMBER	11.14	6.15	7.00	40.30
document plan	WORD	4.43	5.38	3.80	21.50
document plan	CONTEXT	0.57	0.54	0.10	79.77
document plan	NOT CHECKABLE	0.00	0.62	1.60	133.60
document plan	OTHER	0.00	0.00	0.00	-
hierarchical encoder	NAME	4.71	4.69	7.00	29.64
hierarchical encoder	NUMBER	6.14	5.08	4.00	25.82
hierarchical encoder	WORD	3.00	6.00	4.20	41.95
hierarchical encoder	CONTEXT	1.29	1.92	0.60	63.71
hierarchical encoder	NOT CHECKABLE	0.14	0.69	0.90	82.68
hierarchical encoder	OTHER	0.14	0.00	0.00	211.73

System overall errors: The mean error count remained fairly constant for each system, although there were higher coefficients of variation than for the ensemble, ranging from 10.77 to 13.23 as shown in Table 3.

System per-type errors: When looking at the per-type breakdown for per-system errors, we see in Table 4 we see higher variance, especially for the less frequent error types.

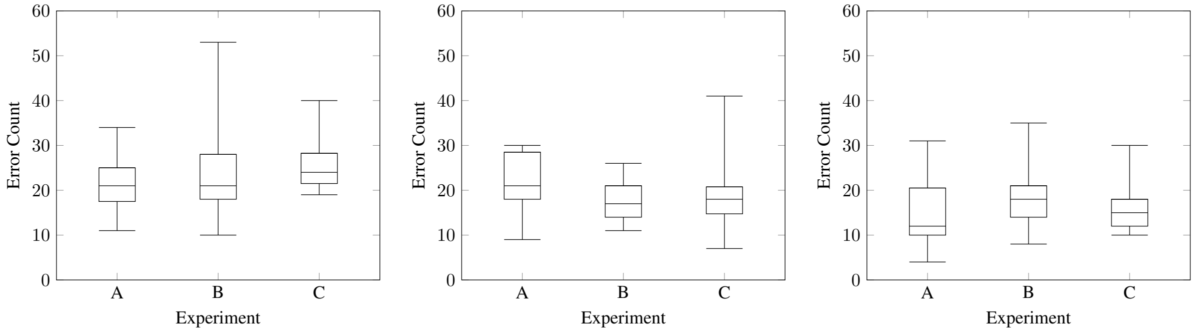
Figure 2 shows the spread of per-document error means for each system, within each experiment. It is worth noting that no generated text was error free and they rarely had fewer than 10 errors.

6 Discussion

The experiments showed that when taking an ensemble of 3 models to create the GSML, the mean error count remained relatively stable between experiments. This adds to the evidence of the gold standard protocol being a reliable method of obtaining instances of errors which can then be used to evaluate alternative methods, such as metrics (Kasner et al., 2021; Nomoto, 2021; Rezgui et al., 2021) or cheaper human evaluations (Garneau and Lamontagne, 2021).

The level of reproducibility for the gold standard protocol when evaluating systems is harder to determine. With a small number of texts per system in each experiment, the means are susceptible to the effects of outlier documents, such as rare cases where the document had 50 or more errors. The per-system coefficient of variation of ranged from about 10 to 13, which is similar to some CV* values reported for other human evaluations (Belz et al., 2021b). The per-type results are limited by the low frequency of some types, but also by the complex resolution problem. In some cases there can be many correct ways in which a text can be annotated for errors, using different combinations of error types.

An alternative way to measure the reproducibility of the protocol would be to run all three experiments again with different annotators. We could then look at how the sets of errors from the original experiment and the reproduction overlap. However, the problem of complex error resolution rears its head again. Just because annotation spans or categories differ, does not necessarily mean that both sets of annotators did not correctly identify the same underlying problem. This addresses the issues of complex error resolution in a way which the exact comparison of token spans and labels does not. Error verifiers can be asked to consider whether a reported error is one valid way to indicate the underlying problem. For discussion of complex annotation see Thomson and Reiter (2021).



(a) Conditional Copy

(b) Document Plan

(c) Hierarchical Encoder

Figure 2: Box plot for each system showing the spread of errors within each experiment.

Measuring reproducibility allows us to determine whether our evaluation protocols are reliable. However, it is not the only method for doing so. An alternative for validating the GSML would be to show individual errors to participants that are familiar with the annotation process, then ask them to indicate whether the highlight represents an error. This would allow us to measure the precision of annotators. We might also check in the same way, any errors reported by a minority of annotators. This would determine whether these errors were false positives, simply missed by the other annotators, or the result of differing annotations for complex errors.

7 Conclusion

This reproduction study showed that there was little variance in the mean error count between the different experiments that were used for the shared task data collection. Increased variance was observed when comparing the mean counts of different error types, and/or when comparing systems. These values do not, however, tell the whole story of this detailed evaluation protocol. For annotation based approaches the agreement between annotators can be measured (Popović and Belz, 2021), although with complex data-to-text, a lack of measurable agreement (based on token overlap) does not necessarily mean that annotators did not find similar underlying problems. An alternative when working at the level of individual errors might be to verify each reported error by asking additional annotators whether they agree with the reported error.

Acknowledgements

We would like to thank the Mechanical Turk annotators for their diligent work, as well as the reviewers for their helpful feedback. The original study was performed as part of Craig Thomson's PhD research, which was funded by an EPSRC NPIF studentship grant (EP/R512412/1).

References

Monya Baker. 2016. Is there a reproducibility crisis? *Nature*, 533:452–454.

Anya Belz, Shubham Agarwal, Anastasia Shimorina, and Ehud Reiter. 2021a. A systematic review of reproducibility research in natural language processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 381–393, Online. Association for Computational Linguistics.

Anya Belz, Maja Popovic, and Simon Mille. 2022. Quantified reproducibility assessment of NLP results. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16–28, Dublin, Ireland. Association for Computational Linguistics.

Anya Belz, Anastasia Shimorina, Shubham Agarwal, and Ehud Reiter. 2021b. The ReproGen shared task on reproducibility of human evaluations in NLG: Overview and results. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 249–258, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Ashwin Devaraj, William Sheffield, Byron Wallace, and Junyi Jessy Li. 2022. Evaluating factuality in text simplification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7331–7345, Dublin, Ireland. Association for Computational Linguistics.

Yao Dou, Maxwell Forbes, Rik Koncel-Kedziorski, Noah Smith, and Yejin Choi. 2022. Is GPT-3 text indistinguishable from human text? scarecrow: A framework for scrutinizing machine text. In *Proceedings of the 60th Annual Meeting of the Association* *for Computational Linguistics (Volume 1: Long Papers)*, pages 7250–7274, Dublin, Ireland. Association for Computational Linguistics.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Nicolas Garneau and Luc Lamontagne. 2021. Shared task in evaluating accuracy: Leveraging pre-annotations in the validation process. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 266–270, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Sebastian Gehrmann, Elizabeth Clark, and Thibault Sellam. 2022. Repairing the cracked foundation: A survey of obstacles in evaluation practices for generated text.

Hayate Iso, Yui Uehara, Tatsuya Ishigaki, Hiroshi Noji, Eiji Aramaki, Ichiro Kobayashi, Yusuke Miyao, Naoaki Okazaki, and Hiroya Takamura. 2019. Learning to select, track, and generate for data-to-text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2102–2113, Florence, Italy. Association for Computational Linguistics.

Zdeněk Kasner, Simon Mille, and Ondřej Dušek. 2021. Text-in-context: Token-level error detection for table-to-text generation. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 259–265, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Margot Mieskes, Karën Fort, Aurélie Névéol, Cyril Grouin, and Kevin Cohen. 2019. Community perspective on replicability in natural language processing. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 768–775, Varna, Bulgaria. INCOMA Ltd.

Tadashi Nomoto. 2021. Grounding NBA matchup summaries. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 276–281, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Maja Popović. 2020. Informative manual evaluation of machine translation output. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5059–5069, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Maja Popović and Anya Belz. 2021. A reproduction study of an annotation-based human evaluation of MT outputs. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 293–300, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with content selection and planning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6908–6915.

Clément Rebuffel, Laure Soulier, Geoffrey Scoutheeten, and Patrick Gallinari. 2020. A hierarchical model for data-to-text generation. In *Advances in Information Retrieval*, pages 65–80, Cham. Springer International Publishing.

Rayhane Rezgui, Mohammed Saeed, and Paolo Papotti. 2021. Automatic verification of data summaries. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 271–275, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Boaz Shmueli, Jan Fell, Soumya Ray, and Lun-Wei Ku. 2021. Beyond fair pay: Ethical implications of NLP crowdsourcing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3758–3769, Online. Association for Computational Linguistics.

Stephanie Strassel, David Graff, Nii Martey, and Christopher Cieri. 2000. Quality control in large annotation projects involving multiple judges: The case of the TDT corpora. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece. European Language Resources Association (ELRA).

Craig Thomson and Ehud Reiter. 2020. A gold standard methodology for evaluating accuracy in data-to-text systems. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 158–168, Dublin, Ireland. Association for Computational Linguistics.

Craig Thomson and Ehud Reiter. 2021. Generation challenges: Results of the accuracy evaluation shared task. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 240–248, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Craig Thomson, Ehud Reiter, and Somayajulu Sripada. 2020. SportSett:basketball - a robust and maintainable data-set for natural language generation. In *Proceedings of the Workshop on Intelligent Information Processing and Natural Language Generation*, pages 32–40, Santiago de Compostela, Spain. Association for Computational Lingustics.

Chris van der Lee, Albert Gatt, Emiel van Miltenburg, Sander Wubben, and Emiel Krahmer. 2019. Best practices for the human evaluation of automatically generated text. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 355–368, Tokyo, Japan. Association for Computational Linguistics.

Emiel van Miltenburg, Miruna Clinciu, Ondřej Dušek, Dimitra Gkatzia, Stephanie Inglis, Leo Leppänen, Saad Mahamood, Emma Manning, Stephanie Schoch, Craig Thomson, and Luou Wen. 2021. Underreporting of errors in NLG output, and what to do about it. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 140–153, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

Reproducing a Manual Evaluation of the Simplicity of Text Simplification System Outputs

Maja Popović, Sheila Castilho, Rudali Huidrom and Anya Belz
ADAPT Centre
School of Computing
Dublin City University, Ireland
name.surname@adaptcentre.ie

Abstract

In this paper we describe our reproduction study of the human evaluation of text simplicity reported by Nisioi et al. (2017). The work was carried out as part of the ReproGen Shared Task 2022 on Reproducibility of Evaluations in NLG. Our aim was to repeat the evaluation of simplicity for nine automatic text simplification systems with a different set of evaluators. We describe our experimental design together with the known aspects of the original experimental design and present the results from both studies. Pearson correlation between the original and reproduction scores is moderate to high (0.776). Inter-annotator agreement in the reproduction study is lower (0.40) than in the original study (0.66). We discuss challenges arising from the unavailability of certain aspects of the original set-up, and make several suggestions as to how reproduction of similar evaluations can be made easier in future.

1 Introduction

Against a background of growing interest in approaches to reproducibility assessment in general, and specific reproduction studies in particular, this paper reports a reproduction study of a human evaluation of text simplicity carried out as part of the ReproGen Shared Task 2022 on Reproducibility of Evaluations in NLG. We participated with a contribution in Track A, carrying out a reproduction study of the human evaluation of sentence simplicity reported by Nisioi et al. (2017), one of the five papers offered in the track.

In the original paper, nine automatic text simplification systems were evaluated by human annotators for four different criteria: Correctness and number of changes, Meaning Preservation, Grammaticality, and Simplicity. In this paper, we concentrate only on Simplicity. We first summarise the original study and describe the details of our reproduction study (Section 2. We then present the results from both studies (Section 3) in terms of the system-level Simplicity scores of the nine systems, and the inter-annotator agreement estimated as quadratic Cohen's Kappa. We also report Pearson's correlation coefficient between the original and the reproduction system scores.

We finish (Section 4) with a discussion of the differences between the two studies and the impact of missing information about the original set-up, and suggest how to make future human evaluations easier to repeat.

2 Experimental Design in Original and Reproduction Study

A commonly cited motivation for automatic text simplification (ATS) systems is that texts containing uncommon words or long and complicated sentences can be difficult to read and understand by people as well as difficult to analyze by machines. ATS is the process of transforming one text into another text which ideally has the same meaning, but is easier to read and understand by a wider audience and also easier to process with NLP tools. ATS systems can be rule-based or corpus-based, namely trained on parallel corpora consisting of original texts and their simplified versions.

For human evaluation of ATS systems, the usual quality criteria are Meaning Preservation (the degree to which the meaning of the original text is retained in the simplified output; analogous to Adequacy in MT), Grammaticality (whether the grammar of the generated output is good), and Simplicity (how difficult/simple the generated output is).

This paper focuses on simplicity evaluation in the form of comparing the automatically simplified output with the original text that was the input: the original sentence is presented together with its automatically simplified version, and the evaluators are asked whether the simplified version is simpler, equally simple/difficult, or more difficult than the original.

2.1 Original experiment

The original paper (Nisioi et al., 2017) reported the first attempt of using neural networks for automatic text simplification. Two basic neural text simplification (NTS) system variants for the English language were developed, one relying only on internal word representations (which we refer to as NTS in tables and results below), and the other additionally using external word2vec representations (NTS-W2V). Each system variant was used to generate outputs in three different ways: (i) by beam search with size 5 (NTS-DEFAULT and NTS-W2V-DEFAULT), (ii) by re-ranking an n-best list using the automatic metric BLEU (Post, 2018) (NTS-BLEU and NTS-W2V-BLEU), and (iii) by re-ranking using the SARI metric (Xu et al., 2016) (NTS-SARI and NTS-W2V-SARI). These six system variants together with an additional three publicly available systems (for which outputs generated in previous work were available), referred to as PBSMT, SARI+PPDB and LIGHTLS in results tables and briefly explained in the next section, were manually evaluated in terms of the three criteria of Meaning Preservation, Grammaticality and Simplicity. In addition, BLEU and SARI scores were calculated.

The outputs from all nine systems, as well as scripts for both automatic evaluation metrics are publicly available.[1] Human sentence-level annotations are however not published, and only the system-level scores were reported in the paper.

2.1.1 Evaluation Data

The developed NTS systems were evaluated on 359 publicly available sentences originating from English Wikipedia[2] and previously released by Xu et al. (2016). These sentences were simplified with the NTS system variants from Nisioi et al. (2017) as well as the three previous systems: PBSMT, a phrase-based SMT system with reranking (Wubben et al., 2012), SARI+PPDB, a paraphrase-based system proposed by Xu et al. (2016), and LIGHTLS, an unsupervised lexical simplification system based on word embeddings (Glavaš and Štajner, 2015).

For each of the nine systems, automatic scores were calculated on all sentences, whereas human evaluation was carried out on the first 70 sentences only. Since each sentence was simplified by 9 systems, 630 sentences were manually evaluated in total.

2.2 Evaluating simplicity

In both original and reproduction study, the manual evaluation of simplicity was performed by three non-native English speakers who were given the original sentence and an automatically generated simplification of it, one pair at a time. They were asked to assign a score to each pair according to the following guidelines:

- +2 if the simplified version is much simpler than the original,
- +1 if the simplified version is somewhat simpler than the original,
- 0 if they are equally simple/difficult,
- -1 if the simplified version is somewhat more difficult than the original, and
- -2 if the simplified version is much more difficult than the original.

The inter-annotator agreement reported by Nisioi et al. (2017) (in the form of quadratic Cohen's Kappa) was 0.66.

The reported aggregated system-level scores (mean sentence-level scores, shown in Table 1, Simplicity/original/score column) indicated that all variants of the newly proposed NTS model substantially outperform all of the comparator systems in terms of simplicity, i.e. generate outputs with a higher level of simplicity than the three previous state-of-the-art ATS systems.

2.3 Reproduction study

Our reproduction experiment was carried out on the same data as the original one, namely the first 70 sentences of the test set simplified by each of the nine systems. The evaluation was carried out by three non-native speakers, too, same as in the original evaluation. They received the same instructions as described in the original paper and in Section 2.1.

Further details about the original evaluation which may or may not have affected results and reproducibility were, however, not available.[3] Such details where we have information only for our reproduction include:

[1] https://github.com/senisioi/NeuralTextSimplification

[2] https://github.com/cocoxu/simplification/

[3] After contacting the authors of the original paper, the responses received were from authors not familiar with the details requested.

- *Native languages of evaluators*

  Reproduction: each evaluator had a different native language (Serbian, Brazilian Portuguese and Manipuri).

- *Evaluators' background*

  Reproduction: all the evaluators were computational linguistics researchers.

- *Evaluators' experience with TS and its evaluation*

  Reproduction: one evaluator had experience with TS evaluation and thus was familiar with the concept of simplicity, whereas the other two did not.

- *Whether the evaluators were able to ask any additional questions or only worked with the above guidelines*

  Reproduction: the two evaluators without experience needed a few additional instructions and examples in order to fully understand the concept of simplicity in this context, and to be able to separate it from meaning and grammar.

- *Number of sentences assessed by each evaluator*

  Reproduction: one evaluator (the one with the experience with TS evaluation) annotated all sentences whereas the other two evaluators annotated half of the sentences each.

  As with the other details in this list, we do not know how the sentences were distributed among the three evaluators in the original study.

- *Number of multiply annotated sentences used for IAA*

  Reproduction: each sentence was annotated by two evaluators, IAA is computed on the whole set.

  We do not know whether this was the case in the original experiment or only a subset of sentences was annotated by more than one evaluator. We also do not know whether any (or all) sentences were evaluated by all three evaluators.

It might also be worth noting that in our reproduction identical sentence pairs (where the output is identical to the input) were not presented to the evaluators but were immediately assigned the score 0. We do not know whether the same was the case in the original evaluation.

3 Results

3.1 Comparing the different ATS systems

The 'original' column in Table 1 presents the ranks and system-level reproduction scores obtained for the nine systems in the original study, and the 'reproduction' column presents the same for the reproduction study. It can be seen that overall, the three ATS systems from previous work, PBSMT, SARI+PPDB and LIGHTLS, have notably lower reproduction scores in both studies, so that the claim from the original paper that the proposed NTS systems generate outputs with higher levels of simplicity is confirmed.

As for comparing the individual NTS systems, the reproduction scores indicate that the NTS-w2v-SARI system (re-ranking with SARI scores) reaches the highest simplicity levels, as well as that the re-ranking is generally beneficial for both model variants. The original scores, on the other hand indicate that re-ranking with automatic metrics was of benefit to the NTS-w2v variant, but for the NTS variant, while re-ranking with BLEU (NTS-BLEU) led to a dramatic improvement in reproduction, re-ranking with SARI (NTS-SARI) actually dropped the reproduction score. In contrast, according to the reproduction scores, re-ranking with SARI had more of a beneficial effect than re-ranking with BLEU.

The last column in Table 1 shows the small-sample coefficient of variation (CV*) for each of the individual system-level reproduction score pairs across the two experiments as a quantified measure of degree of reproducibility (Belz et al., 2022). Lower CV* indicates better reproducibility. Here, the CV* scores show that some systems' human scores are more reproducible than others, but it is not immediately obvious why the human evaluators in the original and reproduction studies should have disagreed particularly about the two systems with the highest CV* (NTS-BLEU and PBSMT).

Pearson correlation coefficient between the original and the reproduction scores is 0.766, i.e. moderate to high. Spearman's rank correlation is slightly higher at 0.787.

3.2 Inter-annotator agreement (IAA)

The IAA in the original experiment was reported as quadratic Cohen's Kappa with a value of 0.66. We also calculated this coefficient for our reproduction, where and the value is lower, 0.40. Unfortunately, we cannot really interpret this discrepancy because,

automatic text simplification system	Simplicity				small-sample coefficient of variation (CV*) ↓
	original		reproduction		
	rank	score	rank	score	
NTS DEFAULT	(3)	0.46	(5)	0.33	5.41
NTS-SARI	(5)	0.38	(3/4)	0.34	1.69
NTS-BLEU	(1)	0.92	(3/4)	0.34	22.0
NTS-W2V-DEFAULT	(6)	0.21	(6)	0.32	4.84
NTS-w2v-SARI	(2)	0.63	(1)	0.46	6.66
NTS-W2V-BLEU	(4)	0.40	(2)	0.36	1.68
PBSMT	(9)	-0.55	(7)	0.08	35.6
SARI+PPDB	(7)	0.03	(9)	0.01	0.99
LIGHTLS	(8)	-0.01	(8)	0.03	1.98

Table 1: System-level Simplicity scores for the nine ATS outputs and system ranks according to these scores, together with CV*s between scores in original and reproduction experiment. Note that CV* is computed on shifted scores, i.e. while the scores assigned by the human evaluators ranged from -2 to +2, before computing CV* they were shifted to range from 0 to 4.

as mentioned in Section 2.3, many of the details of the original experiment are missing, and we do not know what subset of sentences IAA was computed over in the original experiment, or how many individual scores per sentence. If the IAA values do reflect an actual difference, then one possible reason might be the experience of the evaluators with TS and familiarity with the notion of simplicity. In the reproduction study, only one evaluator was already familiar with it while the other two required additional explications. Furthermore, due to how sentences were assigned to evaluators, IAA is calculated only between the experienced and inexperienced annotators and not between the two inexperienced. These factors could generally contribute to a lower IAA. On the other hand, it is possible that all evaluators in the original experiment had experience with TS evaluation so that this is the reason of a higher IAA, however this is only a speculation.

Availability of the sentence-level scores from the original study would have helped to compare the scores for each sentence and potentially find patterns in sentences that make human evaluation more difficult to reproduce.

3.3 Comparison with reproduction of automatic scores

In order to illustrate quantitatively the differences that can arise between reproducing human and reproducing automatic evaluations, Table 2 presents the Simplicity and CV* scores for two NTS system variants, NTS-DEFAULT and NTS-W2V-DEFAULT, together with their automatic metric scores (BLEU and SARI). These results are compared and analysed more comprehensively elsewhere (Belz et al., 2022).

The 'original' column shows the results reported in the original paper, the 'repr1' column shows the results reported in an earlier reproduction paper (Cooper and Shardlow, 2020) at REPROLANG 2020[4], the 'repr2' and 'repr3' columns show the results reported by Belz et al. (2022) when using two different evaluation scripts for BLEU, and the 'repr4' column shows results from the human evaluation carried out in the present work.

It can be noted that, while CV* values for the SARI metric are 0 (perfectly reproduced) and for the BLEU metric are around 1 (reflecting slight differences in implementation and tokenisation), CV* values for human Simplicity scores are over 4, demonstrating that human evaluation was more difficult to reproduce.

4 Conclusions

This paper reported the results of a reproduction study of a human evaluation of text simplicity. The obtained scores confirm some of the findings of the original paper, however findings relating to whether or not re-ranking with BLEU or SARI helped were not aligned in the two studies, in some cases showing opposite effects. Pearson correlation between the studies was moderate to high at 0.766. The inter-annotator agreement was lower in the reproduction study, 0.40 vs. 0.66, but we do not know whether it was computed in a comparable way.

[4] https://lrec2020.lrec-conf.org/en/reprolang2020/

		evaluation round					
metric	output	original	repr1	repr2	repr3	repr4	CV* ↓
BLEU ↑	NTS default	84.51	84.50	85.60	84.20	–	0.838
(automatic)	NTS-w2v default	87.50	–	89.36	88.80	–	1.314
SARI ↑	NTS default	30.65	30.65	30.65	–	–	0
(automatic)	NTS-v2w default	31.11	–	31.11	–	–	0
Simplicity ↑	NTS default	0.46	–	–	–	0.33	5.41
(human)	NTS-v2w default	0.21	–	–	–	0.32	4.84

Table 2: Comparing CV*s of automatic and human system-level scores for two ATS systems, NTS DEFAULT and NTS-W2V DEFAULT. The CV*s indicate that human evaluation is more difficult to reproduce (presumably exacerbated when many experimental details are missing).

A deeper analysis of these differences is unfortunately not possible because we lack too many details for the original set-up. Also, sentence-level human annotations which would be helpful are not published (while the models and the automatic evaluation scripts are).

It appears to be the case that there is a tendency for comprehensive details about the human evaluation process to be reported only in papers dealing with human evaluation itself, although even in these, the provided information is not often fully complete. In papers where human evaluation is not the focus but only a method to assess the system(s), usually only very shallow information is provided, if any. Moreover, it is often the case that the authors themselves perform evaluations, sometimes with no overlap, which makes it impossible to report IAA. Fully reporting such details is disincentivised as doing so may lead to more negative reviews. Human evaluation is time and resource-expensive and it is usually not possible to (i) evaluate large amounts of text, (ii) involve a large number of evaluators, or (iii) evaluate large portions of text by several evaluators for IAA, because all these factors increase cost further.

As in previous work (Howcroft et al., 2020; Belz et al., 2020)), we conclude that reporting more details about human evaluation experiments would be of benefit scientifically. Details of human evaluations should be provided in each paper, even if the conditions were not perfect (and they often are not). It is more scientifically rigorous as well as more useful to provide full details than not providing information for fear of negative review.

Acknowledgments

All authors benefit from being members of the ADAPT SFI Centre for Digital Media Technology. The Centre is funded by Science Foundation Ireland through the SFI Research Centres Programme and is co-funded under the European Regional Development Fund (ERDF) through Grant 13/RC/2106. Popović's work is fully supported by the Centre. Huidrom's work is fully supported by the Faculty of Engineering and Computing, DCU, via a PhD grant.

References

Anya Belz, Simon Mille, and David Howcroft. 2020. Disentangling the properties of human evaluation methods: A classification system to support comparability, meta-evaluation and reproducibility testing. In *International Natural Language Generation Conference 2020 (INLG'20)*.

Anya Belz, Maja Popović, and Simon Mille. 2022. Quantified reproducibility assessment of NLP results. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL 2022)*, pages 16–28, Dublin, Ireland. Association for Computational Linguistics.

Michael Cooper and Matthew Shardlow. 2020. CombiNMT: An exploration into neural text simplification models. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5588–5594, Marseille, France. European Language Resources Association.

Goran Glavaš and Sanja Štajner. 2015. Simplifying lexical simplification: Do we need simplified corpora? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 63–68, Beijing, China. Association for Computational Linguistics.

David M. Howcroft, Anya Belz, Miruna-Adriana Clinciu, Dimitra Gkatzia, Sadid A. Hasan, Saad Mahamood, Simon Mille, Emiel van Miltenburg, Sashank Santhanam, and Verena Rieser. 2020. Twenty years of confusion in human evaluation: NLG needs evaluation sheets and standardised definitions. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 169–182, Dublin, Ireland. Association for Computational Linguistics.

Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P. Dinu. 2017. Exploring neural text simplification models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pages 85–91, Vancouver, Canada. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Sander Wubben, Antal van den Bosch, and Emiel Krahmer. 2012. Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1015–1024, Jeju Island, Korea. Association for Computational Linguistics.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

A reproduction study of methods for evaluating dialogue system output: Replicating Santhanam and Shaikh (2019)

Anouck Braggaar✉, **Frédéric Tomas**, **Peter Blomsma**, **Saar Hommes**, **Nadine Braun**, **Emiel van Miltenburg**, **Chris van der Lee**, **Martijn Goudbeek** and **Emiel Krahmer**
Tilburg University
✉ A.R.Y.Braggaar@tilburguniversity.edu

Abstract

In this paper, we describe our reproduction effort of the paper: *Towards Best Experiment Design for Evaluating Dialogue System Output* by Santhanam and Shaikh (2019) for the 2022 ReproGen shared task. We aim to produce the same results, using different human evaluators, and a different implementation of the automatic metrics used in the original paper. Although overall the study posed some challenges to reproduce (e.g. difficulties with reproduction of automatic metrics and statistics), in the end we did find that the results generally replicate the findings of Santhanam and Shaikh (2019) and seem to follow similar trends.

1 Introduction

Currently, a lot of attention is given to the reproducibility of NLP research. In this paper, we report our contributions to the 2022 ReproGen shared task (Belz et al., 2020).[1] We aim at an exact reproduction of the work by Santhanam and Shaikh (2019) on experiment design for evaluating dialogue system output. No other reproductions of this paper have been published presently. We will first give a brief summary of the paper we aimed to reproduce (§2), and explain how we replicated this research as closely to the original as possible (§3). Next, we will discuss our results and examine how these relate to the original study (§4). Lastly, we will discuss some difficulties we faced during our reproduction efforts (§5). All of our code and data can be found on GitHub.[2]

2 Summary of the original study

The study by Santhanam and Shaikh (2019) focuses on the design of the human evaluation task of evaluating dialogue system output. The purpose of the task is to see which task design yields the most consistent and highest-quality responses. The original study compared Likert scale judgments, Rank-Based Magnitude Estimation (RME), Biased Magnitude Estimation (BME) and Best-Worst Scaling (BWS) using the two metrics of readability and coherence.

Participants. The authors examined four different experimental conditions with 40 participants and 50 items each, yielding a total number of 160 participants.

Task. For each trial, participants were provided with a conversational context consisting of two turns. For each context, participants were asked to either rank or rate four different responses. Three of these responses were generated by three models trained on the Reddit conversation corpus (Dziri et al., 2019a). The other response was human-generated (i.e. the ground truth). In case of the Likert scale, people rate a generated response on a 6-point scale (1 being lowest and 6 highest). For both RME and BME (magnitude estimation) they rate the responses with respect to a given standard value (Bard et al., 1996). In the case of RME this value is always 100 while the value for BME is set by the automatic metrics of Santhanam and Shaikh (2019). In the last experiment design, BWS, participants have to rank the responses from best to worst.

Reported values. Santhanam and Shaikh (2019) report on inter-rater consistency and agreement (intraclass correlations) and also examine if prior experience of rating dialogue system output or engaging with a conversational agent is of any influence. Lastly, Spearman correlations are reported between the human ratings and automatic metrics and between the ratings of readability and coherence on the four designs.

Results. Overall, Santhanam and Shaikh (2019) find that the Likert scale performs worst on intraclass correlation, show that participants without prior experience are more consistent in their ratings,

[1] https://reprogen.github.io/

[2] https://github.com/Anouck96/ReproGen22

Proceedings of the 15th International Conference on Natural Language Generation: Generation Challenges, pages 86 - 93
July 17-22, 2022 ©2022 Association for Computational Linguistics

and report low correlation between the automatic metrics and the human ratings.

3 Study design

3.1 Surveys

Our surveys were made in the online survey platform Qualtrics.[3] We tried to follow the survey design of the original study as closely as possible. Unfortunately, in some cases, the layout was not exactly replicable. An example can be found in the best-worst scaling condition. Qualtrics neither provides the same drag-and-drop ranking question types as used in the original survey nor does it track if an item is ranked or not. We also found that the four original surveys are not completely the same in terms of conversation items and possible replies, e.g., in the Likert survey, the item "Person A: first time watching f1!" occurs twice, while in the best-worst survey, the item only occurs once. There were also some minor layout/style issues that we noticed. For example, some questions in the best-worst survey contained "readability and coherence" in bold while others did not.

3.2 Participants

Participants were recruited using Prolific[4], a crowd-sourcing recruitement platform. Only participants with English as their first language could take part in the study. The participants also were not allowed to participate in more than one of the surveys, beyond this they were also not allowed to participate in the same survey twice. We followed the minimum payment of £6.00 per hour resulting in rewards for the participants of £5.43 (Likert), £4.74 (RME), £4.64 (best-worst) and £4.88 (BME), as prolific uses the median time for payments. As in the original study, we aimed for 40 participants per survey. We started with the Likert-scale survey on Prolific. We set a time based on the mean times reported in Santhanam and Shaikh (2019) (as required by prolific), but we soon discovered that participants tended to take substantially more time, with a median completion time of about 53 minutes. In the case of the Likert scale survey, we even had participants who timed out[5] but were able to finish the survey.[6] These were kept in the dataset. This is why the Likert survey contains 42 participants. For the BME survey, we have 41 participants. In this case we have two submissions with the same Prolific ID but with different answers. As they do have different answers we have decided to keep both submissions. For the other two surveys, we have 40 submissions.

3.3 Intraclass correlation coefficient (ICC)

Intraclass correlation is used to calculate the reliability of raters (Bartko, 1966). In this study we report values for both agreement and consistency. The values can range between 0 and 1 (closer to 1 means stronger reliability) (Koo and Li, 2016). The ICC was calculated using R (R Core Team, 2017) and the irr package (Gamer et al., 2019).

3.4 Automatic metrics

To fully replicate the original results, we re-calculated the automatic metrics used by Santhanam and Shaikh (2019). Since the repository did not provide any code to generate the scores, we first contacted the authors to obtain the exact code that was used for the original paper. However, at the time of writing, the repository did not provide any code to generate the scores. Hence we needed to write our own code to process the data and generate the scores ourselves.

The original paper did not specify what library they used to compute readability. Thus, we explored different options to generate the exact same readability scores.[7] In the end, we did not find an exact match. We decided to calculate the Flesch Reading Ease using the `textacy` Python package.[8] For coherence, Santhanam and Shaikh noted that they used the method proposed by Dziri et al. (2019b). We used their repository to calculate the semantic similarity.[9]

4 Results

In this section, we follow the original study's approach in the data analysis and its structure in the

[3] See: https://www.qualtrics.com. PDF files with the surveys as they were used in Qualtrics can be found on GitHub. We do not know which platform was used for the questions in the original study, although it seems likely the authors used the Mechanical Turk platform itself.

[4] https://www.prolific.co/

[5] To ensure fair payment prolific sets a maximum time based on the set time for the study.

[6] Two participants timed out which meant they were automatically replaced by Prolific, however their completed surveys were collected in Qualtrics.

[7] We looked into textstat, py-readability-metrics, and Microsoft Word, which all generate different readability scores.

[8] https://textacy.readthedocs.io/en/latest/

[9] https://github.com/nouhadziri/DialogEntailment

organization of our results.

4.1 Experiment design and reliability of human ratings

Intraclass correlation coefficient (ICC) scores on consistency (ICC-C) and agreement (ICC-A) for the four experiment tasks can be found in Table 1. Unlike the findings reported in Santhanam and Shaikh (2019), Magnitude Estimation with anchors (RME or BME) does not show more reliable ratings than Likert scale ratings, but it does show more reliable ratings than Best-Worst ranking (BWS). Likert scale ratings result in substantially higher ICC scores in our replication. In fact, the Likert scale condition leads to the most reliable ratings, while Best-Worst ranking (BWS) represents the least reliable ratings in our results. With the exception of RME, all experimental designs show higher ICC scores in our study.

		Likert	**RME**	**BME**	**BWS**
ICC-C	R	0.90	0.89	0.91	0.83
	C	0.94	0.90	0.90	0.87
ICC-A	R	0.87	0.81	0.87	0.83
	C	0.93	0.88	0.88	0.88
Original	*R*	*0.75*	*0.95**	*0.83*	*0.75*
ICC-C	*C*	*0.83*	*0.92*	*0.81*	*0.80*
Original	*R*	*0.59*	*0.95**	*0.83*	*0.75*
ICC-A	*C*	*0.77*	*0.92*	*0.81*	*0.80*

Table 1: ICC scores for readability (R) and coherence (C) for each design. All are significant at $p < .001$. The original study scores are shown in italic with * showing the non-significant values.

4.2 Time and reliability of the rankings

As mentioned in section 3.2, participants in our replication study took a median completion time for the Likert-survey of about 53 minutes, which substantially exceeds the averages reported in the original study (see 5.1 for a more elaborate discussion on experiment times). Table 2 contains the ICC scores for raters who spent more than average time on the task, and Table 3 contains the ICC scores for raters who spent less than average time.

		Likert (n=9)	**RME** (n=16)	**BME** (n=19)	**BWS** (n=17)
ICC-C	R	0.68	0.64	0.82	0.66
	C	0.74	0.70	0.79	0.71
ICC-A	R	0.60	0.47	0.75	0.67
	C	0.72	0.66	0.75	0.71
		n=15	*n=16*	*n=15*	*n=16*
Original	*R*	*0.58*	*0.93*	*0.51*	*0.62*
ICC-C	*C*	*0.74*	*0.85*	*0.55*	*0.64*
Original	*R*	*0.52*	*0.93*	*0.51*	*0.62*
ICC-A	*C*	*0.69*	*0.86*	*0.56*	*0.64*

Table 2: ICC scores for readability (R) and coherence (C) where participants spend **above** average time. All are significant at $p < .001$. Original study scores are in italics.

		Likert (n=33)	**RME** (n=24)	**BME** (n=22)	**BWS** (n=23)
ICC-C	R	0.88	0.88	0.86	0.74
	C	0.93	0.89	0.83	0.83
ICC-A	R	0.83	0.79	0.80	0.74
	C	0.92	0.86	0.80	0.83
		n=25	*n=24*	*n=25*	*n=24*
Original	*R*	*0.61*	*0.88*	*0.81*	*0.65*
ICC-C	*C*	*0.66*	*0.85*	*0.75*	*0.76*
Original	*R*	*0.36*	*0.88*	*0.81*	*0.66*
ICC-A	*C*	*0.55*	*0.85*	*0.75*	*0.76*

Table 3: ICC scores for readability (R) and coherence (C) where participants spend **below** average time. All are significant at $p < .001$. The original study scores are in italic.

We replicate the finding of Santhanam and Shaikh (2019) that consistency and agreement are higher for raters who took less than average time to complete the task, but in all survey conditions, including RME. The RME survey showed the opposite direction in the original study. Additionally, we observe different patterns: the RME condition led to the highest reliability in the original study, both for raters taking above and below average time. In our study, RME actually leads to the lowest reliability for raters taking above average time (the highest being BME), and Likert scale ratings lead to the highest reliability for raters taking below average time (lowest in the original study).

4.3 Prior experience with dialogue system output or conversational agents and reliability of rankings

Tables 4 and 5 show the reliability scores of ratings from participants based on their prior experience with dialogue-system output evaluation. We replicate the findings reported in the original study: ratings from participants without prior experience with evaluating dialogue system output reach better reliability than ratings from participants with such prior experience. We also replicate that no prior experience with conversational agents benefits the consistency and reliability of participants' ratings (Tables 6 & 7).

		Likert (n=10)	**RME** (n=5)	**BME** (n=8)	**BWS** (n=5)
ICC-C	R	0.74	0.75	0.65	0.28*
	C	0.71	0.64	0.60	0.42
ICC-A	R	0.64	0.72	0.55	0.28*
	C	0.67	0.61	0.52	0.42
		n=15	*n=7*	*n=18*	*n=13*
Original	*R*	*0.45*	*0.37*	*0.51*	*0.54*
ICC-C	*C*	*0.38*	*0.48*	*0.55*	*0.63*
Original	*R*	*0.35*	*0.38*	*0.52*	*0.55*
ICC-A	*C*	*0.32*	*0.49*	*0.55*	*0.63*

Table 4: ICC scores for readability (R) and coherence (C) when participants **have** prior experience evaluating dialogue system output. All are significant at $p < .001$, except those indicated with *. Original study scores in italic.

		Likert (n=32)	**RME** (n=35)	**BME** (n=33)	**BWS** (n=35)
ICC-C	R	0.88	0.87	0.89	0.81
	C	0.93	0.89	0.88	0.86
ICC-A	R	0.84	0.77	0.84	0.81
	C	0.92	0.86	0.85	0.86
		n=25	*n=33*	*n=22*	*n=27*
Original	*R*	*0.71*	*0.95**	*0.83*	*0.70*
ICC-C	*C*	*0.82*	*0.92*	*0.76*	*0.72*
Original	*R*	*0.50*	*0.95**	*0.83*	*0.70*
ICC-A	*C*	*0.75*	*0.92*	*0.77*	*0.72*

Table 5: ICC scores for readability (R) and coherence (C) when participants **do not have** prior experience evaluating dialogue system output. All are significant at $p < .001$. The original study scores are shown in italics with * showing the non-significant values.

4.4 Correlation of automated calculation of readability and coherence with human ratings

Santhanam and Shaikh (2019) found low correlations between the automatic metrics and human judgements, ranging from -0.12 to 0.26. We find even lower correlations between readability and coherence scores calculated with automated methods and human ratings (see Table 8).

		Likert (n=16)	**RME** (n=15)	**BME** (n=16)	**BWS** (n=13)
ICC-C	R	0.80	0.62	0.78	0.52
	C	0.86	0.77	0.79	0.57
ICC-A	R	0.72	0.46	0.66	0.52
	C	0.83	0.73	0.74	0.58
		n=18	*n=11*	*n=23*	*n=18*
Original	*R*	*0.46*	*0.69*	*0.60*	*0.57*
ICC-C	*C*	*0.44*	*0.65*	*0.62*	*0.67*
Original	*R*	*0.37*	*0.69*	*0.61*	*0.57*
ICC-A	*C*	*0.38*	*0.65*	*0.62*	*0.67*

Table 6: ICC scores for readability (R) and coherence (C) when participants **have** prior experience engaging with conversational agents. All are significant at $p <$ .001. Original study scores in italic.

		Likert (n=26)	**RME** (n=25)	**BME** (n=25)	**BWS** (n=27)
ICC-C	R	0.85	0.89	0.87	0.78
	C	0.91	0.86	0.84	0.85
ICC-A	R	0.80	0.80	0.83	0.78
	C	0.89	0.83	0.81	0.85
		n=22	*n=29*	*n=17*	*n=22*
Original	*R*	*0.70*	*0.95**	*0.84*	*0.67*
ICC-C	*C*	*0.82*	*0.91*	*0.76*	*0.68*
Original	*R*	*0.48*	*0.95**	*0.84*	*0.67*
ICC-A	*C*	*0.75*	*0.91*	*0.76*	*0.68*

Table 7: ICC scores for readability (R) and coherence (C) when participants **do not have** prior experience engaging with conversational agents. All are significant at $p < .001$. Original study scores in italics with * showing non-significant values.

	Likert	**RME**	**BME**	**BWS**
Readability	0.01	0.01	-0.05	0.04
Coherence	0.06	-0.05	0.01	0.05
Original scores				
Readability	*0.26*	*-0.11*	*-0.12*	*-0.06*
Coherence	*-0.12*	*-0.13*	*-0.11*	*0.01*

Table 8: Spearman correlation between the ratings obtained from the automated metrics to human ratings using raw scores. Original study scores in italic.

4.5 Correlation of readability and coherence by experiment condition

We do not replicate the high correlations between the human ratings of readability and coherence obtained through RME and BME (see Spearman correlations in Table 9). For Likert, RME, and

BME, correlations are weak, while similar to the original paper, we find a moderate correlation for human ratings obtained through BWS.

	Likert	**RME**	**BME**	**BWS**
		Readability		
Coherence	0.13*	0.06	0.24**	0.48***
Original				
	0.1	*0.79****	*0.77****	*0.5****

Table 9: Spearman correlation between the ratings obtained for readability and coherence for each human evaluation method, *$p \leq .05$, **$p \leq .01$, ***$p \leq .001$. Original scores in italic.

		Likert	**RME**	**BME**	**BWS**
Readability	Mean	0.64	0.39	0.47	0.61
	Mode	0.22	0.66	0.36	0.49
Coherence	Mean	0.82	0.77	0.61	0.72
	Mode	0.39	0.57	0.31	0.48

Table 10: Correlations between the original results and the reproduction study results. The correlations were calculated on the average and modal score for each sentence, respectively. All $p \leq .001$.

5 Discussion

The purpose of this study was to gain insights that can aid the Natural Language Generation (NLG) community to increase reproducibility of papers, specifically papers regarding human and automatic evaluation of NLG results. We reproduced the work from Santhanam and Shaikh (2019) including the experiments and the analyses. The results from Santhanam and Shaikh (2019) and our reproduction are equivalent, or at least in the same order of magnitude. Table 10 displays the correlation between their results and ours regarding readability and coherence across all four measures, indicating that, mostly, their measures and ours seem to correlate quite importantly. Additionally, Table 11 discloses an overview of the results. Below we discuss observations and insights gained during this reproduction exercise.

5.1 Participants

As mentioned before in Section 3.2, we used the average time that participants took to calculate our budget on Prolific. As we found out soon with running the first survey, participants took way longer than the estimated average. Our participants took an average of approximately 58 minutes for the Likert-scale survey (SD=24.47), 54.7 minutes for RME (SD=23.39), 48.8 for BME (SD=18.68) and 48.6 for best-worst ranking (SD=22.31). Santhanam and Shaikh (2019) report averages respectively of 33, 42.8, 43 and 32.5 minutes. As can be seen from our standard deviations, the amount of time also varied greatly across participants. Our participants especially seem to take much longer for the Likert and best-worst surveys. We are not sure why the difference is this large. With an online survey where there is no supervision, it is possible that participants get distracted or take breaks during the experiment. Therefore, averages could be lower in a lab-setting where participants are only focused on the task. Other options would be that we just recruited slower participants, or that the Qualtrics survey design makes it more difficult to answer quickly.

5.2 Response quality

Output quality for any annotation task depends on three factors: clarity of the task, ambiguity of the items, and the reliability of the annotators (Aroyo and Welty, 2014). Here we focus on the latter. Not all participants are equally reliable in their responses. If we assume that there is one true ranking or quality score for each dimension,[10] one reasonable way to approximate this true value is to take the average of all responses. We used this intuition to measure the reliability of each participant's scores by comparing their scores to the average scores of all other participants for each item. Figure 1 shows the results for the different metrics.

We observe that there is a fair (0.33) to moderate (0.64) correlation between participants' reliability scores for the relevance and coherence scales. This means that participants who agreed more with other participants on one dimension, also tended to agree more with participants on the other dimension.

We also observe that for each metric, there is a nonzero amount of participants who obtained a Spearman correlation of zero or less with the other participants. We did not exclude any participants from our analysis, to stay true to Santhanam and Shaikh's original report, but depending on the context, one may want to exclude participants who fall below a certain threshold, to obtain a more re-

[10] This may in fact depend on the ambiguity of the item or the perspective of the annotator (Basile et al., 2021), but for this task we believe that we can make this simplifying assumption.

Original result	Replicated?
Magnitude estimation with anchors shows more reliable ratings than Likert scale ratings	No
Magnitude estimation with anchors shows more reliable ratings than Best-Worst ranking	Yes
Consistency and agreement are higher for raters who took less than average time (Likert, BME, BWS)	Yes
Consistency and agreement are higher for raters who took more than average time (RME)	No
Raters without prior experience in evaluating dialogue system output reach greater consistency and agreement than those with experience	Yes
Raters without prior experience with conversational agents reach greater consistency and agreement than those with experience	Yes
The automatic metrics for readability and coherence show low correlation to human judgement ratings	Yes
There is a high correlation between the human ratings for RME and BME	No

Table 11: Results evaluated for replicability in this paper.

liable estimate of output quality (again assuming that there is a single, 'True' quality score that we aim to estimate).

Finally, the distribution of the participant reliability scores seems to differ between metrics. For example, while the participants' reliability scores for the Likert scale seems to cluster together in the top right corner, the RR scores seem to be spread out more.

5.3 Automatic metrics

Another issue that we struggled with was the reproduction of the automatic metrics. While we followed the original paper's descriptions, the calculation of these automatic metrics was not completely clear and resulted in large differences between results. As we had some values from the original study (in their BME-survey), we could compare our metrics to theirs, but we never figured out how to consistently extract the same results. Next to the calculation of the automatic metrics themselves, we were also unsure how the rankings were derived from these metrics. This was not explicitly mentioned in the paper or the supplementary material. Finally, we discovered that they seemed divided into a 25/25/25/25 split. For future work we would suggest to use the code of the original paper for the reproduction of the automatic metrics.

5.4 Standardisation of surveys

To upload the surveys in our survey platform, we had to redesign and retype all four surveys from the supplied PDF files. This task took about four hours per survey. Such a retyping task is a barrier to perform a reproduction, and increases the risk of introducing typos into the surveys. Therefore, we recommend researchers to not only share the PDF files of their original survey, but also other available formats (in case of Qualtrics, the QSF format), such that the retyping task can be prevented.[11]

5.5 Statistical analyses

The statistical analyses were tedious as, despite the sharing of the data and the RMarkdown files, some transformations had been operated on the raw data (i.e., data conversion from raw scores to what we assumed to be ranked scores for BME and RME measures). We could not replicate these transformations despite multiple attempts to contact the authors. We thus ran our statistical analyses based on our own raw data and found the above-mentioned results.

5.6 Study-specific remarks

Santhanam and Shaikh (2019) show that the same content evaluated by four different types of evaluation tasks lead to four different outcomes. The outcomes within each task have a high correlation (high ICC scores). However, the correlation between the outcomes across the evaluation tasks is low. This is possibly because Likert allows for more degrees of freedom in answering a question. A question contains one utterance and four different replies that have to be rated on a 6-point Likert scale. Such question can be answered in $6^4 = 1296$ different ways. In comparison, the best-worst scaling evaluation task allows only $4! = 24$ different ways to answer the same question. Therefore, one would expect a higher ICC for the outcomes of best-worst scaling than those of the Likert evaluation

[11] As far as we know, different survey platforms (SurveyMonkey, Qualtrics, Google Forms, Alchemer) do not have a standard survey file format implemented yet, so some amount of conversion may still be necessary.

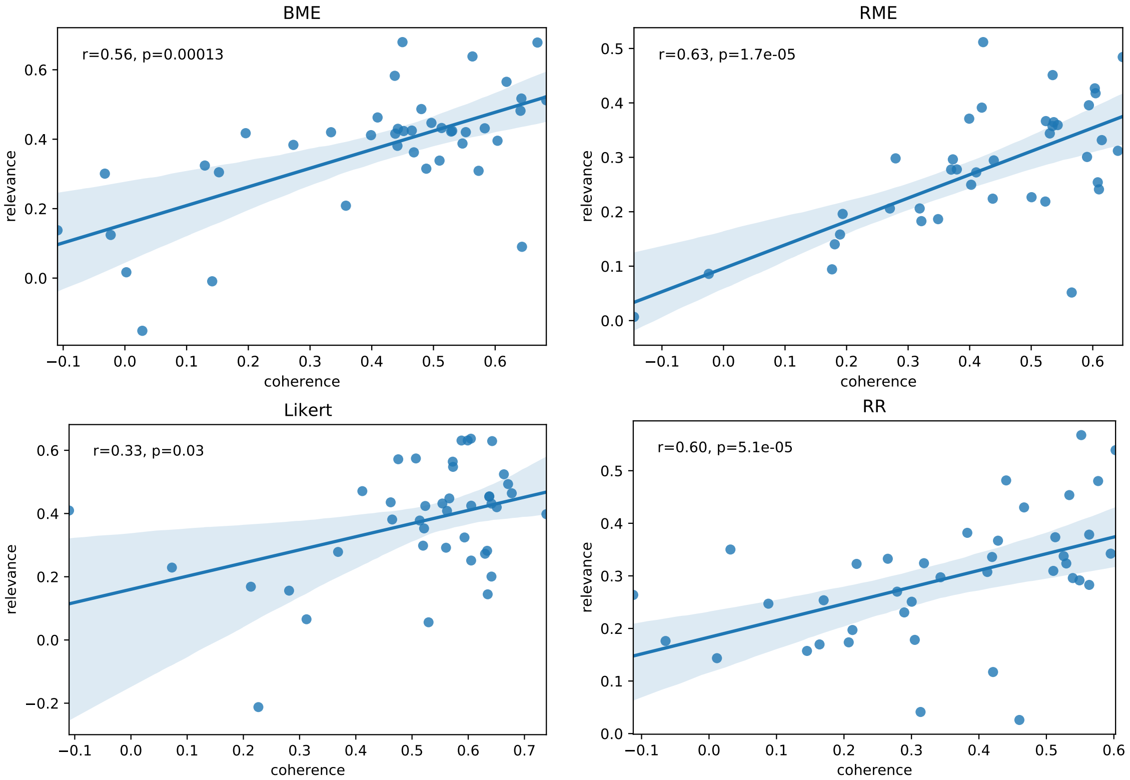


Figure 1: Scatterplot for the correlation between each participant's scores and the average of the other participants' scores. Each point represents one participant, and the axes correspond to the quality dimensions. In other words: these plots show a second-order correlation, measuring whether the reliability of participants (measured as correlations between each participant's scores and the average of other participants' scores) correlates between the two quality dimensions.

task. Furthermore, if we assume that some questions have only low quality replies, then a participant can express that within the Likert evaluation task, but in the best worst scaling task, the participant has to choose a best reply (even if such reply does not exist). The RME and BME evaluation tasks allow an average score. However, the Likert evaluation task is on a 6-point scale, so the participant is forced to evaluate each reply as slightly bad or slightly good. This could influence the correlations between the outcomes of the Likert evalation task on the one hand, and the RME and BME evalation tasks, on the other hand.

6 Conclusion

In this paper, we have tried to reproduce the work of Santhanam and Shaikh (2019). Our results generally replicate the findings of Santhanam and Shaikh (2019) and seem to follow similar trends. As discussed in Section 5, we did run into some difficulties throughout the reproduction process. We hope that our observations are instructive for future researchers in making their work fully reproducible.

Acknowledgements

Funding for paying the participants is supported by the NWO-funded project 628.011.030 Data2Person. Anouck Braggaar's contribution was supported by the NWO Smooth Operators project (KIVI.2019.009).

This research was approved by the Research Ethics and Data Management Committee of the Tilburg School of Humanities and Digital sciences (Tilburg University). Reference number: REDC 2019.40abc (amendment April 5, 2022).

References

Lora Aroyo and Chris Welty. 2014. The three sides of crowdtruth. *Human Computation*, 1(1).

Ellen Gurman Bard, Dan Robertson, and Antonella Sorace. 1996. Magnitude estimation of linguistic acceptability. *Language*, 72(1):32–68.

John J Bartko. 1966. The intraclass correlation coefficient as a measure of reliability. *Psychological reports*, 19(1):3–11.

Valerio Basile, Federico Cabitza, Andrea Campagner, and Michael Fell. 2021. Toward a perspectivist turn in ground truthing for predictive computing. In *Proceedings of the the 18th conference of the Italian Chapter of AIS (Association for Information Systems)*. Available through http://www.itais.org/itais2021-proceedings/pdf/21.pdf or https://arxiv.org/abs/2109.04270.

Anya Belz, Shubham Agarwal, Anastasia Shimorina, and Ehud Reiter. 2020. ReproGen: Proposal for a shared task on reproducibility of human evaluations in NLG. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 232–236, Dublin, Ireland. Association for Computational Linguistics.

Nouha Dziri, Ehsan Kamalloo, Kory Mathewson, and Osmar Zaiane. 2019a. Augmenting neural response generation with context-aware topical attention. In *Proceedings of the First Workshop on NLP for Conversational AI*, pages 18–31, Florence, Italy. Association for Computational Linguistics.

Nouha Dziri, Ehsan Kamalloo, Kory Mathewson, and Osmar Zaiane. 2019b. Evaluating coherence in dialogue systems using entailment. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3806–3812, Minneapolis, Minnesota. Association for Computational Linguistics.

Matthias Gamer, Jim Lemon, and Ian Fellows Puspendra Singh <puspendra.pusp22@gmail.com>. 2019. *irr: Various Coefficients of Interrater Reliability and Agreement*. R package version 0.84.1.

Terry K. Koo and Mae Y. Li. 2016. A guideline of selecting and reporting intraclass correlation coefficients for reliability research. *Journal of Chiropractic Medicine*, 15(2):155–163.

R Core Team. 2017. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Sashank Santhanam and Samira Shaikh. 2019. Towards best experiment design for evaluating dialogue system output. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 88–94, Tokyo, Japan. Association for Computational Linguistics.

DialogSum Challenge: Results of the Dialogue Summarization Shared Task

Yulong Chen [*1], **Naihao Deng** [*2], **Yang Liu, Yue Zhang** [1,3]
[1] School of Engineering, Westlake University
[2] School of Computer Science and Engineering, University of Michigan
[3] Institute of Advanced Technology, Westlake Institute for Advanced Study
yulongchen1010@gmail.com *dnaihao@umich.edu*
inf.yangl@outlook.com *yue.zhang@wias.org.cn*

Abstract

We report the results of *DialogSum Challenge*, the shared task on summarizing real-life scenario dialogues at INLG 2022. Four teams participate in this shared task and three submit their system reports, exploring different methods to improve the performance of dialogue summarization. Although there is a great improvement over the baseline models regarding automatic evaluation metrics, such as ROUGE scores, we find that there is a salient gap between model generated outputs and human annotated summaries by human evaluation from multiple aspects. These findings demonstrate the difficulty of dialogue summarization and suggest that more fine-grained evaluatuion metrics are in need.

1 Introduction

With the power of Pretrained Language Models (PLMs), research on text summarization has made great progresses (Liu and Lapata, 2019). However, previous research focuses on monologue summarization, such as news articles (Paulus et al., 2018; Gehrmann et al., 2018; Liu and Lapata, 2019; Liu et al., 2021), patents (Pilault et al., 2020) and academic papers (Koncel-Kedziorski et al., 2019). However, as an important communicative channel (Bender and Koller, 2020), dialogues receive less attention from the community.

To this end, we propose *DialogSum Challenge* to encourage researchers to investigate different solutuons for real-life summarization (Chen et al., 2021b). Different from previous dialogue summarization tasks (Carletta et al., 2005; Gliwa et al., 2019), *DialogSum Challenge* focuses on diverse real-life scenarios such as schooling, work, medication, shopping, leisure, travel with large scale data.

The challenges of *DialogSum* can be stated from three perspectives. First, *DialogSum* include a variety of topics, requiring models to process text with different real-life scenarios. Second, compared with well-structured monologues, dialogues have unique discourse structures and language styles (Grosz et al., 1995b). The structures and use of languages differ from monologues, for instance, the key information spans over the context (Grosz et al., 1995a), which makes dialogues more difficult to encode. Third, compared with monologue summarization, dialogue summaries are written from a different perspective, usually including speakers' intents and actions (Chen et al., 2021a). Therefore, dialogue summarization is abstractive in nature and requires a high level understanding beyond text semantics (Chopra et al., 2016; Khandelwal et al., 2019). Figure 1 shows an example in *DialogSum*. Apart from the research challenges, *DialogSum Challenge* incentivizes summarization systems that can help end users. Summarizing daily spoken dialogues can help archive the important information in business and personal communication. This presumably lifts the burden of manually taking notes, liberating human beings from the tedious work.

Responding to our calls, four teams participate in the shared task, and three of them submit their system reports. The submitted systems typically employ PLMs, such as BART (Lewis et al., 2020) and PEGASUS (Zhang et al., 2020a). In addition, they explore diverse methods to improve the performance, including integrating additional features, modifying the decoding process for better summary generation, multi-task tuning the model with auxiliary tasks and using data from other sources.

To evaluate the performance, we construct a hidden test set that contains 100 manually annotated samples, and evaluate models on hidden and public test sets using both automatic and manual evaluations. For automatic evaluation, we use ROUGE scores and BERTSCORE. For manual

[*] Equal Contribution.

Proceedings of the 15th International Conference on Natural Language Generation: Generation Challenges, pages 94 - 103
July 17-22, 2022 ©2022 Association for Computational Linguistics

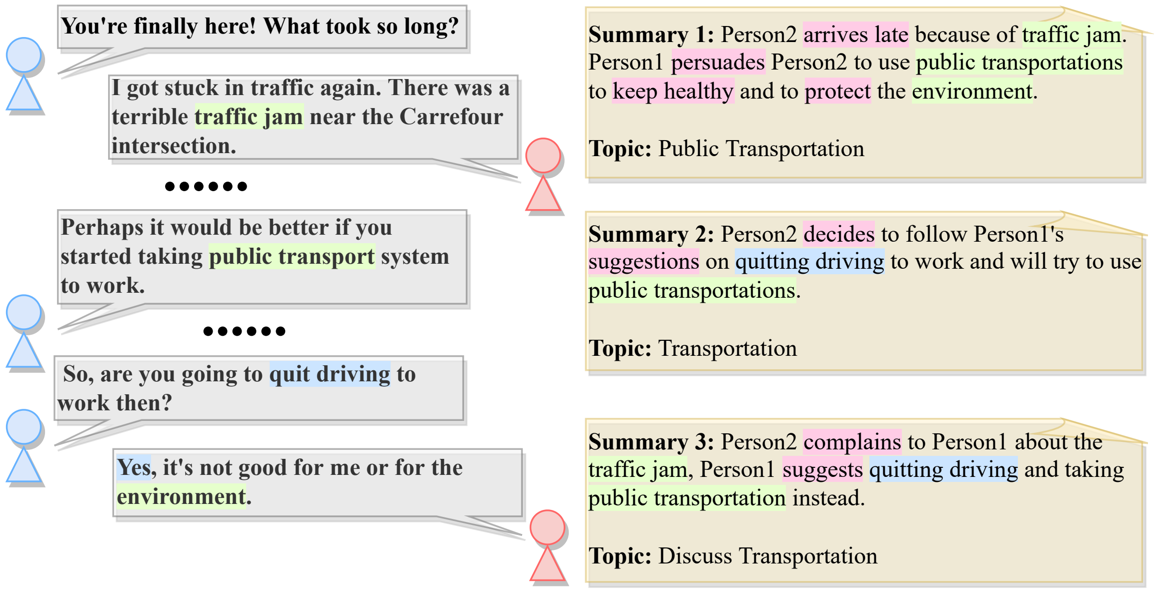

You're finally here! What took so long?
I got stuck in traffic again. There was a terrible traffic jam near the Carrefour intersection.
••••••
Perhaps it would be better if you started taking public transport system to work.
••••••
So, are you going to quit driving to work then?
Yes, it's not good for me or for the environment.
Summary 1: Person2 arrives late because of traffic jam. Person1 persuades Person2 to use public transportations to keep healthy and to protect the environment.
Topic: Public Transportation
Summary 2: Person2 decides to follow Person1's suggestions on quitting driving to work and will try to use public transportations.
Topic: Transportation
Summary 3: Person2 complains to Person1 about the traffic jam, Person1 suggests quitting driving and taking public transportation instead.
Topic: Discuss Transportation

Figure 1: An example in the public test set of the *DialogSum* dataset. Tokens highlighted in red represent tokens that only appear in the summary but not in the dialogue text, requiring the model to summarize with a high level semantic understanding. Tokens hightlighted in blue represent the information that spans across turns. Tokens highlighted in green show the corresponding information in the dialogue text and the summary. Note that such information scatters in various places in the dialogue.

evaluation, we follow *DialogSum Challenge* (Chen et al., 2021b) and evaluate model outputs from multiple aspects. Results show that tuning models on CNN/Daily News corpus (Hermann et al., 2015) or AMI dataset (Carletta et al., 2005), and incorporating topics in summary generation process can improve the model performance. However, there are still rooms for models to improve the metric scores, as well as the quality of the generated summaries in terms of identifying the interlocutors' intents, capturing the discourse relation, etc. Besides, we observe the mismatch between BERTSCORE and human scores, which aligns with the findings by Hanna and Bojar (2021).

Full details of the shared task description and logistics, as well as the dataset can be found at https://cylnlp.github.io/dialogsum-challenge/

2 Task

Given a piece of dialogue text as input, the task is to ask a model to generate a summary that conveys the key information of the dialogue. The output summary should be concise, coherent, consistent and be written from a listener's perspective.

3 Data

Data Sources We use the train, dev and public test data from *DialogSum*. Additionally, we collect 100 summaries as hidden test set from the same website where *DialogSum* crawls the data [1]. We follow the exact same procedure as the annotation for the original *DialogSum* dataset (Chen et al., 2021a). We remove the non-English characters, correct typos and grammatical errors, and filter out duplicated dialogues based on text similarity . The annotators are instructed to write the summaries for each dialogue by: (1) conveying the salient information in the dialogue and; (2) keeping the summary short and; (3) writing from the observer perspective and in formal language. Additionally, we ask annotators to keep tense consistency, preserve important discourse relations, explicitly describe emotion and speaker's intents. Also, annotators are instructed to provide a short topic for each dialogue. Table 3 shows the statistics for the data in *DialogSum Challenge*.

[1] http://tingroom.com

[1] We compute the ROUGE scores between two dialogues and filter out dialogues that have more than 80% ROUGE-1 scores.

	Public Test Set				Hidden Test Set			
Model	R1	R2	RL	BERTScore	R1	R2	RL	BERTScore
Human	**53.35**	**26.72**	**50.84**	92.63	-	-	-	-
GoodBai	**47.61**	**21.66**	45.48	**92.72**	49.66	**26.03**	**48.55**	91.69
UoT	47.29	21.65	**45.92**	92.26	**49.75**	25.15	46.50	**91.76**
IITP-CUNI	47.26	21.18	45.17	92.70	45.89	21.88	43.16	91.13

Table 1: Scores by automatic metrics for each submission and human results. We embolden the top scores among models, as well as the human score if it is the highest among all the scores.

	Public Test Set				Hidden Test Set			
Model	R1	R2	RL	BERTScore	R1	R2	RL	BERTScore
TCS_WITM	47.02	21.20	44.90	90.13	50.32	25.59	47.40	91.81

Table 2: Scores by automatic metrics for the submission from TCS_WITM. The model submitted by TCS_WITM predicts 3 summaries based on the 3 topics in the public test set. We take the highest score among the 3 summaries to calculate the scores.

Train	Dev	$\text{Test}_{\text{public}}$	$\text{Test}_{\text{hidden}}$	Total
12,460	500	500	100	13,560

Table 3: Number of dialogues in each split for *DialogSum Challenge*.

4 Evaluation Set-Ups

Automatic Evaluation We adopt two metrics, Rouge (Lin, 2004) and BERTScore (Zhang et al., 2020b) for automatic evaluation. We use RoBERTa (Liu et al., 2019) large as the backbone to calculate BERTScore.

Manual Evaluation Furthermore, we conduct manual evaluation from various aspects, including standard summarization metrics Kryscinski et al. (2019, 2020), coreference information, intent identification, discourse relation following Chen et al. (2021a), as well as objectiveness (whether the summary is insusceptible to subjectivity such as subjective assumptions in the dialogues). Besides, annotators give an overview score for the predicted summary.

5 Submissions

5.1 IITP-CUNI

The model submitted by Indian Institute of Technology Patna and Charles University employs a multi-task learning set-up to improve model performance. In their experiments, they explore several auxiliary tasks including extractive summarization to classify whether a given sentence belongs to the summary or not, novelty detection (Ghosal et al., 2022) of whether the given summaries correspond to the same dialogue, as well as a masked language modeling (Devlin et al., 2019) task to recover summaries. They find that the BART (Lewis et al., 2020) large model tuned with the auxiliary task of the extractive summarization task with data from AMI (Carletta et al., 2005) corpus performs the best.

5.2 UoT

The participants from the University of Tübingen use the pre-trained BART model which is further tuned on CNN/Daily News corpus (Hermann et al., 2015). Besides, they penalize generating longer summaries in the decoding part of the model, and post-process the summaries to resolve generation errors (e.g. replacing speakers' names who do not appear in the dialogue with #Person_1# or #Person_2#, and fixing duplicated labels such as #Person_1#Person_1# to #Person_1#).

They also explore methods such as intermediate task transfer learning with training on commonsense reasoning task or other summarization task first and then tune the model on *DialogSum*, transforming dialogue structures to news with structures similar to what BART model is trained on, as well as data augmentation by using data from SAMSum (Gliwa et al., 2019) dataset. However, they do not find any improvement using these techniques.

5.3 TCS_WITM

The model from TCS research adopts the pre-trained PEGASUS (Zhang et al., 2020a) large model which is further fine-tuned on CNN/Daily

Dialogue:
#Person1#: Excuse me, could you tell me how to get to the school clinic? I've lost my way.
#Person2#: Yes. Go straight ahead till you come to the traffic lights, turn left there and it's the first turning on the right.
#Person1#: Straight ahead to the traffic lights, left and then right.
#Person2#: That's it. It'll take you about five minutes.
#Person1#: Thank you very much.

Gold:
Summary: #Person1# is lost on the way to the school clinic. #Person2# shows #Person1# the correct direction.

IIPT-CUNI:
Prediction: #Person2# shows #Person1# the way to the school clinic.
UoT:
Prediction: #Person2# shows #Person1# the way to the school clinic.
TCS_WITM:
Prediction: #Person2# tells #Person1# how to get to the school clinic.
GoodBai:
Prediction: #Person2# tells #Person1# how to get to the school clinic.

Figure 2: An example where all the predicted summaries miss the the context #Person1# is lost, while all of the gold summaries contain this context. However, all the predicted summaries successfully capture the event of #Person2# shows #Person1# the direction. [2]

News. They also incorporate the topics provided in *DialogSum* dataset and feed the topic together with the dialogue text to their model. The extra information from the topics boost up the model performance compared to the baseline performance of simply feeding dialogues to the model.

6 Results

Table 1 and Table 2 show the results of human agreement and submissions from participants in *DialogSum Challenge* by the automatic metrics of ROUGE-1 (R1), ROUGE-2 (R2), ROUGE-L (RL), and BERTSCORE. We do not compare TCS_WITM with other models because TCS_WITM uses gold topic information.

In general, scores of the submissions are higher than the baseline models in the original *DialogSum* paper, demonstrating the effort from the participants. Both submissions from UoT and IITP-CUNI tune their models on other dataset then on *DialogSum*. In particular, UoT tunes their model on CNN/Daily News corpus, while IITP-CUNI tunes their model on AMI dataset. The reported results show that the model by UoT outperforms the model by IITP-CUNI. This might be attributed to the different in training size, where AMI (Carletta et al., 2005) has 137 meetings, while CNN/Daily News corpus (Hermann et al., 2015) has 312,000 articles. Thus, the model tuned on CNN/Daily News corpus might have better generalization ability. The model by TCS_WITM which adopts such a method achieves 50.32 in ROUGE-1 score for the hidden test case, showing that generating the summary with the given topic can also help the performance.

However, even the best-performed model underperforms humans by a margin larger than 5.0 in terms of all the ROUGE scores. This indicates that *DialogSum* is challenging and there is still a large room for future improvement.

Although existing works on summarization adopt BERTSCORE (Gabriel et al., 2021), we observe that the BERTSCORE deviates from the human scores. For instance, though GoodBai achieves the best BERTSCORE on the public test set, it is TCS_WITM with a lower BERTSCORE that achieves the best human scores (Overview score in Table 4). Same phenomenon also exists for the hidden test set. This observation aligns with the finding from Hanna and Bojar (2021) that BERTSCORE performance still deviates from human. Thus, the BERTSCORE is still not perfect to serve as the ultimate metric for summarization tasks, and our community might come up with a better automatic metric that aligns with human scores.

7 Human Analysis

We randomly sample 50 examples for the public test set and 20 examples for the hidden test set to conduct manual analysis. As discussed in our proposal, we include the metrics of coreference information, discourse relation, objective description, intent identification, standard summarization metrics and overview scores. We annotate -1, 0, 1 for the metrics of coreference information, discourse relation, objective description, intent identification where 1 means all correct, 0 means partially correct and -1 means all incorrect. We annotate from

	Public Test Set						Hidden Test Set					
Model	CoRef	Dis	Obj	Intent	Summ	Over	CoRef	Dis	Obj	Intent	Summ	Over
Perfect Score	1.00	1.00	1.00	1.00	5.00	5.00	1.00	1.00	1.00	1.00	5.00	5.00
GoodBai	0.96	0.86	**1.00**	0.72	4.12	3.96	**0.90**	**0.90**	**1.00**	0.70	**4.20**	**4.15**
UoT	**0.98**	**0.92**	**1.00**	**0.80**	**4.18**	**4.08**	0.75	0.75	**1.00**	**0.80**	4.00	3.70
IITP-CUNI	0.88	0.66	0.96	0.76	3.94	3.64	0.75	0.85	**1.00**	0.70	3.80	3.70

Table 4: Prediction results by one of the annotators of the *DialogSum* dataset. “CoRef”, “Dis”, “Obj”, “Intent”, “Summ”, “Over” indicates coreference information, discourse relation, objective description, intent identification, standard summarization metrics and overall scores, respectively. We embolden the best scores for each column.

	Public Test Set						Hidden Test Set					
Model	CoRef	Dis	Obj	Intent	Summ	Over	CoRef	Dis	Obj	Intent	Summ	Over
TCS_WITM	0.88	0.90	1.00	0.82	4.20	4.10	0.90	0.80	0.84	0.70	3.95	3.80

Table 5: Prediction results by one of the annotators of the *DialogSum* dataset for TCS_WITM.

1 to 5 for the standard summarization metrics and overview scores. The higher, the better.

Coreference Information Whether the summary aligns interlocutors and their conversation actions or contents.

Discourse Relation Whether the summary captures important relations between main events, identifying discourse relations and using appropriate phrases to express such relations.

Objective Description Whether the summary employs objective languages to describe dialogues.

Intent Identification Whether the summary captures the interlocutors’ intents.

Standard Summarization Metrics (Kryscinski et al., 2019, 2020) Whether the summary is fluent, consistent, relevant and coherent. However, in practice, we find that summaries generated by PLMs are mostly fluent, sometimes better than human annotated summary. And we have already evaluate consistent and coherent with more fine-grained metrics (Coreference Information, Objective Description, etc). Thus, we focus on relevance and judge whether the generated summary is informative and relevant.

Overview Scores Overview score of the summary with the aforementioned metrics.

Table 4 reports the scores from the aforementioned metrics. There is not a universal model which performs the best across all of these metrics, instead, each model excels at certain metrics. Overall, TCS_WITM achieves the best overview score on the public test set, while GoodBai achieves the best overview score on the hidden test set.

Dialogue:

#Person1#: What time is it, Tom?

#Person2#: Just a minute. It’s ten to nine by my watch.

#Person1#: Is it? I had no idea it was so late. I must be off now.

#Person2#: What’s the hurry?

#Person1#: I must catch the nine-thirty train.

#Person2#: You’ve plenty of time yet. The railway station is very close. It won’t take more than twenty minutes to get there.

Gold:

Summary: #Person1# is catching a train. Tom asks #Person1# not to hurry.

IITP-CUNI:

Prediction: #Person1# and Tom are in a hurry to catch the nine-thirty train.

Figure 3: An error example where the model fails to distinguish the intent between #Person1# and Tom (#Person2#). [2]

8 Error Analysis

Table 1, 2, 4, 5 show that the submitted models underperform human beings. Here we analyze some examples where the models make mistakes or fail to capture certain information.

Figure 3 shows an example where the model from IITP-CUNI makes a factual error and fails to reason about who is in the hurry. In order to capture the correct relationship, the model needs to reason that “Tom” is the name of #Person2#, and Tom (#Person2#) is asking #Person1# not to hurry by saying “You’ve plenty of time yet”. However,

[2] We only include one of the gold summaries for demonstration purpose.

the model from IITP-CUNI fails in such reasoning processes. This suggests that reasoning about information across the dialogue discourse is challenging.

Figure 2 shows an example where all of the model predictions deviate from the gold summaries. All of the 3 summaries annotated by human beings include the context of #Person1# being lost on the way. [2] In contrast, none of the model predictions include this context. This is plausible as the majority of the dialogue is dedicated to #Person2# showing #Person1# the correct direction, and the model might only capture such salient information in the dialogue. However, the general pattern when human beings summarize is to lay out the cause (context) of an event before telling the event, which is demonstrated in the gold summaries. Thus, there is still a large room for improvement for the model to generate human-like summaries.

Appendix A gives more examples of models making mistakes in terms of the metrics from § 7.

9 Conclusion

We host *DialogSum Challenge* of summarizing daily dialogue conversation at INLG 2021. Our dataset possesses characteristics distinguished from the existing datasets and poses new challenge to the summarization community. There are 4 teams who submit their models during the challenges. An overview of their methods is provided in this report. We evaluate their predictions by automatic metrics and human analysis. Results show that there are still rooms for models to improve the ROUGE scores, as well as the quality of the generated summaries in terms of identifying interlocutors' intents, capturing the discourse relation, etc. Besides, we observe the mismatch between BERTSCORE and human scores, which aligns with the findings by Hanna and Bojar (2021). Therefore, we advocate to our community to explore automatic metrics that can better align with human scores.

Acknowledgement

Yue Zhang is the corresponding author. We would like to thank Yi Zhang for her help in the human analysis as well as the collection of the hidden test set. This work receives a support from the Tencent AI Lab Rhino- Bird Focused Research Program.

References

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, et al. 2005. The ami meeting corpus: A pre-announcement. In *International workshop on machine learning for multimodal interaction*, pages 28–39. Springer.

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021a. DialogSum: A real-life scenario dialogue summarization dataset. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5062–5074, Online. Association for Computational Linguistics.

Yulong Chen, Yang Liu, and Yue Zhang. 2021b. DialogSum challenge: Summarizing real-life scenario dialogues. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 308–313, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98, San Diego, California. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Saadia Gabriel, Antoine Bosselut, Jeff Da, Ari Holtzman, Jan Buys, Kyle Lo, Asli Celikyilmaz, and Yejin Choi. 2021. Discourse understanding and factual consistency in abstractive summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 435–447, Online. Association for Computational Linguistics.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109, Brussels, Belgium. Association for Computational Linguistics.

Tirthankar Ghosal, Tanik Saikh, Tameesh Biswas, Asif Ekbal, and Pushpak Bhattacharyya. 2022. Novelty detection: A perspective from natural language processing. *Computational Linguistics*, 48(1):77–117.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. 1995a. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2):203–225.

Barbara J Grosz, Aravind K Joshi, and Scott Weinstein. 1995b. Centering: A framework for modelling the local coherence of discourse.

Michael Hanna and Ondřej Bojar. 2021. A fine-grained analysis of BERTScore. In *Proceedings of the Sixth Conference on Machine Translation*, pages 507–517, Online. Association for Computational Linguistics.

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 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1693–1701.

Urvashi Khandelwal, Kevin Clark, Dan Jurafsky, and Lukasz Kaiser. 2019. Sample efficient text summarization using a single pre-trained transformer. *ArXiv preprint*, abs/1905.08836.

Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. 2019. Text Generation from Knowledge Graphs with Graph Transformers. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2284–2293, Minneapolis, Minnesota. Association for Computational Linguistics.

Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. 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 540–551, Hong Kong, China. Association for Computational Linguistics.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

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.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. 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 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Yang Liu, Sheng Shen, and Mirella Lapata. 2021. Noisy self-knowledge distillation for text summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 692–703, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv preprint*, abs/1907.11692.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Jonathan Pilault, Raymond Li, Sandeep Subramanian, and Chris Pal. 2020. On extractive and abstractive neural document summarization with transformer language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9308–9319, Online. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020a. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Dialogue:

#Person1#: How long does it take to get to downtown from here?

#Person2#: It is 15 minutes ’ drive.

#Person1#: What companies do we have in our neighborhood?

#Person2#: Mitsubishi, HP, IBM and many other famous corporations.

#Person1#: Does the 7th floor belong to our company too?

#Person2#: Yes, our company has two floors, the 6th and 7th floor in this building.

#Person1#: It ’ s such a large firm. Do we have our own staff restaurant?

#Person2#: Yes, at the end of the hall.

Gold:

Summary: #Person2# tells #Person1# information about their company and its surroundings.

UoT:

Prediction: #Person2# asks #Person2# about the distance to downtown and the companies in their neighborhood.

Figure 4: An error example for predicting the wrong coreference information.

A More Error Examples

Figure 4, 5, 6, 7, 8 show examples of predicting the wrong coreference information, wrong discourse relation, summarizing with description that is not objective, wrong intent of the interlocutors and with a low standard summarization score as well as a low overview score, respectively.

Dialogue:
#Person1#: Hi! How are things going with you?
#Person2#: I am doing well. What's up with you?
#Person1#: Believe it or not, the company I worked for closed down, so I'm out of a job.
#Person2#: The same with me. Have you given much thought to what you want to do next?
#Person1#: I am not being all that particular right now because I just need to keep a roof over my head.
#Person2#: How about interviews? Have you been on any of those yet?
#Person1#: I wish that I could get the opportunity to be interviewed.
#Person2#: How about the electrician program that they have listed over there?
#Person1#: I read about that, and the position sounded great!
#Person2#: Let's go see how we can apply for those positions.

Gold:
Summary: #Person1# and #Person2# are both unemployed. #Person2# suggests applying for the electrician program and #Person1# agrees.

IITP-CUNI:
Prediction: #Person1# tells #Person2# #Person1# is out of a job because #Person1# needs to keep a roof over #Person1#'s head. #Person2# suggests applying for electrician positions.

Figure 5: An error example for predicting the wrong discourse relation.

Dialogue:
#Person1#: John dates her seven times a week.
#Person2#: Really? That's a straws in the wind.
#Person1#: I think so. Maybe he's fallen for her.
#Person2#: Yeah. They suit each other. A perfect match between a man and a girl.
#Person1#: Right.

Gold:
Summary: #Person1# and #Person2# think that John and the girl are a perfect match.

IITP-CUNI:
Prediction: #Person1# and #Person2# talk about John and the girl he loves.

Figure 6: An error example for summarization with description that is not objective.

Dialogue:
#Person1#: Can I help you?
#Person2#: I need some stamps for this letter.
#Person1#: What kind of stamps do you want?
#Person2#: How much do I need for this letter?
#Person1#: I must weigh it first. Err... It's five grams over weigh, Do you want to send it as an ordinary or registered letter?
#Person2#: I want it registered. How much is it then?
#Person1#: Registration plus overnight... err... seven dollars in all.
#Person2#: Here's a 10 - dollar bill.
#Person1#: Now, your receipt, and the change.
#Person2#: Thanks. Good-bye.

Gold:
Summary: #Person2# wants to send a letter. #Person1# says it's five grams overweight plus overnight so seven dollars in all.

GoodBai:
Prediction: #Person1# helps #Person2# buy some stamps for a registered letter.

Figure 7: An error example for predicting the wrong intent of the interlocutors.

Gold:

Summary: #Person1# and #Person2# are talking about the terrorist attacks on 9-11, which was nightmarish for #Person1#'s family.

UoT:

Prediction: #Person1# and #Person2# talk about where they were for the terrorist attacks on 9-11. #Person1# was at home with #Person1#'s parents in New York City and #Person1# didn't see the crash itself but

Figure 8: An example with a low standard summarization score and a low overview score. The prediction generated by the model seems unfinished.

TCS_WITM_2022 @ DialogSum : Topic oriented Summarization using Transformer based Encoder Decoder Model

Vipul Chauhan, Prasenjeet Roy, Lipika Dey and **Tushar Goel**
TCS Research
New Delhi India
(chauhan.vipul, r.prasenjeet2, lipika.dey, t.goel)@tcs.com

Abstract

In this paper, we present our approach to the DialogSum challenge, which was proposed as a shared task aimed to summarize dialogues from real-life scenarios. The challenge was to design a system that can generate fluent and salient summaries of a multi-turn dialogue text. Dialogue summarization has many commercial applications as it can be used to summarize conversations between customers and service agents, meeting notes, conference proceedings etc. Appropriate dialogue summarization can enhance the experience of conversing with chatbots or personal digital assistants. We have proposed a topic-based abstractive summarization method, which is generated by fine-tuning PEGASUS[1], which is the state of the art abstractive summary generation model.We have compared different types of fine-tuning approaches that can lead to different types of summaries. We found that since conversations usually veer around a topic, using topics along with the diaaloagues, helps to generate more human-like summaries. The topics in this case resemble user perspective, around which summaries are usually sought. The generated summary has been evaluated with ground truth summaries provided by the challenge owners. We use the py-rouge score and BERT-Score metrics to compare the results.

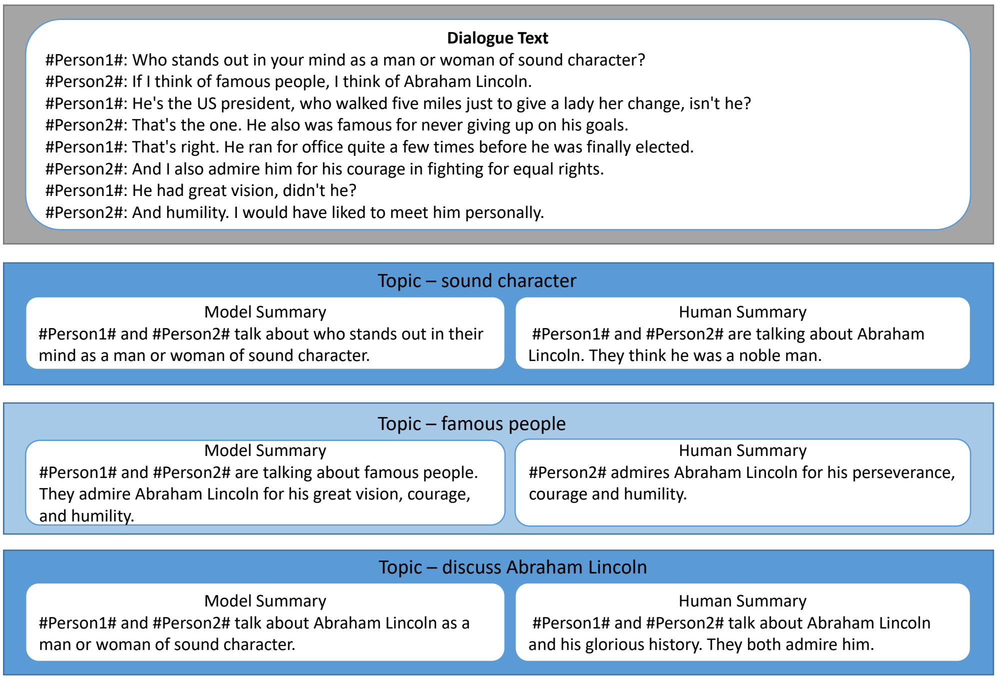
1 Introduction

Automatic text summarization is an important task in natural language processing, and it has been studied for decades. While extractive summarization focused on picking up the most important sentences from the text and create a summary, abstractive summarization generates new concise sentences with the important concepts. The task of abstractive summarization thus has two sub-tasks - identifying the important concepts within content and generating new sentences that are grammatically correct and can cover all important concepts sufficiently without repetition or redundancy. Both the summarization techniques have received attention from researchers of natural language processing. Some of the most cited works in the area of extractive summarization are (Erkan and Radev, 2004; Rai et al., 2021), and for abstractive summarization one may refer to (Lewis et al., 2019; Raffel et al., 2019; Zhang et al., 2020).

However, most of the above-mentioned works has focused on single-speaker documents such as news (See et al., 2017; Nallapati et al., 2016), scientific publications (Nikolov et al., 2018) etc. The documents considered also were short and assumed to contain a limited number of concepts around which summaries were to be generated. On demand summarization based on user queries, summarization of multi-section large reports are some of the problems that are currently being explored in the above area. Content generated through interaction between two or more speakers is known as a dialogue. Dialogues are important forms of communication, which contain lot of information about ideas exchanged and nature of the participants. Dialogue summarization aims to condense a piece of content generated by multiple participants into a short passage. Dialogues are difficult to summarize since the underlying data contains diverse interactive patterns between speakers as well as inherent topic drifts (Feng et al., 2020). Human summarization sometimes focuses only on the content. sometimes gives more attention to the nature of interaction, while at others may be considering both. For example, while summarizing an argument it may be needed to capture the key points made by both the speakers separately and highlight it in the summary. For other scenarios like a customer communication it may be more important to detect dissents, agreements and the topics around which they occur. The difficulty of dialogue summarization stems from the heterogeneity of the

[1] https://huggingface.co/google/pegasus-large

Proceedings of the 15th International Conference on Natural Language Generation: Generation Challenges, pages 104 - 109
July 17-22, 2022 ©2022 Association for Computational Linguistics



Dialogue Text

#Person1#: Who stands out in your mind as a man or woman of sound character?
#Person2#: If I think of famous people, I think of Abraham Lincoln.
#Person1#: He's the US president, who walked five miles just to give a lady her change, isn't he?
#Person2#: That's the one. He also was famous for never giving up on his goals.
#Person1#: That's right. He ran for office quite a few times before he was finally elected.
#Person2#: And I also admire him for his courage in fighting for equal rights.
#Person1#: He had great vision, didn't he?
#Person2#: And humility. I would have liked to meet him personally.

Topic – sound character

Model Summary
#Person1# and #Person2# talk about who stands out in their mind as a man or woman of sound character.

Human Summary
#Person1# and #Person2# are talking about Abraham Lincoln. They think he was a noble man.

Topic – famous people

Model Summary
#Person1# and #Person2# are talking about famous people. They admire Abraham Lincoln for his great vision, courage, and humility.

Human Summary
#Person2# admires Abraham Lincoln for his perseverance, courage and humility.

Topic – discuss Abraham Lincoln

Model Summary
#Person1# and #Person2# talk about Abraham Lincoln as a man or woman of sound character.

Human Summary
#Person1# and #Person2# talk about Abraham Lincoln and his glorious history. They both admire him.

Figure 1: An example of topic focused summarization

underlying content. Dialogue summarization is an important problem that can be further classified into various sub-areas depending on the nature of input considered such as speech summarization, meeting summarization, chat summarization, email thread summarization and so on. A detailed survey on abstractive summarization is presented in (Zhong et al., 2021; Feng et al., 2021).

In recent times, masked language models using transformers that are basically multi-headed attention-based encoder-decoder models, have created a remarkable impact in the area of of text generation (Choi et al., 2019), and consequently tasks like abstractive summarization which are heavily dependent on it. PEGASUS is a transformer based model trained on large **C4** corpora introduced in (Raffel et al., 2020) containing 350M Web-pages and **HugeNews** dataset which consists of 1.5B articles from news-like website(2013-2019). Its pre-training objectives were set as Gap Sentence Generation (GSG), which was more aligned to the downstream task of summarization, and the model thereby is found to achieve much better and faster performance for abstractive summarization tasks, after some fine-tuning. In GSG, top m principal sentences, which are found to be most similar to the other sentences in the document according to ROUGE-F1 score, are initially masked while feeding the document to the model. These sentences are concatenated into a psuedo-summary, and the model is trained to generate these using a sequence to sequence generation task. This pre-training objective has pushed forward state of the art model on 12 diverse summarization datasets. It is found to perform exceptionally well on summarization tasks even when very few training samples are available for fine-tuning (Zhang et al., 2020)

In this challenge, a dialogue is found to contain information related to multiple topics. For example, “#Person2# arrives late because of traffic Jam. #Person1# suggests #Person2# quitting driving and taking public transport” contains two topics- ‘reason for being late’ and ‘benefit of public transport.’ Since the pre-trained PEGASUS model includes the salient information from the input text irrespective of user perspective, it can’t generate a topic-driven or user-perspective driven summary. The novelty of the proposed approach lies in proposing a new fine-tuning task in which a topic is passed as an input along with the dialogue text, to reformulate the task of dialogue summarization. The incorporation of the topic along with the input and a target summary during training allows for additional training of the model to generate topic-focused summaries. This enhances the quality of summary generated by PEGASUS in two ways - it learns to focus on different text segments that are centered around a given topic, and then use those portions to pick up the principal sentences. In the current context, the model learnt to focus on text segments that contained the parts of the conversation that were more relevant to the user-perspectives and thereby generated a topical summary. The significance of the proposed model is that the same text can be summarized differently based on the topics given, by focusing on different portions of the text. Fig 1 shows a sample dialogue from the test set, human-generated summaries around different topics and the outputs generated by our system for each of the given topics.

This paper is organized as follows: Section 2 gives the details of the shared task and the dataset provided. Section 3 provides a detailed description of the proposed methodology. Section 4 gives the details of baseline models and training parameters. Results are discussed in the Section 5 which is followed by the conclusion in Section 6.

2 Shared Task Details and Dataset

The DialogSum Challenge (Chen et al., 2021b) is focused on summarizing real-life dialogues. The task is to generate a fluent, concise, and coherent summary of the multi-turn dialogue text. The DialogSum dataset (Chen et al., 2021a) consists of $13,460$ dialogue conversations collected from three datasets viz Dailydialog (Li et al., 2017), DREAM (Sun et al., 2019), MuTual (Cui et al., 2020), and a few dialogues from English-speaking practice websites. This aggregated dataset [2] consists of a training set of 12460 dialogues, development set of 500 dialogues, and test set of 500 dialogues, where each dialogue was of average length 120 words. Both the training set, and the development set included a topic which usually spans over one to three words, and a human summary whose average length is 19 words. Each dialogue in the test set however had three topics and corresponding topic-focused human-generated summaries, which could be used for evaluating the model. A hidden test set with 100 dialogues and one topic each was provided as the actual challenge task.

3 The Proposed Method

For a given dialogue text $d = d_1, d_2, ...d_n$ of n words and the topic t of the conversation where $t = t_1, t_2, ..., t_k$ consists of k words, the task is to generate a dialogue summary $y = y_1, y_2, ...y_m$ containing m words. The end goal is to find the summary of a dialogue y^* that maximizes the probability $p(y|t, d)$. In order to achieve this objective, we adopt the state of the art pre-trained PEGASUS model [3], which was further fine-tuned on the downstream summarization task using the CNN/Daily News dataset (Nallapati et al., 2016). The target fine-tuning task was designed to generate the News highlights from the text.

The proposed framework used by us is shown in Figure 2 (b), while the standard one is shown in (a). We have fed the topic along with the dialogue text, where the two are separated by a special character. The target summary was a human input that came as a part of the data-set. It was observed that the topics represented human conceptualization of the content succinctly without borrowing key-words from the dialogue itself, unless necessary.

The motivation to use the topic to fine-tune the model was derived from the fact that the test dataset came with three different human summaries, formed around different topics for each dialogue. One such example dialogue with three target summaries are shown in figure1. This clearly indicated that the same conversation could be viewed from different perspectives and hence summarized differently. Though humans inherently tend to map any piece of text to topics, a human summarization tends to occur around these topics. In this dataset, the human annotation contained both the topic and the summary, which we could use to train our model in order to obtain better summaries than default PEGASUS. The idea was that using the topics as input for fine-tuning will be able to generate more topic-oriented summaries, by guiding the model towards sentences that are important for the topic and not by default ROUGE F1 similarity. Since the final hidden dataset also had a topic given, the task could clearly be modeled as one of topic-oriented summarization.

However, not all possible summarization scenarios may come with the topics explicitly mentioned, though the need may still be to do topic-focused summarization. The model in that case may be enhanced to identify the key topics first and then use them for summarization. The present dataset may serve as a good source for training a model to identify topics.

[2] https://github.com/cylnlp/DialogSum

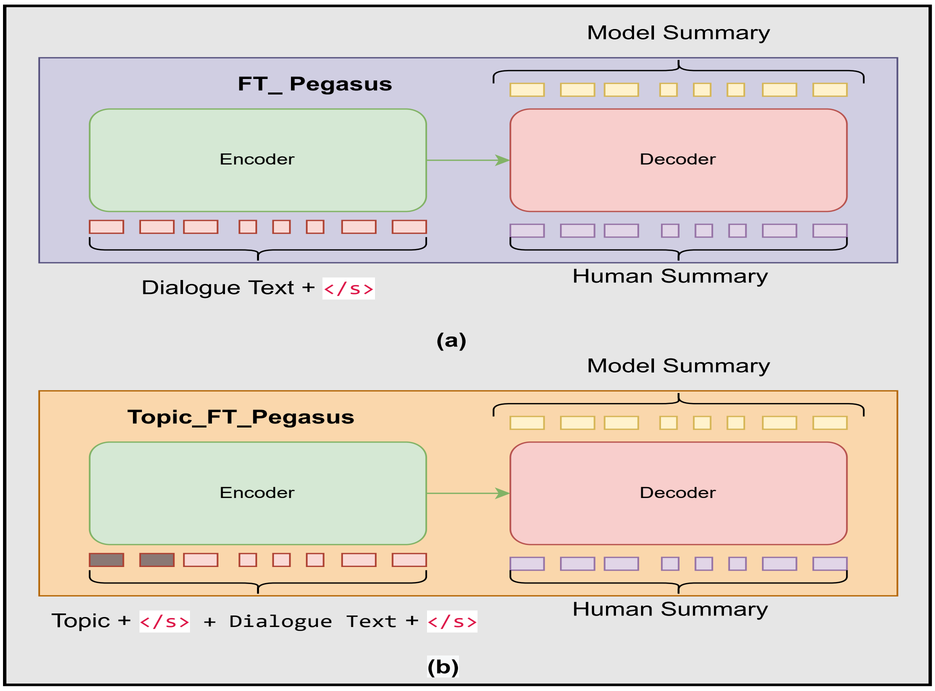


Figure 2: Proposed framework architecture

Model	Average Score				Best Score			
	R1	R2	RL	B-S	R1	R2	RL	B-S
PT_PEGASUS	25.99	6.41	20.97	87.77	37.63	9.63	26.48	88.15
FT_PEGASUS	43.36	18.36	36.23	92.19	51.59	26.58	45.54	92.64
Topic_FT_PEGASUS	**49.42**	**21.81**	**40.85**	**92.22**	**54.53**	**32.00**	**51.47**	**93.22**

Table 1: Evaluated Results over the public Dataset. R1, R2 , RL and BS stands for Rouge-1, Rouge 2, Rouge L and BERT Score respectively.

4 Experiments

This section describes the different baselines we used for comparison, followed by the training parameters used in these experiments.

4.1 Baselines

Following are the models we considered for our baselines:

1. PT_PEGASUS - In this setup, the pre-trained PEGASUS-LARGE model is adopted to generate the summary using the dialogue text as an input.

2. FT_PEGASUS - Here, the pre-trained PEGASUS-LARGE model uses the DialogSum train and development datasets. Only the dialogue text is used as an input.

4.2 Training Parameters

To fine-tune the PEGASUS model on the DialogSum dataset, training epochs is set to 10 with early stopping criteria. Since the PEGASUS is a heavy model and consumes 4 times more memory than the simple BERT model, batch_size is kept at 2 to avoid memory exhaustion. Warm-up steps are chosen at 500 with a $2e-5$ learning rate and weight decay of 0.01.

4.3 Evaluation Metric

The results of our proposed approach and other baselines are shown in Table 1. We have reported the recall of ROUGE (Recall Oriented Understudy for Gisting Evaluation) (Lin, 2004) score. It automatically measures the quality of generated summary by counting the overlapping units like n-grams with reference summary. ROUGE-1,

[3] https://huggingface.co/google/pegasus-large

ROUGE-2 and ROUGE-L [4] have been used for the evaluation. Since rouge scores don't consider semantic similarity, hence BERTScore[5] has also been used as an evaluation metric. It leverages the pre-trained contextual embeddings from BERT and matches the conceptual similarity between the model-generated and human summaries. Since public test set contains three topics and corresponding three human summaries for each dialogue text, hence, we have generated three model summaries corresponding to each topic and reported the average and best scores among the three. It should be noted that the best score is based on the best RL score among the three human summaries.

5 Results and Discussion

When we compared our proposed approach with the baselines, we found that our model outperformed the baselines with significant improvement. ROUGE-L has increased by 4.62% compared to FT_PEGASUS. The improvements indicate that fine-tuning of the PEGASUS model on the DialogSum dataset and topic relevance helped the model in extracting the essential information from the dialogues. We also computed the average length difference between our outputs and ground-truth summaries as recall depends on the length of generated summary. The average length of our model-generated summaries is 22.28 words, which is comparable to the ground-truth summaries, whose average length was 19.99 words.

6 Conclusion

As part of the DialogSum shared task on learning to generate a concise, fluent and topic-oriented summary of dialogues picked up from real-life scenarious, we have enhanced the performance of a pre-trained abstractive summarizer model by incorporating the topic along with the input text, to generate a topic-oriented summary. We have shown that the SOTA pre-trained transformer-based encoder-decoder model PEGASUS can be fine-tuned using the proposed methodology, to generate more human-like summaries of dialogues. Our model performed better in comparison to the baselines. In future, we plan to improve the method further by incorporating nuances of dialogue, speech act theory etc. The model can also be trained to learn the topic before generating a summary.

[4] https://github.com/cylnlp/dialogsum/blob/main/Baseline/rouge.py

[5] https://huggingface.co/metrics/bertscore

References

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021a. DialogSum: A real-life scenario dialogue summarization dataset. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5062–5074, Online. Association for Computational Linguistics.

Yulong Chen, Yang Liu, and Yue Zhang. 2021b. Dialogsum challenge: Summarizing real-life scenario dialogues. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 308–313.

Hyungtak Choi, Lohith Ravuru, Tomasz Dryjański, Sunghan Rye, Donghyun Lee, Hojung Lee, and Inchul Hwang. 2019. Vae-pgn based abstractive model in multi-stage architecture for text summarization. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 510–515.

Leyang Cui, Yu Wu, Shujie Liu, Yue Zhang, and Ming Zhou. 2020. MuTual: A dataset for multi-turn dialogue reasoning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1406–1416, Online. Association for Computational Linguistics.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Xiachong Feng, Xiaocheng Feng, and Bing Qin. 2021. A survey on dialogue summarization: Recent advances and new frontiers. *arXiv preprint arXiv:2107.03175*.

Xiachong Feng, Xiaocheng Feng, Bing Qin, and Xinwei Geng. 2020. Dialogue discourse-aware graph model and data augmentation for meeting summarization. *arXiv preprint arXiv:2012.03502*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. DailyDialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Nikola I Nikolov, Michael Pfeiffer, and Richard HR Hahnloser. 2018. Data-driven summarization of scientific articles. *arXiv preprint arXiv:1804.08875*.

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. *arXiv preprint arXiv:1910.10683*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Akshara Rai, Suyash Sangwan, Tushar Goel, Ishan Verma, and Lipika Dey. 2021. Query specific focused summarization of biomedical journal articles. In *2021 16th Conference on Computer Science and Intelligence Systems (FedCSIS)*, pages 91–100. IEEE.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. 2019. DREAM: A Challenge Data Set and Models for Dialogue-Based Reading Comprehension. *Transactions of the Association for Computational Linguistics*, 7:217–231.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, et al. 2021. Qmsum: A new benchmark for query-based multi-domain meeting summarization. *arXiv preprint arXiv:2104.05938*.

A Multi-Task Learning Approach for Summarization of Dialogues

Saprativa Bhattacharjee
Department of Information Technology
Government Polytechnic Daman
India
saprativa.bhatt@gov.in

Kartik Shinde
Department of Civil Engineering
Indian Institute of Technology Patna
India
kartik_1901ce16@iitp.ac.in

Tirthankar Ghosal
Charles University
MFF, ÚFAL
Czech Republic
ghosal@ufal.mff.cuni.cz

Asif Ekbal
Department of Computer Science and Engineering
Indian Institute of Technology Patna
India
asif@iitp.ac.in

Abstract

We describe our multi-task learning based approach for summarization of real-life dialogues as part of the *DialogSum Challenge* shared task at INLG 2022. Our approach intends to improve the main task of abstractive summarization of dialogues through the auxiliary tasks of extractive summarization, novelty detection and language modeling. We conduct extensive experimentation with different combinations of tasks and compare the results. In addition, we also incorporate the topic information provided with the dataset to perform topic-aware summarization. We report the results of automatic evaluation of the generated summaries in terms of ROUGE and BERTScore.

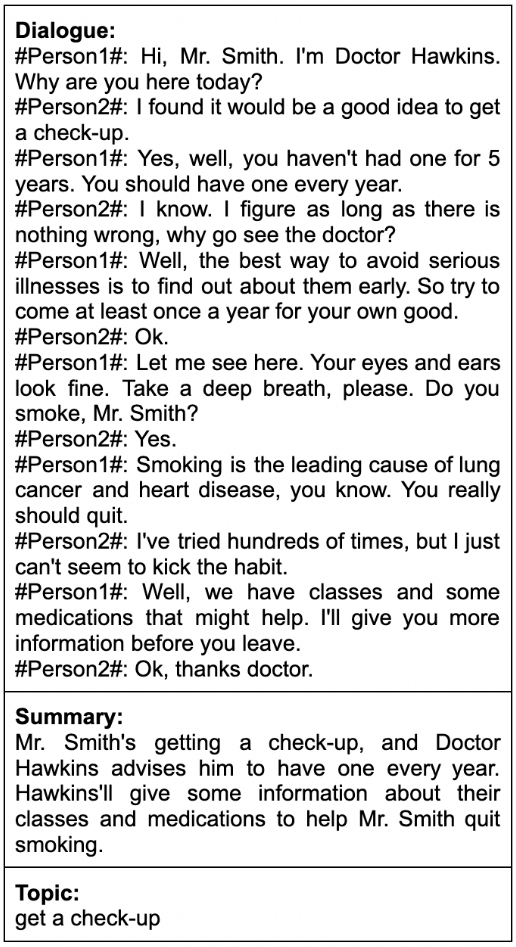
1 Introduction

Much of the early works on summarization devoted attention to *monologues* such as news articles (Nallapati et al., 2016; Narayan et al., 2018), patents (Sharma et al., 2019), Wikipedia articles (Liu et al., 2018; Cohen et al., 2021), scientific research papers (Cohan et al., 2018), Government reports (Huang et al., 2021) and even court judgements (Gao et al., 2019). But more recently, the focus of the summarization community has started shifting from monologues to *dialogues* largely owing to the rising popularity of chatbots, personal assistants, instant messaging platforms and online meetings. While monologues are characterised by the fact that they are authored by a single person, a dialogue involves the utterances of more than one participant (which alone can make them inherently more difficult to summarize). However, the available dialogue summarization datasets (Gliwa et al., 2019; Zhu et al., 2021; Feigenblat et al., 2021) are fewer in number, limited in scale, domain-specific and sometimes even extremely noisy and semi-structured (Carletta et al., 2005; Janin et al., 2003) as compared to the datasets available for monologue texts.

To mitigate these issues a high-quality large-scale dialogue summarization dataset named *DialogSum* was released by Chen et al. (2021a). The dataset consists of a wide variety of task-oriented dialogues from daily-life conversations. One sample dialogue and its corresponding summary from DialogSum's training set is presented in Figure 1, which is a conversation between a doctor and his patient on the topic of getting a check-up. To further encourage research in dialogue summarization, the authors proposed a shared task named DialogSum Challenge (Chen et al., 2021b) as part of INLG 2022, and in this article, we describe our submission to the shared task as Team IITP-CUNI.

Specifically, we attempt to tackle the problem of abstractive dialogue summarization through the use of a mutli-task learning model (Ruder, 2017; Crawshaw, 2020; Vandenhende et al., 2020) based on Transformers (Vaswani et al., 2017). We intend to improve the main task of abstractive summarization of the dialogues through the auxiliary tasks

Proceedings of the 15th International Conference on Natural Language Generation: Generation Challenges, pages 110 - 120
July 17-22, 2022 ©2022 Association for Computational Linguistics



Dialogue:
#Person1#: Hi, Mr. Smith. I'm Doctor Hawkins. Why are you here today?
#Person2#: I found it would be a good idea to get a check-up.
#Person1#: Yes, well, you haven't had one for 5 years. You should have one every year.
#Person2#: I know. I figure as long as there is nothing wrong, why go see the doctor?
#Person1#: Well, the best way to avoid serious illnesses is to find out about them early. So try to come at least once a year for your own good.
#Person2#: Ok.
#Person1#: Let me see here. Your eyes and ears look fine. Take a deep breath, please. Do you smoke, Mr. Smith?
#Person2#: Yes.
#Person1#: Smoking is the leading cause of lung cancer and heart disease, you know. You really should quit.
#Person2#: I've tried hundreds of times, but I just can't seem to kick the habit.
#Person1#: Well, we have classes and some medications that might help. I'll give you more information before you leave.
#Person2#: Ok, thanks doctor.

Summary:
Mr. Smith's getting a check-up, and Doctor Hawkins advises him to have one every year. Hawkins'll give some information about their classes and medications to help Mr. Smith quit smoking.

Topic:
get a check-up

Figure 1: A sample dialogue-summary pair along with the topic information from the DialogSum dataset's training set.

of extractive summarization, novelty detection and language modeling. Additionally, we also explore the usefulness of topic-aware summarization, as in the DialogSum dataset, topics are provided along with the summaries (see Figures 1 and 2).

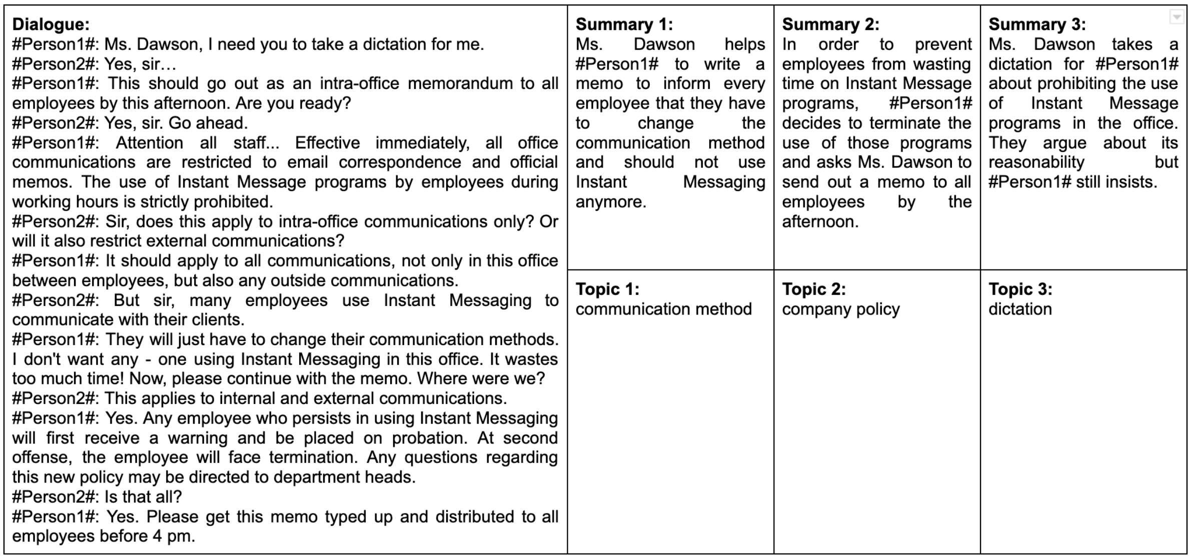
The rest of the paper is organised as follows. Related work is presented in Section 2. The DialogSum Challenge is described in details in Section 3. Section 4 presents our system. Results and discussion are in Section 5. Finally, the conclusion is drawn in Section 6.

2 Related Work

In this section, we discuss some of the most recent works on dialogue summarization and multi-task learning strategies for abstractive summarization. For long dialogue summarization, Zhong et al. (2021) proposed a window-based pre-training strategy using five different types of dialogue-related noise – speaker mask, turn splitting, turn merging, text infilling and turn permutation. At first, the window is corrupted with noise, and then the model is tasked with de-noising and reconstructing the window. On the other hand, Zhang et al. (2022) utilize a multi-stage approach for dealing with long dialogues. In the preliminary stages, they segment the input and produce coarse summaries, while in the final stage, the coarse summaries are used to generate the final fine-grained summary. Zhang et al. (2021) studied the effectiveness of different strategies to deal with long dialogues and concluded that a retrieve-then-summarize pipeline model works better in comparison to Longformer (Beltagy et al., 2020) or HMNet (Zhu et al., 2020). However, in the case of DialogSum, as the input data is well within the limit of the popular pre-trained Transformer models such as BART (Lewis et al., 2020), we are not faced with any such issues. Moreover, Chen et al. (2021a) have shown that the larger version of BART performs better than others on DialogSum. We start our investigation with this strong baseline.

Another direction of work has been the incorporation of topic information to further improve the abstractive dialogue summarization. In this direction, Zou et al. (2021) proposed a novel topic-augmented two-stage dialogue summarizer (TDS) along with a saliency-aware neural topic model (SATM) to perform topic-aware summarization of customer service dialogues. Qi et al. (2021) fused the topic segmentation embedding along with positional embedding in the utterance-level encoder input of a hierarchical Transformer architecture. To capture the topic information of dialogues Liu et al. (2021) came up with two contrastive learning strategies, namely coherence detection and sub-summary generation. And all of them reported performance benefits of taking topic information into account while performing abstractive summarization. We too explore the topic-aware summarization as the DialogSum dataset provides topic information along with the summaries.

A slightly different but closely related task that deserves mention is that of automatic minuting of meeting transcripts. The first shared task on Automatic Minuting (AutoMin) (Ghosal et al., 2021a) at Interspeech 2021 and the SIGDial 2021 Special Session on Summarization of Dialogues and Multi-Party Meetings (SummDial) (Ghosal et al., 2021b) brought out a plethora of interesting works targeting the task such as the attempt to use BART for generation of readable minutes (Shinde et al.,



Dialogue:	**Summary 1:**	**Summary 2:**	**Summary 3:**
#Person1#: Ms. Dawson, I need you to take a dictation for me. #Person2#: Yes, sir… #Person1#: This should go out as an intra-office memorandum to all employees by this afternoon. Are you ready? #Person2#: Yes, sir. Go ahead. #Person1#: Attention all staff... Effective immediately, all office communications are restricted to email correspondence and official memos. The use of Instant Message programs by employees during working hours is strictly prohibited. #Person2#: Sir, does this apply to intra-office communications only? Or will it also restrict external communications? #Person1#: It should apply to all communications, not only in this office between employees, but also any outside communications. #Person2#: But sir, many employees use Instant Messaging to communicate with their clients. #Person1#: They will just have to change their communication methods. I don't want any - one using Instant Messaging in this office. It wastes too much time! Now, please continue with the memo. Where were we? #Person2#: This applies to internal and external communications. #Person1#: Yes. Any employee who persists in using Instant Messaging will first receive a warning and be placed on probation. At second offense, the employee will face termination. Any questions regarding this new policy may be directed to department heads. #Person2#: Is that all? #Person1#: Yes. Please get this memo typed up and distributed to all employees before 4 pm.	Ms. Dawson helps #Person1# to write a memo to inform every employee that they have to change the communication method and should not use Instant Messaging anymore.	In order to prevent employees from wasting time on Instant Message programs, #Person1# decides to terminate the use of those programs and asks Ms. Dawson to send out a memo to all employees by the afternoon.	Ms. Dawson takes a dictation for #Person1# about prohibiting the use of Instant Message programs in the office. They argue about its reasonability but #Person1# still insists.
	Topic 1: communication method	**Topic 2:** company policy	**Topic 3:** dictation

Figure 2: A sample from the DialogSum test set which contains one dialogue and the three reference summaries along with three topics corresponding to each summary.

2021). Singh et al. (2021) present an empirical analysis of the state-of-the-art summarization models for the task of generating meeting minutes and arrive at the conclusion that they are far from being satisfactory. A novel dataset of meetings in English and Czech (Nedoluzhko et al., 2022) is also being released to further encourage the research community to take up the challenging task.

Lee et al. (2021) claim to be the first ones to have applied multi-task learning to dialogue summarization task. Leveraging Part-of-Speech (PoS) information, they constructed a syntax-aware dialogue summarization model on SAMSum corpus (Gliwa et al., 2019). The main intuition behind their approach is that different speaker roles are characterised by different syntactic structures (voiceprints), which could be captured via POS information. More recently, for low-resource datasets Magooda et al. (2021) experimented with several combinations of auxiliary tasks for abstractive summarization in a multi-task setting. They concluded that a certain combination of tasks indeed improved the abstractive summarization results across different datasets and models. Prior to these, in the multi-task setting, the primary task of abstractive summarization has been combined and experimented with several other auxiliary tasks such as entailment generation (Pasunuru et al., 2017); question generation and entailment generation (Guo et al., 2018); extractive summarization (Chen et al., 2019); text categorization and syntax labeling (Lu et al., 2019); dialogue act classification and extractive summarization (Manakul et al., 2020); keyword extraction and key-sentence extraction (Xu et al., 2020). Very recently, Chen et al. (2022) formulated the five different tasks of dialogue understanding (DU) as a unified generation task. These tasks include dialogue summarization, dialogue completion, dialogue state tracking, slot filling and intent detection. Then they experimented with eight different multi-task training strategies and concluded that their proposed method achieves superior performance on both few-shot as well as zero-shot settings. These encouraging results of the multi-task learning strategies on abstractive summarization motivated us to apply the same to the DialogSum Challenge.

3 DialogSum Challenge

In this section, we give a brief overview of the DialogSum Challenge by first describing the dataset and then going through the task description.

3.1 Dataset Description

The DialogSum dataset consists of a total of 13,460 dialogue-summary pairs, out of which 12,460 (92.6%) are in the training set, 500 (3.7%) in the development set and 500 (3.7%) more in the test set, as depicted in Figure 3. The dialogue data has been collected from multiple sources, namely 58.22% from DailyDialogue dataset (Li et al., 2017), 16.94% from DREAM dataset (Sun et al.,

Split	#Dialogues	#Turns	Turn Len.	Dialogue Len.	Summary Len.	%-Compression
train	12460	9.49	20.10	191.37	29.36	83.72
dev	500	9.38	20.17	188.89	27.21	84.74
test	500	9.71	20.04	196.12	23.76	86.70
hidden	100	10.88	19.03	209.42	–	–

Table 1: DialogSum dataset split statistics. '#Dialogues' contains absolute values while rest of the columns report average values. 'Len.' stands for Length. 'hidden' is the hidden test set for which only the dialogues and topics have been released publicly and hence the Summary Length and %-Compression details are not available.

2019), 13.89% from MuTual dataset (Cui et al., 2020) and the rest have been crawled from English speaking practice websites. The dialogues revolve around real-life conversations on topics such as schooling, work, medication, shopping, leisure and travel. The data from these varied sources are cleaned and transformed into a unified format before being annotated.

Some statistics of interest for each split of the dataset are presented in Table 1. Although the training, development and test sets are quite similar in terms of the average number of turns and the average turn length, the test set average dialogue length is larger while the average summary length is smaller than the other two sets. This also gets reflected in the test set's marginally higher compression ratio. Moreover, the average dialogue length of the hidden test set is higher than all other sets, but this may be attributed to the smaller size of the hidden set. In training and development sets, for each dialogue, one human written summary is provided. Figure 1 shows an example dialogue-summary pair from the training set. In addition to the summary, the human annotators also provide the topic information. On the other hand, for each dialogue in the test set, three human written reference summaries are provided. Figure 2 shows an example dialogue from the test set and its three reference summaries. For each reference summary, its corresponding topic is also provided.

In addition to the above, the organizers have also released a hidden test set consisting of 100 dialogues. Only the dialogues and topic information are provided for this hidden set, while the summaries have not been made public. The organizers will use this set for evaluation of the submitted models.

3.2 Task Description

The shared task participants need to design a model which will take as input the dialogue text and produce the corresponding abstractive summary. For automatic evaluation, each system-generated summary will be evaluated against the three human written reference summaries and the average ROUGE scores (Lin, 2004) and BERTScore (Zhang et al., 2020) will be used to determine the position on the DialogSum Challenge's leaderboard. Out of these two metrics, ROUGE (R1, R2 and RL) will be used as the primary metric, while BERTScore will be used as a supplementary metric. Additionally, the generated summaries will also be evaluated against the human-written summaries of the hidden test set. The lowest, highest and averaged scores will be reported for both the multi-reference test sets.

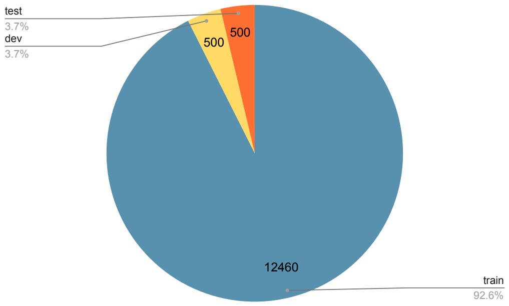


Figure 3: DialogSum dataset distribution.

For human evaluation, the submitted summaries will be judged on the following parameters: (i) fluency, consistency, relevance and coherence; (ii) co-reference information; (iii) intent identification; (iv) discourse relation; and (v) objective description. For more details about these parameters, we would like to refer the readers to the shared task paper (Chen et al., 2021b).

4 Our System

We employ a multi-task learning approach for the DialogSum Challenge. In multi-task learning, a machine learning model is trained simultaneously on more than one related task (Crawshaw, 2020).

Usually, there is a main task and one or more auxiliary tasks. In our case, the main task is abstractive summarization and the auxiliary tasks are extractive summarization, novelty detection and language modeling. There are many variants of multi-task learning. In this work, we employ a hard parameter sharing (Ruder, 2017) Transformers-based architecture in which all tasks share the same encoder layers but have task-specific decoder and/or LM head(s). The multi-task model architecture is depicted in Figure 4. It consists of a single BART encoder which is shared amongst all the tasks. The BART decoder is used for the main task of abstractive summarization, while task-specific heads are used for each of the respective auxiliary tasks. We now describe each of the tasks of our model one-by-one:

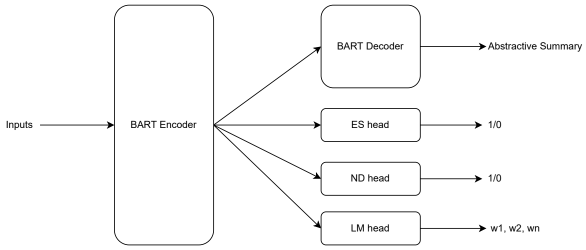


Figure 4: The multi-task learning model based on BART. AS: abstractive summarization; ES: extractive summarization; ND: novelty detection; LM: language modeling.

Abstractive Summarization (AS): For the main task of abstractive summarization, the transcripts are given as input to the BART encoder and the abstractive summaries are obtained as output from the BART decoder. This is a sequence-to-sequence task accomplished with the encoder-decoder architecture. In cases where we want to run only the single task for establishing the baseline, only this task is undertaken while keeping all other auxiliary tasks inactive through the training parameters.

Extractive Summarization (ES): The task of extractive summarization is formulated as a classification task where the goal is to classify a given sentence as either belonging to or not belonging to the extractive summary. The inputs are given in the format `[CLS] SW1, SW2, ..., SWn [SEP] CW1, CW2, ..., CWm`. Here, `[CLS]` is the start token, `[SEP]` is the separator token, `SW1...SWn` is the sentence to be classified as belonging to the extractive summary or not and `CW1...CWm` is the context around the sentence `SW1...SWn`. The sentence and the context around it are chosen in such a way that the maximum combined length does not exceed 1024 tokens.

Novelty Detection (ND): Novelty detection in NLP refers to the identification of novel text, i.e., text containing new information (Ghosal et al., 2022). This task is also formulated as a classification task. For this task, we use data from three different sources: (i) Quora Question Pair (QQP) dataset[1] consisting of more than 400 thousand question pairs. Each such pair is annotated with a binary value which indicates whether or not the questions in the pair are duplicates of each other. (ii) Microsoft Research Paraphrase Corpus (MRPC) (Dolan and Brockett, 2005) is a corpus consisting of 5,801 sentence pairs from news articles where each pair is annotated by humans as being either a paraphrase or not and (iii) data created from the three reference summaries given in the public test set of DialogSum. We assume that the three reference summaries are paraphrases (non-novel) of each other. Since there are 500 dialogues, each with three reference summaries, we obtain 1,500 non-novel samples. We also extract a similar number of novel samples by taking summaries from two different dialogues, as shown in Table 2. The input is given in the form `[CLS] source text [SEP] target text`, and the task of the model is to classify the pair as either novel or non-novel (duplicates).

Source	Target	Novel
Ref. Summary 1	Ref. Summary 2	0
Ref. Summary 2	Ref. Summary 3	0
Ref. Summary 1	Ref. Summary 3	0
Ref. Summary (Dn)	Ref. Summary (Dm)	1

Table 2: Novelty dataset created from the three reference summaries provided in the public test set of DialogSum. Ref. Summary (Dn) & Ref. Summary (Dm) denotes reference summaries from different dialogues.

Language Modeling (LM): We perform masked language modeling on the gold summaries from the training set as per the training strategy adopted by Devlin et al. (2019). For this, 15% of the input tokens are masked and out of this, 80% are replaced by special tokens, 10% with random words and the remaining 10% are left unchanged.

[1] https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs

5 Results and Discussion

In this section, we first describe the experimental setup used and then present the results. Finally, we analyse the summaries generated by our best-performing model.

5.1 Experimental Setup

We run all the experiments on two NVIDIA A100-PCIE-40GB GPUs using a batch size of 4 for both training and evaluation and mostly use the default values for hyperparameters. The BART model is initialized with `facebook/bart-large`[2] and then finetuned using task-specific datasets. Mixed-precision training using fp16 is utilized for faster training and lesser memory footprint. We make use of the summarization script released by Hugging Face[3] and the multi-task learning ideas introduced by Magooda et al. (2021). The ROUGE evaluations are done using `py-rouge`[4] and BERTScore evaluations using `bert_score`[5] as suggested by the organizers of DialogSum Challenge.

5.2 Results

We provide all the results from our experiments in Table 3. The reported performance is the average of the scores of system-generated summaries with respect to the three reference summaries provided in the public test set. We consider the single-task setting where only abstractive summarization (AS) is done without any auxiliary tasks as the baseline. For the topic-aware abstractive summarization (AS[T]), we supply the topic information by prepending it to the input dialogue to the BART encoder as `[CLS] TOPIC [SEP] Dialogue`. We observe a marginal improvement in the scores using this strategy.

In the multi-task setting, we experiment with different combinations of tasks as well as data. The best ROUGE scores are obtained when abstractive summarization is done along with extractive summarization (ES), while the best BERTScore is obtained when abstractive summarization is combined with novelty detection (ND). Since extractive summaries were not provided with the Dialgosum dataset, we used `bert-extractive-summarizer`[6] to obtain the same. Alongside the newly created extractive data from DialogSum, we also experiment with the extractive summary data from AMI (Carletta et al., 2005). Results show that the model trained with auxiliary task of extractive summarization (from AMI) outperforms all others. To explain such a performance, we analyze the outputs and test other configurations with both extractive datasets. However, in our observation, there are no apparent reasons for the model to perform in such a manner on AMI data. Finally, we account this to the fact that AMI is a dataset of meeting transcript and summaries, in which the information is widely dispersed throughout the discourse of the transcript, which have a lot of redundancies. While, dialogues from the DialogSum dataset are relatively shorter, with lesser redundant texts. Moreover, most of the lines from these dialogues (even those that are coherent with parts of summary), have a generic fashion of day-to-day speech. Hence, the BART model learns better from the extractive data from AMI.

Model	**R1**	**R2**	**RL**	**BERTScore**
		Single-Task		
AS	46.15	20.41	43.93	92.40
AS[T]	46.91	20.28	44.26	92.38
		Multi-Task		
AS+ES	46.24	19.42	43.54	92.40
AS+ES(AMI)	**47.26**	**21.18**	**45.17**	92.60
AS+ND(QQP)	46.62	20.12	44.09	**92.72**
AS+LM	45.11	18.92	43.08	92.30
AS+ES+ND(MRPC)	46.85	19.96	44.43	92.57
AS+ES(AMI)+ND	46.60	19.90	44.03	92.40
AS+ES(AMI)+ND(QQP)	46.73	20.30	44.44	92.43
AS+ES+LM	45.51	19.73	43.90	92.52
AS+ND(MRPC)+LM	45.14	19.60	43.20	92.26
AS+ES+ND(MRPC)+LM	45.62	19.80	44.10	92.60

Table 3: Results of single-task and multi-task models on the public test set of the DialogSum dataset. AS: abstractive summarization; ES: extractive summarization; ND: novelty detection; LM: language modeling; AS[T]: topic-aware abstractive summarization; ES(AMI): extractive summarization with AMI data; ND(MRPC): novelty detection with MRPC data; ND(QQP): novelty detection with Quora Question Pair data.

5.3 Analysis

We take our best performing model and manually analyse the summaries generated by it. Figure 5 and Figure 6 present the worst three and best

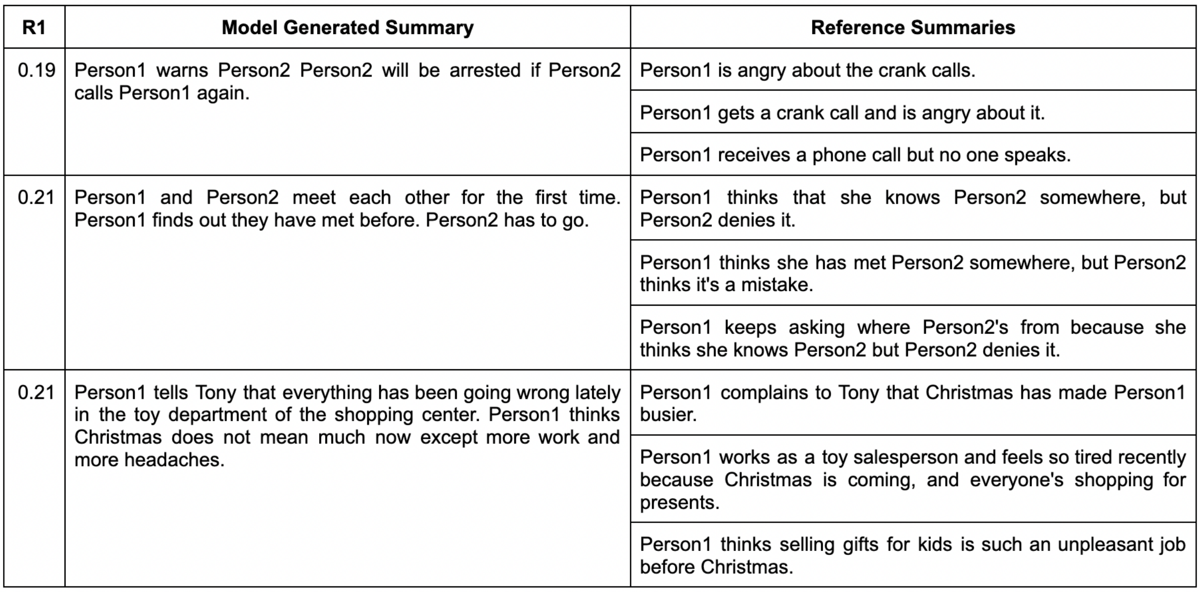
[2] https://huggingface.co/facebook/bart-large

[3] https://github.com/huggingface/transformers/tree/main/examples/pytorch/summarization

[4] https://pypi.org/project/py-rouge/

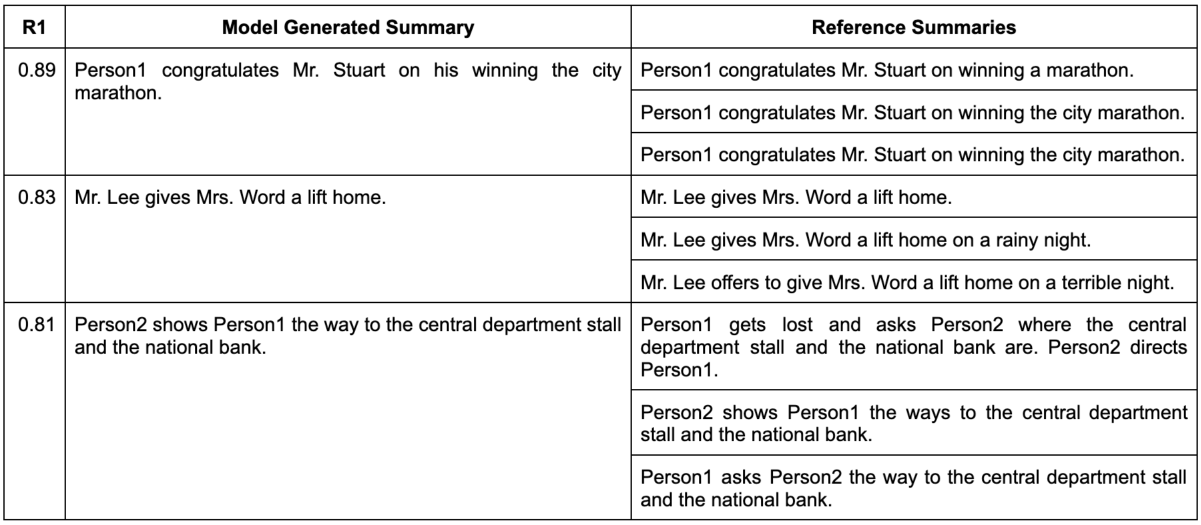
[5] https://github.com/Tiiiger/bert_score

[6] https://pypi.org/project/bert-extractive-summarizer/



R1	Model Generated Summary	Reference Summaries
0.19	Person1 warns Person2 Person2 will be arrested if Person2 calls Person1 again.	Person1 is angry about the crank calls.
		Person1 gets a crank call and is angry about it.
		Person1 receives a phone call but no one speaks.
0.21	Person1 and Person2 meet each other for the first time. Person1 finds out they have met before. Person2 has to go.	Person1 thinks that she knows Person2 somewhere, but Person2 denies it.
		Person1 thinks she has met Person2 somewhere, but Person2 thinks it's a mistake.
		Person1 keeps asking where Person2's from because she thinks she knows Person2 but Person2 denies it.
0.21	Person1 tells Tony that everything has been going wrong lately in the toy department of the shopping center. Person1 thinks Christmas does not mean much now except more work and more headaches.	Person1 complains to Tony that Christmas has made Person1 busier.
		Person1 works as a toy salesperson and feels so tired recently because Christmas is coming, and everyone's shopping for presents.
		Person1 thinks selling gifts for kids is such an unpleasant job before Christmas.

Figure 5: The worst three model-generated summaries in terms of ROUGE-1.



R1	Model Generated Summary	Reference Summaries
0.89	Person1 congratulates Mr. Stuart on his winning the city marathon.	Person1 congratulates Mr. Stuart on winning a marathon.
		Person1 congratulates Mr. Stuart on winning the city marathon.
		Person1 congratulates Mr. Stuart on winning the city marathon.
0.83	Mr. Lee gives Mrs. Word a lift home.	Mr. Lee gives Mrs. Word a lift home.
		Mr. Lee gives Mrs. Word a lift home on a rainy night.
		Mr. Lee offers to give Mrs. Word a lift home on a terrible night.
0.81	Person2 shows Person1 the way to the central department stall and the national bank.	Person1 gets lost and asks Person2 where the central department stall and the national bank are. Person2 directs Person1.
		Person2 shows Person1 the ways to the central department stall and the national bank.
		Person1 asks Person2 the way to the central department stall and the national bank.

Figure 6: The best three model-generated summaries in terms of ROUGE-1.

three summaries generated by the model in terms of ROUGE-1, respectively. It is to be kept in mind that the ROUGE scores reported are the average of the generated summary with respect to the three reference summaries. Let us first consider the case of the three worst summaries shown in Figure 5. In the case of the first system-generated summary, we can see that it is longer than each one of the three reference summaries and the content is quite different. In the second case, our model is unable to figure out that Person1 "thinks" she met/knows Person2. Rather the model generates the phrase "finds out". Moreover, the last line, "Person2 has to go" is totally unnecessary for the summary. In the case of the third summary, although the system-generated summary conveys the same message as the reference summaries, yet the same is not reflected in terms of ROUGE-1 mainly because of the different set of unigrams used.

Let us now consider the best three summaries generated by our model as shown in Figure 6. In all three cases, it can be seen that the generated summary matches almost exactly to one of the three reference summaries. The second system-generated summary matches word-to-word with its first reference summary, while the first and third system-generated summaries differ with their respective best matches on only a single word. The higher score of the first summary can be attributed to the fact that two out of the three reference summaries in this case turn out to be exactly the same, which takes the average score up.

6 Conclusion

In this paper, we describe our submission to the shared task on dialogue summarization named DialogSum Challenge at INLG 2022. DialogSum consists of 13,460 real-life scenario dialogues. We employ a multi-task learning approach for the task and achieve considerable improvement over the single-task baseline. Our best performing model is the multi-task combination of abstractive summarization as the main task and extractive summarization as the auxiliary task. We also incorporate the topic information supplied alongside the summaries to gain marginal improvement in performance over the baseline. In future work, we would like to experiment with other tasks to find the optimal combination. We would also like to explore methods other than multi-task learning for improving the abstractive summarization of dialogues.

Acknowledgements

Tirthankar Ghosal would like to acknowledge the support from the grant 19-26934X (NEUREM3) of the Czech Science Foundation and the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 825460 for the project European Live Translator (ELITR).

References

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *CoRR*, abs/2004.05150.

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Maël Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska, Iain McCowan, Wilfried Post, Dennis Reidsma, and Pierre Wellner. 2005. The AMI meeting corpus: A pre-announcement. In *Machine Learning for Multimodal Interaction, Second International Workshop, MLMI 2005, Edinburgh, UK, July 11-13, 2005, Revised Selected Papers*, volume 3869 of *Lecture Notes in Computer Science*, pages 28–39. Springer.

Yangbin Chen, Yun Ma, Xudong Mao, and Qing Li. 2019. Multi-task learning for abstractive and extractive summarization. *Data Sci. Eng.*, 4(1):14–23.

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021a. DialogSum: A real-life scenario dialogue summarization dataset. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5062–5074, Online. Association for Computational Linguistics.

Yulong Chen, Yang Liu, and Yue Zhang. 2021b. DialogSum challenge: Summarizing real-life scenario dialogues. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 308–313, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Zhi Chen, Lu Chen, Bei Chen, Libo Qin, Yuncong Liu, Su Zhu, Jian-Guang Lou, and Kai Yu. 2022. Unidu: Towards A unified generative dialogue understanding framework. *CoRR*, abs/2204.04637.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Nachshon Cohen, Oren Kalinsky, Yftah Ziser, and Alessandro Moschitti. 2021. WikiSum: Coherent summarization dataset for efficient human-evaluation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 212–219, Online. Association for Computational Linguistics.

Michael Crawshaw. 2020. Multi-task learning with deep neural networks: A survey. *CoRR*, abs/2009.09796.

Leyang Cui, Yu Wu, Shujie Liu, Yue Zhang, and Ming Zhou. 2020. MuTual: A dataset for multi-turn dialogue reasoning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1406–1416, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Guy Feigenblat, Chulaka Gunasekara, Benjamin Sznajder, Sachindra Joshi, David Konopnicki, and Ranit Aharonov. 2021. TWEETSUMM - a dialog summarization dataset for customer service. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 245–260, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shen Gao, Xiuying Chen, Piji Li, Zhangming Chan, Dongyan Zhao, and Rui Yan. 2019. How to write summaries with patterns? learning towards abstractive summarization through prototype editing. 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 3741–3751, Hong Kong, China. Association for Computational Linguistics.

Tirthankar Ghosal, Ondřej Bojar, Muskaan Singh, and Anja Nedoluzhko. 2021a. Overview of the first shared task on automatic minuting (automin) at interspeech 2021. In *Proceedings of the First Shared Task on Automatic Minuting at Interspeech 2021*, pages 1–25.

Tirthankar Ghosal, Tanik Saikh, Tameesh Biswas, Asif Ekbal, and Pushpak Bhattacharyya. 2022. Novelty Detection: A Perspective from Natural Language Processing. *Computational Linguistics*, 48(1):77–117.

Tirthankar Ghosal, Muskaan Singh, Anna Nedoluzhko, and Ondřej Bojar. 2021b. Report on the SIGDial 2021 special session on summarization of dialogues and multi-party meetings (summdial). *ACM SIGIR Forum*, December 2021:1–17.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

Han Guo, Ramakanth Pasunuru, and Mohit Bansal. 2018. Soft layer-specific multi-task summarization with entailment and question generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 687–697, Melbourne, Australia. Association for Computational Linguistics.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1419–1436, Online. Association for Computational Linguistics.

Adam Janin, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke, and Chuck Wooters. 2003. The ICSI meeting corpus. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '03, Hong Kong, April 6-10, 2003*, pages 364–367. IEEE.

Seolhwa Lee, Kisu Yang, Chanjun Park, João Sedoc, and Heuiseok Lim. 2021. Who speaks like a style of vitamin: Towards syntax-aware dialogue summarization using multi-task learning. *IEEE Access*, 9:168889–168898.

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.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. DailyDialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Junpeng Liu, Yanyan Zou, Hainan Zhang, Hongshen Chen, Zhuoye Ding, Caixia Yuan, and Xiaojie Wang. 2021. Topic-aware contrastive learning for abstractive dialogue summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1229–1243, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Yao Lu, Linqing Liu, Zhile Jiang, Min Yang, and Randy Goebel. 2019. A multi-task learning framework for abstractive text summarization. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 9987–9988. AAAI Press.

Ahmed Magooda, Diane Litman, and Mohamed Elaraby. 2021. Exploring multitask learning for low-resource abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1652–1661, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Potsawee Manakul, Mark J. F. Gales, and Linlin Wang. 2020. Abstractive spoken document summarization using hierarchical model with multi-stage attention diversity optimization. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 4248–4252. ISCA.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Anna Nedoluzhko, Muskaan Singh, Marie Hledíková, Tirthankar Ghosal, and Ondřej Bojar. 2022. ELITR Minuting Corpus: A novel dataset for automatic minuting from multi-party meetings in English and Czech. In *Proceedings of The 13th Language Resources and Evaluation Conference*, page To Appear.

Ramakanth Pasunuru, Han Guo, and Mohit Bansal. 2017. Towards improving abstractive summarization via entailment generation. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 27–32, Copenhagen, Denmark. Association for Computational Linguistics.

MengNan Qi, Hao Liu, YuZhuo Fu, and Ting Liu. 2021. Improving abstractive dialogue summarization with hierarchical pretraining and topic segment. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1121–1130, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *CoRR*, abs/1706.05098.

Eva Sharma, Chen Li, and Lu Wang. 2019. BIGPATENT: A large-scale dataset for abstractive and coherent summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2204–2213, Florence, Italy. Association for Computational Linguistics.

Kartik Shinde, Nidhir Bhavsar, Aakash Bhatnagar, and Tirthankar Ghosal. 2021. Team abc @ automin 2021: Generating readable minutes with a bart-based automatic minuting approach. In *Proceedings of the First Shared Task on Automatic Minuting at Interspeech 2021*, pages 1–8.

Muskaan Singh, Tirthankar Ghosal, and Ondrej Bojar. 2021. An empirical performance analysis of state-of-the-art summarization models for automatic minuting. In *Proceedings of the 35th Pacific Asia Conference on Language, Information and Computation*, pages 50–60, Shanghai, China. Association for Computational Linguistics.

Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. 2019. DREAM: A challenge data set and models for dialogue-based reading comprehension. *Transactions of the Association for Computational Linguistics*, 7:217–231.

Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, Dengxin Dai, and Luc Van Gool. 2020. Revisiting multi-task learning in the deep learning era. *CoRR*, abs/2004.13379.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Weiran Xu, Chenliang Li, Minghao Lee, and Chi Zhang. 2020. Multi-task learning for abstractive text summarization with key information guide network. *EURASIP J. Adv. Signal Process.*, 2020(1):16.

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.

Yusen Zhang, Ansong Ni, Ziming Mao, Chen Henry Wu, Chenguang Zhu, Budhaditya Deb, Ahmed Awadallah, Dragomir Radev, and Rui Zhang. 2022. Summ^n: A multi-stage summarization framework for long input dialogues and documents. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1592–1604, Dublin, Ireland. Association for Computational Linguistics.

Yusen Zhang, Ansong Ni, Tao Yu, Rui Zhang, Chenguang Zhu, Budhaditya Deb, Asli Celikyilmaz, Ahmed Hassan Awadallah, and Dragomir Radev. 2021. An exploratory study on long dialogue summarization: What works and what's next. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4426–4433, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ming Zhong, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2021. Dialoglm: Pre-trained model for long dialogue understanding and summarization. *CoRR*, abs/2109.02492.

Chenguang Zhu, Yang Liu, Jie Mei, and Michael Zeng. 2021. MediaSum: A large-scale media interview dataset for dialogue summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5927–5934, Online. Association for Computational Linguistics.

Chenguang Zhu, Ruochen Xu, Michael Zeng, and Xuedong Huang. 2020. A hierarchical network for abstractive meeting summarization with cross-domain pretraining. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 194–203, Online. Association for Computational Linguistics.

Yicheng Zou, Lujun Zhao, Yangyang Kang, Jun Lin, Minlong Peng, Zhuoren Jiang, Changlong Sun, Qi Zhang, Xuanjing Huang, and Xiaozhong Liu. 2021. Topic-oriented spoken dialogue summarization for customer service with saliency-aware topic modeling. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14665–14673. AAAI Press.

Dialogue Summarization using BART

Conrad Lundberg[*] and **Leyre Sánchez Viñuela**[*] and **Siena Biales**[*]
University of Tübingen
conrad.lundberg@student.uni-tuebingen.de
leyre.sanchez-vinuela@student.uni-tuebingen.de
siena.biales@student.uni-tuebingen.de

Abstract

This paper introduces the model and settings submitted to the INLG 2022 DialogSum Challenge, a shared task to generate summaries of real-life scenario dialogues between two people. In this paper, we explored using intermediate task transfer learning, reported speech, and the use of a supplementary dataset in addition to our base fine-tuned BART model. However, we did not use such a method in our final model, as none improved our results. Our final model for this dialogue task achieved scores only slightly below the top submission, with hidden test set scores of 49.62, 24.98, 46.25 and 91.54 for ROUGE-1, ROUGE-2, ROUGE-L and BERTSCORE respectively. The top submitted models will also receive human evaluation.

1 Introduction

Dialogue summarization is a variation of text summarization which aims to generate concise, coherent summaries of conversations. Dialogue summarization requires far deeper insight than summarizing a news article or similar documents, as is done in text summarization. When handling a dialogue, a model must address semantic roles, resolve definite pronouns and coreference, and handle various other complexities (Chen et al., 2021b). We investigate the best methods for summarizing a dialogue while retaining these difficult relations that do not present a problem when summarizing a simple text.

The INLG 2022 DialogSum Challenge is a shared task with the goal of generating summaries of real-life scenario dialogues between two people. In this paper, we will describe our approach to this task using a fine-tuned BART model. Additionally, we explore the effects of using intermediate task transfer learning, reported speech for this task. However, we did not use such a method in our final model, as none improved our results.

2 Background

The field of text summarization has been in focus for decades. Research into automatic text summarization began as early as 1958 with the summarization of magazine articles and technical papers (Luhn, 1958). Text summarization proves challenging for many reasons. The model must be able to identify important topics and condense them in a way that is not redundant, but yet remains readable and cohesive (El-Kassas et al., 2021). Primarily, there are two approaches to text summarization: extractive and abstractive. Extractive summarization seeks to extract the most important information and present it as is. Abstractive summarization, in contrast, may use novel words to create a linguistically correct condensed representation (Zhang et al., 2020). Originally, research on extractive summarization was in the foreground (Murray et al., 2005) but the field is now moving towards abstractive summarization based on neural sequence-to-sequence encoder-decoder models (Sutskever et al., 2014). Top performing models to create summaries have also been based on transformers (Vaswani et al., 2017). Pointer-generator models (See et al., 2017) are another state-of-the-art summarization technique, combining extractive and abstractive methods.

Dialogue summarization is now emerging as a new interest in the field of natural language processing. As early as 2010, Higashinaka et al. were exploring methods of extractive summarization to summarize contact center dialogues using a hidden Markov model called Class Speaker HMM. Since then, more unique and effective methods have emerged. Yuan and Yu (2019) proposed a Scaffold Pointer Network (SPNet), which incorporated three types of semantic scaffolds found in dialogue: speaker role, semantic slot, and dialog domain. Chen and Yang (2020) introduced a multi-view sequence-to-sequence model, which utilized

[*] All authors contributed equally.

Proceedings of the 15th International Conference on Natural Language Generation: Generation Challenges, pages 121 - 125
July 17-22, 2022 ©2022 Association for Computational Linguistics

conversational structures and topic segmentation to assist in better dialogue summarization.

BART is a sequence-to-sequence model that pre-trains by combining Bidirectional and Auto-Regressive Transformers, and achieves good results on a range of abstractive dialogue and summarization tasks (Lewis et al., 2019). Khalifa et al. (2021) found BART to be a viable base model for dialogue summarization and showed additional methods could improve results. For this reason, we selected BART as our base model.

3 System Overview and Methods

In this section, we discuss the setup and hyperparameters of our final model, as well as attempts to improve our results, which included using intermediate task transfer learning, reported speech, and an additional dataset.

3.1 Setup and Hyperparameter Tuning

Our model was made by fine tuning a BART model on 12460 dialogue/summary pairs in the DIALOGSUM dataset provided by the INLG 2022 DialogSum Challenge (Chen et al., 2021a). The training and validation datasets provided to us contain a dialogue, a gold summary, an identifier, and a topic. The dialogue is formatted such that each line represents one dialogue turn. The lines begin with either $\#Person1\#$: or $\#Person2\#$: to identify who is speaking. We pass the full dialogue to the model as input without any further preprocessing apart from randomization of the dataset and tokenization.

In the training dataset, there were 7434 unique topics provided. Some examples of the most common topics are "shopping", "job interview", or "phone call", but even these were only found in about 100 of the 12,460 training instances. The least common topics were only found on one instance and include "job losing", "look ill", "stop doing business", or "the language club". While the topic data could prove useful, we discarded the topic for the purposes of this task.

The BART model described in this paper was first fine-tuned on the CNN/Dailymail corpus (Hermann et al., 2015). We used an NVIDIA Tesla P100 16GB GPU to train our fine-tuned model.

When tuning our hyperparameters, we began with the most impactful settings and documented improvements on each training iteration. In initial training runs with a high learning rate, the model outputted only a few words repeatedly, and appeared overfitted. We opted to use the same learning rate as the task organizers documented in their hyperparameter settings (3e-5) for our final model.

Our best model used a batch size of 2. Other batch sizes (e.g. 3,4,8) were also tested, yet with our settings, using larger batch sizes did not improve results. We trained our model for 3 epochs on the full training dialogue dataset with no early stopping.

3.2 Post-processing

When decoding the generated summary, important adjustments included the minimum and maximum summary lengths, along with a length penalty parameter, which penalizes longer summaries. A very low value for the length penalty tells the model to generate shorter sequences. The perfect summary length is subjective, but these parameters helped to obtain results that were most similar to the target test set summaries. In our final model, we used a minimum length of 14, a maximum length of 64, and a length penalty of 0.04.

To reduce hallucinations in the transformer model, we preemptively replace any instances of speakers who did not appear in the initial dialogues, such as $\#Person3\#$ or $\#Person4\#$, to $\#Person1\#$ or $\#Person2\#$. In addition, we fixed any instances of duplicate labels, such as $\#Person1\#Person1\#$ or $\#Person2\#Person2\#$.

3.3 Intermediate Task Transfer Learning

We experimented with the use of intermediate task transfer learning for this task. Pruksachatkun et al. (2020) studied the effects of multiple intermediate tasks on a variety of target tasks trained on RoBERTa. Although none of the target tasks in the paper were related to text or dialogue summarization, there were some intermediate tasks (Cosmos QA, HellaSwag) that improved target task results across the board, regardless of the task. We decided to investigate the use of one of these generally successful intermediate tasks, HellaSwag, on the dialogue summarization task to see if we could observe any improvement.

The HellaSwag dataset (Zellers et al., 2019) is a natural language inference dataset modeled as multiple-choice questions, where there are four possible answers for continuing the scene set in the "question". This task is easy for humans to determine the correct sentence continuation given the context in the initial sentence, but computers struggle to achieve the same success. In order to alter the HellaSwag question-answer dataset into a sequence-to-sequence problem that our model could solve, we opted to remove all the negative answers and treat the context sentence as the initial sequence, with the correct answer choice as the target sequence.

We trained our BART model for 1 epoch on 10% of the HellaSwag training split and then trained the same model on the DIALOGSUM training dataset exactly as described previously. Unfortunately, the ROUGE scores were all consistently lower using this technique. ROUGE-1, ROUGE-2, and ROUGE-L dropped by 1.2, 1.9, and 1.1 points respectively.

Although training with HellaSwag as an intermediate task did not yield positive results, we also attempted intermediate task transfer learning on a more similar task, namely, news article summarization. For this, we used a portion of the XSum dataset (Narayan et al., 2018). The XSum dataset contains a series of news articles along with one-sentence summaries of each article, making it already ideal for a sequence-to-sequence task with no preprocessing required. Similarly to the HellaSwag dataset, we first trained our BART model for 1 epoch on the XSum training split, and then used this to train on the DIALOGSUM training dataset. Unfortunately, this also resulted in consistently lower ROUGE scores. ROUGE-1, ROUGE-2, and ROUGE-L dropped by 1.5, 1.4, and 0.9 points respectively when using XSum for intermediate task transfer learning.

Our attempt at intermediate task transfer learning did not yield improved results and was not used in our final model, however it did provide valuable information in regard to the question of where intermediate task transfer learning can be applied. In further work, it may be beneficial to further optimize the hyperparameters, such as increasing the number of training epochs on the intermediate task or using larger training splits, before completely ruling out the potential uses of intermediate task transfer learning on the task of dialogue summarization.

3.4 Directed and Reported Speech

The dialogues used for this task and the news articles that the BART model was originally fine-tuned with contain quite different discursive and linguistic structures. The dialogues contain direct speech, using mainly the first and second person verbs conjugations, whereas the news articles have a more narrative style, with a higher use of the third person. We experimented with transforming the structure of our dialogues into reported speech without altering their content to make it more similar to the structure of the news, with the hope that fine-tuning BART with more similar data to what it had been originally fine-tuned with would yield better results.

After fine-tuning BART with these dialogues in their reported-speech form, we had lower ROUGE scores than with the original ones, so we discarded this preprocessing step in our final model. This could be due to the poor quality of our rule-based reported speech transformation algorithm, which results in an excessive use of the verb "says" and some problems in the pronouns reference resolution, but this direct-to-reported-speech task could indeed be interesting to further explore.

3.5 Data Augmentation

Finally, we attempted augmenting our training data by adding a supplementary dataset with similar data to that found in DIALOGSUM. We used the SAMSum dataset (Gliwa et al., 2019), presented as a human-annotated dialogue dataset for abstractive summarization. This dataset presents 16k messenger-like conversations written by linguists fluent in English, together with their summaries.

After merging both datasets, we fine-tuned BART with them, however, we once again achieved results inferior to training on the original dataset alone. This could be due to the shorter length of the SAMSum dialogues and summaries compared to those in DIALOGSUM. It could also be attributed to the different linguistic features between the datasets; the SAMSum dialogues are in a written format, whereas the DIALOGSUM dialogues emulate spoken conversations.

4 Results

Many of the generated summaries produced were close matches to the target summaries. Sometimes generated summaries seemed as though they were a good summarization of the dialogue, but nonetheless had low ROUGE scores. In some cases, this was due to length discrepancies. In other cases, our model generated novel word choices which varied from the gold standard. Examples of a high scoring

TARGET	*#Person1# tells Kate that Masha and Hero get divorced. Kate is surprised because she thought they are perfect couple.*
GENERATED	*#Person1# tells Kate Masha and Hero are getting divorced. Kate is surprised because she thought they are the perfect couple.*
TARGET	*#Person1# and Mike are discussing what kind of emotion should be expressed by Mike in this play. They have different understandings.*
GENERATED	*#Person1# thinks Mike is acting hurt and sad because that's not how his character would act in this situation, but #Person2# thinks Jason and Laura had been together for 3 years so his reaction would be one of both anger and sadness.*

Table 1: Examples of a generated summary close to the target summary (above) and a less ideal generated summary (below)

and low scoring summary can be found in Table 1.

The results were evaluated on ROUGE-1, ROUGE-2, ROUGE-L and BERTSCORE. ROUGE scores measure the *n*-grams shared between the generated and target summaries. ROUGE-L measures the longest shared n-gram. BERTSCORE looks at contextual embeddings instead of exact matches to give a similarity score (Zhang et al., 2019). Our model performed comparable to current leaderboard results on the public test set, and also shows what seem to be respectable results on the hidden test set. Our scores can be found in Table 2.

Our attempts utilizing intermediate task transfer learning, reported speech, and additional datasets all proved unsuccessful. We hypothesize this is a result of insufficient hyperparameter tuning or training. When more complexity is introduced in a model, it often requires specific hyperparameter tuning to result in success, and we suspect this may be one reason our attempts failed.

5 Conclusion

In this paper, we have described our attempt at the INLG 2022 DialogSum Challenge shared task, aimed at generating summaries of real-life scenario dialogues. We utilized a fine-tuned BART model trained on the DIALOGSUM dataset provided to us to achieve our best results.

We explored utilizing intermediate task transfer learning to improve our model, however we speculate that this failed due to a domain mismatch in the datasets, or perhaps due to insufficient hyperparameter tuning and training. Future work could explore intermediate task transfer learning with an intermediate dataset that is better suited for dialogue summarization. Our attempts at altering our data from direct to reported speech, to reflect the dataset that our BART model was fine-tuned with did not work in our favor. We assume this was due to the quality of the reported speech transformation algorithm. Utilizing an additional dataset to increase our number of training samples also did not give desired results. This could be due to differences in the datasets, such as domain, length of texts and summaries, or other factors.

Our results show that it is possible to achieve relatively successful dialogue summarization results using only a basic BART model and fine-tuning on this dataset. In the future, we would further explore the methods we described above.

	R1	R2	RL	BERTSCORE
Public	47.29	21.65	45.92	92.26
Hidden	49.75	25.15	46.50	91.76

Table 2: Scores achieved using the model described in this paper, on both the public and hidden test sets

References

Jiaao Chen and Diyi Yang. 2020. Multi-view sequence-to-sequence models with conversational structure for abstractive dialogue summarization. *arXiv preprint arXiv:2010.01672*.

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021a. DialogSum: A real-life scenario dialogue summarization dataset. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5062–5074, Online. Association for Computational Linguistics.

Yulong Chen, Yang Liu, and Yue Zhang. 2021b. DialogSum challenge: Summarizing real-life scenario dialogues. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 308–313.

Wafaa S El-Kassas, Cherif R Salama, Ahmed A Rafea, and Hoda K Mohamed. 2021. Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, 165:113679.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. *arXiv preprint arXiv:1911.12237*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28.

Ryuichiro Higashinaka, Yasuhiro Minami, Hitoshi Nishikawa, Kohji Dohsaka, Toyomi Meguro, Satoshi Takahashi, and Genichiro Kikui. 2010. Learning to model domain-specific utterance sequences for extractive summarization of contact center dialogues. In *Coling 2010: Posters*, pages 400–408.

Muhammad Khalifa, Miguel Ballesteros, and Kathleen McKeown. 2021. A bag of tricks for dialogue summarization. *arXiv preprint arXiv:2109.08232*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

HP Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, pages 159–165.

Gabriel Murray, Steve Renals, and Jean Carletta. 2005. Extractive summarization of meeting recordings. pages 593—-596.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745*.

Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel R Bowman. 2020. Intermediate-task transfer learning with pretrained models for natural language understanding: When and why does it work? *arXiv preprint arXiv:2005.00628*.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Lin Yuan and Zhou Yu. 2019. Abstractive dialog summarization with semantic scaffolds. *arXiv preprint arXiv:1910.00825*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. BERTscore: Evaluating text generation with BERT. *arXiv preprint arXiv:1904.09675*.

Author Index