

INLG 2023

**The 16th International Natural Language Generation  
Conference: Generation Challenges**

**Proceedings of the Generation Challenges**

September 11 - 15, 2023

©2023 The Association for Computational Linguistics

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](mailto:acl@aclweb.org)

ISBN 979-8-89176-003-5

## Preface by the GenChal Chair

The Generation Challenges (GenChal) aim at bringing together a variety of shared-task efforts that involve the generation of natural language. This year, the Generation Challenges was held during a special session at the 16th International Conference on Natural Language Generation (INLG 2023, September 11-15 2023). The session followed the format of previous GenChal results sessions, with presentations of results by the organisers of recently completed tasks, a poster session for task participants to present their submissions, as well as presentations of proposals for new shared tasks.

In 2023, we received five new shared task proposals; one was withdrawn by the authors and each of the other four proposals was reviewed by three or four PC members. Three proposals were accepted as such, and one proposal was accepted after revision; the four proposals can be found at the beginning of the present proceedings. Three completed shared tasks are also included in these proceedings, with an overview report by the organisers and participants' system descriptions in one single paper (AIWolfDial) or separate papers (AutoMin, FGC). The system descriptions underwent a light touch review from the task organisers and the GenChal chair.

### New Challenge Proposals

- **LOWRECORP**: the Low-Resource NLG Corpus Building Challenge — Khyathi Raghavi Chandu, David M. Howcroft, Dimitra Gkatzia, Yi-Ling Chung, Yufang Hou, Chris Chinenye Emezue, Pawan Rajpoot and Tosin Adewumi
- Long Story Generation Challenge — Nikolay Mikhaylovskiy
- Visually Grounded Story Generation Challenge — Xudong Hong, Khushboo Mehra, Asad Sayeed and Vera Demberg
- The VDG Challenge: Response Generation and Evaluation in Collaborative Visual Dialogue — Nikolai Ilinykh and Simon Dobnik

### Completed Challenges

- The Second Automatic Minuting (AutoMin) Challenge: Generating and Evaluating Minutes from Multi-Party Meetings.
- Shared Task on Feedback Comment Generation for Language Learners (FGC).
- The 5th annual AIWolf international contest (AIWolfDial).

### Further information

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



# Organizing Committee

## **GenChal Chair**

Simon Mille, ADAPT, Dublin City University, Ireland

## **GenChal Publication Chair**

Chung-Chi Chen, National Institute of Advanced Industrial Science and Technology, Japan

## **SIGGEN Exec**

Emiel van Miltenburg, Tilburg University, the Netherlands

## **AutoMin Shared Task Organisers**

Tirthankar Ghosal, Oak Ridge National Laboratory, TN, USA

Ondrej Bojar, Charles University, Czech Republic

Marie Hledíková, Charles University, Czech Republic

Tom Kocmi, Microsoft, Germany

Anna Nedoluzhko, Charles University, Czech Republic

## **FGC Shared Task Organisers**

Ryo Nagata, Konan University / RIKEN, Japan

Masato Hagiwara, Octanove Labs, USA

Kazuaki Hanawa, RIKEN, Japan

Masato Mita, RIKEN, Japan

## **AIWolfDial Shared Task Organisers**

Yoshinobu Kano, Shizuoka University, Japan

Claus Aranha, Tsukuba University

Takashi Otsuki, Yamagata University, Japan

Fujio Toriumi, The University of Tokyo, Japan

Hiroataka Osawa, University of Tsukuba, Japan

Daisuke Katagami, Tokyo Polytechnic University, Japan

Michimasa Inaba, The University of Electro-Communications, Japan

Kei Harada, The University of Electro-Communications, Japan

## **Program Committee**

### **Program Committee**

Jose Alonso, University of Santiago de Compostela  
Guanyi Chen, Utrecht University  
Pablo Duboue, Textualization Software Ltd.  
Kim Gerdes, Paris-Saclay University  
Raquel Hervas, Universidad Complutense de Madrid  
Yufang Hou, IBM Research  
David M. Howcroft, Edinburgh Napier University  
Daphne Ippolito, Carnegie Mellon University  
Yoshinobu Kano, Faculty of Informatics, Shizuoka University  
Elena Lloret, University of Alicante  
Simon Mille, ADAPT, Dublin City University  
Maja Popović, ADAPT, Dublin City University  
Ehud Reiter, University of Aberdeen  
Craig Thomson, University of Aberdeen  
Chris Van Der Lee, Tilburg University

## Table of Contents

<i>LOWRECOP: the Low-Resource NLG Corpus Building Challenge</i>	
Khyathi Raghavi Chandu, David M. Howcroft, Dimitra Gkatzia, Yi-Ling Chung, Yufang Hou, Chris Chinenye Emezue, Pawan Rajpoot and Tosin Adewumi . . . . .	1
<i>Long Story Generation Challenge</i>	
Nikolay Mikhaylovskiy . . . . .	10
<i>Visually Grounded Story Generation Challenge</i>	
Xudong Hong, Khushboo Mehra, Asad Sayeed and Vera Demberg . . . . .	17
<i>The VDG Challenge: Response Generation and Evaluation in Collaborative Visual Dialogue</i>	
Nikolai Ilinykh and Simon Dobnik . . . . .	23
<i>Identifying Feedback Types to Augment Feedback Comment Generation</i>	
Maja Stahl and Henning Wachsmuth . . . . .	31
<i>Error syntax aware augmentation of feedback comment generation dataset</i>	
Nikolay Babakov, Maria Lysyuk, Alexander Shvets, Lilya Kazakova and Alexander Panchenko . . . . .	37
<i>A Report on FCG GenChal 2022: Shared Task on Feedback Comment Generation for Language Learners</i>	
Ryo Nagata, Masato Hagiwara, Kazuaki Hanawa and Masato Mita . . . . .	45
<i>Sentence-level Feedback Generation for English Language Learners: Does Data Augmentation Help?</i>	
Shabnam Behzad, Amir Zeldes and Nathan Schneider . . . . .	53
<i>Retrieval, Masking, and Generation: Feedback Comment Generation using Masked Comment Examples</i>	
Mana Ihori, Hiroshi Sato, Tomohiro Tanaka and Ryo Masumura . . . . .	60
<i>TMU Feedback Comment Generation System Using Pretrained Sequence-to-Sequence Language Models</i>	
Naoya Ueda and Mamoru Komachi . . . . .	68
<i>The Tokyo Tech and AIST System at the GenChal 2022 Shared Task on Feedback Comment Generation</i>	
Shota Koyama, Hiroya Takamura and Naoaki Okazaki . . . . .	74
<i>Feedback comment generation using predicted grammatical terms</i>	
Kunitaka Jimichi, Kotaro Funakoshi and Manabu Okumura . . . . .	79
<i>AIWolfDial 2023: Summary of Natural Language Division of 5th International AIWolf Contest</i>	
Yoshinobu Kano, Neo Watanabe, Kaito Kagaminuma, Claus Aranha, Jaewon Lee, Benedek Hauer, Hisaichi Shibata, Soichiro Miki, Yuta Nakamura, Takuya Okubo, Soga Shigemura, Rei Ito, Kazuki Takashima, Tomoki Fukuda, Masahiro Wakutani, Tomoya Hatanaka, Mami Uchida, Mikio Abe, Akihiro Mikami, Takashi Otsuki, Zhiyang Qi, Kei Harada, Michimasa Inaba, Daisuke Katagami, Hirotaka Osawa and Fujio Toriumi . . . . .	84
<i>Team Zoom @ AutoMin 2023: Utilizing Topic Segmentation And LLM Data Augmentation For Long-Form Meeting Summarization</i>	
Felix Schneider and Marco Turchi . . . . .	101
<i>Team Synapse @ AutoMin 2023: Leveraging BART-Based Models for Automatic Meeting Minuting</i>	
Kristýna Klesnilová and Michelle Elizabeth . . . . .	108

<i>Team Iterate @ AutoMin 2023 - Experiments with Iterative Minuting</i>	
František Kmječ and Ondřej Bojar .....	114
<i>Darbarer @ AutoMin2023: Transcription simplification for concise minute generation from multi-party conversations</i>	
Ismaël Rousseau, Loïc Fosse, Youness Dkhissi, Geraldine Damnati and Gwénolé Lecorvé . . . .	121
<i>Team NTR @ AutoMin 2023: Dolly LLM Improves Minuting Performance, Semantic Segmentation Doesn't</i>	
Eugene Borisov and Nikolay Mikhaylovskiy .....	132
<i>Overview of the Second Shared Task on Automatic Minuting (AutoMin) at INLG 2023</i>	
Tirthankar Ghosal, Ondřej Bojar, Marie Hledíková, Tom Kocmi and Anna Nedoluzhko .....	138



# LOWRECORP: the Low-Resource NLG Corpus Building Challenge

Khyathi Raghavi Chandu<sup>1</sup> David Howcroft<sup>2</sup> Dimitra Gkatzia<sup>2</sup>  
Yi-Ling Chung<sup>3</sup> Yufang Hou<sup>4</sup> Chris Emezue<sup>5</sup> Pawan Rajpoot<sup>6</sup> Tosin Adewumi<sup>7</sup>

<sup>1</sup>Allen Institute of AI, <sup>2</sup>Edinburgh Napier University, <sup>3</sup>The Alan Turing Institute,  
<sup>4</sup>IBM Research, <sup>5</sup>Technical University of Munich, <sup>6</sup>MUST Research, <sup>7</sup>Luleå University of Technology  
khyathic@allenai.org  
lowrecorp.github.io

## Abstract

Most languages in the world do not have sufficient data available to develop neural-network-based natural language generation (NLG) systems. To alleviate this resource scarcity, we propose a novel challenge for the NLG community: low-resource language corpus development (LOWRECORP). We present an innovative framework to collect a single dataset with dual tasks to maximize the efficiency of data collection efforts and respect language consultant time. Specifically, we focus on a text-chat-based interface for two generation tasks – *conversational response generation* grounded in a source document and/or image and *dialogue summarization* (from the former task). The goal of this shared task is to collectively develop grounded datasets for local and low-resourced languages. To enable data collection, we make available web-based software that can be used to collect these grounded conversations and summaries. Submissions will be assessed for the size, complexity, and diversity of the corpora to ensure quality control of the datasets as well as any enhancements to the interface or novel approaches to grounding conversations.

## 1 Introduction

Around the world, people speak about 7000 different languages and nearly all of these have very weak support in language technologies. While about 100 languages are included in recent large language models (e.g. Xue et al., 2021; Devlin et al., 2019), most languages do not have good resources. The situation is especially dire when we examine task-specific datasets, such as for response generation, summarisation, and other forms of natural language generation (NLG).

To address this problem, we propose a new shared task on dataset creation for NLG: LOWRECORP<sup>1</sup> challenge, which invites participants to collect a new dataset combining dialogue

<sup>1</sup>pronounced as <low> <re> /ləʊ ri/, or <Lowry> /ləʊ ri/ followed by <corp> /kɔɪp/.

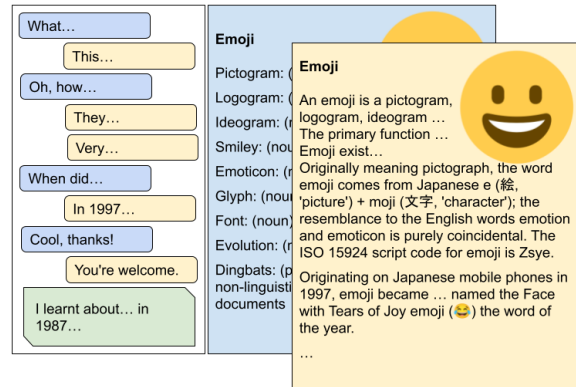


Figure 1: Schematic of the data collection interface. Participants ask questions about a topic (blue, left text bubbles) and answer them (yellow, right), each with access to the same title & image but different grounding text (keyword definitions for the QUESTIONER, a full article for the RESPONDER). After each conversation, each summarises the discussion (green, under chat).

grounded in documents and/or images and dialogue summarisation for a low-resource language (LRL) of their choice. Although conversational response generation and summarization have distinct objectives, they both aim to generate coherent output based on a given context. Drawing from this symbiosis, a new task is proposed that combines the challenges of both tasks into a single framework. The first task is **grounded response generation** and the second is **dialogue summarization**. *Grounded response generation* is the task of generating a conversational response grounded in a context such as documents, images, or other modality, to provide more contextually relevant information (Reddy et al., 2019; Kim et al., 2020; Wu et al., 2021). Similarly, *dialogue summarization* aims to generate a summary of a conversation while preserving its main ideas, and it is particularly useful in scenarios such as meeting notes and doctor-patient conversations where there may be a lot of irrelevant exchange that can obscure informative segments, making the task challenging for traditional approaches (Ghosal et al., 2022).

To facilitate this challenge, we developed a new interface for `slurk` (Götze et al., 2022) which allows paired participants to view different grounding materials such as text, vision, audio, etc., relating to the same topics to engage in a conversation and dialogue summarization task (Figures 1 and 2). Combining these tasks enables the efficient use of participant time and energy, which is especially important when working with LRLs which often have small communities of use, and will serve broader research efforts in linguistic analysis, language documentation, and preservation efforts (Anastasopoulos, 2019). In contrast to mitigation strategies such as data augmentation and multilingual language models (Hedderich et al., 2021; Chandu and Geramifard, 2022), our dual-task design focuses on efficiency during corpus creation. To our knowledge, this is the first work combining both grounded response generation and dialogue summarization to maximize benefits for LRLs data collection.

## 2 Task Description

Engaging human participants to build or annotate corpora typically takes one of two forms. In the first, bulk annotations<sup>2</sup> for a single task are collected via crowdsourcing, but this is challenging to replicate in LRL settings as recruiting, training, and maintaining qualified annotators is more difficult. In the second, annotation tasks are built incrementally, which requires recruiting multiple pools of participants or having the same participants return for multiple tasks – for example multiple sessions to collect annotations for retrieval, localization, and comparison (e.g. Hessel et al., 2022). In contrast, we propose a third approach that combines two tasks within the same experimental session to alleviate challenges in recruiting proficient speakers of low-resourced target languages. The two tasks are:

**Task 1: Grounded Response Generation** Given an image or document and the conversation history as context, respond to the previous utterance.

**Task 2: (Dialogue) Summarization** Given the full grounded conversation, summarize the important points with the goal to maximize reconstructing the original grounding context.

These 2 tasks of short and long form generation complement each other well, for instance, using

---

<sup>2</sup>Throughout, we will use ‘annotations’ equally to refer to annotations on or based upon existing texts as well as the creation of new texts for a corpus and other tasks performed by participants in the process of creating and annotating a corpus.

summarization to identify argument facets in a dialogue (Misra et al., 2015) or dialogue acts for summarization (Goo and Chen, 2018). Note that there can be more such pairs of complementary generation tasks.

### 2.1 Setup

Each instance of data collection involves a context (image/text), keywords with brief descriptions, and 2 participants. We pair participants in the roles of QUESTIONER and RESPONDER. The RESPONDER is given access to a full context, which can be an image, a document, or both. The QUESTIONER is given access to a list of keywords related to the context in order to familiarize the participants so they can ask meaningful and relevant questions. This partially serves the function of world knowledge, as though a participant knows something about the topic but not a lot. Note that the descriptions and the keywords can be in any chosen language.

Both participants are expected to familiarize themselves with their context (the full context for the RESPONDER and the keyword context for the QUESTIONER) at the start of a session. The QUESTIONER then needs to ask questions to learn more about the topic, and the RESPONDER answers those questions *based on the information provided to them in their context*. After the conversation is complete, both participants write a brief summary of the topic discussed, based on their conversation. Figure 2 depicts how our chat interface is used by the QUESTIONER and the RESPONDER to converse (left) about the context (right). An example of task setup is provided in Appendix A.

## 3 Implementation Strategies

The proposed dual-task design provides a framework for efficiently collecting complementary datasets. Noting that many large language models today are trained on a substantial proportion of the data found online in any given language and that these models will likely be used as components in future systems trained on the datasets our participants create, we encourage participants to focus on sources of grounding material that are less likely to be in the training data for such models already (e.g. information about museum exhibits, children’s stories, etc.). We invite participants to leverage this framework to gather datasets in indigenous and local languages grounded on topics of local interest, deciding exactly *what* data to collect and *how* to

Latency: 999 ms  
Users: QASumBot, (QA079) Neach-tadhail, You

## Seòmair-obrach airson QASum

**QASumBot** 0:16  
Cuir a-steach /tòiseachadh gus an deuchainn a thoiseachadh.

**You** 0:16  
toiseachadh

**QASumBot** 0:16  
A-nis a' feitheamh ris a' chompanach agad /'toiseachadh' a thaipeadh.

**QASumBot** 0:16  
Math fhèin! Tòisichidh an còmhraidh a-nis.

**(QA079) Neach-tadhail** 0:16  
Halò a charaid

**You** 0:17  
Halò! A bheil ceistan ann?

**(QA079) Neach-tadhail** 0:17  
Tha. De tha tàileasg?

This room is read-only

Freagairtean do na ceistean air an taisbeanadh.

**Tha thu ag obair aig an taigh-tasgaidh agus a' taisbeanadh na h-ulaidh (exhibit item) inntinnich seo.** Tha an teacsa gu h-ìosal a' riochdachadh an fhiosrachaidh air fad a th' agad mun ulaidh.


(1) **Freagair ceistean do chom-pàirtichean mun taisbeanadh.** Feuch ri freagairtean iomchaidh a thoirt seachad a tha a' riochdachadh co-theacs an fhiosrachaidh a chaidh a thoirt dhut. *Na cleachd an t-eòlas priobhaideach no pearsanta agad fhèin na do fhreagairtean.*

(2) Nuair a tha thu a' faireachdainn gu bheil an còmhraidh air fiosrachadh gu leòr a thoirt seachad agus air àile-stad comhfhurtail a ruighinn, cuir am brath: /ath

Aon uair 's gu bheil thu fhèin agus an com-pàirtiche ag aontachadh gu bheil an còmhraidh agaibh deiseil, sgrìobhaidh gach neach agaibh gearr-chunntas air an fhiosrachadh mun an do bhruidhinn thu.

---

### Tàileasg Leòdhais



Tha na piosan tàileisg meadhan-aoiseil seo à Eilean Leòdhais na h-Alba am meas nan cruinneachaidhean as mòr-chòrdte a th' againn. Bha na h-aon-deug piosan tàileisg a bha air an taisbeanadh ann an Taigh-tasgaidh na h-Alba mar phàirt de chunntas mòr de 93 piosan geama a chaidh a thiodhlacadh ann an Leòdhas.

Thàinig an tasgadh am follais an toiseach nuair a chaidh na piosan a thaisbeanadh

Figure 2: A screenshot of the chat interface used for data collection in Gaelic (Howcroft and Gkatzia, 2023). The chat area is on the left, and the summary of the instructions for the RESPONDER and the beginning of an entry to be discussed on the right.

Implementation Strategy	Technological Access/Literacy	Data volume	Quality Control
Online across network	High	High	Low
In lab or field	Moderate	Moderate	High
Offline in field	Can be low	Moderate	High

Table 1: Tradeoffs of different implementation strategies

collect it based on the considerations of the target language and its community. We foresee at least three possible approaches to data collection, highlighted in Table 1 along with their tradeoffs.

**Online across the network:** In this approach, researchers use a webserver to deliver contextual materials along with an audio, video, or text chat interface. This allows researchers to potentially run multiple pairs of participants simultaneously and provides digital representations of the resulting dialogues and summaries from the start. However, this method relies on linguistic literacy to provide instructions (and potentially for data collection, if using a text-based chat interface) and on technological literacy, since participants must be familiar with using a computer, tablet, or smartphone and feel comfortable interacting with the software.

**Computer-based in the lab or field:** With this approach, researchers are no longer limited to participants with access to technology at home and can be on-hand to answer questions from participants about the interface or troubleshoot any problems. Researchers can use the same kind of software that they would normally deploy online, hosted on

a single laptop. In cases where the aim is to collect spoken dialogues, this also allows the researcher to control the quality of the audio equipment instead of relying on participants to use their own hardware. This method also ensures that the data is immediately available in a digital format and may help address management and/or privacy concerns since data will not need to be transmitted over the internet. This approach requires access to a lab or another controlled space and incurs travel costs either for the participants (to come to a lab of the researcher's choosing) or the researchers (to meet the participants where they are).

**Offline in the field:** Developing corpora for LRLs can be limited by factors such as participant literacy (Hirmer et al., 2021), lack of availability of technology, systemic societal issues (Ahia et al., 2021), and unrepresentative user bases for crowdsourcing. Therefore, it is possible to implement this dialogue+summarization task fully offline, using in-person methods. Researchers can provide contextual materials (keyword explanations and source materials) to participants on printed sheets of paper and use a microphone to record conversa-

tions and summaries. This method may be most helpful when working with participants with very low technological literacy or in communities where electricity or connectivity is limited. In order for participants to use the source materials to answer questions, however, they will need to be literate or to have materials provided to them in a visual format. This approach will also generally require transcription in addition to the kind of normalization and data cleaning required by the other approaches.

**Recruitment Considerations** Recruiting participants for LRLs for co-designing, corpus collection, and system evaluation can be challenging due to the small number of speakers. For example, Scottish Gaelic (*Gàidhlig*) has about 57,000 speakers living in Scotland,<sup>3</sup> implying the difficulty even for a wealthy country with good internet access. In such cases, it can be helpful to collaborate with researchers and institutions that are already involved with the target community. These contacts provide access to community leaders and information exchange structures like community centers and newspapers to promote experiments to potential participants. Being able to reach audiences using spoken or signed media may be especially crucial for LRLs with lower literacy rates (Wu, 2014).

## 4 Submissions and Assessment

**Submission Details** More details about the task and the interface are available at [lowrecorp.github.io](https://github.com/lowrecorp). Researchers interested in participating in the challenge can contact us at [lowrecorp@googlegroups.com](mailto:lowrecorp@googlegroups.com), where participating teams can interact and receive updates and support from the organizers. Researchers will upload their data in May 2024 to allow sufficient time for reporting at INLG 2024. Each team is expected to submit a paper to a special session that discusses their chosen subject, target language, any innovations in their approach, and key corpus statistics along with a data card (Gebru et al., 2021).

**Assessment** A strict evaluation of the diverse submissions expected for this challenge would limit the creativity and flexibility of the teams. Hence, we intend to focus on open evaluations aimed primarily at quality control and description of the data, adapting the data-to-text corpus description guidelines of Perez-Beltrachini and Gardent (2017) to our dialogue and summarisation tasks. For exam-

<sup>3</sup><https://www.scotlandscensus.gov.uk/census-results/at-a-glance/languages/>

Metric or Corpus Property
Grounding material size, complexity, diversity
Conversation length & duration
Lexical diversity (e.g. TTR, bigram TTR)
Corpus & vocabulary size
Lexical and/or syntactic diversity (if possible)
Language typology, geography, community
Creative grounding sources or interface use

Table 2: Parameters for quality control and evaluation

ple, participants will report the number of different grounding contexts used, the size of those grounding contexts<sup>4</sup>, the number of conversations, and the number of summaries. Conversations should be characterized in terms of duration (time; the number of turns) and corpus statistics such as type-token ratio, vocabulary, and corpus sizes. Measures of lexical difficulty or syntactic diversity and complexity available for the language being studied should also be included. Participants should also prepare a datasheet following Gebru et al. (2018).

We will recognize and celebrate submissions based on a variety of dimensions, such as corpus size, lexical diversity, language rarity, most creative grounding source, etc. (cf. Table 2). Submissions involving creative use of local grounding sources especially in areas where technological reach is limited are particularly recognized and commended.

## 5 Conclusion

This challenge aims to bring together researchers interested in corpus building for LRLs to work on a shared, streamlined, and vetted protocol (tested in Scottish Gaelic) to build interesting corpora. We hope that our challenge will contribute toward recent efforts in addressing geographically diverse NLP (Fan et al., 2021; Nekoto et al., 2020; Aldabe et al., 2022) by aiding in the creation of new datasets for NLG tasks from a wider variety of languages. We aim to encourage cross-pollination of ideas and ideally set up for future generation challenges in a variety of low-resourced languages which make use of the data collected. The resulting (future) task will serve as a text-and-image-grounded complementary task to efforts like the semantic-web-grounded WebNLG challenge<sup>5</sup> and the syntactically-grounded Multilingual Surface Realization challenge (Mille et al., 2020).

<sup>4</sup>e.g. number of keywords, length of keyword descriptions, length of full texts for RESPONDER’s, image size/complexity

<sup>5</sup><https://github.com/WebNLG/2023-Challenge>

## Ethical implications

While we believe that our proposed dual-task framework can maximize annotation effort, particularly for low-resource languages, data collection should be handled carefully. We highlight in this section several ethical considerations when collecting data in low-resource languages.

**Bias** The problem of dataset bias is, often, inevitable and can lead to false conclusions and poor generalization of learning methods trained on a given dataset, regardless of modalities (Tommasi et al., 2017). It is important that equal representation is used in the data collection, such as inclusive language, gender, race, and religion (Dhamala et al., 2021). While our approach aims at balancing data availability for low-resource languages by collecting additional data, additional countermeasures would help, for instance, a data statement outlining the data collection process and annotator demographics (Bender and Friedman, 2018). To promote application fairness, researchers are encouraged to quantify dataset bias (Adewumi et al., 2023) and measure the risks of unintended bias.

**Privacy** While our task is not privacy-demanding, we advocate that the resulting dataset/annotation should adhere to privacy policies such as GDPR data privacy mandates from European Union (Europe, 2019). To reduce privacy risks, several measures should be considered. For instance, when collecting conversations between questioners and answerers in this task, it is preferred that annotators address each other in a way that does not disclose their private information. Private (or personally identifiable) information, such as names and social security numbers, can expose individuals to potential harm and should not be captured in data collection unless absolutely necessary (Sokolova and Matwin, 2016). If sensitive information is collected, anonymization and/or pseudonymization techniques should be applied to protect participants (Terrovitis et al., 2012).

**Responsible innovation** Responsible innovation, or responsible AI, refers to careful consideration of the potential impacts and benefits of introducing a new product or service. In the context of research in low-resource languages, researchers will need to consider the impact of using online resources that might be copyrighted (e.g., digital media from museum websites). The societal impact will need

to be considered such as the privacy of speakers of low-resource languages which might be compromised for instance if a dialect is only spoken by a small number of speakers.

**Recruitment and Exploitation** When recruiting participants, high priority should be given to first-language (L1) speakers of the languages of interest. This ensures that the data will be representative of how the language is used by its primary language community and fulfills the inclusiveness principle. Researchers should also consider whether proficiency, regardless of L1 or L2 speaker status, is adequate for inclusion in the corpus collection efforts, depending on the goals of the research. It is important that participants are not overloaded with a high volume of keywords and documents per time, as this may affect the quality of the data collected negatively in addition to being an unreasonable amount of work. Adequate compensation should be established, at a minimum adhering to industry standards or regulatory provisions but preferably aiming at providing a ‘living wage’. In situations where their contribution is based on a voluntary basis, researchers will need to take extra care to ensure that participants’ contributions are freely given and that their needs are respected. One option worth considering for language communities with small numbers of speakers is offering participants the option of being a named contributor to the project, to acknowledge their contribution to the preservation and technological development of their language, which they may appreciate.

**Leveraging Translation** Although translation is not the primary goal of the task, human translation from a high-resource language or one LRL to multiple LRLs can be used as a collection strategy. While it serves several benefits (Adewumi et al., 2022) such as reducing cost while maintaining the correctness of the task, it suffers from challenges such as entrainment (Mizukami et al., 2016; Chandu et al., 2018). Besides the difficulty of recruitment from possibly a low population, another challenge is the representation of local entities in the target languages. Some ways of solving this challenge include replacing such entities with local ones by using the knowledge of native speakers while keeping in mind that semi-automatic alterations of such technologies might include biases from the high resource languages reducing the naturalness of the data (Chandu et al., 2017).

## Acknowledgements

DH and DG are supported under the EPSRC project 'NLG for low-resource domains' (EP/T024917/1).

## References

- Tosin Adewumi, Mofetoluwa Adeyemi, Aremu Anuoluwapo, Bukola Peters, Happy Buzaaba, Oyerinde Samuel, Amina Mardiyyah Rufai, Benjamin Ajibade, Tajudeen Gwadabe, Mory Moussou Koulibaly Traore, et al. 2022. Afriwoz: Corpus for exploiting cross-lingual transferability for generation of dialogues in low-resource, african languages. *arXiv preprint arXiv:2204.08083*.
- Tosin Adewumi, Isabella Södergren, Lama Alkhaled, Sana Sabah Sabry, Foteini Liwicki, and Marcus Liwicki. 2023. Bipol: Multi-axes evaluation of bias with explainability in benchmark datasets. *arXiv preprint arXiv:2301.12139*.
- Orevaoghene Ahia, Julia Kreutzer, and Sara Hooker. 2021. [The low-resource double bind: An empirical study of pruning for low-resource machine translation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3316–3333, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Itziar Aldabe, Jane Dunne, Aritz Farwell, Owen Gallagher, Federico Gaspari, Maria Giagkou, Jan Hajic, Jens Peter Kückens, Teresa Lynn, Georg Rehm, German Rigau, Katrin Marheinecke, Stelios Piperidis, Natalia Resende, Tea Vojtěchová, and Andy Way. 2022. [Overview of the ELE project](#). In *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation*, pages 353–354, Ghent, Belgium. European Association for Machine Translation.
- Antonios Anastasopoulos. 2019. *Computational Tools for Endangered Language Documentation*. Ph.D. thesis.
- Emily M. Bender and Batya Friedman. 2018. [Data statements for natural language processing: Toward mitigating system bias and enabling better science](#). *Transactions of the Association for Computational Linguistics*, 6:587–604.
- Khyathi Raghavi Chandu and Alborz Geramifard. 2022. [Multilingual multimodality: A taxonomical survey of datasets, techniques, challenges and opportunities](#). *CoRR*, abs/2210.16960.
- Khyathi Raghavi Chandu, Ekaterina Loginova, Vishal Gupta, Josef van Genabith, Günter Neumann, Manoj Kumar Chinnakotla, Eric Nyberg, and Alan W. Black. 2018. [Code-mixed question answering challenge: Crowd-sourcing data and techniques](#). In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching@ACL 2018, Melbourne, Australia, July 19, 2018*, pages 29–38. Association for Computational Linguistics.
- Khyathi Raghavi Chandu, Sai Krishna Rallabandi, Sunayana Sitaram, and Alan W. Black. 2017. [Speech synthesis for mixed-language navigation instructions](#). In *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, pages 57–61. ISCA.
- 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.
- Jwala Dhamala, Tony Sun, Varun Kumar, Satyapriya Krishna, Yada Pruksachatkun, Kai-Wei Chang, and Rahul Gupta. 2021. [Bold: Dataset and metrics for measuring biases in open-ended language generation](#). In *ACM FAccT 2021*.
- Interactive Advertising Bureau Europe. 2019. [Gdpr transparency and consent framework](#).
- Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. [Beyond english-centric multilingual machine translation](#). *The Journal of Machine Learning Research*, 22(1):4839–4886.
- Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. 2018. [Datasheets for Datasets](#). In *Proc. of the 5th Workshop on Fairness, Accountability, and Transparency in Machine Learning*, Stockholm, Sweden.
- Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna M. Wallach, Hal Daumé III, and Kate Crawford. 2021. [Datasheets for datasets](#). *Commun. ACM*, 64(12):86–92.
- Tirthankar Ghosal, Muskaan Singh, Anja Nedoluzhko, and Ondřej Bojar. 2022. [Report on the sigdial 2021 special session on summarization of dialogues and multi-party meetings \(summdial\)](#). *SIGIR Forum*, 55(2).
- Chih-Wen Goo and Yun-Nung Chen. 2018. [Abstractive dialogue summarization with sentence-gated modeling optimized by dialogue acts](#). In *Proceedings of 7th IEEE Workshop on Spoken Language Technology*.
- Jana Götze, Maike Paetzel-Prüsmann, Wencke Liermann, Tim Diekmann, and David Schlangen. 2022. [The slurk interaction server framework: Better data for better dialog models](#). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4069–4078, Marseille, France. European Language Resources Association.

- Michael A. Hedderich, Lukas Lange, Heike Adel, Jan-nik Strötgen, and Dietrich Klakow. 2021. [A survey on recent approaches for natural language processing in low-resource scenarios](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2545–2568, Online. Association for Computational Linguistics.
- Jack Hessel, Jena D. Hwang, Jae Sung Park, Rowan Zellers, Chandra Bhagavatula, Anna Rohrbach, Kate Saenko, and Yejin Choi. 2022. [The abduction of sherlock holmes: A dataset for visual abductive reasoning](#). In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXXVI*, volume 13696 of *Lecture Notes in Computer Science*, pages 558–575. Springer.
- Stephanie Hirmer, Alycia Leonard, Josephine Tumweise, and Costanza Conforti. 2021. Building Representative Corpora from Illiterate Communities: A Review of Challenges and Mitigation Strategies for Developing Countries. In *Proc. of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 2176–2189. Association for Computational Linguistics.
- David M. Howcroft and Dimitra Gkatzia. 2023. Building a dual dataset of text- and image-grounded conversations and summarisation in Gàidhlig (Scottish Gaelic). In *Proceedings of the 16th International Conference on Natural Language Generation, Prague, Czech Republic and virtual meeting*. Association for Computational Linguistics.
- Seokhwan Kim, Mihail Eric, Karthik Gopalakrishnan, Behnam Hedayatnia, Yang Liu, and Dilek Hakkani-Tur. 2020. [Beyond domain APIs: Task-oriented conversational modeling with unstructured knowledge access](#). In *Proceedings of the 21st Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 278–289, 1st virtual meeting. Association for Computational Linguistics.
- Simon Mille, Anya Belz, Bernd Bohnet, Thiago Castro Ferreira, Yvette Graham, and Leo Wanner. 2020. [The third multilingual surface realisation shared task \(SR’20\): Overview and evaluation results](#). In *Proceedings of the Third Workshop on Multilingual Surface Realisation*, pages 1–20, Barcelona, Spain (Online). Association for Computational Linguistics.
- Amita Misra, Pranav Anand, Jean E. Fox Tree, and Marilyn A. Walker. 2015. [Using summarization to discover argument facets in online ideological dialog](#). 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 430–440. The Association for Computational Linguistics.
- Masahiro Mizukami, Koichiro Yoshino, Graham Neubig, David R. Traum, and Satoshi Nakamura. 2016. [Analyzing the effect of entrainment on dialogue acts](#). In *Proceedings of the SIGDIAL 2016 Conference, The 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 13-15 September 2016, Los Angeles, CA, USA*, pages 310–318. The Association for Computer Linguistics.
- Wilhelmina Nekoto, Vukosi Marivate, Tshinondiwa Matsila, Timi Fasubaa, Taiwo Fagbohunge, Solomon Oluwole Akinola, Shamsuddeen Muhammad, Salomon Kabongo Kabenamualu, Salomey Osei, Freshia Sackey, Rubungo Andre Niyongabo, Ricky Macharm, Perez Ogayo, Orevaoghene Ahia, Musie Meressa Berhe, Mofetoluwa Adeyemi, Masabata Mokgesi-Seling, Lawrence Okegbemi, Laura Martinus, Kolawole Tajudeen, Kevin Degila, Kelechi Ogueji, Kathleen Siminyu, Julia Kreutzer, Jason Webster, Jamiil Toure Ali, Jade Abbott, Iroro Orife, Ignatius Ezeani, Idris Abdulkadir Dangana, Herman Kamper, Hady Elsahar, Goodness Duru, Ghollah Kioko, Murhabazi Espoir, Elan van Biljon, Daniel Whitenack, Christopher Onyefuluchi, Chris Chinenye Emezue, Bonaventure F. P. Dossou, Blessing Sibanda, Blessing Basse, Ayodele Olabiyi, Arshath Ramkilowan, Alp Öktem, Adewale Akinfaderin, and Abdallah Bashir. 2020. [Participatory research for low-resourced machine translation: A case study in African languages](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2144–2160, Online. Association for Computational Linguistics.
- Laura Perez-Beltrachini and Claire Gardent. 2017. [Analysing data-to-text generation benchmarks](#). In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 238–242, Santiago de Compostela, Spain. Association for Computational Linguistics.
- Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. [CoQA: A Conversational Question Answering Challenge](#). *Transactions of the Association for Computational Linguistics*, 7:249–266.
- Marina Sokolova and Stan Matwin. 2016. Personal privacy protection in time of big data. *Challenges in computational statistics and data mining*, pages 365–380.
- Manolis Terrovitis, John Liagouris, Nikos Mamoulis, and Spiros Skiadopoulos. 2012. Privacy preservation by disassociation. *arXiv preprint arXiv:1207.0135*.
- Tatiana Tommasi, Novi Patricia, Barbara Caputo, and Tinne Tuytelaars. 2017. [A Deeper Look at Dataset Bias](#), pages 37–55. Springer International Publishing, Cham.
- Bin Wu. 2014. [Embedding research in local contexts: local knowledge, stakeholders’ participation and fieldwork design](#). *Field Methods Research Lab Blog*.
- Zequ Wu, Michel Galley, Chris Brockett, Yizhe Zhang, Xiang Gao, Chris Quirk, Rik Koncel-Kedziorski,

Jianfeng Gao, Hannaneh Hajishirzi, Mari Ostendorf, et al. 2021. [A controllable model of grounded response generation](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14085–14093.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. [mT5: A massively multilingual pre-trained text-to-text transformer](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

## A Example task setup

The roles of QUESTIONER and RESPONDER are designed to converse about the given context. Figure 2 is a screenshot of the developed interface in usage to collect the desired data in Gaelic language. The context here is both image and text. An example set up for text based on Wikipedia is demonstrated in Figures 3 (textual context) and 4 (conversation between the QUESTIONER and RESPONDER).

Figure 3 illustrates how information is presented to the QUESTIONER and RESPONDER in textual mode. The RESPONDER has access to the document and its sub-topics (left), while the QUESTIONER is provided with keywords and brief descriptions for each of them (right). The QUESTIONER’s keywords are accompanied by the first sentence of the corresponding Wikipedia page to provide more context but does not have the details contextualized with the main topic at hand. This creates an asymmetry in the amount of information available to the two parties.

The QUESTIONER and RESPONDER discuss the document after studying the information provided to them. An example conversation is shown in Figure 4. The questioner begins the conversation by asking about the keywords starting in the first sub-topic. The order of the keywords within the sub-topic can be randomized but the order of sub-topic itself cannot be changed. This is to increase diversity in the data without affecting the inherent flow of the whole topic. The role of contextualization is further enhanced here as in this example, asking for the birthplace is not natural when we look just at the place, however, in the context of the sub-topic of early life, it is possible to guess that the context is about a person and hence the questioner asks about the place of birth. The conversation includes factual and descriptive answers. At the end, the questioner summarizes the sub-topic using the

information gathered from the conversation. **Data validation can be performed at any stage by an additional annotator.**

This framework offers several advantages such as producing trustworthy and grounded responses, learning surface form style differences, generating multi-sentence long-form responses, and extensibility to multilingual and cross-lingual scenarios with multilingual data. Finally, this framework also offers considerable flexibility, as it can be adapted based on available technological and linguistic resources (cf. Sec. 3). While the example we have provided here uses text as context, other approaches to grounding participant responses fit within this framework as well. For example, in addition to the keywords and source text seen by the QUESTIONER and RESPONDER, respectively, they can be presented with an image relating to the topic to make the topic more concrete and provide some shared context in addition to the individual materials they have available. Although having the advantage of gathering dual task annotations within the same session is advantageous, this framework also has some limitations that can be improved in future iterations. First, the keywords are only approximately cover the content. In the future, the plan is to explore metrics that evaluate summaries around only the keywords. Second, identifying keywords in multimodal contexts is more complex than in textual contexts.



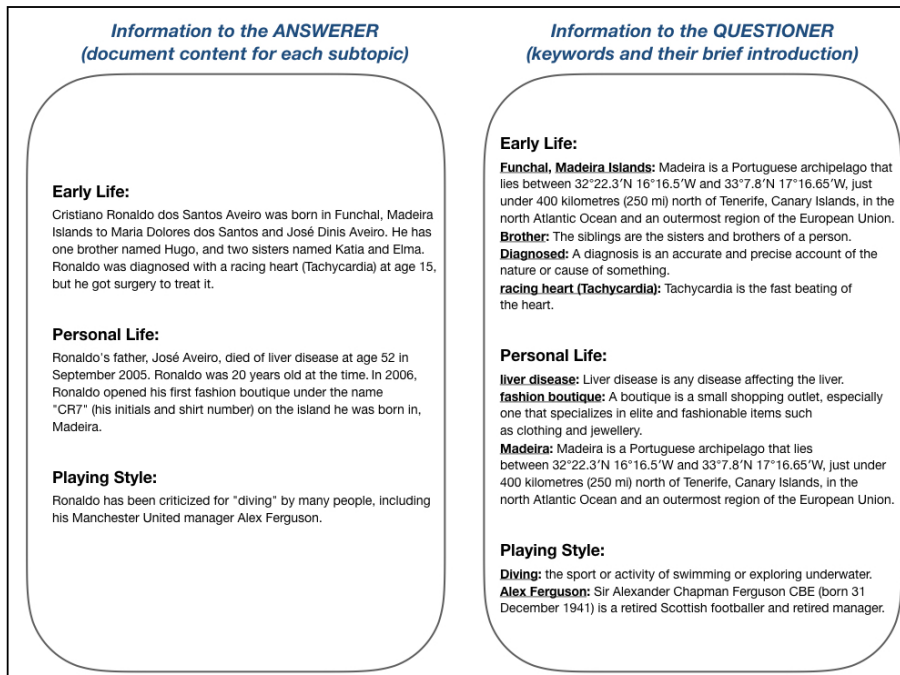


Figure 3: The information provided to the Questioner and the Answerer

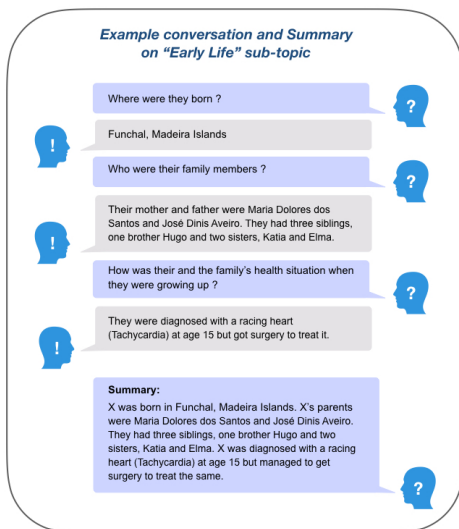


Figure 4: The conversation between the Questioner and the Answerer on an example sub-topic along with the summary.

# Long Story Generation Challenge

Nikolay Mikhaylovskiy

Higher IT School, Tomsk State University, Tomsk, Russia, 634050

NTR Labs, Moscow, Russia, 129594

`nickm@ntr.ai`

## Abstract

We propose a shared task of human-like long story generation, *LSG Challenge*, that asks models to output a consistent human-like long story (a Harry Potter generic audience fanfic in English), given a prompt of about 1K tokens. We suggest a novel statistical metric of the text structuredness, GloVe Autocorrelations Power/Exponential Law Mean Absolute Percentage Error Ratio (GAPELMAPER) and the use of previously-known UNION metric and a human evaluation protocol. We hope that *LSG Challenge* can open new avenues for researchers to investigate sampling approaches, prompting strategies, autoregressive and non-autoregressive text generation architectures and break the barrier to generate consistent long (40K+ word) texts.

## 1 Task Overview

The human-like long story generation (*LSG*) task asks models to output a consistent human-like long story (a Harry Potter generic audience fanfic in English), given a prompt of about 1K tokens. The text will be evaluated by automated metrics described in Section 3.1, and a human evaluation protocol described in Section 3.2.

## 2 Motivation

Autoregressive probabilistic large language models (LLMs) have become a cornerstone for solving every task in computational linguistics through few-shot learning (Brown et al., 2020) or prompt engineering (Sahn et al., 2021). Many users now interact with such models as ChatGPT, Claude, or Google Bard in chat setting regularly. However, these models still have many deficiencies. Despite the targeted effort, they can

generate false information, propagate social stereotypes, and produce toxic language (Taori et al., 2023).

The LLM deficiency we particularly want to attack is their inability to produce a human-grade long text. Current autoregressive language models fail to catch long-range dependencies in the text consistently. Large language models such as GPT-3 (Brown et al., 2020), LLaMA (Touvron et al., 2023), ALPACA (Taori et al., 2023) push the boundary of “short text” rather far, but do not remove the problem. Commercial instruction-following language models such as ChatGPT, GPT-4, Claude and Google Bard are targeted at the use in a dialogue (and probably that is not for nothing). They generate a limited number of tokens per user input, and only generate further text after additional prompting. While the autoregressive window for commercial models at the time of writing reaches 32K tokens for OpenAI and even 100K tokens for Anthropic, which is a lot, it does not allow them to generate long coherent texts.

While relevance, consistency, fluency and coherence are easily achieved by the latest autoregressive generative models on short texts (under 10K tokens), all the current models fail when one tries to generate a long story in a single pass. Modeling long stories requires many additional abilities compared to short texts (Guan et al., 2022), including (1) commonsense reasoning regarding characters’ reaction and intention, and knowledge about physical objects (e.g., “river”) and abstract concepts (e.g., “irony”); (2) modeling discourse-level features such as inter-sentence relations (e.g., causality) and global discourse structures (e.g., the order of events); and (3) the generation coherence and controllability, which require both maintaining a coherent plot and adhering to controllable attributes (e.g., topics).

Mikhaylovskiy and Churilov (2023) have recently studied autocorrelations in long texts using pretrained word vectors. That allowed to study a wide range of autocorrelation distances in human-written and model-generated texts and show that the autocorrelations in human-written literary texts decay according to power laws on distances from 10 to 10K words independently from the language. On the other hand, the behavior of autocorrelations decay in generated texts is quantitatively and often qualitatively different from the literary texts. Large language models often exhibit Markovian (Markov, 1913) behavior with exponential autocorrelations decay.

Several authors have shown theoretically and empirically (Lin and Tegmark, 2017, Alvarez-Lacalle et al., 2006) that the power law autocorrelations decay is closely connected to the hierarchical structure of texts. Indeed, the hierarchical structure of, for example, Leo Tolstoy’s War and Peace consists of at least 7 levels: the whole novel, books, parts, chapters, paragraphs, words, and letters. There are strong reasons to think that this structure reflects an important aspect of human thinking: people do not generate texts autoregressively. Writing a long text requires some thinking ahead, and going back to edit previous parts for consistency. This going back and forth can be reflected by navigating a tree-like structure. The autoregressive nature of the current state-of-the-art models does not reflect this, for example, S4 model (Gu et al., 2021) exhibits clear exponential autocorrelations decay (Mikhaylovskiy and Churilov, 2023).

We hope that this challenge can gain interest from the NLG community and advance sampling approaches, prompting strategies, autoregressive and non-autoregressive text generation architectures and other subfields of text generation.

### 3 Task Description

Formally, the task of LSG Challenge asks participants to provide a system that can output a consistent human-like long story (a Harry Potter generic audience fanfic at least 40K words long), given a prompt of about 1K tokens. A set of at least three dev prompts will be provided by organizers. The systems will be evaluated on a withheld test prompt. The prompts similar to the beginnings of human-written fan fiction will be developed from scratch specifically for the task.

	Power law MAPE	Exp law MAPE	GAPEL-MAPER
The Adventures of Tom Sawyer	0.21	0.55	0.38
The Republic	0.13	0.38	0.34
Don Quixote	0.20	0.44	0.45
War and Peace	0.09	0.42	0.21
Critique of Pure Reason	0.14	0.25	0.56
The Iliad	0.19	0.54	0.35
Moby-Dick or, The Whale	0.15	0.47	0.32
S4 generated text	0.062	0.050	1.24

Table 1: MAPE of power and exp law approximations of texts in English, and resulting GAPEL-MAPER

It is important to note that no copyright-eligible texts will be used in the shared task. The evaluation protocol below does not require using the original Harry Potter texts, and subjective evaluation relies on the fact that judges have read Harry Potter books/seen the films, but no factual knowledge of Harry Potter books is also required for the evaluation criteria below.

Given the open-ended and cutting-edge nature of the generation task and ongoing discussion on the best corpora and approaches to training LLMs, we feel that constraining the training set can be harmful to the task performance and participants are open to train their models on any dataset, as long as it is described in the system report.

We employ both automatic and human evaluation, described below to evaluate the quality of the texts.

#### 3.1 GloVe Autocorrelations Power/Exponential Law Mean Absolute Percentage Error Ratio (GAPEL-MAPER) Metric

Suppose we have a sequence of  $N$  vectors  $V_i \in R^d, i \in [1, N]$ . Autocorrelation function  $C(\tau)$  is the average similarity between the vectors as a function of the lag  $\tau = i - j$  between them. The simplest metric of vector similarity is the cosine distance  $d(V_i, V_j) = \cos \angle(V_i, V_j) = \frac{V_i \cdot V_j}{\|V_i\| \|V_j\|}$ , where  $\cdot$  is a dot product between two vectors and  $\| \cdot \|$  is an Euclidean norm of a vector. Thus,

$$C(\tau) = \frac{1}{N - \tau} \sum_{i=1}^{N-\tau} \frac{V_i \cdot V_{i+\tau}}{\|V_i\| \|V_{i+\tau}\|}. \quad (5)$$

$C(\tau)$  ranges from  $-1$  for perfectly anticorrelated sequence (for  $\tau = 1$  and  $d = 1$  that would be  $1, -1, 1, -1$  etc.) to  $1$  for a perfectly correlated one (for  $\tau = 1$  and  $d = 1$  that would be  $1, 1, 1, 1$  etc.).

A distributional semantic assigns a vector to each word or context in a text. Thus, a text is transformed into a sequence of vectors, and we can calculate an autocorrelation function for the text. Two distributional semantics approaches have been proposed for word-level autocorrelation computations: Alvarez-Lacalle et al. (2006) proposed a bag-of-words (BOW) model, and Mikhaylovskiy and Churilov (2023) have suggested the use of pretrained GloVe (Pennington et al., 2014) vectors. Unlike BOW, which only allows measuring long distance correlations, the latter approach allows to measure autocorrelations at any word distance starting with 1. Thus, we suggest using GloVe for autocorrelation measurement.

Mikhaylovskiy and Churilov (2023) have found that autocorrelations in long human-written texts decay according to a power law at ranges from 10 to 10K words. We suggest measuring the structuredness of a generated text by comparing how well the autocorrelations decay is approximated by power law and exponential law. To do so, one can compute autocorrelations in this range, approximate these points by a straight line in log-log and log-linear coordinates using the least squares regression and evaluate the goodness of fit of these regressions by MAPE (Mean Absolute Percentage Error). The ratio of these two errors constitute a metric we call GloVe Autocorrelations Power/Exponential Law Mean Absolute Percentage Error Ratio (GAPELMAPER):

$$\text{GAPELMAPER} = \frac{\text{MAPE}_{power}}{\text{MAPE}_{exp}}$$

GAPELMAPER less than 1 means that the autocorrelations decay according to a power law and the text is structured in a way. GAPELMAPER more than 1 means that the autocorrelations decay according to an exponential law and the text is unstructured. As a matter of example, we take Table 3 from Mikhaylovskiy and Churilov (2023) and compute GAPELMAPER in Table 1.

The metric proposed above does not require any gold standard, it is a statistical metric of the text

itself. Thus, in terms of Guan and Huang (2020) it is an unreferenced metric.

### 3.2 UNION Metric

UNION is an unreferenced metric for evaluating open-ended story generation, proposed by Guan and Huang (2020). Built on top of BERT, UNION is trained to distinguish human-written stories from negative samples. The negative samples are programmatically constructed using Repetition, Substitution, Reordering and Negation Alteration.

### 3.3 Human Evaluation Approach

A single number is not enough to evaluate the quality of a long story. We adopt multiple human evaluation metrics to better measure model performance. Similarly to Kryscinski et al. (2019), we ask annotators to rate the texts across four dimensions:

1. relevance (of topics in the text to the expected ones),
2. consistency (alignment between the parts of the text),
3. fluency (quality of individual sentences), and
4. coherence (quality of sequence of sentences).

Additionally, extending (Guan et al., 2022), we ask annotators to rate

5. knowledge about physical objects (LLM generated failure example: “I was on shore in a boat; but I was not in the water. I was not in the water. I was in the water.”)
6. knowledge about abstract concepts (LLM generated failure example: “The twenty-eighth one is a twenty-eighth one. The twenty-nineteenth one is a twenty-eighth one. The twenty-ninth one is a twenty-ninth one. The twenty-tenth one is a twenty-tenth one.”)
7. causality (LLM generated failure example: “The first part was pretty easy. The second one, on the other hand, took a lot of practice. I had a lot of difficulty with the first one.”)
8. the order of events (LLM generated failure example: “This is the way all voyages of travel are done in all ages of the earth; they come to it and lay it down in the same fashion: — They get a wind, sail about awhile, and

gather what stores are sufficient for a week, or for one night’s stay.”)

Finally, extending Guan and Huang (2020) we ask annotators to rate

#### 9. repeated plots (repeating similar texts)

A detailed evaluation manual will be developed as a part of the competition preparation and provided to judges, including a checklist conforming to suggestions of Howcroft et al., (2020).

Each text will be rated by 3 distinct judges with the final score obtained by averaging the individual scores. We plan to hire linguistics/philology students with English knowledge level at least C1 as the judges in at least two low-cost countries. Where possible, the judge assignment will be included into coursework. Small non-government/donation funding will be made available to cover judging expenses where the above approach is not possible.

### 3.4 Protocol

We propose the following schedule:

- **Phase 1** (from Sep, 2023): The shared task is announced at the INLG 2023 conference, and the data are available on the shared task website; participants can register to the task.
- **Phase 2** (from Dec, 2023): The leaderboard is open; participants can submit their systems to the organizers and the online leaderboard keeps updating the best performance using automatic evaluation metrics.
- **Phase 3** (from Mar, 2024): The submission is closed; organizers conduct manual evaluation.
- **Phase 4** (Jul, 2024): The LSG Challenge shared task is fully completed. Organizers submit participant reports and challenge reports to INLG 2024 and present at the conference.

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.

Please note that the above schedule can be modified accordingly when the schedule of INLG 2024 is released. The leaderboard and the detailed schedule will be announced on the shared task website.

## 4 Related work

Shaham et al. (2022) introduced SCROLLS, a suite of tasks that require reasoning over long texts. It includes earlier introduced works of Huang et al. (2021), Chen et al. (2022), Zhong et al. (2021), Dasigi et al. (2021), Kočiský et al. (2018), Pang et al. (2022), and Koreeda and Manning (2021). While all are related to long texts, none of these datasets and tasks asks to generate a long text.

Gehrmann et al. (2021) introduced GEM, a living benchmark for natural language Generation (NLG), its Evaluation, and Metrics. GEM provides an environment in which models can easily be applied to a wide set of tasks and in which evaluation strategies can be tested and consists of 11 datasets/tasks. Tay et al. (2020) proposed Long Range Arena, a suite of tasks consisting of sequences ranging from 1K to 16K tokens, encompassing a wide range of data types and modalities such as text, natural, synthetic images, and mathematical expressions requiring similarity, structural, and visual-spatial reasoning. None of these tasks asks to generate a long text as well.

Very recently Köksal et al. (2023) introduced the LongForm dataset, which is created by leveraging English corpus examples with augmented instructions. No evaluation protocol or competition is suggested in the cited paper.

On the unreferenced metrics front, Guan and Huang (2020) proposed UNION metric described in Section 3.2. Huang et al. (2020) proposed a metric dubbed GRADE, which stands for Graph-enhanced Representations for Automatic Dialogue Evaluation. Gao, Zhao, and Eger (2020) suggested SUPERT, which rates the quality of a summary by measuring its semantic similarity with a pseudo reference summary. Vasilyev, Dharnidharka, and Bohannon (2020) suggested BLANC that measures the performance boost gained by a pre-trained language model with access to a document summary while carrying out its language understanding task on the document’s text.

The most similar effort to ours was most likely made by Guan et al. (2022), who proposed a story-centric benchmark named LOT for evaluating Chinese long text modeling. The benchmark aggregates two understanding tasks and two generation tasks. The authors constructed new datasets for these tasks based on human-written Chinese stories. Unlike our proposal, LOT

benchmark is limited to texts hundreds of words long, and Chinese language.

## 5 Conclusion

We propose the LSG Challenge to address the task of long text generation, with the hope that it can open new avenues for researchers to investigate sampling approaches, prompting strategies, autoregressive and non-autoregressive text generation architectures and break the barrier to generate consistent long (40K+ token) texts, and the frontier of text generation can be pushed further.

## Acknowledgments

Thanks to Simon Mille and anonymous referees for the useful comments that helped to improve this paper.

## References

- Enric Alvarez-Lacalle, Beate Dorow, Jean-Pierre Eckmann, and Elisha Moses. 2006. Hierarchical structures induce long-range dynamical correlations in written texts. *PNAS*, 103(21):7956–7961.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volumes 2020–Decem, pages 1877–1901.
- Mingda Chen, Zewei Chu, Sam Wiseman, and Kevin Gimpel. 2022. SummScreen: A Dataset for Abstractive Screenplay Summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8602–8615, Dublin, Ireland. Association for Computational Linguistics.
- Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A Dataset of Information-Seeking Questions and Answers Anchored in Research Papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610, Online. Association for Computational Linguistics.
- Yang Gao, Wei Zhao, and Steffen Eger. 2020. SUPERT: Towards new frontiers in unsupervised evaluation metrics for multi-document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1347–1354, Online. Association for Computational Linguistics.
- Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Anuoluwapo Aremu, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh Dhole, Wanyu Du, Esin Durmus, Ondřej Dušek, Chris Chinenye Emezue, Varun Gangal, Cristina Garbacea, Tatsunori Hashimoto, Yufang Hou, Yacine Jernite, et al. 2021. The GEM Benchmark: Natural Language Generation, its Evaluation and Metrics. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 96–120, Online. Association for Computational Linguistics.
- Albert Gu, Karan Goel, and Christopher Ré. 2021. Efficiently Modeling Long Sequences with Structured State Spaces. *International Conference on Learning Representations*. 2021:1–32.
- Jian Guan and Minlie Huang. 2020. UNION: an unreferenced metric for evaluating open-ended story generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*, Online, November 16-20, 2020, pages 9157–9166. Association for Computational Linguistics.
- Jian Guan, Zhuoer Feng, Yamei Chen, Ruilin He, Xiaoxi Mao, Changjie Fan, and Minlie Huang. 2022. LOT: A Story-Centric Benchmark for Evaluating Chinese Long Text Understanding and Generation. *Transactions of the Association for Computational Linguistics*, 10:434–451.
- 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.
- Lishan Huang, Zheng Ye, Jinghui Qin, Liang Lin, and Xiaodan Liang. 2020. GRADE: Automatic Graph-Enhanced Coherence Metric for Evaluating Open-Domain Dialogue Systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9230–9240, Online. 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.
- Henry W. Lin and Max Tegmark. 2017. Critical behavior in physics and probabilistic formal languages. *Entropy*, 19(7):1–25.
- Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA Reading Comprehension Challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.
- Yuta Koreeda and Christopher Manning. 2021. ContractNLI: A Dataset for Document-level Natural Language Inference for Contracts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1907–1919, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Abdullatif Köksal, Timo Schick, Anna Korhonen and Hinrich Schütze. LongForm: Optimizing Instruction Tuning for Long Text Generation with Corpus Extraction. *ArXiv abs/2304.08460* (2023)
- 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.
- Andrei Markov, 1913. An Example of Statistical Investigation of the Text Eugene Onegin Concerning the Connection of Samples in Chains. *Science in Context*. 2006. Vol. 19, no. 4. pages 591–600. DOI 10.1017/S0269889706001074.
- Nikolay Mikhaylovskiy and Ilya Churilov, 2023. Autocorrelations Decay in Texts and Applicability Limits of Language Models. *Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference “Dialogue 2023”*
- Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and Samuel Bowman. 2022. QuALITY: Question Answering with Long Input Texts, Yes!. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5336–5358, Seattle, United States. Association for Computational Linguistics.
- Jeffrey Pennington, Richard Socher, and Christopher 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, Doha, Qatar. Association for Computational Linguistics.
- Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, et al. 2021. Multitask Prompted Training Enables Zero-Shot Task Generalization. *ICLR*.
- Uri Shaham, Elad Segal, Maor Ivgi, Avia Efrat, Ori Yoran, Adi Haviv, Ankit Gupta, Wenhan Xiong, Mor Geva, Jonathan Berant, and Omer Levy. 2022. SCROLLS: Standardized CompaRison Over Long Language Sequences. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 12007–12021, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford Alpaca: An Instruction-following LLaMA Model. [https://github.com/tatsu-lab/stanford\\_alpaca](https://github.com/tatsu-lab/stanford_alpaca), 2023.
- Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. 2020. Long Range Arena: A benchmark for efficient transformers. In *International Conference on Learning Representations*.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models.
- Oleg Vasilyev, Vedant Dharnidharka, and John Bohannon. 2020. Fill in the BLANC: Human-free quality estimation of document summaries. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pages 11–20, Online. Association for Computational Linguistics.
- Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khachabi, and Hannaneh Hajishirzi. 2022. Self-Instruct: Aligning Language Model with Self Generated Instructions. <https://arxiv.org/abs/2212.10560>
- Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah,

Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021. 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.



# Visually Grounded Story Generation Challenge

Xudong Hong<sup>1,2,4</sup>, Asad Sayeed<sup>3</sup>, Khushboo Mehra<sup>2,4</sup>, Vera Demberg<sup>2,4</sup> and Bernt Schiele<sup>1,4</sup>

<sup>1</sup>Dept. of Computer Vision and Machine Learning, MPI Informatics

<sup>2</sup>Dept. of Language Science and Technology and Dept. of Computer Science, Saarland University

<sup>3</sup>Dept. of Philosophy, Linguistics, and Theory of Science, University of Gothenburg

<sup>4</sup>Saarland Informatics Campus, Saarbrücken

{xhong, kmehra, vera}@lst.uni-saarland.de  
schiele@mpi-inf.mpg.de, asad.sayeed@gu.se

## Abstract

Recent large pre-trained vision-and-language models have achieved strong performance in natural language generation. However, most previous generation tasks neither require coherent output with multiple sentences nor control the output text by grounding the output in the input. We propose a shared task on visually grounded story generation, where the input is an image sequence, and the output is a story that is conditioned on the input images. This task is particularly challenging because: 1) the output story should be a narratively coherent text with multiple sentences, and 2) the protagonists in the generated stories need to be grounded in the images. We aim to advance the study of vision-based story generation by accepting submissions that propose new methods.

## 1 Introduction

Vision-based language generation (VLG) is to generate text from visual input. It is a challenging but interesting task because it requires joint vision and language modeling. Recent large pre-trained vision-and-language models (VLMs) like GPT-4 (OpenAI, 2023) or MiniGPT-4 (Zhu et al., 2023) have shown great success on several multimodal tasks, such as image captioning (Vinyals et al., 2016), visual question answering (Goyal et al., 2017) and visual dialog generation (Das et al., 2017).

Despite recent breakthroughs, current tasks only require models to predict a label or generate short texts (i.e., less than 30 words). It is unclear whether the newest VLMs can generate coherent texts with multiple sentences from visual input. On the contrary, humans can produce long and locally coherent texts from the same visual input. To investigate machine intelligence, we need a task that is more similar to human behavior (Bubeck et al., 2023).

Several previous tasks have been proposed to test the capabilities of VLMs to handle longer

output, such as visual paragraphs (Krause et al., 2017), localized narratives (Pont-Tuset et al., 2020), and video captioning (Voigtlaender et al., 2023). However, these tasks are designed for literal descriptions where sentences are independent of each other, rather than for coherent text. Coherence is a fundamental property of human language. In particular, local coherence, which refers to the relations between entities in context, affects language comprehension and production. Local coherence is essential for vision and language (V&L) research because: **1.** It has many applications in vision and language tasks. For example, a better model of local coherence can improve the performance of text-to-image retrieval (Park and Kim, 2015). **2.** Modeling coherence is a prerequisite for modeling event knowledge as events center around entities. Better event modeling improves vision and language pre-training (Zellers et al., 2021, 2022).

Story generation is a well-studied task in natural language generation, widely used for testing whether large pretrained models can track entities (Paperno et al., 2016) and generate locally coherent texts. Unlike image captions, stories contain several characters and events involving recurrent characters and their interactions with each other and the environment. In addition, *characters* and *relevant content* are among the most critical aspects of story writing (Goldfarb-Tarrant et al., 2020). We argue that story generation is a suitable benchmark for testing whether VLMs can generate coherent texts.

In this work, we propose a new shared task, Visually Grounded Story Generation (VGSG), which requires the VLMs to generate stories with protagonists grounded on images. We aim for coherent and visually grounded stories with high diversity. This task is particularly challenging for two reasons: **1.** The protagonists in the generated stories need to be grounded in the images, meaning that their actions and descriptions should be consistent with the

## Visual Writing Prompts (Ours)

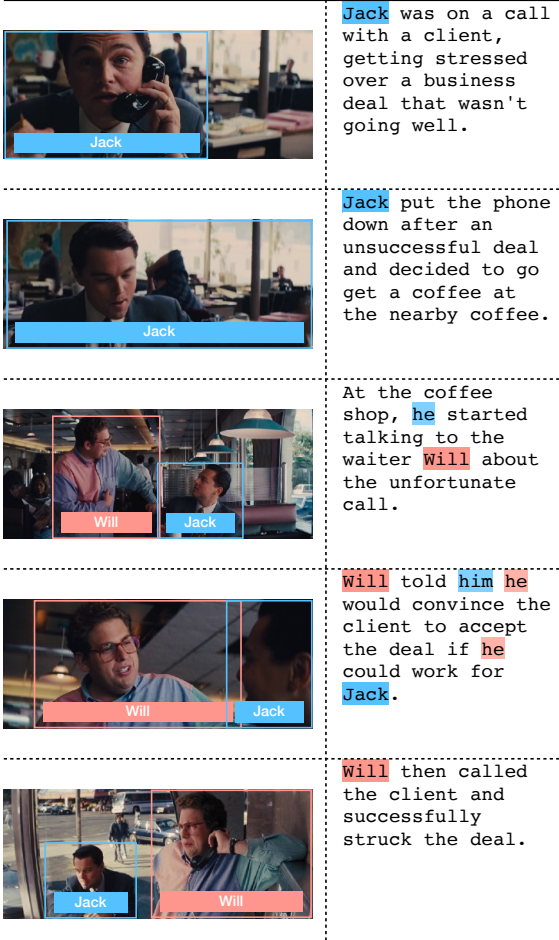


Figure 1: Example of Visual Grounded Story Generation on Visual Writing Prompts dataset. The dataset has recurring characters across all five images and sub-stories. Each occurrence of a character in a sub-story has a bounding box in the corresponding image, which grounds the textual appearance to visual input.

visual information provided. **2.** The output story needs to be a coherent text, meaning that it should have a clear beginning, middle, and end, and flow logically from one sentence to the next.

We hope that this task will help the exploration of VLG by encouraging participants to propose new methods that generate coherent and visually grounded stories. We welcome submissions from researchers around the world who are interested in tackling this exciting challenge. We also seek for researchers who are interested to join the organization of this shared task.

## 2 Related Work

**VLG with Coherence.** One relevant task is Visual Storytelling (Huang et al., 2016), where the

input is a sequence of images and the output is a coherent story. Another task that requires some sort of coherence in the generated text is movie description (Rohrbach et al., 2015), where the input is a video clip from the movie and the output is the corresponding text description of the scene. Chandu et al. (2019) propose a dataset of procedural text from recipes with instructional images, but characters are not explicitly annotated. Unfortunately, the local coherence of the generated text is not evaluated in either of these tasks (Mitchell et al., 2018).

**Visual Story Generation.** Most of the previous tasks for visual story generation have several limitations: there is no sequence of events behind the images (Park and Kim, 2015; Huang et al., 2016) or the dataset is limited in scale (Xiong et al., 2019). None of them can be used for evaluating visual grounding. Mitchell et al. (2018) hosted the first shared task of visual story generation. But there are no automatic evaluations of either coherence or visual grounding. Our shared task is the first to jointly evaluate the coherence and visual grounding of generated stories.

## 3 Task Description

We define the VGSG task as follows: given a sequence of images (like the first column of Figure 1) the system needs to generate a coherent short story conditioned on the image sequence (like the second column of Figure 1). In addition, the generated story should contain the characters seen in the image sequence.

The VGSG shared task focuses on coherent and visually grounded stories with high diversity.

### 3.1 Datasets

To evaluate the submissions, we will use two datasets that provide grounding annotations for characters:

**Visual Writing Prompts** (VWP; Hong et al., 2023b), a vision-based dataset that contains 2K image sequences aligned with 12K human-written stories in English.<sup>1</sup> Each image is corresponding to a part of a story. Instances of each protagonist are annotated with the character’s name (see Figure 1).

**VIST-Character** by Liu and Keller (2023) which has visual and textual annotations for recurring characters in 770 stories from the test split of the

<sup>1</sup><https://vwprompt.github.io/>

Name	Image Genre	Story Genre	Story Source	# Story	# image per Story	# token per Story
VWP	movie	short story	crowdworker	12 K	[5, 10]	83.7
VIST	photo	short story	crowdworker	50 K	5	57.6
Travel blogs	photo	blog	blogger	10 K	1	222.3‡
MSA	movie	movie synopsis	fan	5 K	92	129

Table 1: Statistics of datasets. Numbers with ‡ are obtained from a small sample of the Disney split of the dataset that is available in their repository.

VIST dataset (Huang et al., 2016), along with an importance rating of all characters in any story.<sup>2</sup> We only use it for evaluation.

We also evaluate on these datasets:

**Visual Storytelling** (VIST; Huang et al., 2016) is a widely used dataset with 50K image-story pairs.

**Travel blogs** (TB; Park and Kim, 2015) are two datasets with 10K image sequence-story pairs extracted from travel blogs of visiting New York City or Disneyland.

**Movie Synopses Associations** (MSA; Xiong et al., 2019) contains movie synopses from 327 movies where there are 4494 scenes aligned with corresponding paragraphs in synopses.

These data sets are publicly available so there’s a risk of exposure to the participants. To ensure a fair comparison and make the task more challenging, we collect additional data following the data collection process of these works combine with selected subsets as blind test sets. The statistics of all the datasets are in Table 1.

### 3.2 Tracks

The VGSG shared task contains three tracks: **Strict Track** focuses on exploring Language and Vision Mapping methods and Language Generation models through a controlled experiment. We provide extracted visual features from a pre-trained vision model, which participants can only use as input to train their models with the provided dataset.

**Open Track** aims to test the state-of-the-art of the task. Participants can use all kinds of resources, including pre-trained models and additional text or vision-only datasets. However, they cannot use other vision and language datasets apart from the provided dataset.

**Grounding Track** is based on the Open Track, but participants are required to submit a mapping

<sup>2</sup><https://github.com/iz2late/VIST-Character>

of all entities in the generated text and provided characters (see Figure 2 for an example). The submissions to this track will be evaluated on the VIST-Character dataset (Liu and Keller, 2023).

### 3.3 Schedule

We propose the following tentative schedule:

**Dec 1st, 2023** We will announce the joint task at the INLG 2023 conference (if accepted), with data available on the task’s dedicated website. This is the point when individuals can sign up for the task.

**Feb 1st, 2024** The submission is opened. Participants can submit their systems to the organizers.

**May 1st, 2024** Submission ends at this point and organizers start the process of automatic evaluation on blind test sets and human evaluation of the systems.

**Jun 1st, 2024** The VGSG shared task comes to a conclusion. The organizers will submit reports regarding participant performance and overall challenge outcomes to the INLG 2024 conference and will present these findings at the event. The previously concealed test set will be released to the public.

	Jack	Will
Jack was on a call with a client, getting stressed over a business deal that wasn't going well.	1	-1
Jack put the phone down after an unsuccessful deal and decided to go get a coffee at the nearby coffee.	1	-1
At the coffee shop, he started talking to the waiter Will about the unfortunate call.	1	1
Will told him he would convince the client to accept the deal if he could work for Jack.	1	1
Will then called the client and successfully struck the deal.	-1	1

Figure 2: Example a matching matrix between entities in the generated story and the character in the images.

## 4 Evaluation

We will perform both automatic and human evaluations for the submissions. The scripts for all automatic metrics will be provided after the submission system is open; human evaluation will be conducted after all submissions have been received. We will release the annotator instructions and source code of all metrics after the shared task.

### 4.1 Automatic Evaluation

We will use metrics in the following categories to evaluate the submissions:

**Reference-based metrics** including unigram (B-1), bigram (B-2), trigram (B-3), and 4-gram (B-4) BLEU scores (B; Papineni et al., 2002), METEOR (M; Banerjee and Lavie, 2005), ROUGE-L (R; Lin, 2004), and CIDEr (C; Vedantam et al., 2015), which were used in the previous visual storytelling shared task (Mitchell et al., 2018). We will also use BERTScore (BS; Zhang\* et al., 2020) which is effective in text summarization.

**Grounding** To measure the correctness of referring expressions of human characters in stories, we will use the character-matching (CM) metric defined in (Hong et al., 2023a).

**Event diversity** we will use metrics used by Hong et al., 2023b (based on (Goldfarb-Tarrant et al., 2020)) including the unique number of verbs, verb-vocabulary ratio, verb-token ratio, percentage of diverse verbs not in the top-5 most frequent verbs and unique:total ratios of predicate unigram, bigram, and trigram.

**Coherence** following Hong et al., 2023b we will use the generative Entity Grid model to calculate the log-likelihood based on entity transitions in system outputs.

### 4.2 Human Evaluation

In natural language generation tasks, automatic metrics do not provide a full understanding of the quality of the generated text. Reference-based metrics, in particular, have been shown to not correlate well with human judgment. In addition, several important aspects of narratives such as creativity, and logical coherence are hard to judge using automatic evaluation. Therefore, we will also conduct a human evaluation for the submissions, focussed on narrativity (whether the generation is a story or simply a description of images), character grounding (correctness of referring expressions, model

hallucinations), and coherence. The scale of the evaluation depends on the funding we have. We also encourage participants to perform their own human evaluation and include the results in their reports.

### 4.3 Baselines

Our baselines are:

**Seq2Seq** (Huang et al., 2016) is a simple but powerful model with an encoder-decoder architecture. Visual features are first projected with an encoder which is a feed-forward neural network, then fed to the decoder which is a pre-trained language model.

**TAPM** (Yu et al., 2021) is a Transformer-based model which adapts the visual features with pre-trained GPT-2.

**Other V&L models** We also include other vision and language models that are competitive on similar vision and language tasks like Cho et al. (VL-T5; 2021), Li et al. (BLIP; 2022) and Zhu et al. (MiniGPT-4; 2023).

## 5 Conclusions

This proposal introduces a novel shared task called Visually Grounded Story Generation, which necessitates that Visual Language Models formulate narratives with protagonists based on image inputs, ensuring the production of coherent and visually grounded stories with high diversity. The task poses dual challenges: the need for protagonists’ actions and descriptions to align with the provided visual information and the requirement for the output story to logically progress with a clear beginning, middle, and end. By initiating this task, the authors aim to foster advancements in Visual Language Generation, inviting global researchers to contribute new methodologies that facilitate the creation of visually consistent, logically structured stories.

## References

- Satanjeev Banerjee and Alon Lavie. 2005. **METEOR: An automatic metric for MT evaluation with improved correlation with human judgments**. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.
- Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar,

- Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.
- Khyathi Chandu, Eric Nyberg, and Alan W Black. 2019. [Storyboarding of recipes: Grounded contextual generation](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6040–6046, Florence, Italy. Association for Computational Linguistics.
- Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. 2021. Unifying vision-and-language tasks via text generation. In *International Conference on Machine Learning*, pages 1931–1942. PMLR.
- Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José M.F. Moura, Devi Parikh, and Dhruv Batra. 2017. Visual Dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Seraphina Goldfarb-Tarrant, Tuhin Chakrabarty, Ralph Weischedel, and Nanyun Peng. 2020. Content planning for neural story generation with aristotelian rescoring. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4319–4338.
- Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913.
- Xudong Hong, Vera Demberg, Asad Sayeed, Qiankun Zheng, and Bernt Schiele. 2023a. Visual coherence loss for coherent and visually grounded story generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, Toronto, Canada. Association for Computational Linguistics.
- Xudong Hong, Asad Sayeed, Khushboo Mehra, Vera Demberg, and Bernt Schiele. 2023b. [Visual writing prompts: Character-grounded story generation with curated image sequences](#). *Transactions of the Association for Computational Linguistics*, 11.
- Ting-Hao Kenneth Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Aishwarya Agrawal, Jacob Devlin, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, C. Lawrence Zitnick, Devi Parikh, Lucy Vanderwende, Michel Galley, and Margaret Mitchell. 2016. [Visual storytelling](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1233–1239, San Diego, California. Association for Computational Linguistics.
- Jonathan Krause, Justin Johnson, Ranjay Krishna, and Li Fei-Fei. 2017. A hierarchical approach for generating descriptive image paragraphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 317–325.
- Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR.
- 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.
- Danyang Liu and Frank Keller. 2023. Detecting and grounding important characters in visual stories. In *37th AAAI Conference on Artificial Intelligence*. AAAI Press.
- Margaret Mitchell, Ting-Hao ‘Kenneth’ Huang, Francis Ferraro, and Ishan Misra, editors. 2018. *Proceedings of the First Workshop on Storytelling*. Association for Computational Linguistics, New Orleans, Louisiana.
- OpenAI. 2023. [Gpt-4 technical report](#).
- Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. [The LAMBADA dataset: Word prediction requiring a broad discourse context](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534, Berlin, Germany. 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.
- Cesc C Park and Gunhee Kim. 2015. Expressing an image stream with a sequence of natural sentences. *Advances in neural information processing systems*, 28:73–81.
- Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. 2020. Connecting vision and language with localized narratives. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 647–664. Springer.
- Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. 2015. A dataset for movie description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3202–3212.
- Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4566–4575.

- Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2016. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):652–663.
- Paul Voigtlaender, Soravit Changpinyo, Jordi Pont-Tuset, Radu Soricut, and Vittorio Ferrari. 2023. Connecting vision and language with video localized narratives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2461–2471.
- Yu Xiong, Qingqiu Huang, Lingfeng Guo, Hang Zhou, Bolei Zhou, and Dahua Lin. 2019. A graph-based framework to bridge movies and synopses. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4592–4601.
- Youngjae Yu, Jiwan Chung, Heeseung Yun, Jongseok Kim, and Gunhee Kim. 2021. Transitional adaptation of pretrained models for visual storytelling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12658–12668.
- Rowan Zellers, Jiasen Lu, Ximing Lu, Youngjae Yu, Yanpeng Zhao, Mohammadreza Salehi, Aditya Kusupati, Jack Hessel, Ali Farhadi, and Yejin Choi. 2022. Merlot reserve: Neural script knowledge through vision and language and sound. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16375–16387.
- Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. 2021. Merlot: Multimodal neural script knowledge models. *Advances in Neural Information Processing Systems*, 34:23634–23651.
- Tianyi Zhang\*, Varsha Kishore\*, Felix Wu\*, Kilian Q. Weinberger, and Yoav Artzi. 2020. [Bertscore: Evaluating text generation with bert](#). In *International Conference on Learning Representations*.
- Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. [Minigt-4: Enhancing vision-language understanding with advanced large language models](#).

# The VDG Challenge: Response Generation and Evaluation in Collaborative Visual Dialogue

Nikolai Ilinykh and Simon Dobnik

Centre for Linguistic Theory and Studies in Probability (CLASP),  
Department of Philosophy, Linguistics and Theory of Science (FLoV),  
University of Gothenburg, Sweden  
{nikolai.ilinykh, simon.dobnik}@gu.se

## Abstract

We propose the VDG Challenge: a shared task that addresses and benchmarks the task of utterance generation in collaborative visual dialogue. The task features two challenging datasets, an evaluation protocol and a tentative schedule. Our shared task will allow researchers to unravel problems of modelling multi-modal interaction and fit of the existing approaches in the NLP and NLG communities.

## 1 Introduction

In the early 70s, the SHRDLU system (Winograd, 1971) was a revolutionary development. Many believed it had the ability to understand human language, as it was able to describe what it saw in an environment based on human queries. However, the illusion of intelligence of SHRDLU was dispelled as it became clear that the system did not know, for example, what a “box” is. Fast forward to today’s world, we have made significant progress with more advanced language models. For instance, models like ChatGPT (OpenAI, 2023) and various Transformer-based architectures for language understanding and generation (Devlin et al., 2019; Brown et al., 2020) have demonstrated the ability to understand language to some extent. Moreover, integrating language with other modalities has become essential in improving these models and making them more applicable to real-world scenarios (Bender and Koller, 2020). In fact, language-and-vision research has been making steps in this direction (Bernardi et al., 2016), as its aim is to build language systems that can map language with the world (Wittgenstein, 1953).

Despite the progress made, research efforts in multi-modal NLP have primarily concentrated on more specific tasks, such as referring expression generation (Krahmer and van Deemter, 2012), image description generation (Karpathy and Fei-Fei,

2017), and visual question answering (Antol et al., 2015). This focus is understandable because modelling human-human or human-world interaction is a challenging task, and there are several reasons for this. Firstly, human language use is highly dynamic, adaptive, and difficult to represent computationally. Successful communication relies on understanding the context, which can encompass textual and multi-modal information, as well as grasping the speaker’s intentions. Deciding what to say, how to say it, and when to say it are fundamental aspects of effective communication that require a nuanced understanding of language and context. Secondly, building a robust model requires high-quality dialogue data, which is challenging to gather and ensure that it possesses the properties observable in human language use. Simply put, there are multiple components involved in modelling human dialogue, and they must all be integrated harmoniously to create a truly effective conversational model.

Our proposal, the **Visual Dialogue Generation (VDG)** challenge, aims to create a platform that addresses the challenges in modelling multi-modal human-like situated dialogue (Clark et al., 1991). Specifically, our setup revolves around a collaborative visual dialogue, where two participants are placed in an environment with individual visual scenes and are asked to solve a specific task through language interaction. Within this setup, we focus on a particular task of **next utterance generation**, which is part of a broader communicative context. The primary goal of the challenge is to build and evaluate (neural) modelling proposals that can generate better responses given specific contexts. These contexts are defined as sets of previously generated utterances and visual scenes that collectively form a single language game. Each language game may serve a different purpose, such as describing, asking, or clarifying. Importantly, our

aim is not to build a conversational agent capable of holding a full-scale dialogue with a human. Instead, we narrow our focus to a single step: generating a response given a particular situation. By doing so, we can concentrate on examining the quality and value of the generated texts, which is important for building a model of conversation.

The challenge will use two datasets: the Cups (Dobnik et al., 2020) and MeetUp (Ilinykh et al., 2019), both of which are multi-modal and rich in various dialogue phenomena, such as clarification requests and turn-taking, crucial for a complete collaborative process (Clark and Wilkes-Gibbs, 1986). In addition, Cups corpus has data in two languages, English and Swedish. While there have been a few proposed visual dialogue models and datasets (Das et al., 2017; de Vries et al., 2017), they suffer from rigidity and a lack of many phenomena frequently observed in natural human dialogues. Our proposal aims to learn from better high-quality dialogue data, even though the datasets are relatively small in size. As a result, this challenge specifically focuses on transfer learning, learning from small data, and benchmarking the ability of existing generative models to generate responses in human-like multi-modal dialogues. An important feature of our data is that the dialogues were produced with specific (and different) tasks in mind, resulting in high-quality interactions. This raises questions about how much interactive knowledge is shared between different contexts and domains, and how much of it is specific to certain situations. Additionally, we aim to learn from the Natural Language Generation (NLG) community about the challenges and issues that arise when building generative multi-modal models, including biases, ethical concerns, and the naturalness of generated responses.

## 2 Datasets

Both Cups and MeetUp were collected in a task-oriented setting. In Cups (Figure 1), two participants were asked to locate missing cups on a table in a virtually generated scene. It is worth noting that the cups missing for each participant were not necessarily the same ones. These cups varied in colour, type, and location, and each participant could only see a subset of them from a different view. To communicate and identify each other’s missing cups, participants used the chat interface. Importantly, there were no restrictions on how the task should be approached, allowing participants

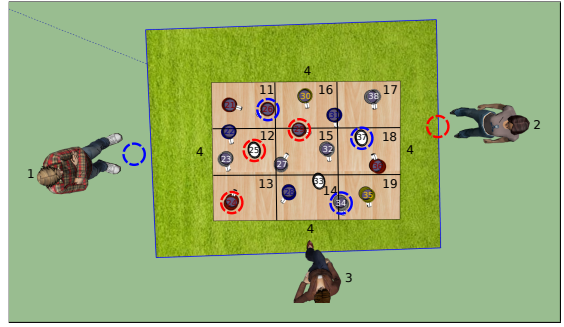


Figure 1: The Cups corpus: an allocentric view of the scene with annotated objects. Participants (labelled as “1” and “2”) cannot see objects marked with their colour (either red or blue). Katie (labelled as “3”) is a passive observer of the scene.

the freedom to choose their strategies. For additional scenes from the Cups corpus, refer to Appendix A.



Figure 2: The MeetUp game interface. A view for player A is presented. The image on the right side changes if the player moves to a different room.

In MeetUp (Figure 2), participants are randomly placed in a room, which is shown as a real image. This room is part of a larger house area, consisting of connected real images. Participants are informed that they can move around the house by executing specific actions, for example, using “/s” to move south from the current room shown in the image. Through navigation and interaction in the chat interface, participants are required to ensure that they end up in the same room, where they both see an identical image. It is worth noting that the house layouts often contained multiple rooms of the same type, and participants were frequently asked to meet in a room of this specific type. Similar to Cups, there were no restrictions placed on



the participants regarding how they should solve the task.

### 3 Task description

The main objective of this challenge is to benchmark and evaluate generative models in the context of multi-modal dialogues. Specifically, the task is to generate the utterance  $u_i$  given the context  $C$ , where  $i$  represents the utterance number. The context  $C$  is formed by the dialogue history and visual scene(s). An important question arises: how effectively can the models utilise the context for generating the next utterance? To address this, we first split each dialogue in the dataset into conversational games (Dobnik and Storckenfeldt, 2018). Each game corresponds to a specific sub-task such as asking for more information or describing a scene. In the Cups corpus, game annotations for dialogues in Swedish are already available, and we plan to prepare annotations for dialogues in English and MeetUp dialogues as well. These conversational games can be seen as mini-contexts that help achieve a particular goal (Kowtko et al., 1991). The sentences we select belong to the context of a specific conversational game  $C_L$ , which, in turn, is part of a broader dialogue context  $C_D$ . We provide two types of contexts: (i) conversational game context  $C_L$ : this includes the utterances preceding the target utterance  $u_i$  within a single conversational game, and (ii) dialogue context  $C_D$ : this provides information on how the target utterance fits into the overall context of all other utterances and games. It is important to include the full dialogue context, as the flow of dialogues is not always linear; sometimes, a single conversational game can be embedded within another game. We emphasise that a single utterance can often be just a fragment of a broader set of utterances that together convey a specific idea.

#### 3.1 State of the data and statistics

The challenge will be conducted on the data available in public repositories. The MeetUp repository <sup>1</sup> contains 430 dialogues, where each dialogue is a sequence of events. An event can be a message from either a bot or a player. This can also be a navigation action executed by one of the players. Each valid navigation action changes position of the player in the house resulting in change of the

<sup>1</sup><https://github.com/clp-research/meetup>

scenery that the player sees. The images of house environments are taken from the ADE20k corpus (Zhou et al., 2017) and can be referred back to it. The MeetUp dialogues have on average 13.2 turns per dialogue, with each turn consisting on average of 5.1 tokens. There are 28.3 navigation actions performed on average per dialogue which means that there are approximately 2 moves per message. There are a few instances in the dataset when a single participant played the game multiple times, e.g. one worker participated in the game 49 times. Novice players played with each other only in 22 games. This information can be potentially useful for modelling because, participants adopt and change their strategy based on the familiarity with the game and they carry some of that knowledge to new conversations<sup>2</sup>.

The Cups corpus<sup>3</sup> consists of dialogues and corresponding individual static views of the same scene. We provide the views for each participant along with the ground-truth top-down view of the scene with no missing objects. We will also provide files with bounding box annotations of object ids as shown in Figure 1. The textual part of the dataset includes annotations of turns, dialogue acts, frame of reference (FoR), repair, and dialogue games (Swedish only) with the goal of capturing situated collaborative referring (Dobnik et al., 2015). Cups also contains annotations of reference and co-reference to scene entities Dobnik and Silfversparre (2021) using the CoNLL 2011/2011 annotation scheme (Pradhan et al., 2011). The Cups dataset has a fewer dialogues: 2 dialogues in English and 6 dialogues in Swedish. However, as they can take over an hour they are much longer and are structured in more dialogues games than MeetUp dialogues. There are on average 299 turns per dialogue in the English data and 171 turn per dialogue in the Swedish data.

The information on downloading the data will be available to the participants. We plan to complete annotations of dialogues with conversational games before INLG 2023. In terms of the splits, we are planning to follow the standard 80/10/10 split for training, validation and test data. Note that these splits are not for dialogues themselves,

<sup>2</sup>Based on our observations expert players tend to produce fewer messages, instead relying on the strategy of asking the other player to stay in the room and describe it, while they are looking for it.

<sup>3</sup><https://github.com/sdobnik/cups-corpus>

but for target utterances  $U$ , which are part of annotated conversational games. Our datasets are relatively small. However, they contain rich natural interaction data (rather than short crowd-sourced interactions or artificially generated dialogue data found in some popular datasets). We would like to encourage challenge participants to exploit the possibility of applying transfer learning by training the multi-modal dialogue model first on the other (larger) datasets (Zhang et al., 2018; Galetzka et al., 2020) and then fine-tuning them on our data and evaluate the possibility of such transfer as well as compare the datasets with each other.

Visual dialogue is a task that has previously been addressed in the Visual Dialog Challenge<sup>4</sup> where the goal is to answer a question given an image and a dialogue history. The challenge has attracted several submissions and has been conducted three times. However, the data used in this challenge lacks several linguistic phenomena found in Cups and MeetUp (Byron, 2003) which go beyond simple question-answer pairs (Das et al., 2017; Dong et al., 2021). In appendix B we provide a linguistic analysis of dialogues from both Cups and MeetUp and demonstrate that complexity and richness of dialogue phenomena found in our data.

## 4 Evaluation campaign

Table 1 presents the preliminary schedule for the proposed challenge. Initially, we will provide a description of the available infrastructure, which will serve as the hosting server for managing system submissions. Participants are expected to adhere to these requirements, and they should specify the use of GPUs, external APIs, and other components in their systems. For the submission and review of papers, we will use the OpenReview platform. For evaluating the generated responses, we will compare the outputs of each model against a held-out test set, using various metrics such as BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), ROUGE (Lin, 2004) and CIDEr (Vedantam et al., 2015), BERTScore (Zhang et al., 2020), and BARTScore (Yuan et al., 2021).

We will conduct a human evaluation and develop an evaluation protocol for the system submissions. To collect human judgments, we will leverage Amazon Mechanical Turk, and we might also explore using Prolific as an alternative platform, which could

<sup>4</sup><https://visualdialog.org/challenge/2020>

potentially provide more qualified human crowd-workers<sup>5</sup>. The human evaluation procedure will be modelled after the one used in the WebNLG+ 2020 challenge (Castro Ferreira et al., 2020), given that one of the authors of this proposal has experience in running human evaluation. For the human evaluation, crowd-workers will assess the quality of the generated responses based on the dialogue history and visual history (in the case of MeetUp) or scene views (in the case of Cups). These evaluations may be compared against the ground-truth human responses. To rank the responses, a Likert scale from 1 to 5 will be employed, where a higher score indicates better quality of the generated response. In human evaluation we will focus on several aspects including:

1. **Relevance:** Does the response align with the available history of linguistic interaction between participants? Does the utterance sound like something a person would say? We refer to this criterion as **local relevance**. Additionally, we will consider context provided after the generated response (from the ground-truth dialogue) and ask human evaluators to assess if the response fits the overall topic of conversation (**global relevance**).
2. **Usefulness:** Does the response contribute to solving the task? Can people identify the visual elements that the utterance refers to? We will present human crowd-workers with examples of situations where a response is relevant but not useful, and vice versa. The aim is to measure the degree of informativeness of responses, considering the nature of the task.
3. **Correctness:** Is the response well-structured, grammatical, and written in fluent language?

Additionally, we will ask human evaluators to perform fuzzy matching of the generated utterances with the ground-truth responses. This approach takes into consideration that the system’s output might not be entirely relevant locally but could still be relevant globally. By doing so, the models will not be penalised by automatic metrics for generating responses that differ from the ground-truth

<sup>5</sup>As our focus is not solely on conversational agents, the evaluation will be on assessing the quality of the generated utterances within a given context. Implementing a more sophisticated evaluation setup, where a submitted system actively plays the full game, would require a different type of challenge task.

Period	Phase
September 2023	Announcement at INLG 2023 along with the call for participation. The training and validation data are made available on the challenge website. Release of automatic evaluation scripts. Registration of participants is open.
December 2023	Test data is released, system submission. The baseline model is released along with its results for automatic evaluation.
January 2024	Deadline for system submission.
February 2024	Results of automatic evaluation are announced.
April 2024	Results of human evaluation are announced. Authors are asked to submit their system reports.
May–June 2024	System report reviewing and notification. Camera-ready submission of the system reports.
June 2024	The challenge is completed. Participant reports and challenge report are submitted to INLG 2024 and presented at the conference.

Table 1: Tentative protocol for the challenge. The schedule might change depending on the timeline of INLG 2024.

significantly, as long as they remain relevant to the conversation itself.

To ensure the quality of human evaluation, we will prepare a set of utterances in contexts that clearly represent both low and high points on the Likert scale for each of the aspects mentioned earlier. These examples will be shown to crowdworkers before they begin evaluating the actual outputs of the submitted systems. Conducting a few test rounds for human evaluation will help us understand the workers’ performance and the level of guidance they require to perform well in our task. This process will help us build a pool of highly skilled workers who are trained to evaluate challenge submissions<sup>6</sup>.

The challenge winners will be selected based on multiple criteria. Instead of focusing solely on models that perform well overall, we will also consider models that excel in specific tasks. For instance, we will look for better transfer learning approaches, multi-lingual models, or uni-modal approaches that perform well across various metrics. By examining individual properties of the submitted systems, we aim to document and benchmark the task of utterance generation in visual dialogue from multiple perspectives.

## 5 Conclusion

We present the VDG Challenge as a platform to advance research in grounded situated dialogue. We believe that the task of generating the next utterance in collaborative visual dialogue holds significant value for the NLG (Natural Language Generation) community, especially considering the remarkable performance and attention achieved by large language models in the NLP field. Our primary objective is to establish a comprehensive task bench-

<sup>6</sup>We will also recruit workers from external websites such as <https://www.mturkcrowd.com>.

mark, and as such, we welcome novel ideas for multi-modal dialogue modelling. We would be delighted to host the challenge at INLG 2024.

## Acknowledgments

The challenge that is described in this paper will be supported by a grant from the Swedish Research Council (VR project 2014-39) for the establishment of the Centre for Linguistic Theory and Studies in Probability (CLASP) at the University of Gothenburg. We would like to thank reviewers for their valuable and insightful comments.

## References

- Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. [VQA: visual question answering](#). In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 2425–2433. IEEE Computer Society.
- Satanjeev Banerjee and Alon Lavie. 2005. [METEOR: An automatic metric for MT evaluation with improved correlation with human judgments](#). In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.
- 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.
- Raffaella Bernardi, Ruket Cakici, Desmond Elliott, Aykut Erdem, Erkut Erdem, Nazli Ikizler-Cinbis, Frank Keller, Adrian Muscat, and Barbara Plank. 2016. [Automatic description generation from images: A survey of models, datasets, and evaluation measures](#). *J. Artif. Int. Res.*, 55(1):409–442.

- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Donna K Byron. 2003. [Understanding referring expressions in situated language some challenges for real-world agents](#). In *Proceedings of the First International Workshop on Language Understanding and Agents for Real World Interaction*, pages 39–47.
- Thiago Castro Ferreira, Claire Gardent, Nikolai Ilinykh, Chris van der Lee, Simon Mille, Diego Moussallem, and Anastasia Shimorina. 2020. [The 2020 bilingual, bi-directional WebNLG+ shared task: Overview and evaluation results \(WebNLG+ 2020\)](#). In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 55–76, Dublin, Ireland (Virtual). Association for Computational Linguistics.
- Herbert H Clark, Susan E Brennan, et al. 1991. Grounding in communication. *Perspectives on socially shared cognition*, 13(1991):127–149.
- Herbert H. Clark and Deanna Wilkes-Gibbs. 1986. Referring as a collaborative process. *Cognition*, 22:1–39.
- Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, Jose M. F. Moura, Devi Parikh, and Dhruv Batra. 2017. [Visual dialog](#). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 326–335.
- 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.
- Simon Dobnik, Christine Howes, and John Kelleher. 2015. [Changing perspective: Local alignment of reference frames in dialogue](#). In *Proceedings of the 19th Workshop on the Semantics and Pragmatics of Dialogue - Full Papers*, Gothenburg, Sweden. SEMDIAL.
- Simon Dobnik, John D. Kelleher, and Christine Howes. 2020. Local alignment of frame of reference assignment in english and swedish dialogue. In *Spatial Cognition XII*, pages 251–267, Cham. Springer International Publishing.
- Simon Dobnik and Vera Silfversparre. 2021. [The red cup on the left: Reference, coreference and attention in visual dialogue](#). In *Proceedings of the 25th Workshop on the Semantics and Pragmatics of Dialogue - Full Papers*, Potsdam, Germany. SEMDIAL.
- Simon Dobnik and Axel Storckenfeldt. 2018. [Categorisation of conversational games in free dialogue over spatial scenes](#). In *Proceedings of the 22nd Workshop on the Semantics and Pragmatics of Dialogue - Poster Abstracts*, Aix-en-Provence, France. SEMDIAL.
- Tianai Dong, Alberto Testoni, Luciana Benotti, and Raffaella Bernardi. 2021. [Visually grounded follow-up questions: a dataset of spatial questions which require dialogue history](#). In *Proceedings of Second International Combined Workshop on Spatial Language Understanding and Grounded Communication for Robotics*, pages 22–31, Online. Association for Computational Linguistics.
- Fabian Galetzka, Chukwuemeka Uchenna Eneh, and David Schlangen. 2020. [A corpus of controlled opinionated and knowledgeable movie discussions for training neural conversation models](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 565–573, Marseille, France. European Language Resources Association.
- Nikolai Ilinykh, Sina Zarriß, and David Schlangen. 2019. [Meet up! a corpus of joint activity dialogues in a visual environment](#). In *Proceedings of the 23rd Workshop on the Semantics and Pragmatics of Dialogue - Full Papers*, London, United Kingdom. SEMDIAL.
- Andrej Karpathy and Li Fei-Fei. 2017. [Deep visual-semantic alignments for generating image descriptions](#). *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4):664–676.
- Jacqueline C. Kowtko, Stephen D. Isard, and Gavin Doherty. 1991. Conversational games within dialogue. In *Proceedings of the ESPRIT Workshop on Discourse Coherence*, University of Edinburgh.
- Emiel Kraemer and Kees van Deemter. 2012. [Computational generation of referring expressions: A survey](#). *Computational Linguistics*, 38(1):173–218.
- 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.
- OpenAI. 2023. Chatgpt: Large-scale unsupervised language modeling for open-domain conversational agents. <https://openai.com/blog/chatgpt>. Accessed on June 9, 2023.

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.

Sameer Pradhan, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. 2011. *CoNLL-2011 shared task: Modeling unrestricted coreference in OntoNotes*. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–27, Portland, Oregon, USA. Association for Computational Linguistics.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. *Cider: Consensus-based image description evaluation*. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4566–4575.

Harm de Vries, Florian Strub, Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron Courville. 2017. *Guesswhat?! visual object discovery through multi-modal dialogue*. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5503–5512.

Terry Winograd. 1971. *Procedures as a representation for data in a computer program for understanding natural language / by Terry Winograd*. Massachusetts Institute of Technology Cambridge.

Ludwig Wittgenstein. 1953. *Philosophical Investigations*. Basil Blackwell, Oxford.

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.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. *Personalizing dialogue agents: I have a dog, do you have pets too?* In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. 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.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. 2017. *Scene parsing through ADE20K dataset*. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5122–5130. IEEE Computer Society.

## A Cups corpus: individual views

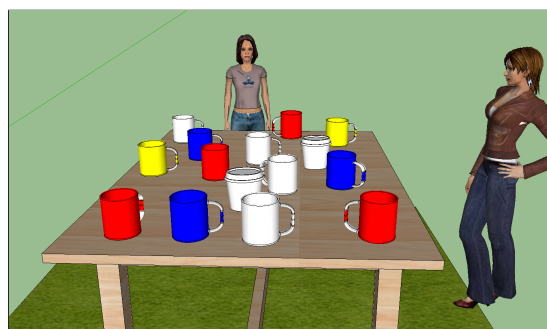


Figure 3: Individual scene view of the speaker 1 from Figure 1.



Figure 4: Individual scene view of the speaker 2 from Figure 1.

## B MeetUp and Cups examples

Here is an example from the MeetUp corpus:

- (1)
  - a. Game Master: You have to meet in a room of type *utility room*.
  - b. A: Hi. I'm in a bedroom with pink walls.
  - c. B: I seem to be in a kitchen.
  - d. A: I'll go look for a utility room.
  - e. A (privately): *north*
  - f. A (privately): *west*
  - g. B (privately): *east*
  - h. A: Found a room with a washing machine. Is that a utility room?
  - i. B: Was wondering as well. Probably that's what it is.
  - j. B: I'm in the pink bedroom now. I'll come to you.
  - k. B (privately): *north*
  - l. B (privately): *west*
  - m. B: Poster above washing machine?
  - n. A: Mine has a mirror on the wall.
  - o. B: yeah, could be mirror. Plastic chair?
  - p. A: And laundry basket.
  - q. A: *done*
  - r. B: Same
  - s. B: *done*

MeetUp has many high-level descriptions of the scene ((1-b)–(1-c)), which require understanding of

room types. Strategy coordination ((1-d)–(1-j)) and revisions of previously made references (“poster” to “mirror” in (1-m)–(1-o)) occur as well. There is also a need for memory (1-j), which is important for co-reference. Another property of the dataset is the presence of indirect dialogue acts ((1-p)–(1-o)).

Below is an excerpt from the Cups corpus:

- (2)
- a. A: i see lots of cups and containers on the table
  - b. B: me too
  - c. A: some white, some red, some yellow, some blue
  - d. B: I see six white ones
  - e. A: i see seven
  - f. A: but maybe we should move in one direction...
  - g. B: ok, lets do that
  - h. A: shall we take it from katies point of view?
  - i. B: ok
  - j. ...
  - k. B: so what do you see in the “second row” from my perspective?
  - l. A: i see red, then space, then white and blue (same as katie’s)
  - m. A: no yellow
  - n. B: is it on the edge of the table?
  - o. B: on your left
  - p. A: ok, yes!

We observe reference to the same (or different!) objects using attributes such as colour and identification of object mismatch, e.g. (2-a)–(2-e). In (2-f)–(2-i) participants negotiate interactive strategy. Adjusting a perspective (or frame of reference) for spatial relations is also important in dialogue games, e.g. (2-k)–(2-p).

# Identifying Feedback Types to Augment Feedback Comment Generation

Maja Stahl and Henning Wachsmuth

Leibniz University Hannover, Hannover, Germany

Institute of Artificial Intelligence

{m.stahl, h.wachsmuth}@ai.uni-hannover.de

## Abstract

In the context of language learning, feedback comment generation is the task of generating hints or explanatory notes for learner texts that help understand why a part of text is erroneous. This paper presents our approach to the Feedback Comment Generation Shared Task, collocated with the 16th International Natural Language Generation Conference (INLG 2023). The approach augments the generation of feedback comments by a self-supervised identification of feedback types in a multitask-learning setting. Within the shared task, other approaches performed more effectively, yet the combined modeling of feedback type classification and feedback comment generation is superior to performing feedback generation only.

## 1 Introduction

Several studies have dealt with identifying and correcting grammatical errors to help language learners improve their writing skills (Imamura et al., 2012; Bryant et al., 2017; Rozovskaya and Roth, 2019; Grundkiewicz et al., 2019). However, these approaches do not provide learners with a rationale for why a piece of text is erroneous. To help learners better understand and adapt the underlying writing rules, Nagata (2019) introduced the task of *feedback comment generation*: Given a learner text in which some span is known to be erroneous, automatically generate a comment containing helpful hints and explanations. Specifically, the comment should prompt the learner to come up with a solution rather than pointing out an error (grammatical error detection) or correcting it (grammatical error correction).

Towards this end, the Feedback Comment Generation Shared Task (Nagata et al., 2021) at the 16th International Natural Language Generation Conference (INLG 2023) has provided a corpus of erroneous English sentences written by non-native learners of English. Each sentence comes with a

feedback comment that is targeted towards a given position of the sentence. The focus is on errors related to the use of prepositions in order to restrict the extensive task of generating feedback to a manageable setting. The generated comments are supposed to explain to the writer why the text part in question is erroneous, possibly with related writing rules. One exemplary instance of the task looks as follows:

**Input Text** “They can help their father or mother about money that we must use in the university too.”

**Feedback Comment** “«About» is not the appropriate <preposition> to be used when a <noun> follows the structure <help + someone>. Look up the use of the <verb> «help» in a dictionary to learn the appropriate <preposition> to be used.”

As our contribution to the shared task, we present an approach that relies on multitask-learning to simultaneously (a) classify the *type* of the target feedback for the given erroneous input sentence and (b) generate an appropriate feedback comment of this type. Since no feedback type labels are given in the data, we tackle the type classification in a self-supervised manner. In particular, we apply an unsupervised clustering based on TF-IDF vector representations of the feedback comments. Each cluster is assumed to represent one feedback type. We then learn a mapping from input texts to feedback types. The rationale is that an explicit distinction between different types of feedback may help to generate targeted feedback comments per type and, hence, more diverse comments for different types. Overall, the generated feedback comments may then better match the input text by exploiting the feedback patterns per comment type.

Our evaluation results in the shared task suggest that the combined modeling of feedback type classification and feedback comment generation is

superior to performing feedback generation only. Our approach improves over sequence-to-sequence baselines in automatic and manual evaluation.

## 2 Related Work

Supporting non-native speakers of a language to improve their writing skills has been approached from multiple perspectives. So far, however, the main focus has been on detecting and correcting grammatical errors in text.

Early research often targeted only on one common error type, such as incorrect article usage (Han et al., 2006), preposition and determiner usage (Gamon et al., 2008; De Felice and Pulman, 2008), singular and plural usage (Nagata et al., 2006), or false friends (Katrenko, 2012). More recent work proposed approaches to detecting (Nagata et al., 2022) and correcting (Chollampatt et al., 2016; Takahashi et al., 2020; Junczys-Dowmunt et al., 2018) grammatical errors in general using large-scale neural networks, including transformer-based language models. Some works go beyond grammar to assess argumentative structures in learner texts (Wachsmuth et al., 2016; Stab and Gurevych, 2016; Chen et al., 2022). Creutz and Sjöblom (2019) proposed the usefulness of rewriting language learner texts not only to correct errors but also to improve the fluency and naturalness of a text.

The task of feedback comment generation, as proposed by Nagata (2019), goes beyond detecting and correcting errors in that it includes to provide explanations for why some text part is erroneous. With this, language learners can better understand and adapt the underlying writing rules. Hanawa et al. (2021) compared a neural retrieval-based method to a sequence-to-sequence model and a hybrid of these two that edits retrieved feedback comments. They found that the sequence-to-sequence model works best in a setting with few feedback variations, for example, concerning preposition use only. At the same time, the hybrid approach seems most promising for general feedback generation.

## 3 Task and Data

This section summarizes the Feedback Comment Generation Shared Task as well the data provided as part of the task.

### 3.1 Task

In the context of the Feedback Comment Generation Shared Task, the definition of feedback com-

ment generation can be summarized as follows (Nagata et al., 2021):<sup>1</sup>

Given an input text and a position known to be erroneous regarding preposition use, automatically generate hints or explanatory notes (feedback comment). The generated feedback comment should explain to the writer why the input text is erroneous at the specified position, possibly with related writing rules. Alternatively, the special token `<NO_COMMENT>` can be generated if an approach cannot generate reliable feedback.

### 3.2 Data

Each instance in the dataset provided by the organizers consists of an English erroneous input sentence, the position of the error, and a manually written feedback comment targeted towards the error position, as described in Nagata (2019). A total of 4868 training, 170 development, and 215 test instances was provided.

The sentences come from essays of the International Corpus Network of Asian Learners of English (ICNALE) that were written by Asian college students with proficiency levels in English estimated to be between A2 and B2+ in the CEFR metric (Ishikawa, 2013). The essays discuss two topics: (a) “It is important for college students to have a part-time job”, and (b) “Smoking should be completely banned at all restaurants in the country”. The feedback comments were written by professional annotators with good English skills. They were asked to use special symbols in their writing to highlight specific tokens: (`<`, `>`) to surround grammatical terms, (`<<`, `>>`) to surround citations from the input text.

## 4 Approach

We now present our approach to feedback comment generation. Its core idea is to classify the type of feedback to be given and to generate an according feedback comment simultaneously.

### 4.1 Overview

As illustrated in Figure 1, our approach consists of two main stages:

1. *Feedback Clustering*. We first perform clustering on the TF-IDF vector representation of the training feedback comments in order to identify different feedback types.

<sup>1</sup><https://fcg.sharedtask.org/task/>, last accessed: 2022-09-12



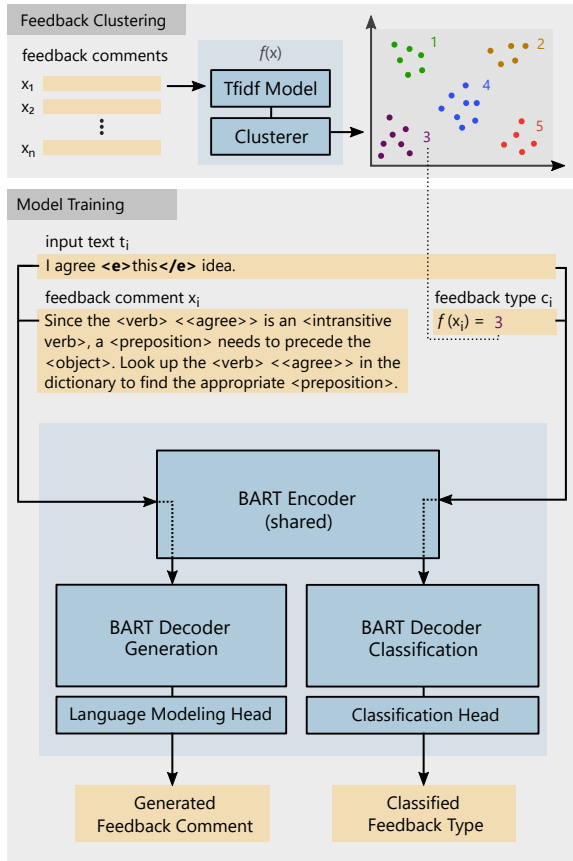


Figure 1: Overview of our approach: First, the training feedback comments are clustered into *feedback types* based on their TF-IDF vector representations. Given an input text and the position of an error, a multitask-learning model then jointly classifies the feedback type and generates the target feedback comment.

2. *Model Training.* Then, a pre-trained language model is trained jointly on feedback type classification and feedback comment generation, using the cluster number from Step 1 as the target label for the classification.

Notice that the feedback type classification is performed based on the erroneous input sentences and not on the target feedback comments, since the latter are not available at inference time. The model is therefore expected to infer the feedback type to be given from the input text only.

## 4.2 Details

For the feedback clustering, we remove citations from the erroneous input texts as highlighted with ( $\langle\langle, \rangle\rangle$ ) from the feedback comments, to improve the generalizability. For model training and inference, we provide the model with the error position by surrounding the erroneous text part with special tokens ( $\langle e \rangle, \langle /e \rangle$ ), as shown in Fig-

ure 1.

For the joint classification and generation, we use a transformer-based encoder-decoder model in a multitask-learning setting. Multitasking is performed by sharing the encoder between the two tasks and combining it with task-specific decoders and language modeling and classification heads, respectively. The training of the model is performed alternately for both tasks, so the encoder weights are updated in each step, while only one decoder and one model head are updated at a time. The hypothesis is that this setting leads to encodings that differ more between different types of feedback comments and are more similar for similar target feedback comments compared to a single task setting. We expect this to help generate more targeted feedback towards the feedback comment types identified in the training data.

## 5 Evaluation

This section reports on our experiments with joint feedback type classification and feedback comment generation before presenting the evaluation methods and results.

### 5.1 Experimental setup

In our evaluation, we relied on the following setup:

**Feedback Clustering** For clustering feedback comments, we use the scikit-learn implementation (Pedregosa et al., 2011) of TF-IDF to transform the training feedback comments into their vector representations. We excluded vocabulary entries with an absolute document frequency below 5 and a relative document frequency above 95% in order to remove rare tokens and stop words. On this basis, we ran  $k$ -means clustering with pseudo-random centroid initialization (seed 42). We optimized the number of clusters against the BLEU score (Papineni et al., 2002) of the generated feedback comments and found  $k = 12$  clusters to perform best in this regard.

**Feedback Type Classification** Next, we employed the TF-IDF model and the  $k$ -means model to infer feedback types for the validation examples, which we then used to evaluate classification performance. On the validation set, our model achieved a macro-averaged  $F_1$ -score of 0.80 for feedback type classification. The score varied between 0.59 and 0.89 for numbers of clusters between 6 and 30.

Approach	Automatic (BLEU)	Manual (F <sub>1</sub> )
Generation-BART	0.394	n/a
Generation-Pointer (Nagata et al., 2021)	0.334	0.312
Multitask-BART (our model)	<b>0.437</b>	<b>0.358</b>

Table 1: Automatic and manual evaluation results: Our model outperforms both baselines in terms of BLEU score, and it also improves over the shared task baseline of Nagata et al. (2021) in the manual evaluation.

**Feedback Comment Generation** In our language model experiments, we used the HuggingFace implementation (Wolf et al., 2020) of the pre-trained BART language model with 139M parameters (Lewis et al., 2020). Together with the cluster number optimization, we tuned the hyperparameters for the training of the model and found a learning rate of  $5^{-5}$ , batch size of 4, 8 training epochs, and length penalty of 1.0 to perform best regarding the feedback comment generation. Below, our model is called *Multitask-BART*.

**Baselines** We compare the Multitask-BART model against to two baselines:

- *Generation-BART*. A modification of our model, trained only on feedback comment generation.
- *Generation-Pointer*. The baseline model provided by the shared task organizers, which is an encoder-decoder model with a copy mechanism based on a pointer generator network (Nagata et al., 2021).<sup>2</sup>

## 5.2 Results

Table 1 presents the results of both the automatic and the manual evaluation.

**Automatic Evaluation** We automatically assessed the feedback comment generation quality of all models on the test set using BLEU score (Papineni et al., 2002), as suggested by the organizers. Among the evaluated approaches, our proposed model achieves the highest BLEU score (0.437), that is, its output has the highest overlap with the human-written reference comments.

**Manual Evaluation** In addition, our submitted shared task approach was manually evaluated by the organizers, who compared the generated feedback comments to the corresponding reference

<sup>2</sup>[https://github.com/k-hanawa/fcg\\_genchal2022\\_baseline](https://github.com/k-hanawa/fcg_genchal2022_baseline), last access: 2022-09-12

feedback comments. A generated feedback comment was considered correct when (1) it contains information similar to the reference and (2) it does not contain information irrelevant to the error position. The overall performance was then measured as F<sub>1</sub>-score based on the correctness labels (Nagata et al., 2021).

With an F<sub>1</sub>-score of 0.358, our model outperforms over the strong baseline based on a pointer generator network (0.312), even though the performance difference is not as big as in the automatic evaluation. Compared to the other submissions to the shared task, our model achieved the sixth place in the automatic evaluation and the seventh place in the manual evaluation.

**Error Analysis** To obtain insights into the weaknesses of our approach, we finally looked at those feedback comments generated by the model that were flagged as incorrect by the organizers. We found that the main contents of the comments are often correct or somewhat correct, but the important details, which were highlighted in the target feedback comments by brackets, are wrong. For example, a wrong word is cited from the input text, or a word not present in the input is generated as if it was a citation from the input (using the brackets <<, >>). The generated grammatical terms (surrounded by <, >) are the other common error of our model, which is more problematic as it cannot be identified easily as an error by a language learner. The organizers made the same observations when they assessed our model output.

## 6 Conclusion

This paper has described our approach to the Feedback Generation Shared task Collocated with the 16th International Natural Language Generation Conference (INLG 2023). The key idea of our approach is to jointly model the classification of feedback types and the generation of feedback comments in order to exploit found patterns per comment type during the generation. Our experiments suggest that the generation quality improves by modeling both tasks together. We also observed open issues, though, that indicate a wrong integration of parts of the input into the generated output. A refined control of the interaction of input and output may resolve such issues in future work.

## References

- Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. [Automatic annotation and evaluation of error types for grammatical error correction](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.
- Wei-Fan Chen, Mei-Hua Chen, Garima Mudgal, and Wachsmuth. 2022. Analyzing culture-specific argument structures in learner essays. In *Proceedings of the 9th Workshop on Argument Mining*.
- Shamil Chollampatt, Duc Tam Hoang, and Hwee Tou Ng. 2016. [Adapting grammatical error correction based on the native language of writers with neural network joint models](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1901–1911, Austin, Texas. Association for Computational Linguistics.
- Mathias Creutz and Eetu Sjöblom. 2019. [Toward automatic improvement of language produced by non-native language learners](#). In *Proceedings of the 8th Workshop on NLP for Computer Assisted Language Learning*, pages 20–30, Turku, Finland. LiU Electronic Press.
- Rachele De Felice and Stephen G. Pulman. 2008. [A classifier-based approach to preposition and determiner error correction in L2 English](#). In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 169–176, Manchester, UK. Coling 2008 Organizing Committee.
- Michael Gamon, Jianfeng Gao, Chris Brockett, Alexandre Klementiev, William B. Dolan, Dmitriy Belenko, and Lucy Vanderwende. 2008. [Using contextual speller techniques and language modeling for ESL error correction](#). In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-1*.
- Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. [Neural grammatical error correction systems with unsupervised pre-training on synthetic data](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 252–263, Florence, Italy. Association for Computational Linguistics.
- Na-Rae Han, Martin Chororow, and Claudia Leacock. 2006. [Detecting errors in english article usage by non-native speakers](#). *Natural Language Engineering*, 12(2):115–129.
- Kazuaki Hanawa, Ryo Nagata, and Kentaro Inui. 2021. [Exploring methods for generating feedback comments for writing learning](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9719–9730, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Kenji Imamura, Kuniko Saito, Kugatsu Sadamitsu, and Hitoshi Nishikawa. 2012. [Grammar error correction using pseudo-error sentences and domain adaptation](#). In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 388–392, Jeju Island, Korea. Association for Computational Linguistics.
- Shin’ichiro Ishikawa. 2013. [The icnale and sophisticated contrastive interlanguage analysis of asian learners of english](#). *Learner corpus studies in Asia and the World*, 1:91–118.
- Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018. [Approaching neural grammatical error correction as a low-resource machine translation task](#). 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 595–606, New Orleans, Louisiana. Association for Computational Linguistics.
- Sophia Katrenko. 2012. [“could you make me a favour and do coffee, please?”: Implications for automatic error correction in English and Dutch](#). In *\*SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 49–53, Montréal, Canada. 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.
- Ryo Nagata. 2019. [Toward a task of feedback comment generation for writing learning](#). 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 3206–3215, Hong Kong, China. Association for Computational Linguistics.
- Ryo Nagata, Masato Hagiwara, Kazuaki Hanawa, Masato Mita, Artem Chernodub, and Olena Nahorna. 2021. [Shared task on feedback comment generation for language learners](#). In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 320–324, Aberdeen, Scotland, UK. Association for Computational Linguistics.
- Ryo Nagata, Atsuo Kawai, Koichiro Morihiro, and Naoki Isu. 2006. [A feedback-augmented method for detecting errors in the writing of learners of English](#). In *Proceedings of the 21st International Conference*

- on *Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 241–248, Sydney, Australia. Association for Computational Linguistics.
- Ryo Nagata, Manabu Kimura, and Kazuaki Hanawa. 2022. [Exploring the capacity of a large-scale masked language model to recognize grammatical errors](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 4107–4118, Dublin, Ireland. 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.
- Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Alla Rozovskaya and Dan Roth. 2019. [Grammar error correction in morphologically rich languages: The case of Russian](#). *Transactions of the Association for Computational Linguistics*, 7:1–17.
- Christian Stab and Iryna Gurevych. 2016. [Recognizing the absence of opposing arguments in persuasive essays](#). In *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*, pages 113–118. Association for Computational Linguistics.
- Yujin Takahashi, Satoru Katsumata, and Mamoru Komachi. 2020. [Grammatical error correction using pseudo learner corpus considering learner’s error tendency](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 27–32, Online. Association for Computational Linguistics.
- Henning Wachsmuth, Khalid Al Khatib, and Benno Stein. 2016. [Using argument mining to assess the argumentation quality of essays](#). In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1680–1691. The COLING 2016 Organizing Committee.
- 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.

# Error syntax aware augmentation of feedback comment generation dataset

Nikolay Babakov<sup>1,\*</sup>, Maria Lysyuk<sup>2</sup>, Alexander Shvets<sup>3</sup>, Lilya Kazakova<sup>4</sup>,  
and Alexander Panchenko<sup>2,5</sup>

<sup>1</sup> Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS),  
Universidade de Santiago de Compostela

<sup>2</sup> Skolkovo Institute of Science and Technology, <sup>3</sup> Pompeu Fabra University

<sup>4</sup> HSE University, <sup>5</sup> Artificial Intelligence Research Institute  
[nikolay.babakov@usc.es](mailto:nikolay.babakov@usc.es)

## Abstract

This paper presents a solution to the GenChal 2022 shared task dedicated to feedback comment generation for writing learning. In terms of this task given a text with an error and a span of the error, a system generates an explanatory note that helps the writer (language learner) to improve their writing skills. Our solution is based on fine-tuning the T5 model on the initial dataset augmented according to syntactical dependencies of the words located within indicated error span. The solution of our team “nigula” obtained second place according to manual evaluation by the organizers.

## 1 Introduction

Feedback comment generation for language learners is the task of generating an explanatory note that helps the writers (language learners) improve their writing skills for text with error and a span of this error. In terms of GenChal2022<sup>1</sup>, the target language is English, but this task applies to any language. Moreover, this task is mostly designed not for such cases as simple typos or misspellings (something which can be easily detected by standard grammar-error detection systems) but for erroneous, unnatural, or problematic words. See Table 1 for examples of such types of errors and corresponding comments.

Detecting the aforementioned misuse of words is not enough to prevent the same error in the future. It is important to provide some explanation and reference to the error-specific rule and give a direct hint on which correct word should be used in this particular case.

While any language follows certain rules which can be encoded manually to obtain the solution relying on rules and templates, in our work we try to use the benefit of the availability of parallel data and explore the limits of modern seq2seq models

with the minimal number of rules and manual labor involved.

The contributions of our work are as follows:

- We present our solution based on tuning the T5-large model on the dataset augmented in the special error syntax-based approach.
- We opensource the model on Huggingface Model Hub<sup>2</sup> and code for experiments on GitHub<sup>3</sup>.

## 2 Related works

The first attempts to provide feedback on a particular error type were based on rules (Nagata et al., 2014). In Morgado da Costa et al. (2020) authors used the English Resource Grammar parser to analyze the learner’s sentence. If the parser fails to process a sentence, this sentence is supposed to have an error, and, so-called, mal-rules are used to detect the particular type of error. If the mal-rule works, the user is provided with a mal-rule-specific comment.

The steps towards the usage of more modern approaches, such as neural networks, have also been performed in this task. In Andersen et al. (2013), the authors use a combination of basic machine learning approaches to detect errors and rules to provide feedback on some common types of errors. Nagata (2019) showed that a neural retrieval-based method can be effective in preposition feedback comment generation. Lai and Chang (2019) proposed a method that provides context-sensitive explanations using grammatical error correction and templates. Gkatzia et al. (2014) suggested methods for automatically choosing feedback templates based on learning history. In Kaneko et al. (2022), the sentence with an error is first corrected with

<sup>2</sup>[https://huggingface.co/SkolkovoInstitute/GenChal\\_2022\\_nigula](https://huggingface.co/SkolkovoInstitute/GenChal_2022_nigula)

<sup>3</sup>[https://github.com/skoltech-nlp/feedback\\_generation\\_nigula](https://github.com/skoltech-nlp/feedback_generation_nigula)

Work mostly has been done while at Skoltech

<sup>1</sup><https://fcg.sharedtask.org/task/>

Learner’s sentence	Golden feedback / Our system’s output
Maybe at holiday and have free time then I can <b>to</b> have part-time job .	<Verbs> that follow an <auxiliary verb> are used in their <infinitive form> instead of a <to infinitive>. <hr/> <verbs> that come right after an <auxiliary verb> are used in their <infinitive form>.
<b>Become</b> college student requires a lot of money .	A <verb phrase> needs to be converted into into a <noun phrase> in the form of a <to infinitive> or a <gerund> to be used as the <subject>. <hr/> A <verb phrase> needs to be converted into into a <noun phrase> in the form of a <to infinitive> or a <gerund> to be used as the <subject>.
They might face <b>with</b> the danger of exploring of the battery or the problems of the electronic .	Since the <verb> «face» is a <transitive verb> and its <direct object> indicates the confronted object, it does not require a <preposition>. <hr/> Since the <verb> «face» is a <transitive verb>, the <object> does not require a <preposition>.
There are many advantages <b>to have</b> a part-time job .	Use <preposition + gerund> instead of a <to-infinitive> to describe the “advantage”. Look up the use of the <noun> «advantage» in a dictionary. <hr/> Use the structure <preposition+gerund> instead of a <to-infinitive> with the <noun> «advantage».

Table 1: **Input and output (golden and system)**. Example of feedback comments from the competition test set and generated without system. The word with an error is highlighted in red. < > signs indicate grammar terms, « » mean reference to the word in the learner text.

the grammar error correction system and then the K-nearest neighbors algorithm is used to provide the learner with the pair of an incorrect and corrected sentence which contains a similar kind of error. In [Getman \(2021\)](#), the authors use unusual n-grams, out-of-vocabulary words, and several pre-trained models to find an error in the learner’s text. This system does not provide text feedback in natural language, but it generates a structured report of found errors in the text.

The useful subtask of the feedback comment generation is grammar error classification. The information on the particular type of error made in the text could be used either directly by creating a template comment to this error or by using the error type as an additional signal in training data. One example of such work is [Bryant et al. \(2017\)](#) which automatically extracts the edits between parallel original and corrected sentences using a linguistically-enhanced alignment algorithm. In this paper, a rule-based framework that relies solely on dataset-agnostic information such as lemma and part-of-speech is developed as well. Beyond this, the paper of [Choshen et al. \(2020\)](#) uses universal dependencies syntactic representation scheme.

The main limitation of using the most modern text-to-text models had been the non-availability of parallel datasets with errors and corresponding annotation. In [Pilan et al. \(2020\)](#), a unique dataset where feedback comments on linking words were annotated was released. The dataset used in GenChal 2022 was collected in [Nagata \(2019\)](#); [Nagata et al. \(2020\)](#) for the English language. The main types of errors in this dataset are misuse of prepositions and other writing items such as discourse and lexical choice.

### 3 Task description

In this section, we introduce the formal definition of the task and the dataset provided for it.

#### 3.1 Task definition

The system is provided with the text that by default contains an error. Moreover, the exact error span is provided as well. The output of the system should be the text which provides explanatory feedback on the error. If the system fails to generate the feedback it is supposed to return the <NO\_COMMENT> string.

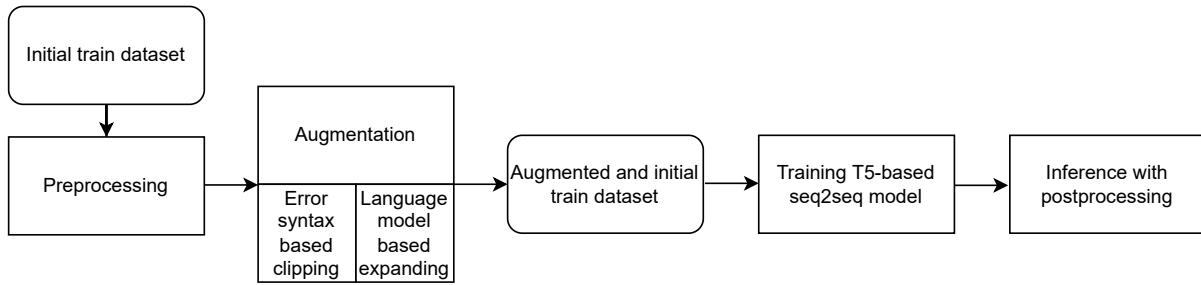


Figure 1: **Method workflow.** Description of the main steps of our feedback generation approach. The initial dataset is first preprocessed, then it is augmented by clipping the learner sentences according to the syntax relations of the word within error spans and the clipped sentences are then expanded with a large language model. The initial and augmented versions of the dataset are then used to train the T5 model in seq2seq mode. After that, the trained model is used to generate the feedback comments on the test data and the final text is post-processed.

### 3.2 Dataset

The form of the data in the task dataset is as follows:

- I agree it . \t 3:10 \t < <Agree > > is an <intransitive verb> and thus it requires a <preposition> before its object.

where \t stands for the tab character. If a sentence contains more than one error, it appears two or more times with different spans, so the input text always consists of only one sentence with one span. Also, the texts are pre-tokenized where tokens are separated by whitespace.

The feedback texts contain special symbols: <, > for grammatical terms (e.g. <intransitive verb>, <noun>, etc.) and < <, > > for citations of the words from learners' sentence (e.g. «agree»).

The dataset consists of the train (4867 samples) and development (169 samples) sets which were provided at the beginning of the competition and the test set (214 samples) which was provided in the last week of the competition and had only text and error span information.

## 4 Our method description

In this section, we introduce the main steps for training the model used for the final submission. These steps are also shown in Figure 1.

### 4.1 Preprocessing

Even though the text in the dataset is pre-tokenized, the special symbols described in 3.2 (<, >, < <, > >) can interfere with the tokenizer of a large pre-trained model. Thus, we lowercase the text and insert whitespace between words and the special symbols. Refer to Table 2 to have a look at the example of an initial and preprocessed sample.

As mentioned in 3.2, one sentence can have multiple errors, but according to the task definition, the system is supposed to provide feedback only to one slot of the error. To explicitly point out the exact error span in the learner's sentence we put similar special symbols (< <, > >) around the error span.

### 4.2 Augmentation

Even though the dataset has a limited amount of error types the variability of natural language yields an almost unlimited amount of situations in which each particular error can occur.

Let's demonstrate it in the following example of learner's text - *They can help their father or mother about money that we must use in the university too.* According to the span, the error is in the misuse of the preposition "about". To be more specific, the student has used an incorrect preposition after the "help + someone" construction. So, if we generate a new sentence that would be similar to the initial one by 'help someone about something' construction and would be different from other points of view, it will still correspond to the initial feedback and it could be applied to training the model in seq2seq mode as an additional training sample.

Our approach to augmentation consists of two parts. First, we cut the initial sentence by the last word which is syntactically related to the words within an error span. Second, we use the remaining text as a prompt to the language model, so it generates an alternative end to the sentence with a given prefix. Refer to Figure 2 for the principal scheme of the augmentation approach. More details can be found below in Sections 4.2.1 and 4.2.2.

Sentence	Comment
They can help their father or mother about money that we must use in the university too .	< <About > >is not the appropriate <preposition>to be used when a <noun>follows the structure <help + someone >. Look up the use of the <verb> < <help > > in a dictionary to learn the appropriate <preposition> to be used.
they can help their father or mother < < about > > money that we must use in the university too .	< < about > > is not the appropriate < preposition > to be used when a < noun > follows the structure < help + someone > . look up the use of the < verb > < < help > > in a dictionary to learn the appropriate < preposition > to be used .

Table 2: **Preprocessing.** Example of the preprocessing of learner’s text and the corresponding feedback comment (the initial one is at the top, the preprocessed one is at the bottom).

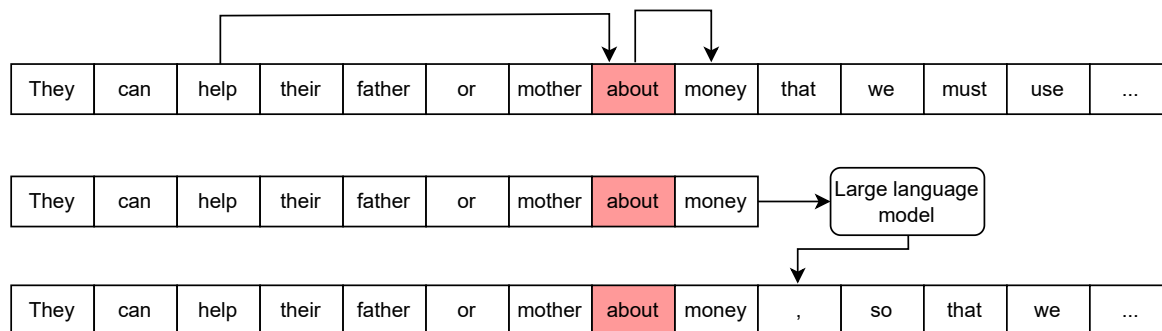


Figure 2: **Augmentation workflow.** The sentence with an error passes automatic syntactical analysis. The sentence is cropped by the last word that is syntactically connected to any word within the error span or by the error span itself (if no syntactically dependent words are located after the span). The cropped sentence is used as a prompt for a large language model to generate an alternative phrase with a similar error.

#### 4.2.1 Learner sentence clipping

We use the `spacy`<sup>4</sup> package to perform syntactic parsing of the learner text. In the case of the sentence from our example, the error word "about" is syntactically connected to the verb "help" and the noun "money". We assume that these words form the skeleton of the error in this particular sentence. Thus, we crop everything after the last connected word 'money' and the cropped sentence becomes *they can help their father or mother about money*. If the syntactically connected words are located before the error span we crop the sentence by the error span.

<sup>4</sup>[https://spacy.io/models/en#en\\_core\\_web\\_md](https://spacy.io/models/en#en_core_web_md)

#### 4.2.2 Text generation

The cropped version of the text can be used as a prompt to infer an entire sentence with a large pre-trained language model. We use EleutherAI/gpt-neo-1.3B (Gao et al., 2020; Black et al., 2021)<sup>5</sup> to generate new texts.

This approach allows us to get a new sample consisting of the new sentence (which consists of the prefix similar to the initial one and the extension generated by the language model) and the feedback similar to the initial sample.

Here are some examples of the sentences generated by the model from the prompt *they can help their father or mother about money*:

- they can help their father or mother about

<sup>5</sup><https://huggingface.co/EleutherAI/gpt-neo-1.3B>



Exp. #	Data	BLEU
1	Initial	0.64
2	Augmented	0.65
3	Initial&augmented	0.67

Table 3: **Training steps.** Sequence of experiments conducted to train the final model. BLEU is shown for the validation dataset.

money, so that we can be independent. We have to work hard to earn our bread.

- they can help their father or mother about money." "Well, if I do that I'll have to buy clothes, and pay my own expense." "The girl has
- they can help their father or mother about money, they can help the mother, who can give us some medicine? We are able to keep the household alive from the old and the sick with

Some examples in the original dataset have similar feedback comments. If the learner's sentence is related to the group of samples which have ten or more similar feedback sentences, we assume that there is enough information for training the final model and do not apply augmentation to it.

Thus, we apply the augmentation technique to more than 4700 samples from the initial dataset. Each sample is augmented with 8-10 sentences. The final versions of the augmented dataset consist of 43,174 samples. We use these samples as additional data for training the final model.

### 4.3 Model training

The final solution is based on tuning T5 model (Raffel et al., 2020)<sup>6</sup>. The model's training input is the preprocessed sentences (see Section 4.1) and its target output is the corresponding feedback comments. Our default training parameters are batch size 8, Adam optimizer, gradient clipping by 1, and learning rate 1e-5. During training, we evaluate the current version of the model by calculating the BLEU score (Papineni et al., 2002) on the validation dataset.

As shown in Table 3 the main training steps are as follows:

- We train the model on the initial dataset. The best validation BLEU score is 0.64

- We tune the best version of the model on the augmented dataset (just newly generated samples). The best validation BLEU score is 0.66
- We merge both datasets, decrease the learning rate to 1e-6 and tune it for 4000 steps. This increases the validation BLEU score up to 0.67

The final version of the model is open-sourced to HuggingFace ModelHub.

### 4.4 Postprocessing

During inference, the final model demonstrates unusual behavior in terms of placing the special marks. To be more specific, the model correctly generates the closing brackets (>, > >) but fails to generate opening brackets (<, < <). We apply some postprocessing to the generated text. For doing this, we collect all possible grammar term references within train and validation texts, find the longest possible term before the closing bracket, and put opening brackets before the found term.

### 4.5 Other experiments

Except for the data augmentation approach, we also tried other hypotheses, which did not work well according to the preliminary analysis. We do not provide a full comparison with our main solution, but we find it useful to share them because this is the first generation challenge of feedback comment generation and there seems to be no or very little relevant work.

First, we tried various ways to enrich the training samples with auxiliary information, such as part of speech of the words inside the error span, corrected words, or grammar error classes (obtained with ERRANT<sup>7</sup> classification model). The results of this group of approaches were slightly worse than the main solution, however, we may assume that we did not dedicate much time to that, so it could be promising to conduct further experiments in this direction.

Second, we tried to find some easy heuristics that can be used for retrieving the existing suitable comment from the train data. We clustered learners' texts using similar words or parts of speech within the error span or vector representation of the text and then manually analyzed the feedback comments corresponding to every cluster. We also

<sup>6</sup><https://huggingface.co/t5-large>

<sup>7</sup><https://pypi.org/project/errant/>

Preprocessed input	Reference comments	System output	Manual evaluation by orgs
most < < of > > restaurant separate smoking are and non smoking are , but the smoking smell is remain this place .	Something defined follows < < most of > > . When referring to restaurants in general, use < < most > > as an <adjective>instead of a <noun>.	A group of something specific follows < < most of > > . Use < < most > > as an <adjective>rather than a <noun>when simply referring to something in general.	correct
most of restaurant separate smoking are and non smoking are , but the smoking smell is < < remain > > < < this > > place .	The <verb> < < remain > > is an <intransitive verb>and cannot take <direct objects>without a <preposition>of place.	When the <verb> < < remain > > is used as an <intransitive verb>, a <preposition>needs to precede the <object>. Look up the <verb> < < remain > > in the dictionary to find the appropriate <preposition>.	correct
part time job is a very important < < of > > people in present .	Use the <preposition>'for' to express "to be important for people".	It is not grammatically incorrect to use the <preposition> < < of > > , but using < < of > > emphasizes the part in one's opinion". It is more natural to use the <preposition>'for'to express to be important for..."	incorrect
part time job is a very important of people < < in > > present .	The <preposition>used in the <idiom>with the <noun> < < present > > to express "now" is not < < in > > .	The <preposition>used in the <idiom>with the <noun> < < present > > to express for the moment" or now" is not < < in > > .	correct

Table 4: Examples of system output to similar sentences with different error span

tried to do similar experiments in the opposite direction (clustered feedback comments and analyze the learners' sentences). This approach let us find some heuristics that were used as an auxiliary tool for the language model-based feedback generation. However, the decision that used this tool with the trained model did not show any significant improvement over the pure model-based approach, which most probably means that such heuristics can be learned by the language model itself.

## 5 Results

The results of the system output are scored automatically and manually.

Automatic and manual scoring compares the system's outputs with manually created feedback comments. The automatic approach uses the BLEU score. In terms of manual scoring, a system output is regarded as appropriate if it contains information similar to the reference and does not contain information that is irrelevant to the offset; it may

contain information that the reference does not contain as long as it is relevant to the offset. If these conditions are met, the output is labeled as correct. The task definition (see Section 3.1) allows systems to generate <NO\_COMMENT> phrase which is excluded from both the numerator and the denominator of precision and the numerator of recall. That is why the final score is calculated as precision, recall, and F1-score.

We do not make any filtering of the generated feedback, thus there is no case when our system generates <NO\_COMMENT> phrase, so all metrics are equal. Refer to Table 5 for the results of manual evaluation by organizers of top-3 solutions. Our solution took second place.

It is also worth mentioning that the approach of using double brackets as a signal of the exact location of the error span to train the model worked well. To be more precise, the system always generates different feedback for similar sentences with different error spans and 12 out of 20 sentences in the test set (similar sentences with different slots were presented by pairs, so in total there are ten pairs of such samples) were scored as correct by organizers. Examples of such sentences can be found at Table 4.

## 6 Future work

There are several possible ways to improve the proposed data augmentation approach, that we leave for future work.

In our approach, we use the large language model to generate a new text using the prompt that contains a grammar error. The error could generally affect the quality of the generated text, which is why it could be interesting to first automatically correct the error in the clipped sentence, use the language model to generate a new version of the sentence, and then replace the corrected word with the erroneous word in the new sentence.

Another promising direction of the improvement of the augmentation approach is to apply changes not only to the right part of the error span but also to the left part. This could be done, for example by filling the masks placed on the position of random words that are not syntactically related to the words within the error span.

The amount of data generated for our experiments was based on a "the good the better" basis. However, it is also worth studying the relation between the amount of augmented data and the

#	Team ID	Precision	Recall	F1
1	ihmana	0.6244	0.6186	0.6215
2	nigula (ours)	0.6093	0.6093	0.6093
3	TMUED	0.6132	0.6047	0.6089

Table 5: **Results.** Manual evaluation by organizers.

improvement in the quality of the trained model.

## 7 Conclusion

In this paper, we present our solution for GenChal 2022 shared task dedicated to feedback comments generation to improve the English language learning experience. Our solution uses the error span based preprocessing of the learner’s text, augmentation of the dataset by clipping of the learner’s text w.r.t syntactic dependency to the words within the error span, and then the inference of large language model, using clipped text as a prompt, and finally training large T5-based model with both initial and augmented version of the dataset. Our solution took second place in this competition according to manual evaluation by organizers. The model and code of our experiments are open-sourced.

We also share the track of unsuccessful experiments and general ideas about alternative approaches to this task to prepare the ground for future researchers.

## 8 Acknowledements

This work was supported by a joint MTS-Skoltech laboratory on AI, the European Union’s Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 860621, and the Galician Ministry of Culture, Education, Professional Training, and University and the European Regional Development Fund (ERDF/FEDER program) under grant ED431G2019/04.

## References

- Øistein E. Andersen, Helen Yannakoudakis, Fiona Barker, and Tim Parish. 2013. [Developing and testing a self-assessment and tutoring system](#). In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 32–41, Atlanta, Georgia. Association for Computational Linguistics.
- Sid Black, Gao Leo, Phil Wang, Connor Leahy, and Stella Biderman. 2021. [GPT-Neo: Large](#)

- Scale Autoregressive Language Modeling with Mesh-Tensorflow. If you use this software, please cite it using these metadata.
- Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. [Automatic annotation and evaluation of error types for grammatical error correction](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.
- Leshem Choshen, Dmitry Nikolaev, Yevgeni Berzak, and Omri Abend. 2020. [Classifying syntactic errors in learner language](#). In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 97–107, Online. Association for Computational Linguistics.
- Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.
- Yaroslav Getman. 2021. Automated writing support for swedish learners. In *Swedish Language Technology Conference and NLP4CALL*, pages 21–26.
- Dimitra Gkatzia, Helen Hastie, and Oliver Lemon. 2014. [Comparing multi-label classification with reinforcement learning for summarisation of time-series data](#). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1231–1240, Baltimore, Maryland. Association for Computational Linguistics.
- Masahiro Kaneko, Sho Takase, Ayana Niwa, and Naoaki Okazaki. 2022. [Interpretability for language learners using example-based grammatical error correction](#).
- Yi-Huei Lai and Jason Chang. 2019. [TellMeWhy: Learning to explain corrective feedback for second language learners](#). 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): System Demonstrations*, pages 235–240, Hong Kong, China. Association for Computational Linguistics.
- Luís Morgado da Costa, Roger V P Winder, Shu Yun Li, Benedict Christopher Lin Tzer Liang, Joseph Mackinnon, and Francis Bond. 2020. [Automated writing support using deep linguistic parsers](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 369–377, Marseille, France. European Language Resources Association.
- Ryo Nagata. 2019. [Toward a task of feedback comment generation for writing learning](#). 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 3206–3215, Hong Kong, China. Association for Computational Linguistics.
- Ryo Nagata, Kentaro Inui, and Shin’ichiro Ishikawa. 2020. [Creating corpora for research in feedback comment generation](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 340–345, Marseille, France. European Language Resources Association.
- Ryo Nagata, Mikko Vilenius, and Edward Whittaker. 2014. [Correcting preposition errors in learner English using error case frames and feedback messages](#). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 754–764, Baltimore, Maryland. 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.
- Ildiko Pilan, John Lee, Chak Yan Yeung, and Jonathan Webster. 2020. [A dataset for investigating the impact of feedback on student revision outcome](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 332–339, Marseille, France. European Language Resources Association.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.

# A Report on FCG GenChal 2022: Shared Task on Feedback Comment Generation for Language Learners

**Ryo Nagata**

Konan University, Japan  
RIKEN, Japan  
nagata-genchal@ml.hyogo-u.ac.jp.

**Masato Hagiwara**

Octanove Labs, USA\*  
masato@octanove.com

**Kazuaki Hanawa**

RIKEN, Japan<sup>†</sup>  
k-hanawa@kodansha.co.jp

**Masato Mita**

RIKEN, Japan<sup>‡</sup>  
mita\_masato@cyberagent.co.jp

## Abstract

We report on the results of the first ever shared task on *feedback comment generation for language learners* held as Generation Challenge (GenChal) in INLG 2022, which we call FCG GenChal. Feedback comment generation for language learners is a task where, given a text and a span, a system generates, for the span, an explanatory note that helps the writer (language learner) improve their writing skills. We show how well we can generate feedback comments with present techniques. We also shed light on the task properties and the difficulties in this task, with insights into the task including data development, evaluation, and comparisons of generation systems.

## 1 Introduction

Feedback comment generation for language learners is a task where, given a text and a span, a system generates, for the span, an explanatory note that helps the writer (language learners) improve their writing skills as exemplified in Fig. 1 (for convenience, the task will be abbreviated as *feedback comment generation*, hereafter). In this regard, feedback comment generation is related to grammatical error detection and correction. In many cases, however, it is not enough to just point out an error with its correct form in order to help language learners with writing learning. Instead, it is often essential for them to explain the underlying rules. In other words, it is essential in feedback comment generation to include more information than grammatical error detection and correction provide.

\*Currently also with Earth Species Project, USA

<sup>†</sup>Currently with KODANSHA LTD., Japan

<sup>‡</sup>Currently also with CyberAgent, Inc., Japan

We report on the results of the first ever shared task on feedback comment generation held as Generation Challenge (GenChal) in INLG 2022, which we call FCG GenChal. One of the goals of this report is to reveal how well we can generate feedback comments with present techniques. There is a wide variety of choices for generation methods that are applicable to this task. Nevertheless, they have not yet been explored (at least, much less than in other generation tasks). Another goal is to shed a light on the task properties and the difficulties in this task. Specifically, we show, based on the results, insights into the task including data development, evaluation, and comparisons of generation systems.

## 2 Related Work

Generally speaking, feedback comment generation is a task of text-to-text generation. The input text, which is written by a language learner, is transformed into another text explaining the writing rules. This implies that generation methods employed in other generation tasks such as Machine Translation (MT) may be effective in the present task. For example, feedback comments often refer to words and phrases appearing in the input text, and techniques for referring to words in the source text (e.g., copy mechanisms) will likely be beneficial.

Feedback comment generation has its own unique aspects. It should be emphasized that a feedback comment is generated against a span (of the input text or sentence) whereas only a text (e.g., a sentence or utterance) is dealt with in other major text-to-text generation tasks such as MT and dialog systems. In consequence, feedback comment generation systems have to output different texts for the exact same source sentence, depending on the

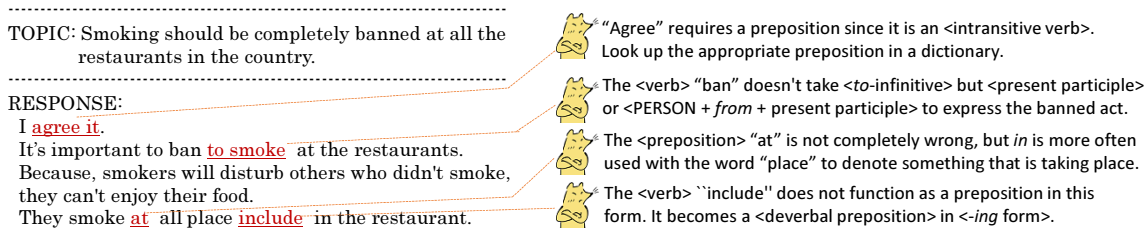


Figure 1: Example of Feedback Comments.

given spans.

The source and target languages are also unique. In this challenge, both are English, but there is room for discussion whether they fall into the same language class. The former is learner English, and inevitably it contains erroneous/unnatural words. Even within correct sentences, grammar, expressions, and style are expected to be used differently from canonical English. This brings out further research questions related to the source and target languages. For example, which is the best setting of vocabularies — only one common vocabulary for the source and target, or one for each? Does a pre-trained general (or native) language model work well to model learner English? There are a number of unaddressed research questions like these.

Feedback comment generation is also related to grammatical error detection/correction. The state-of-the-art methods typically solve the problems as sequence labeling (e.g., Kaneko et al. (2017)) or MT with DNNs (e.g., Junczys-Dowmunt et al. (2018); Napoles and Callison-Burch (2017); Rothe et al. (2021)). Recently, a DNN-based sequence labeling method is combined with symbolic transformations (Omelianchuk et al., 2020), which can be a good source of information to generate feedback comments.

Some researchers (Kakegawa et al., 2000; McCoy et al., 1996; Nagata et al., 2014) made an attempt to develop rule-based methods for diagnosing errors in line with grammatical error correction. However, this line of work suffered from the difficulty of improving coverage of errors.

More recently, researchers started to apply more modern techniques. Nagata (2019) showed that a neural-retrieval-based method was effective in preposition feedback comment generation. Lai and Chang (2019) proposed a method that used grammatical error correction and templates to generate detailed comments. Gkatzia et al. (2013) and

Gkatzia et al. (2014) proposed methods for automatically choosing feedback templates based on learning history. Hanawa et al. (2021) compared several neural-based generation methods with insights into feedback comment generation.

The availability of datasets for research in feedback comment generation has also been increasing. Nagata (2019) released a dataset consisting of feedback comments on preposition use. They marked up erroneous prepositions and annotated them with feedback comments. Nagata et al. (2020a) extended it to other grammatical errors and also other writing items such as discourse and lexical choice. Pilan et al. (2020) released a unique dataset where feedback comments on linking words were annotated.

### 3 Task Definition

#### 3.1 General Definition

This subsection describes the general task definition of feedback comment generation, which is somewhat different from the one used in FCG Genchal. The task definition that was actually used is described in Subsec. 3.2, which is a reduced version of the general definition.

In the general task definition, a unit of the input in feedback comment generation consists of a text and spans of the text. Spans, which are counted by 1-based index based on characters, correspond to where to comment. An example input text would be:

(1) *I agree it.*

as shown on the left-hand side of Figure 1. A span would be 3 to 10, which will be abbreviated as 3:10, hereafter.

The output for a span is a string that explains why the span is not good, together with the underlying rule. To make the task different from grammatical error detection/correction, the output string has to

contain more information than what grammatical error detection/correction provide. In other words, just indicating the error position, the erroneous word(s), and/or the correct form are not enough as a valid feedback comment, details of which are discussed in Subsection 3.2.

### 3.2 Task Definition Used in FCG GenChal

The above task definition is too general and abstract to be a practical one. For this reason, we put some constraints on it.

First, the target language(s) can be any language, but we limit ourselves to English input texts and English feedback comments in this challenge. As shown in Figure 1, a feedback comment is typically made about erroneous, unnatural, or problematic words in a given text so that the writer can understand why the present form is not good together with the underlying rule.

Second, we limit the target only to errors related to preposition usages, as in the examples in Figure 1. It should be emphasized that the target preposition errors involve a much wider range of errors than in the conventional definition of preposition errors (such as the one provided by ERRANT (Bryant et al., 2017)). Examples include verb phrases used as a subject (e.g., *\*Lean English is difficult.*) and comparison between a phrase and a clause (e.g., *\*because an error → because of an error*); see the work (Nagata et al., 2020b) for the details.

Third, we also limit the input to a narrower unit. Specifically, the input text always consists of only one sentence with one span. Also, they are pre-tokenized where tokens are separated by whitespace. For example, the first sentence in Figure 1 would give an input:

(2) *I agree it .* \t 3:10

where \t stands for the tab character. If a sentence contains more than one preposition error, it appears two or more times with different spans.

Under these settings, participants develop a system that automatically generates an appropriate feedback comment in English for an input sentence and a span. The length of a generated feedback comment should be less than 100 tokens. If a system cannot generate an appropriate feedback comment for a given span, it may generate the special token `<NO_COMMENT>`, which is not counted as a system output. This allows us to calculate recall, precision, and  $F_1$ , as explained below. An example output would be:

(3) *I agree it .* \t 3:10 \t **“agree” is an intransitive verb and thus it requires a preposition before its object.**

Also note that the input sentence and its span are included in the system output for evaluation convenience.

Evaluation is probably the hardest challenge in this task. We adopt automated and manual evaluation methods. In the former, we simply take BLEU between a system output and its corresponding reference (manually created feedback comment)<sup>1</sup>. In the latter, human evaluators examine whether a system output and its corresponding reference are equivalent in meaning. To be precise, a system output is regarded as appropriate if (1) it contains information similar to the reference and (2) it does not contain information that is irrelevant to the span; it may contain information that the reference does not contain as long as it is relevant to the span. This way of manual evaluation inevitably involves human subjectivity to some extent. In practice, however, the results of a pilot study show that inter-evaluator agreement is high.

The final manual evaluation measures are recall, precision, and  $F_1$ . Recall is defined as the number of appropriate system outputs divided by the number of target spans. Similarly, precision is defined as the number of appropriate system outputs divided by the number of system outputs where the special output `<NO_COMMENT>` is excluded.  $F_1$  is the harmonic mean of recall and precision.

We can do the same for BLEU. Simply, we replace the binary human judgment with the normalized, continuous BLEU value.

## 4 Data

Based on the work (Nagata, 2019; Nagata et al., 2020a), we created two versions of new datasets for this generation challenge: feedback comments written in the same language as the target (input) text (i.e., English) and in a different language (specifically, Japanese). The input texts (written by learners) are excerpts from the essays in ICNALE (Ishikawa, 2011). We had experts, who had experience in English teaching, manually annotate all preposition errors in the input texts with feedback comments in English and Japanese.

<sup>1</sup>An official score is available at the FCG GenChal Official webpage: [https://nagata-github.github.io/fcg\\_genchal/](https://nagata-github.github.io/fcg_genchal/)

Split	No. of feedback comments
Training	4,868
Development	170
Test	215

Table 1: Statistics on Datasets.

After having finished all annotations, we looked into the results. It turned out that the overall quality of the obtained data was much higher in the Japanese version than in the English. For this reason, we decided to use the Japanese version in this FCG GenChal; we translated the Japanese Feedback comments into English. Overall, it took us approximately three years to create the final datasets.

The results were split into training, development, and test sets. If a sentence contains more than one preposition error, it appears two or more times with different spans (in different lines). The split sets were provided for the participants, which are also available on the official FCG GenChal web site. Table 1 shows their statistics.

## 5 Participants and Results

### 5.1 Timeline and Summary of FCG GenChal

As shown in Figure 2, we initially had 12 registrations from seven countries. After registration, we released the training and development sets on 28 January, 2022. We let the participants have approximately four months to prepare their system.

After four months, we released the test set on 2 May, 2022. The participants had one week to prepare their generation results for final submission. In the end, seven teams submitted their results<sup>2</sup>. Four out of the seven systems are available on the

<sup>2</sup>Probably, feedback comment generation is a relatively new task and we guess four months were not enough for some teams to develop their systems.

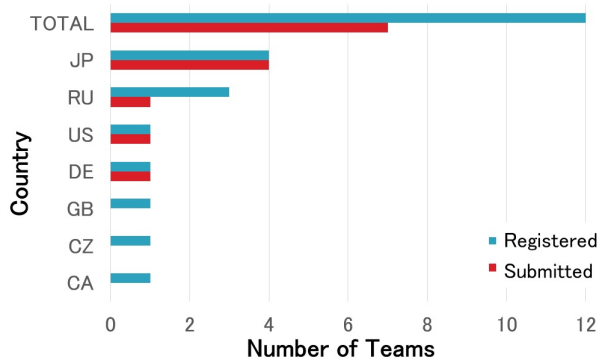


Figure 2: Statistics on Registration and Submission.

official FCG GenChal website<sup>3</sup>. Table 2 summarizes the seven systems. Also, a short description of each system is as follows:

**ihmana:** This system consists of three modules: retrieval, masking, and generation modules. The first module retrieves the instance most similar to the input learner sentence from the training data. Then, the second module masks tokens in the retrieved feedback comment that do not fit the input sentence well. Finally, the generation module generates a feedback comment given the input sentence and the retrieved, masked feedback comment. The retrieval and masking modules are based on BERT while the generation module uses a pre-trained T5 (Raffel et al., 2020). This system is capable of generating <NO\_COMMENT>.

**nigula:** This system is based on a pre-trained T5. The generator is trained on the official training set and also on an extra set obtained by data augmentation. Data augmentation is done by completing clipped input learner sentences using a language model. This system is capable of generating <NO\_COMMENT>.

**TMUUED:** This system is also based on a pre-trained T5. It takes the Part-Of-Speech labels for the input learner sentence as an extra source of information. It also uses a synonym dictionary to determine if the generation result is appropriate or not. This system is capable of generating <NO\_COMMENT>.

**kjimichi:** This system also uses a pre-trained T5 as a generator. It also uses RoBERTa (Liu et al., 2020) as a classifier to obtain grammatical term labels such as noun and preposition. The predicted grammatical term labels are used as an additional source of information in the T5 generator. This system is not capable of generating <NO\_COMMENT>.

**shotakoyama:** This system is based on GPT-2. Its approach is unique compared to the other systems in that it focuses on cleaning training data rather than improving the generation module itself. Specifically, it automatically corrects errors in feedback comment annotation such as incorrect spans. Also, it uses error type tags obtained via GECToR (Omelianchuk et al., 2020) as an extra source of information. This system is capable of generating <NO\_COMMENT>.

**stahl:** This system uses BART (Lewis et al., 2020) as a generator. It is also unique in that only

<sup>3</sup>[https://nagata-github.github.io/fcg\\_genchal/](https://nagata-github.github.io/fcg_genchal/)



Participant ID	Generator	Other Information
ihmana	T5 (t5-base)	Retrieve and masking modules: BERT (bert-base-cased)
nigula	T5 (t5-large)	Data augmentation: T5 (t5-large)
TMUUED	T5 (t5-base)	NLTK to obtain POS tags.
GU	T5 (t5-large)	Data augmentation: EleutherAI/gpt-neo-1.3B
kjimichi	T5 (t5-base)	Grammar term prediction: RoBERTa (roberta-large)
shotakoyama	GPT-2 (gpt2-large)	Data cleaning, error correction operation tags (GECToR)
stahl	BART	Clustering of training instances ( $k$ -means clustering)
Baseline	BiLSTM	—

Table 2: Summary of Participating Systems.

it exploits clustering. Specifically, before training, feedback comments in the training set are automatically grouped by clustering where TF-IDF vectors are used. This system is not capable of generating `<NO_COMMENT>`.

We ourselves implemented a baseline system for comparison. It was a text generation system based on a BiLSTM-based encoder-decoder with a copy mechanism (Hanawa et al., 2021). It is also available on the official website.

We initially had two months for manual evaluation. It actually took us approximately one month to evaluate the results of all systems including our baseline system. It took some more time to double check the evaluation results and to perform related tasks such as summarizing the results. We released the results on 25 June, 2022 as planned.

## 5.2 Results

Table 3 and Table 4 show the manual and automatic evaluation results, respectively. Both tables show a similar overall tendency. However, if we look at the details, we can see differences between them.

In BLEU-based evaluation, the system rankings are reversed compared to manual evaluation in some cases. This means that we cannot use BLEU to obtain strict system rankings as in shared tasks. We will get back to this point in Sect. 6.

In manual evaluation, the performance values tend to be larger than the corresponding automatic evaluation values. This suggests that even if  $n$ -gram overlap rate is not so high between a generated feedback and its reference, it can be judged to be appropriate by human evaluators. In other words, a feedback comment can be described by different words and phrases as expected.

Participant ID	Precision	Recall	$F_{1.0}$
ihmana	0.6244	0.6186	0.6215
nigula	0.6093	0.6093	0.6093
TMUUED	0.6132	0.6047	0.6089
GU	0.5860	0.5860	0.5860
kjimichi	0.5628	0.5628	0.5628
shotakoyama	0.5756	0.5488	0.5619
stahl	0.3581	0.3581	0.3581
Baseline	0.3116	0.3116	0.3116

Table 3: Results of Manual Evaluation.

Participant ID	Precision	Recall	$F_{1.0}$
ihmana	0.486	0.482	0.484
TMUUED	0.477	0.471	0.474
GU	0.471	0.471	0.471
nigula	0.463	0.463	0.463
kjimichi	0.460	0.460	0.460
stahl	0.437	0.437	0.437
shotakoyama	0.444	0.424	0.434
Baseline	0.334	0.334	0.334

Table 4: Results of Automatic Evaluation (BLEU).

## 6 Discussion

As shown in Sect. 5, all participating systems are based on a pre-trained, transformer-based generator while the baseline system uses a non-pre-trained BiLSTM. This partially answers one of the research questions raised in Sect. 2 (i.e., Does a pre-trained general (native) language model work on learner writings?). The results show that pre-training and/or the architectures (likely both) contribute to performance improvement, although we need more investigation to confirm this argument.

The top five systems use T5 as a generator while the rest use either GPT-2 or BART. The results prefer T5 as a generator for feedback comment

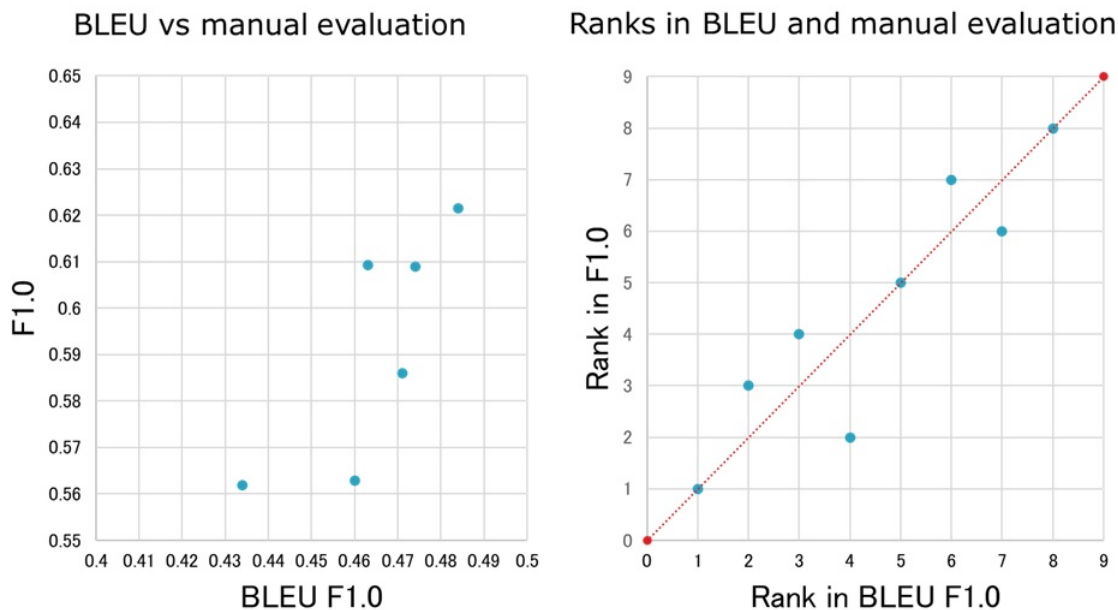


Figure 3: Comparison between Automatic (BLEU-based  $F_{1.0}$ ) and Manual ( $F_{1.0}$ ) Evaluation Results.

generation. Having said that, we need more investigations to confirm this argument, considering the amount of training, development, and test data.

Interestingly, some teams report that smaller models perform equal to, or even better than, the corresponding larger model (see their reports for the details). For example, they report that under the same condition, ‘t5-base’ achieves a better BLEU-based  $F_{1.0}$  than ‘t5-large’ while ‘bart-base’ and ‘bart-large’ achieve a similar BLEU-based  $F_{1.0}$ . A possible reason for this is that the training set is not so large and that the amount is not enough to fine-tune a large model properly. Here, one thing we should note is that these comparisons are based on BLEU because manual evaluation was applied only to the final submission results (thus, one has to use automatic evaluation (e.g., BLEU) to compare their system variants). Manual evaluation may lead to a different conclusion.

Comparison between BLEU and manual evaluation results provide an interesting insight into this task, which is summarized in Fig. 3. BLEU and manual evaluation results correlate well (correlation coefficient: 0.85). However, the system rankings differ from those by manual evaluation when the difference in BLEU is small. Specifically, even if the difference is more than 0.01 (i.e., TMUUED: 0.474 vs. nigula: 0.463), a reversal of a system ranking occurs. According to the obtained results, when the difference is larger than a certain

value (e.g., 0.02 in this case), BLEU might be a reliable measure to choose a better system (or a better method, or a better hyper-parameter setting). We need more investigations to confirm that this argument is correct. For the time-being, we do not have enough data to do so and we need manual evaluation to obtain reliable system rankings. At the same time, manual evaluation is costly and time-consuming. One of the necessary research directions is to explore more efficient ways of evaluation.

Another challenging direction is to pursue methods for generating `<NO_COMMENT>` (i.e., *not possible to generate a reliable feedback comment*). Considering practical use, it is important to decide not to generate when the system is not confident enough. In FCG GenChal, four out of the seven systems are capable of generating `<NO_COMMENT>`. Their implementations are rather simple (e.g., simple rule-based) and their effects are limited; the difference between precision and recall is rather small as shown in Table 3.

## 7 Conclusions

In this paper, we have reported on the results of a new generation challenge called *feedback comment generation for language learners*. The best-performing system achieves an  $F_{1.0}$  of 0.62 in manual evaluation. The results suggest that pre-

training and/or transformer-based methods are effective. They also suggest that smaller models within transformer-based methods perform better with the training data available. We have also reported insights into automatic and manual evaluation in feedback comment generation.

## Acknowledgements

We thank all participants for their efforts. Thanks to their work, we now have a unique testbed including datasets, system outputs, and system source codes. Without their contributions, we could not have held this generation challenge. We also thank the GenChal and INLG organizers for their support. This work was supported by JSPS KAKENHI Grant-in-Aid for Scientific Research(C) Grant Number JP22K12326 and Japan Science and Technology Agency (JST), PRESTO Grant Number JPMJPR1758, Japan. This work was partly conducted by using computational resource of AI Bridging Cloud Infrastructure (ABCI) provided by National Institute of Advanced Industrial Science and Technology (AIST).

## References

- Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. [Automatic annotation and evaluation of error types for grammatical error correction](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.
- Dimitra Gkatzia, Helen Hastie, Srinivasan Janarthnam, and Oliver Lemon. 2013. Generating student feedback from time-series data using reinforcement learning. In *Proc. of 14th European Workshop on Natural Language Generation*, pages 115–124.
- Dimitra Gkatzia, Helen Hastie, and Oliver Lemon. 2014. Comparing multi-label classification with reinforcement learning for summarisation of time-series data. In *Proc. of 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1231–1240.
- Kazuaki Hanawa, Ryo Nagata, and Kentaro Inui. 2021. [Exploring methods for generating feedback comments for writing learning](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9719–9730, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Shinichiro Ishikawa. 2011. *A new horizon in learner corpus studies: The aim of the ICNALE project*, pages 3–11. University of Strathclyde Publishing, Glasgow.
- Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018. Approaching neural grammatical error correction as a low-resource machine translation task. In *Proc. of 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 595–606.
- Jun’ichi Kakegawa, Hisayuki Kanda, Eitaro Fujioka, Makoto Itami, and Kohji Itoh. 2000. Diagnostic processing of Japanese for computer-assisted second language learning. In *Proc. of 38th Annual Meeting of the Association for Computational Linguistics*, pages 537–546.
- Masahiro Kaneko, Yuya Sakaizawa, and Mamoru Komachi. 2017. Grammatical error detection using error- and grammaticality-specific word embeddings. In *Proc. of 8th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 40–48.
- Yi-Huei Lai and Jason Chang. 2019. TellMeWhy: Learning to explain corrective feedback for second language learners. In *Proc. of 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 235–240.
- 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.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Ro{bert}a: A robustly optimized {bert} pretraining approach](#).
- Kathleen F. McCoy, Christopher A. Pennington, and Linda Z. Suri. 1996. English error correction: A syntactic user model based on principled “mal-rule” scoring. In *Proc. of 5th International Conference on User Modeling*, pages 69–66.
- Ryo Nagata. 2019. Toward a task of feedback comment generation for writing learning. In *Proc. of 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3197–3206.
- Ryo Nagata, Kentaro Inui, and Shin’ichiro Ishikawa. 2020a. Creating Corpora for Research in Feedback Comment Generation. In *Proc. of the 12th Language Resources and Evaluation Conference*, pages 340–345.

- Ryo Nagata, Kentaro Inui, and Shin'ichiro Ishikawa. 2020b. [Creating corpora for research in feedback comment generation](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 340–345, Marseille, France. European Language Resources Association.
- Ryo Nagata, Mikko Vilenius, and Edward Whittaker. 2014. Correcting preposition errors in learner English using error case frames and feedback messages. In *Proc. of 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 754–764.
- Courtney Napoles and Chris Callison-Burch. 2017. Systematically adapting machine translation for grammatical error correction. In *Proc. of 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 345–356.
- Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. [GECToR – grammatical error correction: Tag, not rewrite](#). In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA → Online. Association for Computational Linguistics.
- Ildiko Pilan, John Lee, Chak Yan Yeung, and Jonathan Webster. 2020. A Dataset for Investigating the Impact of Feedback on Student Revision Outcome. In *Proc. of 12th Language Resources and Evaluation Conference*, pages 332–339.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(1).
- Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. [A Simple Recipe for Multilingual Grammatical Error Correction](#). In *Proc. of 59th Annual Meeting of the Association for Computational Linguistics and 11th International Joint Conference on Natural Language Processing*, pages 702–707.

# Sentence-level Feedback Generation for English Language Learners: Does Data Augmentation Help?

**Shabnam Behzad**

Georgetown University

shabnam@cs.georgetown.edu

**Amir Zeldes**

Georgetown University

{amir.zeldes, nathan.schneider}@georgetown.edu

**Nathan Schneider**

## Abstract

In this paper, we present strong baselines for the task of Feedback Comment Generation for Writing Learning. Given a sentence and an error span, the task is to generate a feedback comment explaining the error. Sentences and feedback comments are both in English. We experiment with LLMs and also create multiple pseudo datasets for the task, investigating how it affects the performance of our system. We present our results for the task along with extensive analysis of the generated comments with the aim of aiding future studies in feedback comment generation for English language learners.

## 1 Introduction

Grammatical error correction has been vastly studied recently in the NLP community (Wang et al., 2021), but it is not always sufficient to merely provide the learner with a correction; in many cases, explicit feedback can facilitate the learning process. Language learners can revise improperly employed linguistic elements by reviewing feedback containing information on the error such as an explanation of why the usage is incorrect and suggestions on how to correct it. This will also help the user avoid making similar errors in the future (Pilan et al., 2020).

In this paper, we focus on preposition errors made by English language learners. Some studies have shown that the majority of syntactic errors made by English language learners are prepositional errors of substitution, omission, and addition (Lorincz and Gordon, 2012). Prepositions are challenging for language learners to master since they are highly frequent; short, unstressed and perceptually weak; and can have several different senses which may not map onto their native languages (Tyler and Evans, 2003; Morimoto and Loewen, 2007; Johansson Falck, 2015).

The task of feedback generation hasn't been explored much until recently when Nagata (2019) proposed the feedback comment generation task and a corpus (Nagata et al., 2020) and then organized the GenChal 2022: FCG (Feedback Comment Generation for Writing Learning) shared task (Nagata et al., 2021). In this task, a system generates an explanation note, given a sentence and a span that indicates the error in the sentence.

Later, Hanawa et al. (2021, 2022) explored different baselines for this task, including a neural-retrieval-based method, a pointer-generator-based seq2seq model, and a retrieve-and-edit method. For preposition-related errors, they found the pointer-generator-based seq2seq model performs the best.

In this paper, we describe our submission to GenChal 2022: FCG (Nagata et al., 2021). We use a simple encoder-decoder model to tackle the task and provide extensive analysis of the different aspects of the task. Our contributions in this paper are as follows:

- We present a simple but strong baseline for the FCG task which is currently ranked third on the leaderboard (team *GU*, BLEU score 0.472; top leaderboard score is 0.486).
- We look into data augmentation techniques and their usefulness for this task.
- We analyze samples that were marked as incorrect by human evaluators and categorize the errors made by our system.
- We further investigate the automatic evaluation metric used for the task and whether or not it is in line with human evaluations.

## 2 Experiments

### 2.1 Data

We use data provided by Nagata et al. (2021). The sentences come from essays in ICNALE (The International Corpus Network of Asian Learners of English; Ishikawa, 2013). ICNALE contains es-

says on two topics: “It is important for college students to have a part-time job” and “Smoking should be completely banned at all the restaurants in the country”.

Nagata et al. (2021) hired annotators to annotate a subset of the data for preposition errors. Annotators manually annotated all preposition errors with feedback comments in Japanese (Nagata, 2019) and later translated these comments to English for the FCG shared task. The corpus consists of 4868, 170, and 215 sentences in the train, dev, and test sets respectively. The input for the task is a sentence and a span of the text which contains the error. The output is a string that explains why the span is erroneous. Example:

*Input:* And we can put posters to remind the smokers the risks they are taking .  
37:48

*Output:* When the <verb> «remind» is used to express “ to cause someone to remember something”, “someone” is an <object> and a <preposition> needs to precede “something”. Look up the use of the <verb> «remind» in a dictionary and add the appropriate <preposition> according to the context.

If a sentence contains more than one preposition error, it can appear more than once in the training set, each time with a different span offset. We incorporate span offsets by adding special characters before and after the erroneous span before encoding the text. For example, the above input sentence becomes: *And we can put posters to remind the \*\*\* smokers the \*\*\* risks they are taking* . We do not perform any further preprocessing since the text is already processed and tokenized. We used \*\*\* as special characters in our setting but the tokenizer behaved the same way when using other characters such as {.

## 2.2 Experimental Setting

As a baseline for this task, we use T5 (Raffel et al., 2020) as our model. T5 is an encoder-decoder model built on top of the transformer architecture (Vaswani et al., 2017) which is pretrained using a combination of masked language modeling and multitask training such as summarization, machine translation, and sentiment classification.

In our experiments, we encode the essay sentences and fine-tune the model to decode feedback comments. We fine-tune *T5-Large* (770M param-

eters) with the following hyper-parameters: batch size = 8, learning rate = 0.0001 and maximum training epoch = 50.<sup>1</sup>

## 2.3 Pseudo Data

We experiment with a few other settings, trying to leverage pseudo data. To create the pseudo data, we select random sentences that are in the same corpus as the gold data (an ICNALE subset that includes correction of sentences; Ishikawa, 2018) but are not included in the FCG shared task train/dev/test sets. Since the focus of the FCG shared task is on preposition errors, we use ERRANT (Felice et al., 2016; Bryant et al., 2017) to annotate error types in these sentences. Then we keep the samples that have preposition-related errors. This gave us 544 additional sentences. To obtain comments for these new sentences, we use our fine-tuned T5 model and generate comments for these samples. We experiment with the pseudo data in two ways:

**Multi-stage fine-tuning** Fine-tune T5 on pseudo data, and then fine-tune that model on gold training data.

**Combined fine-tuning** Combine pseudo and gold data, and fine-tune T5 on the combination.

Other than experimenting with pseudo data created from the same learner corpus, we create a large pseudo dataset from other learner corpora, W&I+LOCNESS (Bryant et al., 2019; Granger, 2014). W&I (Write & Improve) is an online web platform in which users from around the world submit letters, stories, articles, and essays, and the system provides automated feedback. Some of these submissions have been further corrected by annotators. LOCNESS consists of essays written by native British and American undergraduates on different topics.

Using ERRANT, we select sentences from W&I+LOCNESS that have preposition errors. This resulted in 6,973 sentences. For the grammatical error correction task, Kiyono et al. (2019) suggests that when the amount of pseudo data and gold data is balanced, concatenating them for training works better (combined fine-tuning), but when the amount of data is unbalanced, a multi-step approach works better (multi-stage fine-tuning). Here, we investigate this by comparing conditions where the pseudo data is limited to 5,000 samples (balanced) versus conditions with all 6,973 pseudo samples (unbalanced).

<sup>1</sup><https://github.com/shabnam-b/GU-FCG-2022>

Model	Dev BLEU	Test BLEU	Human Evaluation F1 (Test)
FCG Shared Task Baseline	46.30	33.40	31.16
F/t T5 Large (No pseudo data)	<b>57.29</b>	47.11	58.60
Multi-stage f/t (ICNALE)	55.01	46.76	–
Combined f/t (ICNALE)	55.55	<b>47.25</b>	<b>61.90</b>
Multi-stage f/t (WIL, balanced)	55.46	45.95	–
Combined f/t (WIL, balanced)	57.05	46.91	61.40
Multi-stage f/t (WIL, unbalanced)	55.05	44.97	–
Combined f/t (WIL, unbalanced)	<b>57.29</b>	45.36	–

**Table 1:** Comparison of models on dev and test sets. *WIL* refers W&I+LOCNESS. The gold training data on which T5 is fine-tuned contains 4,868 samples. The multi-stage fine-tuning and combined fine-tuning conditions make use of data augmentation, supplementing the gold training data with pseudo data. The pseudo data consists of 5,000 samples in the balanced setting and 6,973 samples in the unbalanced setting. There are 170 and 215 samples in the dev and test sets, respectively. Best scores in each column are bolded.

### 3 Results and Analysis

Results of our experiments are available in Table 1. We compared against the official shared task baseline system, which was an encoder-decoder with a copy mechanism based on a pointer generator network.

#### 3.1 Automatic Evaluation

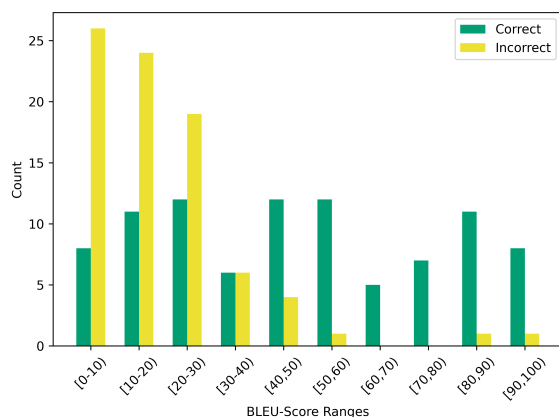
We use test set BLEU scores to compare all the conditions in Table 1. On this metric, all systems based on T5 give improvements of 12+ points over the official baseline. The gain for our best model (which uses pseudo data from ICNALE for combined fine-tuning) is almost 14 points.

**Multi-stage vs. combined fine-tuning** In all our experiments, *Combined f/t* showed better performance compared to *Multi-stage f/t* (by a difference of 1 BLEU point or less).

**Balanced vs. unbalanced** In our experimental setup, using a larger pseudo dataset hurt the performance in both *Combined f/t* and *Multi-stage f/t* settings. One possible explanation is the amount of noise that is being introduced to the system by pseudo data. Creating pseudo data with different techniques might show different results.

#### In-domain vs. out-of-domain pseudo data

Even though our in-domain pseudo data was very small (544 sentences), it was more effective than larger amounts of out-of-domain pseudo data. An intuitive explanation for this case is that ICNALE contains essays on only two specific topics: “It is important for college students to have a part-time job” and “Smoking should be completely banned at all the restaurants in the country”. Since the FCG shared task test set comes from ICNALE, a more



**Figure 1:** Comparison of human evaluations (correct or incorrect system generated feedback comment) with automatic evaluation metric (BLEU score)

general model fine-tuned on pseudo data from other corpora might not necessarily perform well on this test set. It seems likely that the model trained on multiple datasets would be more robust in realistic settings testing on other domains.

#### 3.2 Human Evaluation

Shared task organizers provided us with the human evaluation of three of our systems (4th column in Table 1). In this evaluation, each system output is compared to the corresponding reference. System output is regarded as appropriate if the following criteria are met: A) it contains information similar to the reference and B) it does not contain information that is irrelevant to the erroneous span. The performance is measured by recall, precision, and F1 based on correct/incorrect outputs.<sup>2</sup>

Based on this human evaluation, our best model achieved an F1 score of 61.09 (this was not our

<sup>2</sup><https://fcg.sharedtask.org/task/>

official submission to the shared task, but falls just behind the top leaderboard score<sup>3</sup> of 62.15). Comparing the performance of different systems, human evaluation results and test set BLEU scores seem to be consistent. We investigate this further for our top system, comparing human labels (correct or incorrect) with the BLEU score for each sample in the test set. Results are available in Figure 1. Based on this analysis, when BLEU score is higher than 60%, it is mostly in line with the human evaluations. We also observe that about 49 samples (23% of the test set) are indeed correct, but get a BLEU score below 50. This is due to system-generated comments not having much overlap with the gold feedback comment, despite being correct (Sulem et al., 2018; Nema and Khapra, 2018).

Lastly, we look at 50% of samples where the model-generated comment was labeled as incorrect in human evaluation. We observed that generated comments are very fluent and follow the templates FCG annotators used. In cases where the system output was labeled as incorrect, some of the patterns we observed are as follows:

*Completely incorrect comment (≈54%):* The model’s generated comment includes incorrect suggestions and explanations (first and second example in Table 2). Interestingly, we noticed that the model made the same wrong suggestion in different sentences, containing the same type of error (for example, usage of “on” when it means *sticking to, or hanging from a surface* such as “on the door”, “on the wall”). Possible explanations for these cases are that 1) similar errors were not seen during training and 2) in most cases, the sentence contains other errors within the same span or nearby tokens, which presumably makes it hard for the model to understand what the learner was trying to say.

*Correct explanation, but incorrect suggestion (≈22%):* In these cases, the model gives the right correction, but the explanation is incorrect or incomplete (third example in Table 2).

*Correct suggestion, but incorrect evaluation (≈14%):* In many cases, the model gives the correct suggestion but the comment starts with something along the lines of “It is not grammatically incorrect to use the ...”, even though the usage is indeed incorrect (fourth example in Table 2).

*Human annotation errors (≈12%):* In a few cases, we believe the system-generated comment is correct, but wrongly labeled as incorrect.

Looking at some positive examples, there are many cases where the model generates completely valid comments. In 19% of cases (41 samples), the model generates exactly the same comment as the reference. In all of these instances, the exact comment was seen during training. There were another 51 comments in the test set that were seen during training, and the model was able to generate a correct comment (but not exactly the same) in 38 cases of those. In many cases, the system output has minor differences compared to the gold output but there are also cases where the generated comment is completely different. Examples appear in Table 3.

## 4 Discussion

In this paper, we presented a simple baseline for sentence-level feedback generation for English language learners. We investigated the effect of using pseudo data for the task and provided an analysis of system outputs and metrics used.

**Does data augmentation help?** Incorporating pseudo data gave us very slight BLEU score improvements (0.14) over the T5 model with no pseudo data. But in human evaluations, we observe a 3.3% improvement in F1. Our experiments show that pseudo data could potentially improve results if carefully created and incorporated. Studies on similar tasks such as grammatical error correction have shown that there are many important factors when creating pseudo data such as choice of learner corpora, the method of generating the pseudo data, and error tendency of learners (Kiyono et al., 2019; White and Rozovskaya, 2020; Takahashi et al., 2020). Future studies should further investigate similar important factors for this task and their effectiveness.

**Is this a general evaluation of grammatical error feedback capability?** Even though our model outperforms the baseline for this task by a large margin, we think it is not a good indicator of current models’ performance for the feedback comment generation task in general, due to the following limitations: 1) The topics covered in the train/dev/test sets are very limited and only related to smoking and part-time jobs. Considering the topics and the fact that learners may have a limited vocabulary in the target language, the model is likely performing well because it has seen most of the errors during training, not because it has developed the capacity to recognize and comment on grammat-

<sup>3</sup>As of 14 December 2022



Input Sentence	Gold Output	System Output
Where there is food <b>on</b> the smell of cigarette smoke , food soaks Damage to people who go around If you smoke , you know everyone is very often not a good point .	Use the <preposition> 'with' instead of «on» to express "food with the smell of cigarettes".	The <preposition> «on» is not used with 'smell' as it is meant to indicate a connection. Use the <preposition> 'in' instead.
The direct means when you smoking , some smoke and chemical access <b>to</b> your body after that all bad diseases will show the results .	Since the <verb> «access» is a <transitive verb>, a <preposition> is not necessary.	«To» is not the <preposition> used with the <noun> «access» to indicate a point of arrival. Look up the <noun> «access» in the dictionary and find the appropriate <preposition>.
And , it is the best <b>of</b> us to acquire various skills , manners without lazing about study .	<For + noun> should be used as the subject of a <to-infinitive> describing the <formal subject> «it».	«Of» is not the <preposition> used with the <adjective> «best» to indicate the recipient of an action. Think of a <preposition> that indicates destination.
We are as engineer usually work in an industry or <b>on</b> a factory , we use our attitude , knowledge to get a job which is comfortable with us .	When the <verb> «work» is followed by the <preposition> «on», it means "to spend time repairing or improving something." Use the <preposition> "in" or "at" to indicate where you work.	It is not incorrect to use the <preposition> «on» in this case, but «on» describes the state of being physically inside the factory building. Use 'in' instead to refer to business as a situation or an organization".

**Table 2:** Examples of incorrect system-generated comments. Error span tokens are bold and in red.

Input Sentence	Gold Output	System Output
Smoking sometimes makes people uncomfortable , so smoking should be banned <b>at</b> the public place like the restaurants .	The <preposition> «at» may be used with «place» to indicate a place where something takes place, but it is more common to use 'in' instead.	The <preposition> «at» may be used with «place» to indicate a place where something takes place, but it is more common to use 'in' instead.
In the past , most <b>of</b> college students probably had no part-time job	A group of something specific follows «most of». Use «most» as an <adjective> rather than a <noun> when simply referring to students in general.	«Most of» is followed by a <pronoun> or a <noun> with a <determiner> such as 'the', 'these', or 'my'. Otherwise, «most» comes just before the <noun> to be qualified as an <adjective>.
We meet new people and learn about service for customer , how difficult or hard to get money before we <b>graduate school</b> .	«Graduate» needs to be followed by a <preposition> when used as an <intransitive verb>.	Since the <verb> «graduate» is an <intransitive verb>, the <object> requires a <preposition>. Look up the use of the <verb> «graduate» in the dictionary.
Parents worry <b>about</b> they waste time and hurt themselves .	The <preposition> «about» follows the <verb> «worry» when a <noun> follows. When there is a <clause> in the structure of <S + V>, use a <that-clause>.	The <verb> «worry» is a <transitive verb> that takes a <that-clause> as an <object> which does not require a <preposition>.

**Table 3:** Examples of correct system-generated comments. Error span tokens are bold and in red.

ical errors in general. 2) The task is focused on preposition-related errors, which makes the errors and comments even more limited than in realistic settings. 3) Most of the comments follow a specific template, which made it easier for the model to learn the patterns. 4) Many reference comments consist of boilerplate—very general suggestions such as *Look up the use of the <verb> «prohibit» in a dictionary and rewrite the sentence using the appropriate structure.* The model also generates many such boilerplate sentences. Of course, a general correct comment is better than a detailed incorrect comment, but we think a model that is able to give more specific suggestions would be of more use to learners.

With these limitations in mind, we still believe this could be a first step toward better and more robust feedback comment generation systems, and we view the organization of the shared task and the release of the data as important milestones for making progress in this research area.

## 5 Acknowledgements

We thank the shared task organizers and anonymous reviewers for their insightful comments. This research was supported in part by NSF award IIS-2144881.

## References

- Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. [The BEA-2019 shared task on grammatical error correction](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. Association for Computational Linguistics.
- Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. [Automatic annotation and evaluation of error types for grammatical error correction](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.
- Mariano Felice, Christopher Bryant, and Ted Briscoe. 2016. [Automatic extraction of learner errors in ESL sentences using linguistically enhanced alignments](#). In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 825–835, Osaka, Japan. The COLING 2016 Organizing Committee.
- Sylviane Granger. 2014. [The computer learner corpus: a versatile new source of data for SLA research](#). In *Learner English on computer*, pages 3–18. Routledge.
- Kazuaki Hanawa, Ryo Nagata, and Kentaro Inui. 2021. [Exploring methods for generating feedback comments for writing learning](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9719–9730, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Kazuaki Hanawa, Ryo Nagata, and Kentaro Inui. 2022. [Analyzing methods for generating feedback comments for language learners](#). *Journal of Natural Language Processing*, 29(3):901–924.
- Shinichiro Ishikawa. 2018. [The ICNALE edited essays: A dataset for analysis of L2 English learner essays based on a new integrative viewpoint](#). *English Corpus Studies*, 25:117–130.
- Shin’ichiro Ishikawa. 2013. The ICNALE and sophisticated contrastive interlanguage analysis of Asian learners of English. *Learner corpus studies in Asia and the world*, 1(1):91–118.
- Marlene Johansson Falck. 2015. [Linguistic theory and good practice: How cognitive linguistics could influence the teaching and learning of English prepositions](#). Lindgren, Eva, & Janet Enever (ed.), *Språkdidaktik: researching language teaching and learning (pp. 61-73)*. Umeå: Umeå Universitet.
- Shun Kiyono, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui. 2019. [An empirical study of incorporating pseudo data into grammatical error correction](#). 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 1236–1242, Hong Kong, China. Association for Computational Linguistics.
- Kristen Lorincz and Rebekah Gordon. 2012. [Difficulties in learning prepositions and possible solutions](#). *Linguistic Portfolios*, 1(1):14.
- Shun Morimoto and Shawn Loewen. 2007. [A comparison of the effects of image-schema-based instruction and translation-based instruction on the acquisition of L2 polysemous words](#). *Language Teaching Research*, 11(3):347–372.
- Ryo Nagata. 2019. [Toward a task of feedback comment generation for writing learning](#). 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 3206–3215, Hong Kong, China. Association for Computational Linguistics.
- Ryo Nagata, Masato Hagiwara, Kazuaki Hanawa, Masato Mita, Artem Chernodub, and Olena Nahorna. 2021. [Shared task on feedback comment generation for language learners](#). In *Proceedings of the 14th*

*International Conference on Natural Language Generation*, pages 320–324, Aberdeen, Scotland, UK. Association for Computational Linguistics.

*Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 198–208, Seattle, WA, USA → Online. Association for Computational Linguistics.

Ryo Nagata, Kentaro Inui, and Shin'ichiro Ishikawa. 2020. [Creating corpora for research in feedback comment generation](#). In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 340–345, Marseille, France. European Language Resources Association.

Preksha Nema and Mitesh M. Khapra. 2018. [Towards a better metric for evaluating question generation systems](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3950–3959, Brussels, Belgium. Association for Computational Linguistics.

Ildiko Pilan, John Lee, Chak Yan Yeung, and Jonathan Webster. 2020. [A dataset for investigating the impact of feedback on student revision outcome](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 332–339, Marseille, France. European Language Resources Association.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018. [BLEU is not suitable for the evaluation of text simplification](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 738–744, Brussels, Belgium. Association for Computational Linguistics.

Yujin Takahashi, Satoru Katsumata, and Mamoru Komachi. 2020. [Grammatical error correction using pseudo learner corpus considering learner's error tendency](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 27–32, Online. Association for Computational Linguistics.

Andrea Tyler and Vyvyan Evans. 2003. *The semantics of English prepositions: Spatial scenes, Embodied meaning, and Cognition*. Cambridge University Press.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). *Neural Information Processing Systems (NIPS)*.

Yu Wang, Yuelin Wang, Kai Dang, Jie Liu, and Zhuo Liu. 2021. [A comprehensive survey of grammatical error correction](#). *ACM Transactions on Intelligent Systems and Technology (TIST)*, 12(5):1–51.

Max White and Alla Rozovskaya. 2020. [A comparative study of synthetic data generation methods for grammatical error correction](#). In *Proceedings of the*

# Retrieval, Masking, and Generation: Feedback Comment Generation using Masked Comment Examples

Mana Ihori and Hiroshi Sato and Tomohiro Tanaka and Ryo Masumura

NTT Computer and Data Science Laboratories, NTT Corporation

1-1 Hikarinooka, Yokosuka-Shi, Kanagawa 239-0847, Japan

mana.ihori@ntt.com

## Abstract

In this paper, we propose a novel method, retrieval, masking, and generation, for feedback comment generation. Feedback comment generation is a task in which a system generates feedback comments such as hints or explanatory notes for language learners, given input text and position showing where to comment. In the conventional study, the retrieve-and-edit method for retrieving feedback comments in the data pool and editing the comments has been thought effective for this task. However, the performance of this method does not perform as well as other conventional methods because its model learns to edit tokens that do not need to be rewritten in the retrieved comments. To mitigate this problem, we propose a method for combining retrieval, masking, and generation based on the retrieve-and-edit method. Specifically, tokens of feedback comments retrieved from the data pool are masked, and this masked feedback comment is used as a template to generate feedback comments. The proposed method should prevent unnecessary conversion by using not retrieved feedback comments directly but masking them. Our experiments on feedback comment generation demonstrate that the proposed method outperforms conventional methods.

## 1 Introduction

Feedback comment generation is a task in which, given an input text and position that shows where to comment, a system generates feedback comments such as hints or explanatory notes for non-native language learners (Nagata, 2019). In this task, it is not enough to simply point out or correct errors; the system should also explain why they are wrong. Such a system would be extremely beneficial for language learners, but there is currently no effective method for generating comments.

Conventionally, retrieval-based (Nagata, 2019), simple generation (See et al., 2017), and retrieve-and-edit (Hashimoto et al., 2018) methods have

been used for the feedback comment generation task. With the retrieval-based method, the method retrieves a feedback comment from the data pool with maximum cosine similarity. Although the modeling for this method is simple, the output is not flexible; for example, this method often retrieves feedback comments that were correct as hints or explanatory notes but focused on tokens that are not in the input text. The simple generation method generates feedback comments directly given the input text and position using an encoder-decoder model. Since this method can generate flexible feedback comments, it mitigates the problem with the retrieval-based method. However, generating comments with simple generation is more difficult than with the retrieval-based method because this method should generate comments from scratch. Thus, Hanawa et al. (2021) used the retrieve-and-edit method that combines these two methods for better performance. With this method, the feedback comments retrieved with the retrieval-based method are edited using the simple generation method. However, the experimental results indicated that the retrieve-and-edit method did not perform well because its model learned excessive conversions and converted unnecessary tokens (Hanawa et al., 2021).

To mitigate this problem, we consider extending the retrieve-and-edit method with which unnecessary tokens in the retrieved feedback comments are actively edited. Our idea to specify where to edit the retrieved feedback comment is masking tokens that should be edited. By masking tokens that should be edited, the method should be able to predict only masked tokens and not change the other tokens in the retrieved feedback comments. Specifically, we mask tokens in retrieved feedback comments obtained with the retrieval-based method, and the method outputs a feedback comment given this masked feedback comment and the input text. We can create a template for comment generation,

as the tokens of the retrieved feedback comment that are irrelevant to the input text are eliminated.

In this paper, we propose a novel method, *retrieval, masking, and generation*, for feedback comment generation. It consists of three modules: retrieval, masking, and generation. First, the retrieval module retrieves a feedback comment from the data pool, as with the conventional retrieval-based method. Next, the masking module executes binary classification, i.e., masking or not, for each token in the retrieved feedback comment. The masking module learns to mask tokens of the retrieved comment that are not in the reference feedback comment. Finally, the generation module generates a feedback comment given the input text and masked feedback comment. To generate feedback comments, the proposed method cascades the results of each module, which means that the performance of each module depends on the subsequent performance of the module output. To mitigate this potential problem, we introduce a multi-decoding operation that uses not only the top result but also the top  $k$  results. Our experiments on feedback comment generation demonstrate that the proposed method performs better than the above three conventional methods.

## 2 Feedback Comment Generation

In feedback comment generation, given the input text that has grammatical errors and position, a system outputs a feedback comment. We define the input text as  $\mathbf{X} = \{x_1, \dots, x_M\}$  and feedback comment as  $\mathbf{Y} = \{y_1, \dots, y_N\}$ , where  $x_m$  and  $y_n$  are tokens, and  $M$  and  $N$  are the number of tokens in the input text and feedback comment, respectively. The position represents the range of the feedback target in character units and consists of integers. We make  $\tilde{\mathbf{X}}$  emphasize the target characters by adding brackets to the input text on the basis of the given position. For example, when  $\mathbf{X}$  is ‘‘He agrees the opinion.’’ and the position is 3:13,  $\tilde{\mathbf{X}}$  is ‘‘He «agrees the» opinion.’’ Note that in  $\tilde{\mathbf{X}}$ , the input text and position are not treated separately, but the position is also treated as a token.

## 3 Retrieval, Masking, and Generation Modules

The proposed method consists of three modules: retrieval, masking, and generation. Figure 1 shows an overview of the proposed method.

### 3.1 Retrieval Module

The retrieval module outputs a candidate of feedback comments, given the input text, position, and data pool of feedback comments. To obtain the candidate, the module calculates the cosine similarity between vectors of the input text  $\mathbf{X}$  and feedback comment  $\mathbf{Y}$  selected from the data pool. Since  $\mathbf{X}$  does not include the position information, we add it by converting the position into position label  $\mathbf{P} = \{p_1, \dots, p_M\}$ . The  $p_m \in \{0, 1\}$  is the  $m$ -th binary label represented by 1 if each token in the input text is in the position range, and 0 otherwise. For example, when  $\mathbf{X}$  is {He, agrees, the, opinion, .} and the position is 3:13,  $\mathbf{P}$  is {0, 1, 1, 0, 0}. Therefore, given the  $\mathbf{X}$ ,  $\mathbf{P}$ , and selected  $\mathbf{Y}$ , the module outputs the cosine similarity between  $\mathbf{X}$  and  $\mathbf{Y}$ , as

$$s = \text{ret}(\mathbf{X}, \mathbf{P}, \mathbf{Y}; \Theta_{\text{ret}}), \quad (1)$$

where  $\text{ret}()$  is the function of the retrieval module and  $\Theta_{\text{ret}}$  is a trainable parameter set.

This module is constructed using a Transformer-based encoder model. First, the module encodes  $\mathbf{X}$  and  $\mathbf{Y}$  into hidden representations  $\mathbf{Q} = \{q_1, \dots, q_M\}$  and  $\mathbf{R} = \{r_1, \dots, r_N\}$  as

$$\mathbf{Q} = \text{TransformerEncoder}(\mathbf{X}; \Theta_{\text{ret}}), \quad (2)$$

$$\mathbf{R} = \text{TransformerEncoder}(\mathbf{Y}; \Theta_{\text{ret}}), \quad (3)$$

where  $\text{TransformerEncoder}()$  is the Transformer encoder that consists of an embedding layer, scaled dot product multi-head self-attention layer, and position-wise feed-forward network (Vaswani et al., 2017). We also use a pre-trained BERT that performs well for various natural language understanding tasks (Devlin et al., 2018) as  $\text{TransformerEncoder}()$  in the module.

Next, it multiplies  $\mathbf{Q}$  and  $\mathbf{P}$  to convert the hidden representations into a single vector  $\mathbf{u}$  as

$$\mathbf{u} = \sum_{m=1}^M \mathbf{q}_m \cdot p_m. \quad (4)$$

We also use  $r_1$ , which is the embedding of the [CLS] token at the beginning of the feedback comment as a single vector of  $\mathbf{R}$ .

Finally, we calculate the cosine similarity of  $\mathbf{u}$  and  $r_1$  as

$$\text{ret}(\mathbf{X}, \mathbf{P}, \mathbf{Y}; \Theta_{\text{ret}}) = \frac{\mathbf{u} \cdot r_1}{\|\mathbf{u}\| \|r_1\|}. \quad (5)$$

The module outputs a candidate of the feedback comment that has the highest similarity in the data pool.

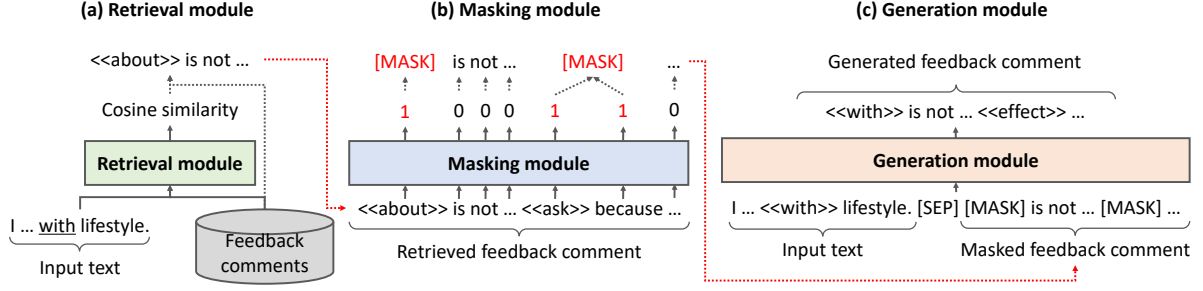


Figure 1: Overview of proposed method. Underlined words in input text represent position.

**Training:** The retrieval module is trained using dataset  $\mathcal{D}_{\text{ret}} = \{(X^1, P^1, \mathcal{Y}, \mathcal{S}^1), \dots, (X^C, P^C, \mathcal{Y}, \mathcal{S}^C)\}$ . Here,  $C$  is the number of the input text and position,  $\mathcal{Y} = \{Y^1, \dots, Y^D\}$  is the data pool of feedback comments,  $\mathcal{S}^c = \{s^{c,1}, \dots, s^{c,D}\}$  is the sets of Levenshtein similarities (Levenshtein et al., 1966) for  $c$ -th reference feedback comment  $Y^c$ , and  $s^{c,d}$  is the Levenshtein similarity of  $Y^c$  and selected comment  $Y^d$ . This module is trained to approach the cosine similarity of  $X^c$  and  $Y^d$  for  $s^{c,d}$  using mean squared error. The training loss function  $\mathcal{L}_{\text{ret}}$  is defined as

$$\mathcal{L}_{\text{ret}} = \frac{1}{C \cdot D} \sum_{c=1}^C \sum_{d=1}^D (s^{c,d} - \text{ret}(X^c, P^c, Y^d; \Theta_{\text{ret}}))^2, \quad (6)$$

**Retrieval:** The retrieval module outputs a candidate of feedback comment  $\tilde{Y}$  with the highest similarity per input text from the data pool  $\mathcal{Y}$  as

$$\tilde{Y} = \arg \max_{Y \in \mathcal{Y}} \text{ret}(X, P, Y; \Theta_{\text{ret}}). \quad (7)$$

### 3.2 Masking Module

The masking module executes binary classification, i.e., masking or not, for each token in the retrieved feedback comment that is output of the retrieval module. Given the  $X$ ,  $P$ , and  $\tilde{Y}$ , the module outputs the masking label  $L = \{l_1, \dots, l_N\}$  where  $l_n \in \{0, 1\}$  is the  $n$ -the binary label as

$$P(L|X, P, \tilde{Y}; \Theta_{\text{mask}}) = \prod_{n=1}^N P(l_n|X, P, \tilde{Y}; \Theta_{\text{mask}}), \quad (8)$$

where  $\Theta_{\text{mask}}$  is the trainable parameter set.

This module has a common architecture with the retrieval module. Thus, we convert  $X$  and  $P$  into

a single vector  $u$  and convert  $Y$  into hidden representations  $R = \{r_1, \dots, r_N\}$  with Eqs. (2), (3), and (4). Next, we compute a binary classification for each token in the retrieved comment by using these vectors as

$$P(l_n|X, P, \tilde{Y}; \Theta_{\text{mask}}) = \text{softmax}(v_n; \Theta_{\text{mask}}), \quad (9)$$

$$v_n = [u^T, r_n^T]^T, \quad (10)$$

where,  $\text{softmax}()$  is a linear transformational function with a softmax activation.

The final output is a masked feedback comment with masked tokens labeled 1. The module masks tokens of the retrieved comment that are not in the reference feedback comment. It also replaces tokens that should be masked with the special token [MASK]. The generation module should then predict the same number of tokens as the number of masked tokens; however, the numbers of masked tokens and tokens that should be generated are not necessarily the same. Therefore, we use span-mask denoising (Raffel et al., 2020), which replaces consecutive tokens with a single special token.

**Training:** The masking module is trained using dataset  $\mathcal{D}_{\text{mask}} = \{(X^1, P^1, \mathcal{Y}, \mathcal{L}^1), \dots, (X^C, P^C, \mathcal{Y}, \mathcal{L}^C)\}$ , where  $C$  is the number of input text and position, and  $\mathcal{L}^c = \{L^{c,1}, \dots, L^{c,D}\}$  is the label sets. Label set  $L^{c,d}$  is a set with label 1 for tokens of the  $Y^d$  selected from the data pool that are not in the reference feedback comment  $Y^c$  and 0 for all others. The training loss function  $\mathcal{L}_{\text{mask}}$  is defined as

$$\mathcal{L}_{\text{mask}} = - \sum_{c=1}^C \sum_{d=1}^D \log P(L^{c,d}|X^c, P^c, Y^d; \Theta_{\text{mask}}). \quad (11)$$

**Masking:** The decoding problem is defined as

$$\hat{L} = \arg \max_L P(L|X, P, \tilde{Y}; \Theta_{\text{mask}}). \quad (12)$$

Finally, the module outputs a masked feedback comment  $\tilde{Y}$  using  $\hat{L}$  and  $\tilde{Y}$  as

$$\tilde{Y} = \text{MASK}(\tilde{Y}, \hat{L}), \quad (13)$$

where MASK() is the operation to mask tokens of the retrieved feedback comment using span-mask denoising. Note that when the output results are all 0 (no tokens are masked), we directly use the retrieved comment as the feedback comment.

### 3.3 Generation Module

The generation module outputs  $Y$  given the input text  $\tilde{X} = \{\tilde{x}_1, \dots, \tilde{x}_m\}$  and masked feedback comment  $\tilde{Y} = \{\tilde{y}_1, \dots, \tilde{y}_n\}$ . The input is the concatenated sequence of the input text and masked comment with a separator token,  $Z = \{\tilde{x}_1, \dots, \tilde{x}_m, [\text{SEP}], \tilde{y}_1, \dots, \tilde{y}_n\}$ . The generation probability of  $Y$  is defined as

$$P(Y|Z; \Theta) = \prod_{n=1}^N P(y_n|y_{1:n-1}, Z; \Theta_{\text{gen}}), \quad (14)$$

where  $\Theta_{\text{gen}}$  is the trainable parameter set.

This module is constructed using a Transformer-based encoder-decoder model. First, the encoder converts the  $Z$  into hidden representations  $H$  as

$$H = \text{TransformerEncoder}(Z; \Theta_{\text{gen}}). \quad (15)$$

Next, the decoder computes the generation probability of a token from the preceding tokens and the  $H$ . The predicted probabilities of the  $n$ -th token  $y_n$  are calculated as

$$P(y_n|y_{1:n-1}, Z; \Theta_{\text{gen}}) = \text{softmax}(w_n; \Theta_{\text{gen}}). \quad (16)$$

The hidden representations  $w_n$  are calculated using  $H$  and  $y_{1:n-1} = \{y_1, \dots, y_{n-1}\}$  as

$$w_n = \text{TransformerDecoder}(y_{1:n-1}, H; \Theta_{\text{gen}}), \quad (17)$$

where TransformerDecoder() is the Transformer decoder that consists of an embedding layer, scaled dot product multi-head self-attention and source target attention layers, and a position-wise feed-forward network (Vaswani et al., 2017).

**Training:** The generation module is trained using dataset  $\mathcal{D}_{\text{gen}} = \{(Z^1, Y^1), \dots, (Z^{|\mathcal{D}_{\text{gen}}|}, Y^{|\mathcal{D}_{\text{gen}}|})\}$ . The training loss function  $\mathcal{L}_{\text{gen}}$  is defined as

$$\mathcal{L}_{\text{gen}} = - \sum_{(Z, Y) \in \mathcal{D}_{\text{gen}}} \log P(Y|Z; \Theta_{\text{gen}}). \quad (18)$$

**Decoding:** The decoding problem is defined as

$$\hat{Y} = \arg \max_Y P(Y|Z; \Theta_{\text{gen}}). \quad (19)$$

### 3.4 Multi-Decoding

Since the proposed method cascades the output of three modules, the performance of each module is directly related to that of the next module. However, it is difficult to fully guarantee the output of each module. When only the top output is used, we might not be able to take full advantage of each module. To mitigate this problem, we use multi-decoding to generate a feedback comment with high confidence.

In the multi-decoding operation, the top  $k$  feedback comments per input text are first retrieved using the retrieval module. Next, masking is executed on each token of  $k$  retrieved comments using the masking module. Then, the duplicated ones for the same input text are excluded. When only the unmasked tokens remain, the comment is stored as a candidate feedback comment. Next, given each masked feedback comment and the input text, the generation module outputs the feedback comment per masked comment. As a result of this process, multiple feedback comments are generated for a single input text, including the candidates that are outputted in the masking module. Thus, we extract only one feedback comment from these comments using Algorithm 1. In this algorithm, lev() is the function to calculate Levenshtein similarity.

In the feedback comment generation task, the special output token <NO\_COMMENT> indicates that a system cannot generate any reliable feedback comment. In this study, unreliable feedback comments are converted into <NO\_COMMENT> in accordance with the following rules.

- Feedback comments in which bracketed tokens in the comment are not in the input text.
- Feedback comments include “ the preposition is not necessary,” but the input text does not have the preposition.

---

**Algorithm 1** Multi-decoding operation

---

**Require:**  $X$ , candidates =  $\{\hat{Y}^1, \dots, \hat{Y}^k\}$ 

```
1: all_ave = 0, all_cnt = 0
2: for  $i \leftarrow 0$  to  $k$  do
3:   ave = 0, cnt = 0
4:   for  $j \leftarrow 0$  to  $k$  do
5:     if  $i \neq j$  then
6:       ave = ave + lev( $\hat{Y}^i, \hat{Y}^j$ )
7:       if  $\hat{Y}^i = \hat{Y}^j$  then
8:         cnt = cnt + 1
9:       end if
10:    end if
11:  end for
12:  ave = ave + lev( $\hat{Y}^i, X$ )
13:  ave = ave/ $k$ 
14:  if all_cnt < cnt then
15:    all_cnt = cnt
16:     $Y_c = \hat{Y}^i$ 
17:    flag = 0
18:  else if cnt = all_cnt then
19:    flag = 1
20:  end if
21:  if all_ave < ave then
22:    all_ave = ave
23:     $Y_a = \hat{Y}^i$ 
24:  end if
25: end for
26: if flag = 1 then
27:    $Y = Y_a$ 
28: else
29:    $Y = Y_c$ 
30: end if
31: return  $Y$ 
```

---

## 4 Experiments

### 4.1 Datasets

We used a dataset provided by Generation Challenge 2022 that contains input text, position, and feedback comments. The dataset has 4,868 sentences in a training set, 170 sentences in a validation set, and 215 sentences in a test set. In the dataset, the errors in the input text only cover preposition uses. The three modules of the proposed method require individual datasets for training; thus, we created datasets for each module from this provided dataset.

**Retrieval module:** The dataset for the retrieval module consists of the input text, position, feedback comment selected from the data pool, and Levenshtein similarity. The input text and position are the same as the provided dataset. Also, the Levenshtein similarity is calculated using reference feedback comments and selected comments from the data pool (feedback comments in the training data of the provided dataset). In the provided dataset, there are many low-similarity combinations of the reference feedback comment and selected comments. Thus, the dataset would be unbalanced if we used all combinations for training. To prevent this problem, we randomly removed samples so there would be less than five with the same first decimal place value of similarity. Finally, we divided the dataset into 139,687 sentences in a training set and 5,001 sentences in a validation set.

**Masking module:** The dataset for the masking module consists of the input text, position, feedback comments selected from the data pool, and masking labels. We use the sets of input text, position, and selected comments that were created in the retrieval module for the masking module to prevent data imbalance. To make masking labels, we took a word-by-word alignment for all comments and reference feedback comments. We then labeled tokens that were not in the reference feedback comments as 1 and others as 0 for the selected comments. Finally, there were 139,687 sentences in a training set and 5,001 sentences in a validation set, the same as the dataset of the retrieval module.

**Generation module:** The dataset for the generation module consists of the sequence that was concatenated with the input text and masked feedback comments, and reference feedback comments. The input text and masked feedback comment are connected using a separator token [SEP]. To create the dataset, we used the datasets of the retrieval and masking modules. First, we extracted the top five and five random feedback comments per input text by using the Levenshtein similarity in the dataset of the retrieval module. Next, we extracted the masked feedback comments that correspond to the above ten feedback comments from the dataset of the masking module. Then, we deleted the duplicated masked comments. Finally, these input text and masked feedback comments were concatenated into a single sequence and paired with the reference comment. In addition, we divided the



dataset into 48,309 sentences in a training set and 1,000 sentences in a validation set.

## 4.2 Setup

We implemented the proposed method (with all three modules); the retrieval-and-generation method, which is the proposed method without the masking module and regarded as the retrieve-and-edit method; the retrieval module only, regarded as the retrieval-based method; and the generation module only, regarded as the simple generation method. We also used the pointer-generator network (See et al., 2017) provided by Generation Challenge 2022 as a baseline for comparison. We converted the comments generated with all methods into <NO\_COMMENT> when they met the rules discussed in Subsection 3.4.

These methods were fine-tuned using a pre-trained model. The retrieval and masking modules used a pre-trained BERT (Devlin et al., 2018) (bert-based-cased from the HuggingFace Transformers library (Wolf et al., 2020)). The generation module used a pre-trained T5 (Raffel et al., 2020) (t5-base from the HuggingFace Transformers library (Wolf et al., 2020)). We fine-tuned these pre-trained models using the dataset constructed in Subsection 4.1 and used the RAdam optimizer (Liu et al., 2019) with the mini-batch size set to 64. In the multi-decoding operation, we set  $k$  to seven. We also fine-tuned only the generation module using the provided dataset and the retrieval and generation method using the dataset that had unmasked feedback comments in the dataset for the generation module. Note that only the pointer-generator network was not pre-trained.

## 4.3 Results

Table 1 lists the experimental results of the feedback comment generation. The values represent BLEU scores (Papineni et al., 2002), where precision is calculated by dividing the sum of BLEU for each generation by the number of expected feedback comments, and recall is calculated by dividing the sum of BLEU for each generation by the number of generations excluding <NO\_COMMENT>. The precision and recall results are then used to calculate F1.

The table shows that the proposed method with multi-decoding outperformed the other methods. Specifically, the performance of the proposed method improved using multi-decoding. With the proposed method, we believe that the multi-

Method	Precision	Recall	F1
pointe-generator	0.334	0.334	0.334
retrieval	0.424	0.422	0.423
generation	0.464	0.464	0.464
retrieval-and-generation	0.482	0.482	0.482
+ multi-decoding	0.480	0.480	0.480
proposed	0.483	0.481	0.482
+ multi-decoding*	<b>0.495</b>	<b>0.493</b>	<b>0.494</b>

\* This is our best result, although it differs from officially published results.

Table 1: Results of feedback comment generation.

decoding improved the probability of generating a feedback comment that was close to the correct comment because the method could generate different feedback comments by using different masked comments. It generated different feedback comments for the same input text using different masked comments, as shown in Table 2. The table shows that these comments were generated by predicting mask tokens of masked feedback comments. These results indicate that masking tokens in the retrieved feedback comments are important for rewriting these comments for the input text.

With the retrieval-and-generation method, when we used different retrieved feedback comments for the same input text, it generated the same feedback comment, as shown in Table 3. Therefore, even if we used multi-decoding for this method, the performance would almost be the same without multi-decoding. We assume it would be difficult to rewrite the retrieved feedback comment for the input text with this method. It also performed better than the generation module. This indicates that a large amount of training data was effective, not the use of retrieved feedback comments without masking.

The retrieval module underperformed the other methods. This is because it often retrieved comments that were correct as feedback comments but focused on tokens that were not in the input text, as was the problem with the conventional study.

## 4.4 Ablation Study

Table 1 shows that the performances of the proposed method and retrieval-and-generation method were equivalent. We believe that the performance of the masking module adversely affected the performance of the proposed method because the proposed method cascades the results of three modules. Thus, at the inference, we compared the final results using predicted masking with correct masking. Table 4 shows the results, and the proposed method

Input text	Reference	
After all , as a student , he or she needs to put the study <u>at</u> the first place .	«At» is not the correct <preposition> to be used with the set phrase formed using «the first place» meaning “to prioritize something”. ‘In’ is the <preposition> to be used with the <noun> «place».	
Retrieved	Masked	Generated
The <preposition> «at» is normally used to indicate a relatively short period of time such as the time of day. Look up the <noun> «time» in a dictionary to learn the appropriate <preposition> to be used to indicate a period.	The <preposition> «at» is normally used to indicate [MASK] relatively short period [MASK] time [MASK] the time [MASK] day. Look up the <noun> « <b>time</b> » in a dictionary to learn the [MASK] <preposition> to [MASK] used to indicate [MASK] <b>period</b> .	The <preposition> «at» is normally used to indicate <b>a</b> relatively short period <b>of</b> time <b>such as</b> the time <b>of</b> day. Look up the <noun> « <b>place</b> » in a dictionary to learn the <b>appropriate</b> <preposition> to <b>be</b> used to indicate <b>a place</b> .
Using the <preposition> «in» makes the expression literally mean “to have in one hand”. Look up the <noun> «hand» in a dictionary to learn the appropriate <preposition> to be used to form an <idiom> introducing one of two ideas.	[MASK] the <preposition> [MASK] <b>the expression</b> [MASK] Look up the <noun> [MASK] in a dictionary to learn the [MASK] <preposition> to [MASK] used <b>to</b> [MASK]	«At» is not the <b>appropriate</b> <preposition> to be used with the <noun> « <b>place</b> » to express “to put something in the first place”. Look up the <noun> « <b>place</b> » in a dictionary to learn the <b>appropriate</b> <preposition> to <b>be</b> used.

Table 2: Example of proposed method’s output. Underlined words in input text represent position. **Bold words** in masked comments mean that they were edited but not masked. **Red words** in generated comments mean that they were not in masked comments.

Input text	Reference
... colleagues is totally different <u>with</u> the way ...	The <preposition> «with» is often used to indicate concordance. Consult a ...
Retrieved	Generation
«In» is not the <preposition> used with ‘bad’ to qualify the subsequent ...	The <preposition> «with» is often used to indicate concordance. Consult a ...
«to» is not the correct <preposition> to be used to refer to the target. Look ...	The <preposition> «with» is often used to indicate concordance. Consult a ...

Table 3: Example of retrieval-and-generation method’s output. Underlined words in input text represent position.

Method	Precision	Recall	F1
proposed	0.483	0.481	0.482
with correct masking	<b>0.539</b>	<b>0.539</b>	<b>0.539</b>

Table 4: Results using predicted or correct masking.

with correct masking was significantly improved in performance. Table 5 also shows the generated feedback comments using predicted and correct masked feedback comments. The predicted masking was wrong. Although the generated feedback comment using the correct masked comment was the same as the reference feedback comment, using the predicted masked comment was different. In addition, when we used predicted masking, the proposed method also edited tokens other than those in the masked comments. We assume its model determined that predicting only the masked tokens would generate unnatural feedback comments. In other words, when the wrong masked feedback comment was used, the generation task was more difficult than using correct masking. Therefore, it is inferred that the design of the masking module is important for the proposed method, and we should improve this for future work.

## 5 Conclusion

In this paper, we proposed a novel method, *retrieval, masking, and generation*, for feedback comment generation. The proposed method has three modules, retrieval, masking, and generation, and generates feedback comments by cascading each module output. First, the retrieval module extracts an example of feedback comments appropriate for the input text from the data pool. Next, the masking module masks tokens of the retrieved example that are not in the reference feedback comment. Finally, given a concatenated sequence of the input text and masked feedback comment, the generation module outputs the feedback comment in the generation module. With this three-module method, the performance of each module depends on the subsequent performance of the module output. Since it is difficult to guarantee the perfect performance of each module, we use a multi-decoding operation that uses not only the top result but also the top  $k$  results for each decoding. Our experimental results indicate that the proposed method outperforms the three conventional methods.

Input text	Reference	Retrieved
... please keep consider <u>about</u> that.	The <verb> «consider» is a <transitive verb> when used to express “to think about” and does not need the <preposition> «about».	Since the <verb> encounter is a <transitive verb> when used to express “to meet someone”, a <preposition> does not need to precede the <object>.
	Masked	Generation
Predicted Masking	[MASK] the <verb> [MASK] is <transitive verb> <b>when used to</b> [MASK] a <preposition> <b>does not need to precede the</b> [MASK]	<b>When</b> the <verb> «consider» is <b>used as</b> a <transitive verb> <b>in a sense of</b> “care for someone”, a <preposition> <b>is unnecessary</b> .
Correct Masking	[MASK] <verb> [MASK] is a <transitive verb> when used to express “to [MASK] does not need [MASK]	<b>The</b> <verb> «consider» is a <transitive verb> when used to express “to <b>think about</b> ” and does not need <b>the</b> <preposition> «about».

Table 5: Example of proposed method using predicted or correct masking. Underlined words in input text represent position. **Bold words** in masked comments mean that they were edited but not masked. **Red words** in generated comments mean that they were not in masked comments.

## References

- 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*.
- Kazuaki Hanawa, Ryo Nagata, and Kentaro Inui. 2021. Exploring methods for generating feedback comments for writing learning. In *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9719–9730.
- Tatsunori B Hashimoto, Kelvin Guu, Yonatan Oren, and Percy S Liang. 2018. A retrieve-and-edit framework for predicting structured outputs. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, page 10052–10062.
- Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, pages 707–710.
- Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2019. On the variance of the adaptive learning rate and beyond. In *Proc. International Conference on Learning Representations (ICLR)*.
- Ryo Nagata. 2019. Toward a task of feedback comment generation for writing learning. In *Proc. Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3206–3215.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proc. the Association for Computational Linguistics (ACL)*, pages 311–318.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.
- Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proc. Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1073–1083.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proc. Advances in neural information processing systems (NIPS)*, pages 5998–6008.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proc. Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP)*, pages 38–45.

# TMU Feedback Comment Generation System Using Pretrained Sequence-to-Sequence Language Models

Naoya Ueda and Mamoru Komachi

Tokyo Metropolitan University

ueda-naoya@ed.tmu.ac.jp, komachi@tmu.ac.jp

## Abstract

In this paper, we introduce our Tokyo Metropolitan University Feedback Comment Generation system submitted to the feedback comment generation task for INLG 2023 Generation Challenge. In this task, a source sentence and offset range of preposition uses are given as the input. Then, a system generates hints or explanatory notes about preposition uses as the output. To tackle this generation task, we finetuned pretrained sequence-to-sequence language models. The models using BART and T5 showed significant improvement in BLEU score, demonstrating the effectiveness of the pretrained sequence-to-sequence language models in this task. We found that using part-of-speech tag information as an auxiliary input improves the generation quality of feedback comments. Furthermore, we adopt a simple postprocessing method that can enhance the reliability of the generation. As a result, our system achieved the F1 score of 47.4 points in BLEU-based evaluation and 60.9 points in manual evaluation, which ranked second and third on the leaderboard.<sup>1</sup>

## 1 Introduction

This paper describes our submission to the feedback comment generation task for INLG 2023 Generation Challenge (Nagata et al., 2021). Feedback comment generation is a task of automatically generating hints or explanatory notes about errors for the purpose of helping the language learner improve their writing skills (Nagata, 2019). In this task, the target of the feedback comment generation is limited to preposition uses, such as missing prepositions, to-infinitives, and deverbal prepositions. Table 1 shows the overview of this task.

In the previous study (Hanawa et al., 2021), Pointer Generator Network (See et al., 2017) was used as a sequence-to-sequence method and found

<sup>1</sup>Our source code is available at [https://github.com/NOIRUED/T5\\_FCG.git](https://github.com/NOIRUED/T5_FCG.git)

### Input

**Source sentence:** I can not agree you in this case.

**Offset Ranges:** 9:18

### Output

Since the <verb> «agree» is an <intransitive verb>, a <preposition> needs to precede the <object>. Look up the <verb> «agree» in the dictionary to find the appropriate <preposition>.

Figure 1: Overview of the feedback comment generation task.

to be effective in a setting with few variations of feedback comments such as preposition uses. While this study shows the effectiveness of non-pretrained sequence-to-sequence models such as Pointer Generator Network, no experiments using pretrained language models have been conducted. Since pretrained sequence-to-sequence language models, such as T5 (Raffel et al., 2020), show significant performance in the generation task, it is conceivable that using pretrained sequence-to-sequence language models improves the generation quality.

In this paper, we examined the performance of pretrained sequence-to-sequence language models in the feedback comment generation task. We employ BART (Lewis et al., 2020) and T5 (Raffel et al., 2020) as the pretrained sequence-to-sequence language models. Both models have improved the generation quality compared with the non-pretrained sequence-to-sequence model. Also, we confirmed that using part-of-speech (POS) tags as an auxiliary input improves the generation quality of feedback comments in the T5 model. Furthermore, we adopted a simple postprocessing method

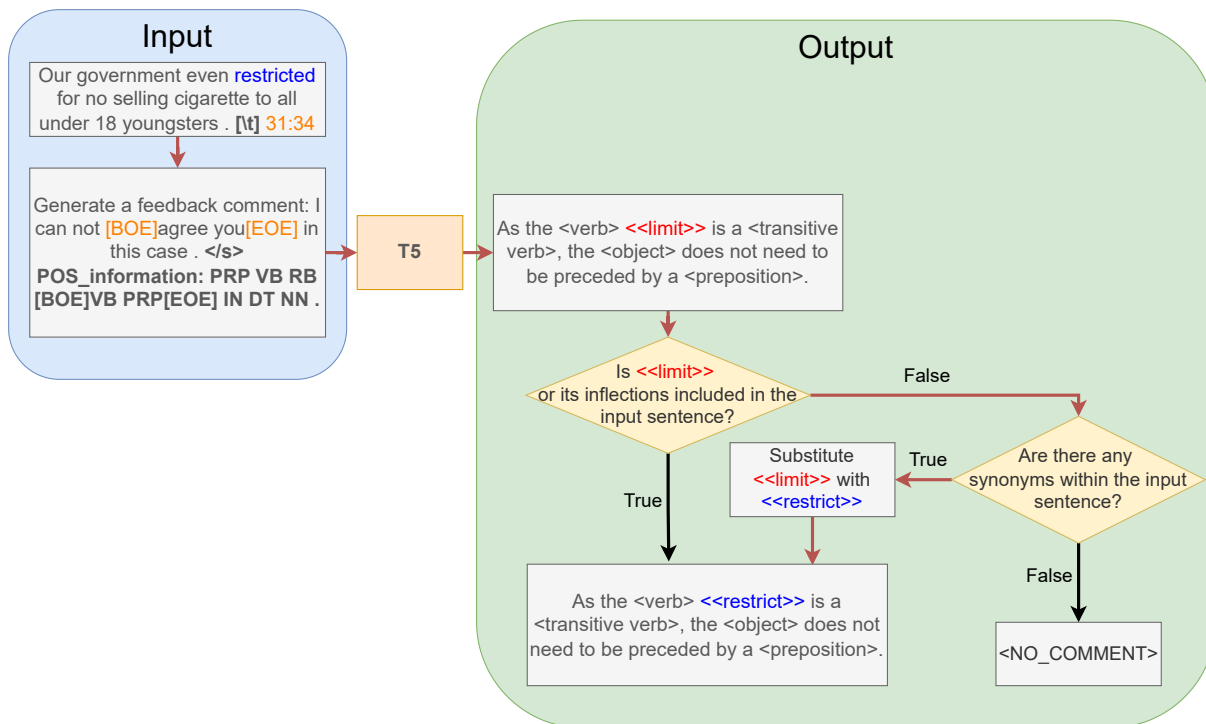


Figure 2: Overview of our method.

to enhance the reliability of the generation. By using this model and methods, we achieved the F1 score of 47.4 points in the BLEU-based evaluation and 60.9 points in the manual evaluation.

## 2 Feedback Comment Generation Task

### 2.1 Task Description

The task focuses on the feedback comment generation targeted on preposition uses. As the input, the source sentence and the offset ranges indicating where to comment is given. From the input, a system is required to generate an appropriate feedback comment or the special token `<NO_COMMENT>` indicating that the system cannot generate any reliable feedback comment.

### 2.2 Evaluation

The performance of the system is evaluated automatically and manually. As an automatic evaluation, BLEU (Papineni et al., 2002) score is calculated between the system output and the reference using SacreBLEU (Post, 2018). A manual evaluation is done by the shared task organizers on the final submission. Both evaluations are measured by recall, precision, and F1. System outputs with `<NO_COMMENT>` are excluded from both the numerator and the denominator of precision and the numerator of recall.

### 2.3 Official Baseline System

The official baseline system is Pointer Generator Network model (See et al., 2017) implemented based on fairseq (Ott et al., 2019). It is a sequence-to-sequence neural network with attention and copy mechanisms. We refer to this model as a non-pretrained sequence-to-sequence model and compare it with pretrained sequence-to-sequence models.

## 3 Our Method

We frame the feedback comment generation task as a sequence-to-sequence generation task. We finetuned the pretrained sequence-to-sequence language models with the official distributed datasets. Since it is difficult for the models to learn the meaning of the offset ranges, instead of using offset as it is, we inserted the special tokens `[BOE]` and `[EOE]` in the position of offset ranges. Figure 2 shows the overview of our proposed method.

### 3.1 Auxiliary Input

As shown in Figure 2, there are cases that POS information is needed in the output. However, the input sequence does not contain such information, which might lead a system to generate a feedback comment with wrong POS information.

	Train	Dev	Test
Official Datasets	4,868	170	215

Table 1: Number of data instances used in the experiment.

To address this problem, we used POS tag information as an auxiliary input in the T5 model. We used Natural Language Toolkit (NLTK) (Bird et al., 2009) to obtain POS tags of the source sentence. Using the obtained POS tags, we concatenated them with the source sentence as follows:

[Source sentence] <\s> POS: [POS tags]

where <\s> is special token in T5. This method (we will refer to as POSTAG hereafter) allows the T5 model to learn the POS information of the source sentence, which makes better auxiliary inputs.

### 3.2 Postprocessing

In this task, the quotations from the source sentence should be bracketed using double-angle brackets. Conversely, if the double-angle bracketed words are not present in the source text, the feedback comment is considered unreliable. However, there are cases where the T5 model quotes the words that do not exist in the source sentence. To overcome this problem, we adopted a simple postprocessing method (we will refer to it as EDIT hereafter). In this postprocessing method, if the double-bracketed words do not exist in the source sentence, it finds the 10-best synonyms using FastText (Bojanowski et al., 2017). If any of the 10-best synonyms are included in the text, the system replaces the bracketed word with the synonym. Conversely, if none of the 10-best synonyms are included in the text, it changes the outputs to <NO\_COMMENT>.

## 4 Experimental Settings

### 4.1 Dataset

In this paper, we only used the official datasets distributed in the shared task. Since there are some typographical errors and orthographic variants in the datasets, we preprocessed the datasets to correct typographical errors and unify orthographic variants. The number of data instances is shown in Table 1.

### 4.2 Model

In this study, we employ BART (Lewis et al., 2020) and T5 (Raffel et al., 2020) as the pretrained

System	BLEU		
	Precision	Recall	F1
Official Baseline	46.3	46.3	46.3
BART-base	51.9	51.9	51.9
BART-large	51.6	51.6	51.6
T5-base	<b>64.0</b>	<b>64.0</b>	<b>64.0</b>
T5-large	60.4	60.4	60.4

Table 2: Experimental results for each system.

System	BLEU		
	Precision	Recall	F1
T5-base	64.0	64.0	64.0
+POSTAG	64.7	<b>64.7</b>	<b>64.7</b>
+EDIT	<b>64.9</b>	64.4	64.6

Table 3: Experimental results for POSTAG and EDIT settings.

sequence-to-sequence language models. We used the Huggingface Transformer (Wolf et al., 2020) to implement the models.

**BART** For the BART-based model, we use the BART-base<sup>2</sup> and BART-large<sup>3</sup>. For fine-tuning, the models are optimized using AdamW optimizer with the constant learning rate of 1e-5, the batch size 16, and trained for 20 epochs.

**T5** For the T5-based model, we use the T5-base<sup>4</sup> and T5-large<sup>5</sup>. For fine-tuning, the models are optimized using AdamW optimizer with a constant learning rate of 5e-4, a batch size of 16, and trained for 30 epochs. To specify a task, the prefix “Generate a feedback comment: ” is added at the beginning of input sequences.

## 5 Results

### 5.1 Experimental Results

Table 2 shows the experimental results against the development set. Compared with the official baseline system, BART and T5 models improved the BLEU scores, demonstrating the effectiveness of the pretrained sequence-to-sequence language models in this task. In our case, the T5-base model

<sup>2</sup><https://huggingface.co/facebook/bart-base>

<sup>3</sup><https://huggingface.co/facebook/bart-large>

<sup>4</sup><https://huggingface.co/t5-base>

<sup>5</sup><https://huggingface.co/t5-large>

Source sentence	But smoking in the restaurant will cause both the smokers and surrounding people <u>facing</u> with the those problems more than public places .	
System	System Output	BLEU
Gold	A <verb> part representing the cause of <verb> «cause» takes the form of a <to-infinitive> rather than the <ing-form>.	100.00
T5-base	A <verb> part representing the cause of <verb> «cause» takes the form of a <to-infinitive> rather than the <base form>.	89.53
POSTAG	A <verb> part representing the cause of <verb> «cause» takes the form of a <to-infinitive> rather than the <ing-form>.	100.00

Table 4: Example of the result in POSTAG setting. The underline indicates the offset ranges.

Source sentence	With the development of society , we , college students , should do more to <u>adjust it</u> .	
System	System Output	BLEU
Gold	As the <verb> «adjust» is an <intransitive verb> when used to express “to adapt to something” , [...]	100.00
POSTAG	The <verb> «adapt» does not take an <indirect object> to indicate what one adjusts to. Use the <verb> «adapt» as an <intransitive verb> with a <preposition>. [...]	37.72
EDIT	The <verb> «adjust» does not take an <indirect object> to indicate what one adjusts to. Use the <verb> «adjust» as an <intransitive verb> with a <preposition>. [...]	44.97

Table 5: Example of the result in EDIT setting. The underline indicates the offset ranges.

performed best in this task. We expected large-sized models to perform better than the base-sized models, but contrary to our expectations, the base-sized models outperformed the large-sized models. We consider this odd finding comes from a lack of sufficient parallel data or unreliability of BLEU scores in the feedback comment generation task. We leave for future work a more detailed examination of these model differences.

Table 3 shows the experimental results in the POSTAG and EDIT settings. Compared with the T5-base model, POSTAG setting improved the score by 0.7 points. The improvements of the BLEU score are relatively small because the superficial differences in the generated outputs were small. Table 4 shows the example that the model has successfully used POS tag information. From the table, we can confirm that POSTAG setting generated feedback comments with correct POS information, but the BLEU score only improved by 10.5 points. These results indicate that using POS tag information as an auxiliary input does not improve the overall

BLEU score, but is effective in this task to generate reliable feedback comments.

Compared with the POSTAG setting, EDIT setting improved the precision, but lowered recall and F1 score. Although, the EDIT setting does not improve the BLEU score, it actually enhances the reliability of the feedback comments. Table 5 shows the example that had successfully edited an unreliable feedback comment into a reliable feedback comment. These results show that our postprocessing method is effective to enhance the reliability of the generation.

## 5.2 Official Results

From the experimental results, we submitted the T5-base with POSTAG and EDIT as our final submission to the shared task. As shown in Table 6, our system obtained a BLEU score of 47.4 and a manual evaluation score of 60.9, which ranked second and third on the leaderboard.

System	BLEU			Manual Evaluation		
	Precision	Recall	F1	Precision	Recall	F1
Official Baseline	33.4	33.4	33.4	31.2	31.2	31.2
Our System	<b>47.7</b>	<b>47.1</b>	<b>47.4</b>	<b>61.3</b>	<b>60.5</b>	<b>60.9</b>

Table 6: Official results.

## 6 Conclusion

In this paper, we described our submission to the feedback comment generation task for INLG 2023 Generation Challenge. The result of the experiments showed that using pretrained sequence-to-sequence language models is effective in the feedback comment generation for preposition uses. Furthermore, we found that using POS tags as an auxiliary input improves the generation quality, and confirmed that our postprocessing method enhances the quality of the feedback comments by editing unreliable feedback comments into reliable feedback comments. Future work will explore additional postprocessing methods that can better identify and appropriately edit unreliable feedback comments.

## References

- Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. O’Reilly Media, Inc.
- 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.
- Kazuaki Hanawa, Ryo Nagata, and Kentaro Inui. 2021. [Exploring methods for generating feedback comments for writing learning](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9719–9730, Online and Punta Cana, Dominican Republic. 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.
- Ryo Nagata. 2019. [Toward a task of feedback comment generation for writing learning](#). 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 3206–3215, Hong Kong, China. Association for Computational Linguistics.
- Ryo Nagata, Masato Hagiwara, Kazuaki Hanawa, Masato Mita, Artem Chernodub, and Olena Nahorna. 2021. [Shared task on feedback comment generation for language learners](#). In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 320–324, Aberdeen, Scotland, UK. Association for Computational Linguistics.
- Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. [fairseq: A fast, extensible toolkit for sequence modeling](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. 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.
- 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.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. [Get to the point: Summarization with pointer-generator networks](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. 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.

# The Tokyo Tech and AIST System at the GenChal 2022 Shared Task on Feedback Comment Generation

Shota Koyama<sup>1,2</sup>, Hiroya Takamura<sup>2</sup>, Naoaki Okazaki<sup>1,2</sup>

<sup>1</sup>Tokyo Institute of Technology

<sup>2</sup>National Institute of Advanced Industrial Science and Technology

shota.koyama@nlp.c.titech.ac.jp

takamura.hiroya@aist.go.jp

okazaki@c.titech.ac.jp

## Abstract

This paper describes the Tokyo Tech and AIST system in the GenChal 2022 shared task, which is the first shared task of feedback comment generation. We adopted five methods: data cleaning, fine-tuning pre-trained models, correcting errors in learners' sentences, appending a correcting operation, and filtering out irrelevant outputs. Our system achieved  $F_1 = 43.4$  on the test dataset.

## 1 Introduction

Recently, Nagata (2019) proposed a novel task called feedback comment generation (FCG), wherein feedback is provided to help writers improve their skills, especially in the context of computer-assisted language learning. The input of an FCG model is a learner's sentence, and the output is a comment given as feedback to the learner.

The GenChal 2022 shared task is the first shared task of the FCG task. This paper describes the system developed in our study, which encompasses the following five methods:

(1) **Data cleaning** (§3.1): We corrected mistakes in the annotations within the training dataset.

(2) **Fine-tuning pre-trained models** (§3.2): We fine-tuned pre-trained models to address the low-resource aspect of this task.

(3) **Correcting errors in learners' sentences** (§3.3): We corrected errors in the input sentences outside of the target words for the FCG, thus preventing errors in the model output.

(4) **Appending a correcting operation** (§3.4): We appended a correcting operation (such as "delete") to the input with the aim of generating more accurate feedback comments.

(5) **Filtering** (§3.5): We removed irrelevant feedback comments using simple heuristics.

This paper is organized as follows. Section 2 describes the shared task and its dataset. Section 3 details the methods outlined above. Section 4 presents

the experimental setup. Section 5 shows the results. Section 6 concludes this paper.

All of our code has been publicly released for reproducibility<sup>1</sup>.

## 2 Task and Dataset Description

The GenChal 2022 shared task was proposed by Nagata et al. (2021) to address FCG. The organizers released a new dataset for this task, wherein original texts written by English learners were borrowed from ICNALE (Ishikawa, 2011).

The input of this task is a pair consisting of the learners' text and a span indicating the feedback comment's location. The input text is written in English and tokenized. The span is provided as input in this task, although it can be detected by grammatical error detection models. For example, the sentence "It is a problem for health ." has an error, which we can correct by replacing for to of. The span is character-level and colon-separated, and the position indicates a 0-indexed point between characters including whitespace. Therefore, the span of this example is 16:19, wherein the start position is 16 and the end position is 19.

The output of this task is feedback comment, which must be informative beyond merely an indication of 'correct' or 'incorrect'. Furthermore, specific words and phrases in feedback comments are annotated using brackets. Grammar terms and idiomatic patterns are bracketed using <>, whereas quotations from the learner's sentence are bracketed using << >>. Miscellaneous quotations and words or phrases to highlight can be annotated using '" "'. For example, the feedback comment for the sentence in the last paragraph can be: The <preposition> <<for>> should precede a person. Simply use '"of' in this case. The special output <NO\_COMMENT> indicates that the system cannot generate any reliable feedback

<sup>1</sup><https://github.com/shotakoyama/fcgtools>

comment. In this task, feedback comments are written in English, while another choice is using the learner’s native language as in Nagata et al. (2020).

The released dataset was split into training, development, and test subsets, which contain 4868, 170, and 215 sentence pairs, respectively. Feedback comments in the test dataset were not released during the shared task period.

The BLEU (Papineni et al., 2002) score is adopted as the automatic evaluation metric. To incorporate the use of <NO\_COMMENT> into the evaluation, task submissions are evaluated by BLEU-based  $F_1$  score. The precision ( $P$ ), recall ( $R$ ) and  $F_1$  score are calculated as follows:

$$\begin{aligned} \mathcal{R} &: \text{reference sentences } (r_1, r_2, \dots), \\ \mathcal{S} &: \text{system outputs } (s_1, s_2, \dots), \\ \mathcal{S}' &= \{s_i \in \mathcal{S} \mid s_i \neq \text{<NO\_COMMENT>}\}, \\ P &= \frac{1}{|\mathcal{S}'|} \sum_{i \in \{i \mid s_i \in \mathcal{S}'\}} \text{BLEU}(s_i, r_i), \\ R &= \frac{1}{|\mathcal{R}|} \sum_{i=1}^{|\mathcal{R}|} \text{BLEU}(s_i, r_i), \\ F_1 &= 2 \times \frac{P \times R}{P + R}. \end{aligned}$$

### 3 Method

#### 3.1 Data Cleaning

Because we found the training data contain many annotation mistakes, we manually cleaned all erroneous feedback comments. 1,770 data samples ( $\approx 37\%$ ) were affected by this process.

##### 3.1.1 Illegal Span

The start and end of the target’s span must correspond to the start and end of a word, respectively. For example, a span of 10:12 is correct for the input “It is fun to me .”, whereas a span of 9:12 is illegal.

##### 3.1.2 Wrong Annotation

Annotations using brackets must satisfy the bracket correspondence. Illegal brackets (e.g., <verb>>  $\rightarrow$  <verb>) and illegal quotations (e.g., 'of '  $\rightarrow$  ‘ ‘of ' ') are corrected.

##### 3.1.3 Others

Some trivial mistakes include grammatical errors and the usage of non-ASCII characters. Please

refer to the source code for all modifications<sup>2</sup>.

#### 3.2 Fine-Tuning Pre-Trained Models

Recently, many NLP studies have focused on the use of pre-trained models that are trained on unlabeled data. Pre-training and subsequently fine-tuning is a simple and effective approach widely adopted for low-resource NLP tasks. We fine-tuned a Transformer decoder model, GPT-2 (Radford et al., 2019), and an encoder-decoder model, BART (Lewis et al., 2020).

Because the model requires access to the target position in the input, we added double brackets to the target of feedback comment generation. For example, if the input is “I agree the issue .” and the span is 2:11<sup>3</sup>, the model input is “I <<agree the>> issue .”.

Figure 1 illustrates the use of the BART and GPT-2 models for this task. Because the learner’s sentence and feedback comment must be unified as the input of the GPT-2 decoder, the two sentences are concatenated with #<sup>4</sup>. In training, the GPT-2 model predicts entire concatenated sentences<sup>5</sup>.

#### 3.3 Correcting Errors in Learners’ Sentences

Learners’ sentences may have many errors outside of the target range, which can negatively impact performance. For example, the input “I want go <<to>> abroad .” has a non-targeted error and would be modified to “I want to go <<to>> abroad .”. To address this issue, we corrected any non-targeted errors using GECToR (Omelianchuk et al., 2020), one of the state-of-the-art grammatical error correction models.

#### 3.4 Appending a Correcting Operation

The GECToR model corrects errors by predicting correcting operations, such as “delete” and “from base form to -ing form”, which can benefit more accurate feedback comment generation. We extracted the GECToR tag for the input sentence’s target word, replaced it with a more intuitive form

<sup>2</sup><https://github.com/shotakoyama/fcgtools/blob/main/fcgtools/cli/prepare.py>

<sup>3</sup>This span indicates that this correction inserts a word (in this case on) between agree and the.

<sup>4</sup>We tried various other separation tokens and found that the token selection is not significant with regards to the performance.

<sup>5</sup>In our preliminary experiments, conducting back-propagation of both learner’s sentence and feedback comment yielded better performance than that of only feedback comment.

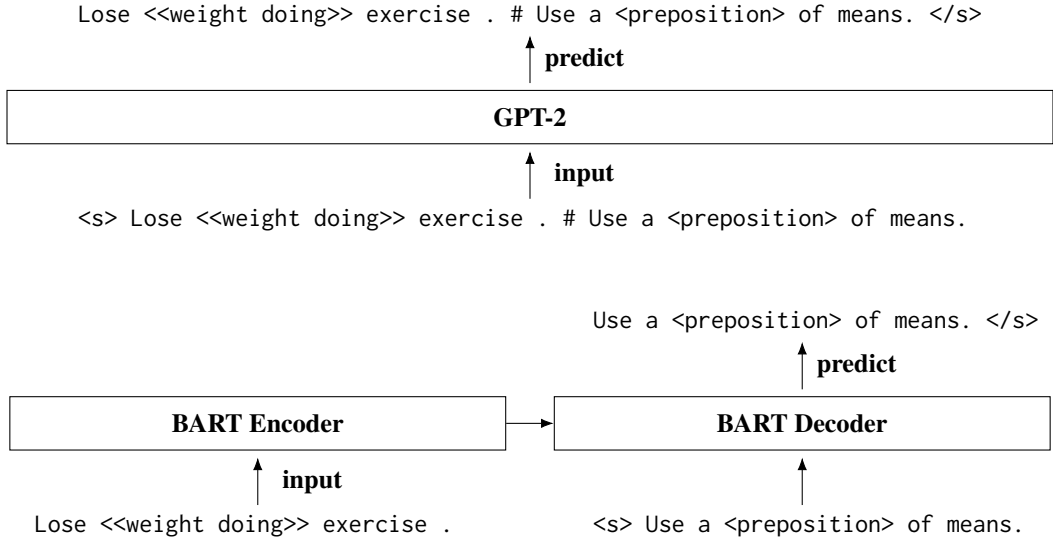


Figure 1: GPT-2 and BART models for the feedback comment generation task.

GECToR tag	tag
DELETE	delete
REPLACE_*	replace *
APPEND_*	append *
TRANSFORM_AGREEMENT_PLURAL	plural
TRANSFORM_AGREEMENT_SINGULAR	singular
TRANSFORM_CASE_CAPITAL	titlecase
TRANSFORM_CASE_CAPITAL_1	capitalcase
TRANSFORM_CASE_LOWER	lowercase
TRANSFORM_CASE_UPPER	uppercase
TRANSFORM_SPLIT_HYPHEN	split hyphen
TRANSFORM_VERB_*_*	from * to *

Table 1: Replacement rules of GECToR tag.

according to the rules listed in Table 1, and concatenated it with //. For example, the GECToR tag for the input “You cannot stop <<to smoke>> .” should be “TRANSFORM\_VERB\_VB\_VBG”, and the input would be converted to “You cannot stop <<to smoke>> . // from VB to VBG”.

### 3.5 Filtering Out Irrelevant Outputs

Filtering represents a simple heuristic to improve performance. If the quoted part in the feedback comment does not appear in the input sentence, the whole output is replaced with <NO\_COMMENT>, as the comment is obviously irrelevant. This procedure was introduced to prevent the score from dropping. For example, when the model generated the feedback comment “Since <<ahead>> is an <adverb>, ...” for the input “I want to go <<to>> abroad .”, this comment was filtered out and replaced with <NO\_COMMENT> because ahead

does not appear in the input sentence.

## 4 Experimental Setup

We used the dataset released by the shared task and selected the best epoch for each training trial using the validation dataset. We fine-tuned models for 100 epochs with saving checkpoints at five-epoch intervals.

We used the cross-entropy loss, AdamW optimizer (Loshchilov and Hutter, 2019) with a weight decay of 0.01 and gradient clipping of 1.0, and a learning rate of 0.0001 with a constant scheduler.

We employed BART small and GPT-2 small to compare both models and GPT-2 large for the final submission. We used NVIDIA A100 GPU with 40 GiB memory for all experiments and varied the batch size for model size to ensure an efficient use of the GPU memory. We set the maximum tokens per batch to 2,000 for BART/GPT-2 small and 250 for GPT-2 large, and accumulated every four batches for BART/GPT-2 small and 32 for GPT-2 large, thus setting the number of maximum tokens for each step to 8,000.

## 5 Experimental Results

### 5.1 Comparison Between BART and GPT-2

First, we conduct experiments to compare the performance of BART and GPT-2 and verify the effectiveness of the methods introduced in Section 3.

Table 2 lists the average scores on the validation set obtained by the five models. Correcting non-target errors (+ correction, § 3.3), appending

	BART small	GPT-2 small
fine-tuning	47.74	49.45
+ correction	47.36	50.23
+ operation	47.58	50.70
+ both	47.32	<b>51.80</b>

Table 2: Comparison between BART and GPT-2.

	w/o filtering	w/ filtering
fine-tuning	49.45	49.97
+ correction	50.23	50.71
+ operation	50.70	51.14
+ both	<u>51.80</u>	<b>52.44</b>

Table 3: Effect of filtering.

a correcting operation (+ operation, § 3.4), and applying both methods improves the performance of GPT-2 and decreases that of BART. Furthermore, GPT-2 performs better than BART in all settings. Accordingly, we selected GPT-2 for the task submission.

## 5.2 Impact of Filtering

We verified the effectiveness of filtering (§ 3.5) on GPT-2 small. Table 3 lists the average scores on the validation set obtained by the five models. We confirmed that filtering improves the performance by approximately 0.5 points in every setting.

## 5.3 Final Submission

We compared the results obtained by GPT-2 small and large, to determine the final submission. The results listed in Table 4 represent the best scores on the validation set obtained by the five models. We adopted GPT-2 large, appending a correcting operation and filtering for the final submission. In the shared task, our final submission achieved 43.4 in  $F_1$  score on the blind test set.

## 6 Conclusion

In this paper, we described our system for the GenChal 2022 shared task. We employed five methods: data cleaning, fine-tuning pre-trained models, correcting errors in learners’ sentences, appending a correcting operation, and filtering. We fine-tuned BART and GPT-2 and then selected GPT-2 for submission. We verified that filtering `<NO_COMMENT>` using a simple heuristic improves performance. Our final submission was obtained using GPT-2 large with appending a correcting operation and filtering without correcting non-target errors. Our

correction	operation	filtering	small	large
✓	✓		52.84	53.96
✓		✓	52.56	52.15
	✓	✓	51.79	<b>54.73</b>
✓	✓	✓	53.19	54.33

Table 4: Comparison between GPT-2 small and large.

system achieved an  $F_1$  score of 54.73 on the validation set, and 43.4 on the test set.

## 7 Acknowledgments

This paper is based on results obtained from a project JPNP20006, commissioned by the New Energy and Industrial Technology Development Organization (NEDO). For experiments, computational resource of AI Bridging Cloud Infrastructure (ABCI) provided by National Institute of Advanced Industrial Science and Technology (AIST) was used.

## References

- Shin’ichiro Ishikawa. 2011. A new horizon in learner corpus studies: The aim of the ICNALE project. *Corpora and Language Technologies in Teaching, Learning and Research*, pages 3–11.
- 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.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.
- Ryo Nagata. 2019. [Toward a task of feedback comment generation for writing learning](#). 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 3206–3215, Hong Kong, China. Association for Computational Linguistics.
- Ryo Nagata, Masato Hagiwara, Kazuaki Hanawa, Masato Mita, Artem Chernodub, and Olena Nahorna. 2021. [Shared task on feedback comment generation for language learners](#). In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 320–324, Aberdeen, Scotland, UK. Association for Computational Linguistics.

- Ryo Nagata, Kentaro Inui, and Shin'ichiro Ishikawa. 2020. [Creating corpora for research in feedback comment generation](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 340–345, Marseille, France. European Language Resources Association.
- Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. [GECToR – grammatical error correction: Tag, not rewrite](#). In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA → Online. 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.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

# Feedback comment generation using predicted grammatical terms

Kunitaka Jimichi<sup>1</sup>, Kotaro Funakoshi<sup>1</sup>, Manabu Okumura<sup>1,2</sup>

<sup>1</sup>Tokyo Institute of Technology <sup>2</sup>RIKEN Center for Advanced Intelligence Project  
{kunitaka, funakoshi, oku}@lr.pi.titech.ac.jp

## Abstract

The purpose of feedback comment generation is to provide useful feedback comments for a wide range of errors in learners' essays from a language learning perspective. Since it is difficult to obtain appropriate comments at a practical level with rule-based or retrieval-based methods, we explore neural-based generative methods with pre-trained models. We further assume the effectiveness of considering grammatical terms in generating feedback comments. Specifically, this paper proposes T5-based models using predicted grammatical terms, submitted to FCG GenChal, and presents their results. By using correct grammatical terms, our model could improve the BLEU score by 19.0 points, compared with the baseline T5 without grammatical terms on the development dataset. Furthermore, by using predicted grammatical terms, our model could improve the manual evaluation score by 2.33 points, compared with the baseline T5 without grammatical terms on the test dataset.

## 1 Introduction

Feedback comment generation (FCG) for writing studies is a task to generate explanations of why ungrammatical sentences written by language learners are incorrect and on what basis a correction was made. Related research has been mainly conducted on the basis of a dataset published by Nagata (2019). However, the accuracy, which is based on *manual evaluation* (ME), so far is insufficient for real-world use. One reason is that the data in the dataset are not necessarily sufficient to cover all error types and feedback comments. Since annotating feedback comments would require specialized knowledge in writing, constructing larger datasets is time-consuming and labour-intensive.

FCG GenChal (Nagata et al., 2021) targets the generation of feedback comments for prepositional errors. An example feedback comment for the prepositional error is shown in the following.

Target sentence: I agree on the idea.

Feedback comment: «Agree on» indicates that more than one person agrees on a certain matter. <verb> «agree» to find the <preposition> used to indicate that the same idea is shared.

Here, the words marked by <> are *grammatical terms* (GTs). Underlined words are the target word with an error that the feedback comment is generated for.

To achieve this task, methods using an Encoder-Decoder model, which generates feedback comments from scratch, are considered effective because they can deal with various learners' error types even in unsophisticated settings. Therefore, it is a good idea to develop the Encoder-Decoder model for generating feedback comments.

State-of-the-art (SOTA) results have been achieved in various natural language processing tasks by using pre-trained models. However, there has been no previous work on FCG using pre-trained models. A logical concern is what level of accuracy the SOTA pre-trained models can achieve in FCG. In grammatical error correction, which is highly related to FCG, methods using T5 (Raffel et al., 2020) achieve good results. Thus, in this work, we also utilize T5 for FCG.

The system can more easily generate a feedback comment when some words in the target feedback comment sentence to be generated are known. One of such clue words for better generating feedback comments might be grammatical terms (GTs), since commonly used GTs are limited, and it might be possible to predict and use them to generate feedback comments. However, no research has focused on GTs for FCG.

Therefore, we propose the following procedure for generating feedback comments in this study. First, GTs to be used in the feedback comment are selected. As several GTs are used in a feedback

comment, this becomes a multi-label classification task. Next, the selected GTs are used to generate the feedback comment.

To predict GTs, we use RoBERTa (Liu et al., 2019) because RoBERTa often achieves better accuracy than other pre-trained models in the multi-label classification task. T5 is then used to generate feedback comments since it can be used for text-to-text tasks.

The contributions of this research are therefore as follows:

- We investigate the extent to which the use of GTs improves the ME in FCG by using T5.
- We demonstrate the use of correct GTs using the  $T5_{base}$  model improves the BLEU (Papineni et al., 2002) score by 19.0 points on the development dataset, and predicted GTs using the  $T5_{base}$  model improves the ME score by 2.33 points on the test dataset.

## 2 Related work

Grammatical error correction is closely related to the FCG task. Rothe et al. (2021) have achieved a high accuracy in grammatical error correction by using a pre-trained generative language model, T5. This suggests that FCG could also be handled by T5.

A survey (Hanawa et al., 2021) of the methods used in the FCG task investigated three methods: retrieval-based, retrieve-and-edit, and simple generation. The survey shows that the simple generation method performs best in generating feedback comments for prepositional errors and the retrieval-based method alone cannot cope with various errors present in the training examples in generating feedback comments.

In generating feedback comments using a generative model, prompting the model with the predicted GTs corresponding to the target error is likely to guide the direction for the generation. However, to the best of our knowledge, there have been no studies taking such an approach to FCG.

## 3 Grammatical term prediction

### 3.1 Task definition and notations

This section describes the prediction task of GTs. To define the task formally, we introduce the following symbols. The learners’ sentence, its length (the number of tokens), and the  $i$ -th token are denoted by  $S$ ,  $N$  and  $w_i$ , respectively. That is,  $S =$

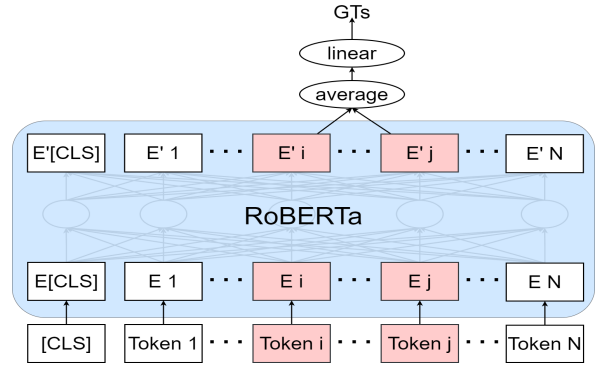


Figure 1: Schematic diagram of multi-label classification for GTs with RoBERTa.

$w_1, \dots, w_i, \dots, w_N$ . The span where to comment is denoted by  $s$ ,<sup>1</sup> which indicates the position of several consecutive words. The task input is defined as  $x = (S, s)$ . The GTs and their number in the reference feedback comment  $y$  corresponding to  $S$  are denoted by  $T$  and  $M$ , respectively. That is,  $T = t_1, \dots, t_i, \dots, t_M$ . Here,  $T$  is sorted in lexicographic order.  $M = 0$  means that  $y$  does not contain any GTs. The grammatical term prediction task is to predict  $T$  given  $x$ .

### 3.2 Prediction method

We use RoBERTa as the base model. Each input word in  $S$  is transformed into the corresponding embedding, which is then transformed into a context-aware embedding by RoBERTa. The embeddings of the words in  $s$  are then averaged and fed into a linear layer. The GTs whose probabilities are above a certain threshold  $\theta$  are to be included as the prompt used in feedback comment generation, described in the next section. An overview of the model is shown in Figure 1.

## 4 Feedback comment generation

In FCG, T5 is used as the base model to predict  $\hat{y}$  given  $x$  and  $T$ . The input sequence to T5 is given in the following format:

$$\text{fbc: } w_1 \dots [ \dots w_i \dots ] \dots w_N \langle \text{GTs} \rangle : t_1 \dots t_i \dots t_M$$

Here, a special token “fbc:”, which stands for a *feedback comment*, is used as a prompt to train the T5 model. The target span  $s$  is marked by square brackets. Predicted GTs for  $S$  are listed after  $S$  with another special token “<GTs>”.

<sup>1</sup>These spans are referred to as *offsets* in the shared task.



Data	Instances	Sent toks.	Com toks.
Train	4868	110906	127439
Dev.	170	3142	4516
Test	215	4446	-

Table 1: Statistics of the dataset. Instances, sent toks., and com toks. denote the number of instances, tokens in learners’ sentences and tokens in reference feedback comments, respectively. The information for the reference feedback comments in the test dataset is not included because FCG GenChal participants cannot get access to them.

## 5 Experiments

### 5.1 Dataset and metrics

We used the shared task data (Nagata et al., 2021). The data was originally divided into three sets, which are training, development, and test sets, by the FCG Organizers. The statistics of the dataset are shown in Table 1. The training dataset was used for fine-tuning RoBERTa and T5. The development and test datasets were used for evaluating the methods for FCG.

For grammatical term prediction, considering the relatively small size of the data, we used only the top 10 most frequent GTs and discarded the others. The top ten most frequent GTs are <preposition>, <verb>, <noun>, <object>, <transitive verb>, <intransitive verb>, <to-infinitive>, <noun phrase>, <adjective>, and <to infinitive>.<sup>2</sup> For evaluation of grammatical term prediction, only the development dataset was used.

The precision, recall, and F1 of ME, and BLEU (N=4) are the evaluation measures used in accordance with FCG GenChal. The ME scores are based on the human judgment of whether a system output is correct or not. More details are found on the page of the FCG GenChal task description.<sup>3</sup> Since all values of precision, recall, and F1 are the same, only F1 is listed in the table. GTs (full) means the case using all GTs rather than only top-10 GTs.

To evaluate grammatical term prediction, we used exact match ratio (EMR), which indicates the percentage of instances that have all their labels classified correctly. In addition, we used micro averaged precision, recall, and F1 with GTs as a

<sup>2</sup>In these experiments, <to-infinitive> and <to infinitive> were used as separate terms.

<sup>3</sup><https://fcg.sharedtask.org/task/>

Data	Method	BLEU	ME	
Dev.	T5 <sub>small</sub>	47.6	-	
	T5 <sub>small</sub> + predicted GTs (top-10)	45.9	-	
	T5 <sub>base</sub>	49.6	-	
	T5 <sub>base</sub> + predicted GTs (top-10)	49.0	-	
	T5 <sub>small</sub> + correct GTs (top-10)	61.0	-	
	T5 <sub>small</sub> + correct GTs (full)	64.7	-	
	T5 <sub>base</sub> + correct GTs (top-10)	63.0	-	
	T5 <sub>base</sub> + correct GTs (full)	68.6	-	
	Test	Baseline system	33.4	31.16
		T5 <sub>small</sub> + Predicted GTs (top-10)	46.0	56.28
T5 <sub>base</sub>		-	58.14	
T5 <sub>base</sub> + Predicted GTs (top-10)		-	60.47	

Table 2: Feedback comment generation results on the development and test datasets. ME: manual evaluation.

unit.

### 5.2 Hyperparameters

**Grammatical term prediction** The RoBERTa model used in the experiments was roberta-large.<sup>4</sup> We tuned the learning rate from 0.00001, 0.00003, and 0.0001, and the threshold  $\theta$  with the highest EMR on the training dataset. The learning rate was fixed to 0.00003 and the threshold  $\theta$  was fixed to 0.68604184.

AdamW was used as the optimisation function. A batch size of 8 was used and a drop-out rate of 0.1 was used for each linear layer. The maximum sentence length was set to 256. We added one linear layer not included in RoBERTa, with a size of  $1024 \times 10$ . The hidden layer size of roberta-large is 1024 and the number of GT types is 10. No drop-out was applied to the linear layer. Each word was lowercased. The number of epochs used for training was 5. We applied a weight to each GT label when calculating the loss for it. We used the inverse document frequency (IDF) of each label as the weight and calculated it within the training dataset.

**Feedback comment generation** The T5 model used in the experiments was T5<sub>small</sub> and T5<sub>base</sub>.<sup>5</sup> A learning rate of 0.0001 was used. AdamW was used as the optimisation function. A batch size of 8 was used and a drop-out rate of 0.1 was used for each linear layer. The maximum sentence length was set to 512. Each word was lowercased. The number of epochs used for training was 50. “GTs”, “`”, “` ”, “` ”, “` ”, “<”, “>”, “< ” and “> ” were added to the T5 dictionary as special tokens.

<sup>4</sup><https://huggingface.co/roberta-large>

<sup>5</sup>[https://huggingface.co/docs/transformers/model\\_doc/t5](https://huggingface.co/docs/transformers/model_doc/t5)

EMR	P	R	F
8.23	44.68	42.61	42.65

Table 3: Grammatical term prediction results on the development dataset. EMR=exact match ratio, P=micro averaged precision, R=micro averaged recall, and F=micro averaged F1-measure.

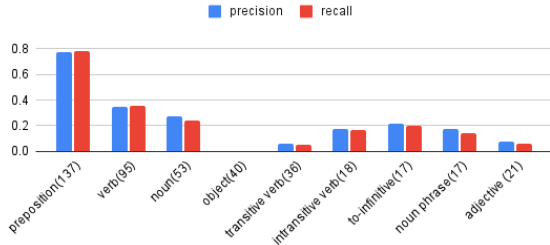


Figure 2: Precision and recall for each label in grammatical term prediction; blue bars are precision and red bars are recall. The number in brackets next to the label indicates the number of occurrences on the development dataset.

### 5.3 Results

The results for the FCG are shown in Table 2. The table shows that  $T5_{small}$  using correct GTs (top-10 and full) improves the BLEU score by 13.4 and 17.1 points on the development dataset, respectively. We also found that  $T5_{base}$  using correct GTs (top-10 and full) improves the BLEU score by 13.4 and 19.0 points on the development dataset, respectively. Furthermore,  $T5_{base}$  using predicted GTs improves the ME score by 2.33 points on the test dataset. This indicates that incorporating predicted GTs in FCG is beneficial for T5.

The results of grammatical term prediction are shown in Table 3. The table shows the multi-labelling performance of the submitted model on the development dataset.

We independently investigated the precision and recall for each GT in the top 10 most frequent words. The results are shown in Figure 2 (Since <to infinitive> was not present in the development dataset, it was not included in the target GTs to be predicted and is excluded from the figure). The figure shows that the precision and recall for each GT do not depend on its frequency in the training dataset. The highest precision and recall are for <preposition>, followed by <verb>, <noun>, and <to-infinitive>. The high prediction performance for <to-infinitive> would be related to the ease of

Reference	The <compound preposition> «because of» should be followed by a <noun>. «Of» is unnecessary when a <clause> follows.
Our Model	The <compound preposition> «because of» should be followed by a <noun>. «Of» is unnecessary when a <clause> follows.
Predicted GTs	<noun> <preposition>
Reference	When a <noun> is qualified by another <noun> that follows, a <preposition> is necessary between the two nouns. Think of the most common <preposition> of association.
Our Model	The <preposition> to indicate the direction of negative influence is missing. Look up the use of the <noun> «future» in the dictionary and add the appropriate <preposition>
Predicted GTs	(no output)

Table 4: Case study: Two qualitative examples; one for which the model predicted the GTs <noun> and <preposition>, and one for which the model failed to predict any GTs (no output). In the top example, only the GT <noun> was successfully predicted and the generated feedback comment was correct. In the bottom example, no GTs were predicted and the generated feedback comment was incorrect.

predicting the error type in an English sentence. When we find a case of two consecutive verbs or ‘to infinitive’ + the ‘ing’ form of a verb in a sentence, we can simply determine there is an error in it.

## 6 Case study

We investigated whether our model could generate correct feedback comments with the predicted GTs in the development dataset. Table 4 shows examples where our model produced correct and incorrect feedback comments in the top and bottom rows, respectively. In the top example, while our model correctly predicted <noun>, it also incorrectly predicted <preposition>, a GT similar to the correct <compound preposition>. In the bottom example, it did not predict any GTs, and as a result, gains no benefit from them and generates an incorrect feedback comment.

## 7 Conclusion

We explored neural methods for FCG using pre-trained models. In this study, we showed predicting the GTs and using them in generating feedback comments can be useful for feedback comment generation with T5. The results also suggested that further improvement in grammatical term prediction would be beneficial for FCG.

## Acknowledgements

The authors are grateful to Prof. Ryo Nagata in Konan University for suggesting the topic addressed in this paper. We also thank the FCG Organizers for sharing their dataset and hosting the FCG GenChal for us.

## References

- Kazuaki Hanawa, Ryo Nagata, and Kentaro Inui. 2021. [Exploring methods for generating feedback comments for writing learning](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9719–9730, Online and Punta Cana, Dominican Republic. 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](#). *CoRR*, abs/1907.11692.
- Ryo Nagata. 2019. [Toward a task of feedback comment generation for writing learning](#). 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 3206–3215, Hong Kong, China. Association for Computational Linguistics.
- Ryo Nagata, Masato Hagiwara, Kazuaki Hanawa, Masato Mita, Artem Chernodub, and Olena Nahorna. 2021. [Shared task on feedback comment generation for language learners](#). In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 320–324, Aberdeen, Scotland, UK. 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.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. [A simple recipe for multilingual grammatical error correction](#). 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 702–707, Online. Association for Computational Linguistics.

# AIWolfDial 2023: Summary of Natural Language Division of 5th International AIWolf Contest

Yoshinobu Kano<sup>1\*</sup>, Neo Watanabe<sup>1</sup>, Kaito Kagaminuma<sup>1</sup>, Claus Aranha<sup>2</sup>, Jaewon Lee<sup>3</sup>,  
Benedek Hauer<sup>3</sup>, Hisaichi Shibata<sup>4</sup>, Soichiro Miki<sup>4</sup>, Yuta Nakamura<sup>4</sup>,  
Takuya Okubo<sup>3</sup>, Soga Shigemura<sup>3</sup>, Rei Ito<sup>3</sup>, Kazuki Takashima<sup>3</sup>, Tomoki Fukuda<sup>3</sup>,  
Masahiro Wakutani<sup>3</sup>, Tomoya Hatanaka<sup>3</sup>, Mami Uchida<sup>3</sup>,  
Mikio Abe<sup>5</sup>, Akihiro Mikami<sup>5</sup>, Takashi Otsuki<sup>5</sup>, Zhiyang Qi<sup>6</sup>,  
Michimasa Inaba<sup>6</sup>, Kei Harada<sup>6</sup>, Daisuke Katagami<sup>7</sup>, Hirotaka Osawa<sup>8</sup>, Fujio Toriumi<sup>3</sup>,  
<sup>1</sup>Shizuoka University, <sup>2</sup>University of Tsukuba, <sup>3</sup>The University of Tokyo,  
<sup>4</sup>The University of Tokyo Hospital, <sup>5</sup>Yamagata University,  
<sup>6</sup>The University of Electro-Communications, <sup>7</sup>Tokyo Polytechnic University, <sup>8</sup>Keio University

## Abstract

We held our 5th annual AIWolf international contest to automatically play the Werewolf game “Mafia”, where players try finding liars via conversations, aiming at promoting developments in creating agents of more natural conversations in higher level, such as longer contexts, personal relationships, semantics, pragmatics, and logics, revealing the capabilities and limits of the generative AIs. In our Natural Language Division of the contest, we had six Japanese speaking agents from five teams, and three English speaking agents, to mutually run games. By using the game logs, we performed human subjective evaluations and detailed log analysis. This paper is jointly written by the organizers and the participants. We found that the entire system performance has largely improved over the previous year, due to the recent advantages of the LLMs. However, it is not perfect at all yet; the generated talks are sometimes inconsistent with the game actions, it is still doubtful that the agents could infer roles by logics rather than superficial utterance generations. It is not explicitly observed in this log but it would be still difficult to make an agent telling a lie, pretend as a villager but it has an opposite goal inside. Our future work includes to reveal the capability of the LLMs, whether they can make the duality of the “liar”, in other words, holding a “true” and a “false” circumstances of the agent at the same time, even holding what these circumstances look like from other agents.

## 1 Introduction

Recent achievements of generation models, e.g. ChatGPT (OpenAI, 2023), are gathering greater

attentions. However, it is not fully investigated whether such a huge language model can sufficiently handle coherent responses, longer contexts, common grounds, and logics. Our shared task, AIWolfDial 2023, is an international open contest for automatic players of the conversation game “Mafia”, which requires players not just to communicate but to infer, persuade, deceive other players via coherent logical conversations, while having the role-playing non-task-oriented chats as well. AIWolfDial 2023 is one of the INLG 2023 Generation Challenges for this year. We believe that this contest reveals not just achievements but also current issues in the recent huge language models, showing directions of next breakthrough in this area.

“Are You a Werewolf?”, or “Mafia” (hereafter “werewolf game”), is a communication game conducted solely through discussion. Players must exert their cognitive faculties fully in order to win. In the imperfect information games (Bowling et al., 2015), players must hide information, in contrast to perfect information games such as chess or Go (Silver et al., 2016). Each player acquires secret information from other players’ conversations and behavior and acts by hiding information to accomplish their objectives. Players are required persuasion for earning confidence, and speculation for detecting fabrications.

We propose to employ this werewolf game as a novel way of evaluations for dialog systems. While studies of dialog systems are very hot topics recently, they are still insufficient to make natural conversations with consistent context, or with complex sentences. One of the fundamental issues is

Correspondence to kano@kanolab.net

a lack of an appropriate evaluation. Because the Werewolf game forces players to deceive, persuade, and detect lies, neither inconsistent nor vague response are evaluated as “unnatural”, losing in the game. Our werewolf game competition and evaluation could be new interesting evaluation criteria for dialog systems, but also for imperfect information game theories. In addition, the werewolf game allows any conversation, so the game includes both task-oriented and non-task-oriented conversations.

We have been holding an annual series of competition to automatically play the Werewolf game since 2014 (Toriumi et al., 2017), as the AIWolf project<sup>1</sup>. Our competitions were linked with other conferences such as the competitions in IEEE Conference On Games (CoG), ANAC (Automated Negotiating Agents Competition) (Aydođan et al., 2020)(Lim, 2020) in International Joint Conference on Artificial Intelligence (IJCAI), Computer Entertainment Developers Conference (CEDEC), etc., in addition to our AIWolfDial 2019 workshop at INLG 2019 (Kano et al., 2019). These mean that our contests attract interests from communities of many areas including dialog system, language generation, task- and non-task-oriented conversations, imperfect information game, human-agent interactions, and game AI.

We have been providing two divisions in the contests: the protocol division and the natural language division. The protocol division uses our original AIWolf protocol which is designed for simplified language specific to the Werewolf game player agents. In the natural language division, player agents should communicate in the natural languages (English or Japanese). The natural language division is simple and natural goal of our project, but very difficult due to its underlying complexity of human intellectual issues. We focus on this natural language division in this report.

In the natural language division of our contest, we ask participants to make self-match games as preliminary matches, and mutual-match games as final matches. Agents should connect to our server to match, i.e. participants can run their systems in their own servers even if they require large computational resources. The game logs are evaluated by human subjective evaluations.

Eight agents of seven teams participated in this AIWolfDial 2023 shared task. Because our games are held by five players, we held a mutual match

game in the Japanese language by six agents from five teams, and another mutual match game in the English language by three teams.

In the following sections, we explain the game regulations of the AIWolf natural language division in Section 2, detailed system designs for each agent in Section 3, results of subjective evaluations in Section 4.2 followed by discussions in Section 5, finally conclude this paper in Section 6. This paper is jointly written by the organizers and the participants, i.e. Section 3 is written by each participant, the other sections are by the organizers, thus “we” stand for the organizers except for i.e. Section 3.

## 2 Werewolf Game and Shared Task Settings

We explain the rules of the werewolf game in this section. While there are many variation of the Werewolf game exists, we only explain the our AIWolfDial shared task setting in this paper.

### 2.1 Player Roles

Before starting a game, each player is assigned a hidden role from the game master (a server system in case of our AIWolf competition). The most common roles are “villager” and “werewolf”. Each role (and a player of that role) belongs either to a villager team or a werewolf team. The goal of a player is for any of team members to survive, not necessarily the player him/herself.

There are other roles than the villager and the werewolf: a seer and a possessed. A seer belongs to the villager team, who has a special talent to “divine” a specified player to know whether the player is a human or a werewolf; the divine result is notified the seer only. A possessed belongs to the villager team but his/her goal is win the werewolf team.

A game in the AIWolfDial 2023 shared task have five players: a seer, a werewolf, a possessed, and two villagers.

### 2.2 Day, Turn and Winner

A game consist of “days”, and a “day” consists of “daytime” and “night”. During the daytime phase, each player talks freely. At the end of the daytime, a player will be executed by votes of all of the remained players. In the night phase, special role players use their abilities: a werewolf can attack and kill a player, and a seer can divine a player.

---

<sup>1</sup><http://aiwolf.org/>

In the shared task, Day 0 does not start games but conversations e.g. greetings. A daytime consists of several turns; a turn is a synchronized talks of agent, i.e. the agents cannot refer to other agents' talks of the same turn. We set a maximum limit of ten turns per day in AIWolfDial 2023.

The victory condition of the villager team is to execute all werewolves, and the victory condition of the werewolf team is to make the number of villager team less than the number of werewolf team.

### 2.3 Talk

An AIWolf agent communicates with an AIWolf server to perform a game. Other than vote, divine, and attack actions, an agent communicates in natural language only.

We intend to design our shared task to be played by physical avatars in real time in future, rather than to limit to communications in the written language. Therefore, a talk text should be able to pronounce verbally, while symbols, emojis, and any other non-pronounceable letters are not allowed.

Because of the same reason, we set the maximum response time to be five seconds in the prior contests. However, we set the response timeout to be five minutes in this year, because we expected that many participants would use external web APIs such as ChatGPT, which could cause longer response time. We hope to shorten this talk timeout again in future.

In this text-base multiple player game, it is not clear that an agent speaks to which specific agent, or speaks to everyone. Human players can use their faces and bodies to point another player. In order to specify which agent to speak to, an agent may insert an anchor symbol (e.g. ">>Agent[01]") at the beginning of its talk.

Player agents are asked to return their talks agent by agent in a serial manner, which order is randomly changed every turn. This is different from the humans' verbal turn taking in that humans can speak (mostly) anytime.

## 3 Participant Systems

Six agents from five teams participated our shared task in the Japanese language, which agent names are **am**, **ChatWolf**, **k2b\_ara**, **k2b\_shigemura**, **kanolab**, **sUper\_IL**). Three agents from three teams participated in the English language, which agent names are **HowlGPT**, **MIV**, and **kanolab**,

where **kanolab** is their English version of their original Japanese agent. Most of the agents used ChatGPT in their system, while its usage is different between the agents; **ChatWolf** uses another LLM, **am** employed a rule-based system.

We, the organizers, provided a template agent code in Java and Python, in addition to the server codes.

We describe each participant system in an alphabetical order in the following subsections. where "we" stands for the corresponding participants, only limited in this section.

### 3.1 am

**am** is created by Mikio Abe, and Akihiro Mikami in Yamagata University.

We used an agent by `m_cre`<sup>2</sup>, the runner-up in the Natural Language Division of the 4th AIWolf Competition held in 2018, which was a rule-based agent system. For natural language analysis, `m_cre` use the morphological analyzer "Juman"<sup>3</sup>, "KNP"<sup>4</sup> which performs syntactic analysis, case analysis and reference resolution of Japanese sentences, and a dictionary to which we added special words that appear in Werewolf games. Our goal was to build an agent that could speak naturally like a human by adding the utterance patterns to the `m_cre` agent, which are seen in a Werewolf game between humans. We have added the following five elements.

**The reaction against a CO of Seers** When another player makes a CO (Coming Out of roles) of Seers, our agent responds to that player.

**The reaction against a report of divination results** When another player makes a report of divination results, our agent responds to that player.

**The utterance declaring the candidates for voting at the end of the day** Our agent declares who to vote for at the end of the day.

**The utterances when distressed** If there is nothing to say, our agent says something like "ummm".

**The questions that follow the flow of the game** Our agents speak when they have a question about a game situation. For example, we ask other agents who is the Seer at the beginning of the day, or we ask the agent who divined our agent why he divined us.

<sup>2</sup><https://github.com/mcre/aiwolf-4th-nlp>

<sup>3</sup><https://nlp.ist.i.kyoto-u.ac.jp/?JUMAN>

<sup>4</sup><https://nlp.ist.i.kyoto-u.ac.jp/?KNP>

## 3.2 ChatWolf

**ChatWolf** was developed by Hisaichi Shibata, Soichiro Miki, Yuta Nakamura of the University of Tokyo Hospital.

### 3.2.1 Design

We designed the agent to have two models: the talk model and the voting model. The talk model can talk with other agents while the voting model can vote (including attack voting) for the other agents. These models run separately, and respond to queries from the game server. We did not handle the divination in the game with models and ChatWolf divines other agents at random. We adopted one of the LLMs (Large Language Models) open to the public from CyberAgent (Open-Calm-7b<sup>5</sup>) both for the talk model (we used `AutoModelForCausalLM` in hugging face transformers) and the voting model (we used `AutoModelForSequenceClassification` in hugging face transformers). Each model contains approximately 7 billion parameters. For the talk model, to adapt the LLM to the Werewolf game, we executed a LoRA (Low-Rank Adaption for LLM (Hu et al., 2021)) tuning with Japanese Werewolf game logs<sup>6</sup> newly collected by the developers of ChatWolf. To systematically collect the game logs, we developed UTRAD Werewolf, which is a web browser-based platform to play the Werewolf game with or without artificial intelligence agents. We collected 48 game logs (not open to the public). For details of the log collection, see (Shibata et al., 2023). For conditional text generation with the talk model, we enabled the sampling to generate diverge texts. The temperature of the sampling was experimentally set to 0.7 and the repetition penalty to 1.05. For the voting module, which can interpret intermediate Werewolf game logs in Japanese and vote for agents to attack or eliminate, we attached a classification head on the LLM and trained parameters in it with the same logs. The training method with the classification head is well known as transfer learning whereas LoRA is one of the fine-tuning methods. The prompt (input) to the both models contained instructions of the

<sup>5</sup><https://huggingface.co/cyberagent/open-calm-7b>

<sup>6</sup>The log collection was partially supported by JST, CREST Grant Number JPMJCR21M2, including the AIP challenge program (Necessary conditions that personal claims are objective facts; PI: H. Shibata), Japan. We thank many participants who played Werewolf again and again to generate game logs.

game, the latest 15 talks by all the participants, the role and agent number of ChatWolf, and the previous voting result if it exists. We executed the inference with those models on a single NVIDIA RTX-A6000 graphics processing unit with 48 GB memory.

### 3.2.2 Background of ChatWolf design

Because the evaluation criteria of the regulation for the natural language division do not refer to the win ratio of the Werewolf game, we decided to give priority on fluently talking natural language than to winning the game.

If we adopt GPT-4 (Generative Pre-trained Transformers 4; (OpenAI, 2023)) a.k.a. ChatGPT<sup>7</sup>, it is expected to very fluently talk natural languages. However, we specified disadvantages of GPT-4 as follows:

- The tuning of the agents ends up in prompt-engineering and not very interesting.
- Many teams would participate in the contest with GPT-4. In the case, agents powered by GPT-4 must be highly tuned up to differentiate with the other GPT-4-based agents.
- Agents powered by GPT-4 could not be significantly novel.
- Should the server of GPT-4 be temporarily down, we cannot play the game at all.

On the basis of this insight, we decided to develop ChatWolf running on a local computer.

## 3.3 HowIGPT

HowIGPT is, when boiled down, a simple 3 layer chain-of-thought reasoner.

At the lowest layer are the language models. An abstraction layer allows our system to direct templated patterns of (recursive) queries either to a locally running transformer, a distributed "Petals" backend, or the OpenAI API (gpt-3.5-turbo-16k).

Context lengths are fixed to 8K tokens for prefix input and 8K generation when using OpenAI. Chunk sizes of 16-64 are used for top-k generation. These calls are used for "large context" reasoning and decision making.

Local models are used for reasoning and decision-making. Local models deployed include: `mpt-7b-storywriter`,

<sup>7</sup><https://chat.openai.com/auth/login>

mpt-7b-instruct, gpt-medium,  
gpt-xxl, Nous-Hermes-13b,  
vicuna-7b-1.1.

For local models, context lengths are sized appropriately to the model in use.

Different models were used for their different strength. For example, gpt and bloom are largely used for knowledge-management and summarization while mpt is used for introducing creative elements to roleplay effort and Nous is used for "smaller context" commonsense reasoning.

At the second layer, calls to/between these models are coordination by a query language runtime.

Two or three values from a large table of "character descriptors" are chosen for each instance, to give the agent some personality direction. An initial description and small "backstory" is generated for the instance character and given along with each query to the language models.

Four "temperature" values are also chosen randomly - these are used to control sampling and beam search for queries. These values are a "summarization" temperature which controls sampling for internal summarization processes, a "thought" temperature which controls sampling during chain-of-thought queries, a "choice" temperature (fixed to 0 for the competition play but allowed to vary during training) which controls temperature when making vote decisions, and a "talking" temperature for variability of spoken statements. (This makes agents less predictable and their roles less identifiable to other agents.)

A simple sockio.ai socket is opened to the game server and a small FSM handles the game protocol messaging. The events observed are translated from their json representation into english representations, embedded and saved into a knowledge retrieval store, and stored in in-memory structures for use in prompt constructions. Whenever new conversation is seen, or new context is integrated, lengths of queries are re-checked to confirm that they still fit within the language model context widths. If thresholds are passed (for example, 80% of the gpt-3.5-turbo-16k's 8k input window is consumed) then the agent summarizes existing context information before appending the newly received information.

A third RNN model layer is used as a supervisory, to modify/reject any outputs from the models which are deemed likely low quality or otherwise problematic. This supervisory layer was trained with a combination of self-play and human super-

vision.

### 3.4 k2b\_ara

**k2b\_ara** is created by Takuya Okubo, Kazuki Takashima, Tomoya Hatanaka, Mami Uchida, Rei Ito in the University of Tokyo.

We have developed an agent that performs the following functions using BERT (Devlin et al., 2018), GPT-3 (text-davinci-003) and GPT-4 (gpt-4-0613) (OpenAI, 2023).

- Infer roles
- Plan strategies
- Accept or reject requests
- Answer questions

To actualize these features, our agent is composed of eight different modules:

**RoleEstimationModule** Estimates the role distribution of each agent based on game information (the number of each role, talk content, and divination results if available) utilizing BERT<sup>8</sup>. It considers the sum of attentions from the other tokens to the first token ([CLS]) across all 12 multi-head attention layers in BERT as the basis for these estimates and makes a list of word-attention pairs for other modules to use<sup>9</sup>.

**RoleInferenceModule** Infers the roles of each agent by GPT-4, which is given the estimation basis (a list of word-attention pairs obtained from the fine-tuned BERT model above) and an certain agent's role distribution derived from the **RoleEstimationModule**. It receives a list of word-attention pairs as the basis for inference, selecting the top 10 phrases from this list as the rationale. Then, it gets the response (inference result) from the GPT-4 with the prompt, which includes the above rationale, followed by an additional prompt, *"Please infer the agent's role based on the above information and state the logical reason why you think so."*

<sup>8</sup>we used a pretrained model <https://huggingface.co/cl-tohoku/bert-base-japanese-whole-word-masking> and fine-tuned it with about 500 werewolf game logs scraped from an online werewolf game server <https://ruru-jinro.net/>

<sup>9</sup>i.e. if "hello, I divined Agent[01] werewolf" is in the talk history, then the list includes [{"hello",0.01},{"I",0.03},{"divined",0.9},{"Agent[01]",0.1}, {"werewolf",0.7}]



**StrategyModule** Determines whom to vote for, whom to divine, and how to persuade other agents with GPT-4, based on the role estimation results obtained from **RoleEstimationModule** and **RoleInferenceModule**. For instance, it selects the agent with the highest probability of being a werewolf or a possessed as the voting target calculated by **RoleEstimationModule**, and generates the reason for the role estimation using **RoleInferenceModule**. These information are then provided to GPT-4, with an additional prompt, "*Based on the above reasons, please persuade other agents to vote for Agent[number].*", to generate a persuasive sentence. The divination target is also chosen as the agent with the highest probability of being a werewolf or a possessed.

**RequestProcessingModule** Categorizes the requests from others into three types (whom to vote for, whom to divine, or other requests) using GPT-3 with few shot prompt, such as "*Q: Shall we vote for Agent[01] A: whom to vote for*", "*Q: Would you mind divining Agent[01] next? A: whom to divine*" and "*Q: Could you believe me a villager? A: other requests*". Then, it determines whether to comply with the request using the **StrategyModule** and **RoleEstimationModule** and generates the response based on rule-based procedures.

**QuestionProcessingModule** Classifies the questions from others into four types (past behaviour, future plan, its role prediction, and other questions) with GPT-3 with few shot prompt, such as "*Q: Who do you think as a possessed? A: other requests*", "*Q: Why did you vote him? A: whom to vote for*" and "*Q: Could you tell me the reason you divined Agent[01]? A: whom to divine*". Then, it determines the response to the question using the **StrategyModule** and **RoleEstimationModule** and generates the response based on rule-based procedures.

**InfluenceConsiderationModule** Determines whether the statement of a certain agent is directed at itself using GPT-3 with few shot prompt, such as "*Q: I think Agent[01] is a werewolf A: no*", "*Q: please believe me a seer! A: request*" and "*Q: Who do you all vote for? A: question*". If it's a request or a question, then calls **RequestProcessingModule** or **QuestionProcessingModule** respectively and returns the response to its caller.

**SpeakerModule** Transforms personality-less speech content into personality-rich speech content using GPT-4.

**IntegrationModule** Integrates above seven modules and then sends information to the server. For instance, when in the talk phase, this module selects speech content from **InfluenceConsiderationModule** if it is not empty. Otherwise, it selects speech content from **StrategyModule**. Then, it enriches the speech content with **SpeakModule** and send it to the server. When in the vote phase (divine phase), it asks **StrategyModule** for whom to vote for (divine) and send it to the server.

### 3.5 k2b.shigemura

**k2b.shigemura** is created by Soga Shigemura, Tomoki Fukuda, and Masahiro Wakutani in the University of Tokyo.

#### 3.5.1 Design

We utilized a mixed model approach, using both gpt-4-0613 and gpt-3.5-turbo-0613, which are OpenAI's models<sup>10</sup>. This allowed us to leverage the precision of GPT-4 and the speed of GPT-3.5. In terms of parameters, we found that the best results were achieved by retaining default values, aside from `max_tokens` which we set at 300 (`temperature=1.0`, `top_p=1.0`, `presence_penalty=0`, `frequency_penalty=0`). With the exception of prompts related to conversational expressions, most of the prompts are written in English. This is because prompts in English tend to have higher accuracy and logical coherence, while prompts related to conversational expressions in Japanese are better conveyed in Japanese, resulting in greater naturalness and creativity by using the same language.

We also incorporated the OpenAI's **Function calling** feature, which was made public in June 2023<sup>11</sup>. This Function calling feature can prevent unintended responses when making requests to OpenAI's API, by sending a list of functions specified with JSON schema<sup>12</sup> that defines the types (such as string, integer, enum) as parameters.

<sup>10</sup><https://platform.openai.com/docs/models/overview>

<sup>11</sup><https://openai.com/blog/function-calling-and-other-api-updates>

<sup>12</sup><https://json-schema.org/understanding-json-schema/>

### 3.5.2 Model

**Conversation Summarization Model** Since it was impossible to incorporate all game conversation into the prompt, we developed a **Conversation Summarization Model** using `gpt-3.5-turbo` to summarize each agent’s statement. To prevent usage of the term ”agent” within the dialogue and to simplify the conversation history, we converted “Agent[01]” to “[1]” in advance, for example, then reverted “[1]” back to “Agent[01]” after LLMs processed. Inserting this **Conversation Summarization Model** allowed us to transform self-perspective statements like “*I AM ABSOLUTELY THE SEER!*” into sentences like “*Agent[01] claims to be the seer.*” This adjustment reduced the influence of blindly trusting other agents’ statements when generating our own dialogue or making decisions. It also prompted the addition of logical coherence to our statements, such as “... *because Agent[03] has revealed Agent[05] as innocent*”.

**Dialogue Generation Model** We employed GPT-4 for dialogue generation to ensure logical coherence and richness of expression. We provide the following information to the model:

- Explanation of the rules of the werewolf game
- Each agent’s survival status
- Divination results
- Voting results
- The agent’s role and recommended behavior in the current situation

For example, if it is the first day as a seer, this agent should announce its divination results. If it is the first day as a villager, this agent would wait for the seer’s results. In case of a werewolf or a possessed, if no other agent has claimed the seer role, then this agent should claim to be the seer.

Instead of including all the “if” conditions in the prompt, we varied the prompt depending on the game situation. This is because LLMs are not proficient at handling conditional branching.

Initially, our agent primarily make non-committal and vague statements, such as “*More information is needed,*” or “*Let’s discuss this carefully.*” In response to this, we have add prompts to express doubts about other players even when the agent is not completely sure in scenarios with limited information. By doing this, our agent could

influence the dynamics of the game and stimulate more revealing interactions, which in turn could lead to more data for making informed decisions.

**Action Decision Model** This model is used for making decisions such as voting, determining the victim as a werewolf and a divination target as a seer. We utilized the OpenAI’s **Function calling** feature, which can limit the type of response, to avoid selecting dead players or giving non-committal answers like “*I cannot decide which agent to vote due to a lack of information*”. The information provided to this model is the same as our **Dialogue Generation Model**.

### 3.5.3 Discussion

Utilizing the functionalities of GPT-4, our model could generate statements that were both natural and logical, such as “*If Agent[02] were a werewolf, they wouldn’t engage in such conspicuous behavior,*” and “*Both Agent[01] and Agent[02] claim to be the seer, but they both present the same results, leaving us in a dilemma over whom to trust.*” These instances demonstrate the advanced language understanding and generation capabilities of the model.

There were no agents that performed prompt injection in this tournament. However, we aim to incorporate countermeasures in upcoming competitions. Nevertheless, it is conceivable that such practices may be prohibited by the rules from the beginning.

In future, a primary obstacle to achieving our goal of humans and AI playing the Werewolf game competitively is the latency often associated with Large Language Models (LLMs) and machine learning technologies. For reference, our agent currently takes approximately 15-30 seconds to generate a complete statement. Yet, by using stream setting (in OpenAI’s API, `stream=True`), we can have statements generated incrementally, thus minimizing the waiting time for humans, irrespective of whether they are communicating in text or speech. We believe that the day when AI and humans can play the Werewolf game without interruption is not too distant.

## 3.6 kanolab

**kanolab** is created by Neo Watanabe, Kaito Kagaminuma, and Yoshinobu Kano in Shizuoka University.

We focused on the two main features of ChatGPT: its ability to generate fluent natural language, and its ability to perform some inference. We explain our game behaviors below, i.e. **talk**, **vote**, **attack**, and **divine** in this order.

### 3.6.1 GPT-4 and its settings

We used GPT-4 with setting parameters to `model = gpt-4-0613`, `temperature = 0.7`, `top_p = 0.75`, `frequency_penalty = 2`, `max_tokens = 300`, leaving other parameters at their defaults.

### 3.6.2 talk

In the **talk** behavior, the prompts were given five elements to generate natural conversation in the Werewolf game: **character setting**, **game strategy**, **game conversation examples**, **conversation history**, and **instructions regarding the specific content of speech**.

**character settings** This prompt includes *name*, *nickname*, *gender*, *personality*, *hobbies*, and *occupation*; *name* specifies the Agent’s name, and *nickname* specifies the name by which the agent is called by other agents, such as Agent[01]; The other *gender*, *personality*, *hobbies*, and *occupation* are randomly selected from our predefined ones.

**game strategy** This prompt includes prompts important for the game to proceed strategically, such as *Werewolf game roles and number of roles*, *assigned role*, *strategic direction*, and *specific strategy*.

The *werewolf game roles and number of roles* prompt is in the form of “*Villager:1, Possessed:1*” to clarify the roles assigned in the game and how many people are in them. *assigned role* includes a prompt as *Your position is villager*. There was a concern that the agent would openly disclose (come out) his or her role, especially problematic in *possessed* and *werewolf*; we give a prompt as *seer* if *possessed* assigned, and *villager* if *werewolf* assigned to avoid such behaviours.

The *strategic direction* prompt is “*Lead the villagers to victory by what you say.*” when *villager* or *seer* is assigned, and “*Lead the Werewolf camp to victory with your statement*” when *possessed* or *werewolf* are assigned.

The *specific strategies* prompt includes strategies that are generally known to be effective in the werewolf games in order to encourage rational behavior. We give two strategies: if there are more than two

seers come out, the following one divines the same player; When there are multiple seers, players vote around the agent who comes out as a seer.

**werewolf game conversation examples** This is a few-shot prompting to give five or around examples from the logs of the past werewolf games, expecting that the use of anchors and ways of speech during game behavior are learned.

**conversation history** We give the all agents’ previous conversation history as a prompt, aiming at contextualizing the conversation with other agents. Because the input token length of ChatGPT is 4096, we make a summary of the conversation history when the token length of the conversation history exceeds every 1000 tokens, excluding last five talks; this summary is incrementally generated by another GPT-4, inputting that latest conversation history and the previous summary if any exists. The main GPT-4, which is in charge of generating talks, is given the summary and a part of conversation history which is not summarized yet. We also give information which agents are exit from the game by votes and attacks.

**specific speech instructions** We give instructions on what kind of speech we wanted the agents to say. For example, on Day 0, the participants are not engaged in a conversation about the Werewolf game but in the greeting phase with other agents, so they are prompted to greet everyone before the game. From Day 1, the agents are expected to engage in conversations related to the Werewolf game, so we give the following prompts: *organizing the situation based on the conversation history*, *predicting the positions of other agents and discussing who can be trusted*, *soliciting opinions from other agents*, *not repeating the same conversation*, and *being logical in what you say*. When *seer* is assigned, we asked to speak that “*As a result of the fortunetelling, Agent[01] was white. Please tell this result to the other players*” in the first talk of the day from the first day.

### 3.6.3 vote, attack, and divine

The agent is require to nominate another agent in the *vote*, *attack*, and *divide* game actions, which are expected to be reasoned with based on the conversation history. We give a prompt to GPT-4, which consists of a summary of the previous conversations, the conversation history, to choose another agent who is considered to be a threat to our agent’s

role. For example, we asked the agents in the *vote* action that “Based on the summary and conversation history, answer the number of the player you consider to be the most werewolf among Agent[01], Agent[02], and Agent[03].” and asked them to nominate one agent. If no agent nomination was obtained from ChatGPT, we randomly choose an agent.

When *possessed* is assigned, we use a special pattern of prompts. On Day 1, our agent is asked to nominate a player who should be a *seer*; If ChatGPT finds a *seer* from the conversation history, we ask generating a talk that he/she is a *werewolf*; Else we randomly choose a agent and ask generating a talk that he/she is a *werewolf*. On Day 2 and later, we ask to nominate which agent to divine, and ask generating that agent is a *villager*.

### 3.7 MIV

MIV is developed by Jaewon Lee and Benedek Hauer in The University of Tokyo.

#### 3.7.1 Background

The emergent capabilities of Large Language Models (LLMs) are attracting attention (Wei et al., 2022). Emergent capability is defined as, “an ability is emergent if it is not present in smaller models but is present in larger models.” This ability manifests when the number of parameters in a language model surpasses a certain threshold. Specifically, it refers to the In-Context-Learning ability for learning situational information within a particular context.

The capability required of agents in this competition is a “human-like speaking ability.” Until now, computational machinery has lacked the ability to “read” context like humans. However, it is proposed that by fully harnessing the emergent capabilities of large language models, agents can learn situational information, engage in speech, and plan and execute actions (Park et al., 2023). In this competition, our focus is on fully exploiting this emergent capability.

#### 3.7.2 Generative Agent

In this competition, the “agent” must fulfill four major roles. That is to say, each agent must maintain consistency in order to fulfill their expected roles. Only when this consistency is ensured can the acquired situational information be fully utilized. To embody this condition, we employ the

“Generative Agent” framework proposed by Park et al. 2023.

Utilizing this framework, based on the defined agent roles, allows for customization of intentions, personality, and other aspects while maintaining consistency. Additionally, by providing the agent with a memory structure, information can be extracted based on what has already been communicated.

The memory structure of the agent works in a weighted way, where each memory (string) is assigned a weight. This weight depends on the importance of that memory and on the recency of the memory. Therefore, an important information that was recently given to the agent will affect the agent’s next behavior/action more than an old less important information.

In our case, the way our main agent system interacts with our helper agent is through prompts (natural language) as it is probably the easiest way for the human system designers; we can accentuate some words or tell the agent if some task/information is relevant or not, which seems very well suited for this task. For creating our generative agent, we used a Python library called [Langchain](#), which contains a section about [Generative Agents](#) that provides more precision on their mechanism and how to implement them.

#### 3.7.3 Implementation

We implemented an agent of the following roles: Werewolf, Seer, Villager and Possessed. In this section, we explain the general skeleton of our agent first, that is common in these roles. Then we explain details for each role.

**General Skeleton** When creating an agent, the first thing that we do is initialize its memory and fill it with information related to the game rules and mechanism.

In [Langchain](#), there is a module called `GenerativeAgentMemory`, that will generate the memory for a particular agent, by specifying an LLM, a reflection threshold and a function that acts as a memory retriever. For the memory retrieval, we used [Faiss](#), a library for efficient similarity search coupled with a time-weighted vector store retriever which evaluates memory elements (strings) based on their importance. For the LLM we used ChatGPT of OpenAI, and for the reflection threshold we used 8, however we don’t know how this parameter changes the behaviour of the agent

since we did not experiment with other values. At every game action (talk, vote, divine, attack), we tell corresponding information to our agent to update its memory. For example, if `Player[01]` votes to eliminate `Player[02]` on Day 3, then our agent will be told: *"Player[01] has voted to eliminate Player[02] on day 3"*.

Following subsections describe our implementation for each game action; then describe our implementation for each role, and "personalities".

**talk** Our core agent program behaves like a helper that gives our main agent program hints on what to do.

Our main agent program interprets what the agent is saying, and transforms into a talk text to be returned; our core agent will look through its memory, say something relevant, or ask a question (or provide an answer to another player if necessary) in natural language. Since our core agent acts as a helper, it will return statements such as *"I think you should say the following: "Do you think that Player[02] is a werewolf?""* or *"You can say "I am not a werewolf because I was asleep that night" so that people don't realize you are a werewolf"*. Since what we should say is inside quotation marks, we will simply return the text that is contained within the quotation marks, which corresponds to the transformation process.

**vote** We tell our agent that it is now time to vote, and it should vote for the most dangerous player out there. That is, if a werewolf knows who a seer is, then we would like the werewolf to vote out the seer; if a seer (or a villager) knows who a werewolf is, then that agent would vote out the werewolf.

Suspiciousness of another player is not determined by some metric, rather it is a decision that is completely up to the generative agent to tell based on the context it has received throughout the whole game so far. When it is our turn to vote, we prompt the agent: *"It is your time to vote, pick a player who thinks we should eliminate based on the events that happened so far"*. We parse the agent's output and look for a particular player's name, i.e. if the sentence contains `Agent[05]` then we will vote to eliminate `Agent[05]`.

**divine** A seer agent picks a player whose role is unknown for the agent yet. The process works in the same way as the vote. We prompt the agent: *"Since you are the seer you can get to know another player's role. Don't choose yourself or*

*a player whose role you already know"*. Then, from the reply we extract the name in the same manner as for the vote mechanism. When the divined player's role is revealed, we will tell our seer agent that role. For example, if the divination result was that `Player[01]` is a werewolf, we supply the following information to our core agent: *"You have just discovered that Player[01] is a werewolf. This is an important information and you should let the other players know."*

**attack** A seer is the most dangerous role for a werewolf because a seer can reveal the werewolf's identity by divination. Therefore, when our werewolf agent is requested to attack, our agent will most likely attack the seer if it knows who the seer is. The way our agent is prompted similar to the following: *"It is now your time to attack someone as the werewolf. Tell me the name of the player you want to attack. If you know who the seer is, then it is better to attack them to increase your chances of victory"*. This way, if the seer has already revealed themselves, the werewolf will most likely make the decision to attack them.

In the following subsections, we describe the general personalities and behaviours of our agents.

Langchain's generative agents can be supplied two different string fields:

- `traits`: traits, characteristics and personalities of that agent
- `status`: the goal of the agent, what its purpose is

This is very convenient for us, because in the Werewolf game, each player has to have a different personality and each player has a different goal (or at least not all of them have the same goal).

**Seer** For the seer role, we set the `traits` to be very positive, i.e. they are honest, transparent, patient, etc. as they are basically the leader of the villagers (non-malicious). We set its `status` just to find out who a werewolf is.

**Villager** The villager role is the most basic one, and their `trait` do not really matter, but we designed them to be somewhat aggressive towards other players (i.e. getting straight to the point), but also transparent and honest. Their goal (`status`) is to find out who the werewolf is.

**Possessed** The possessed role is one of the trickiest, and we believe that our implementation is still not perfect for that role, as its main goal is to deceive, and language models are either too obvious or do not achieve deception at all, at least from what we experienced. We therefore implemented its `traits` as a compulsive liar, weird, funny and dishonest, and its `status` as "pretends to be a werewolf".

**Werewolf** The werewolf role is basically the complete opposite of the seer. Its `traits` are "dishonest, liar, non-transparent", and its `status` is "blend in with other humans and act like you are not a werewolf".

**Worst-Case-Scenario** Sometimes, the response of our agent (operated using the OpenAI API GPT3.5-Turbo) is not very precise with its game actions, so we cannot always guarantee that our agent will make an appropriate decision. Therefore, when we are not able to infer the intention of the agent during the `vote`, `divine` and `attack` phases, we will override the core agent's response by providing a random player in the pool, though such a worst case scenario is rarely used. This situation can happen when the agent does not include another player's name in its answers. For example, if it answers "I don't want to attack anyone", or "I'm not sure who is the werewolf so I cannot decide who to vote out", then in this case we will choose a random player in the list of players, using the same mechanism as a `RandomTalkAgent` provided by the organizers.

**Disadvantages** A disadvantage of our agent is passiveness, sometimes being fixated on a single issue during the whole game. In our self-match games (five MIV agents playing together), we noticed that our agents are not very good at making decisions and taking leadership, sometimes they were so fixated on one small detail that they kept talking about it throughout the whole day without making much progress. We believe that a workaround to this issue would be to filter out some information from the agent's memories, which is long-term and irrelevant. In order to do so, we would need to manually implement a mechanism that filters out irrelevant our outdated information from the agent's memory. However, since this mechanism would be very dependent on the situation, the conversation, and the game, defining such a rule was out of our reach, because we would still need to keep the "important" and "most-relevant" information for the

agent to be functional.

### 3.8 sUper\_IL

**sUper\_IL** is created by Zhiyang Qi in The University of Electro-Communications.

The `sUper_IL` system is based on the ChatGPT API (OpenAI, 2023). It is widely known that using suitable prompts with the ChatGPT API often leads to good text generation. Therefore, this system adjusts the prompt according to the role and game progress to facilitate dialog generation in different situations.

Specifically, for the four roles of werewolf, villager, seer, and possessed, four different prompts were created and adjusted based on the "day" progression. These prompts include the basic rules of the game, the agent number, the assigned role, and role-specific instructions. For example, for the werewolf, the prompt could be as follows: "The basic rules are... you are Agent[x], and your role is werewolf. The following are essential guidelines: 1. Never reveal that you are a werewolf to conceal your identity; 2. If someone accuses you of being a werewolf, resist strongly; 3. Provide false information or view other players with suspicion to protect yourself; 4. Actively participate in conversations to gain the trust of the villagers."

On the second "day", part of the prompt will be replaced with specific information such as "It is the second day, and there are three players left. The strategy is three: 1. If the seer is present and identifies you as a werewolf, call for a vote by accusing the seer of being a werewolf; 2. If the possessed shows up, confidently state that you are a werewolf; 3. Otherwise, choose one of the remaining players, contradict their statements, and vote to oust them."

Furthermore, to ensure the consistency of the generated responses, the entire conversation history of the day was appended to the prompt each time a response was generated. Additionally, this system utilizes zero-shot prompt, which implies that the prompt do not incorporate any conversation history beyond the current game.

Due to ChatGPT's tendency to generate safe and non-advancing responses like "I agree with all your opinions. By sharing information and promoting discussion, we will find inconsistencies and suspicious points" or "I will actively participate in the discussion and help find the werewolves. I will elicit your opinions and questions and share detailed information with you", efforts were made

to minimize this behavior in self-matches. The first sentence of each day’s response is generated based on rules to avoid these responses. Additionally, different strategies were employed to ensure game variety. For example, for the werewolf, the following three possible responses were generated:

- I am a villager who is not well informed. Let’s work together to protect our village.
- I am a seer, I divined Agent[x] and the result was human. Please take this information into account in future discussions and votes.
- I am a seer, and I divined Agent[x], but the result was a werewolf. Be vigilant against Agent[x].

Each of these three responses leads to a different direction for the game, and after that response is generated, the prompt text is modified accordingly. Regarding the ChatGPT API model, the system used `gpt-3.5-turbo-0613` in the preliminary contest, and `gpt-4` in the final contest.

Lastly, the `sUper_IL` system is installed with only talk module, voting, divination and attacks based on random selection.

## 4 Subjective Evaluation Results

All of our shared task runs are in a five players werewolf games as described earlier. Our shared task runs were performed in self-matches and mutual matches. The same five player agents play games in the self-matches; different five player agents play games in the mutual-matches. The shared task reviewers are required to perform subjective evaluations based on game logs of these matches. The game logs will be available from the our website <sup>13</sup>.

### 4.1 Evaluation Metrics

We performed subjective evaluations by the following criteria, five level scores (5 for best, 1 for worst) for each:

- A Naturalness of utterance expressions
- B Naturalness of conversation context
- C Coherency (contradictory) of conversation
- D Coherency of the game actions (vote, attack, divine) with conversation contents

- E Diversity of utterance expressions, including coherent characterization

This subjective evaluation is based on both self-match games and mutual match games. This subjective evaluation is same as the evaluations in the previous AIWolf natural language contests.

### 4.2 Results

Table 1 and Table 2 show the results of the human subjective evaluations for Japanese language and English language, respectively. Four organizers, who do not commit to the participant systems, evaluated the Japanese agents; three English fluent evaluators including external staffs evaluated the English agents. Each cell ranges from 1 (lowest) to 5 (highest), the All-Average column shows averages over these human evaluators. Cells of highest scores are highlighted in bold for each metric and in total.

## 5 Discussion

### 5.1 Score-wise Analysis and Generative AIs

In this subsection, we discuss the subjective evaluation scores shown in Table 1 and Table 2.

Most of the participant systems rely on OpenAI ChatGPT, while **am** is a rule-based system and **ChatWolf** uses another LLM. **sUper\_IL** obtained the best score in average, A (expression), B (context), and C (coherency); these scores are higher in other ChatGPT-based systems, showing the natural generation performance of ChatGPT, even in context and coherency in this mostly sentence-pair level.

Regarding D (game action), **am** obtained the best score, suggesting that their hand-crafted precisely tuned rules work better than prompt-based generations. While the talk history is input as prompts, some of the talks might mislead the generation results due to the agent’s unstable superficial talks and the other agent talks; when the talk history exceeds the maximum input length, some of the talk history could be missed which are important to decide the game actions.

Regarding E (diversity), **ChatWolf** and **Kanolab** obtained the best scores in Japanese, **HowIGPT** in English. The reason would be that **ChatWolf** does not rely on ChatGPT but uses a smaller LLM with LoRA, **kanolab** created many prompts to play different characters, and **HowIGPT** uses not just ChatGPT but other local LLMs.

<sup>13</sup><https://kanolab.net/aiwolf/>

Table 1: Subjective Evaluation Results for Japanese Language Games

Team	A Expression	B Context	C Coherency	D Game Action	E Diversity	All Average
<b>am</b>	3.400	3.350	3.450	<b>3.800</b>	2.100	3.220
<b>ChatWolf</b>	3.050	2.400	2.600	2.700	<b>4.150</b>	2.980
<b>k2b_ara</b>	4.075	3.825	3.250	3.075	3.425	3.530
<b>k2b.shigemura</b>	3.625	3.250	3.125	3.375	3.375	3.350
<b>kanolab</b>	3.575	3.900	3.750	3.500	<b>4.150</b>	3.775
<b>sUper_IL</b>	<b>4.450</b>	<b>4.200</b>	<b>4.050</b>	3.550	3.800	<b>4.010</b>

Table 2: Subjective Evaluation Results for English Language Games

Team	A Expression	B Context	C Coherency	D Game Action	E Diversity	All Average
<b>HowlGPT</b>	2.667	1.667	2.667	2.000	<b>3.333</b>	2.460
<b>MIV</b>	<b>3.333</b>	<b>3.667</b>	<b>3.000</b>	<b>3.667</b>	2.333	<b>3.200</b>
<b>kanolab</b>	<b>3.333</b>	2.667	<b>3.000</b>	2.667	2.667	2.860

We cannot directly compare the evaluation scores between Japanese and English because evaluators are different, the **kanolab** agent in English is the same as it in Japanese other than it adds a prompt instruction to speak in English, thus we could compare the results using the scores of **kanolab** as a pivot. Because **MIV** obtained better scores than **kanolab**, **MIV** might show good performance in Japanese as well, though not sure due ChatGPT shows better performance in English than in Japanese.

## 5.2 Log Analysis

We pick one of the mutual-game logs<sup>14</sup> to analyze in detail. Table 4 shows the game’s log of Day 1, Table 5 shows the game’s log of Day 2, and Table 3 shows the game’s players with their roles and game actions. We translated the original log in Japanese into English, cut off some of the logs which would not affect the game and talk contents.

Table 4 and Table 5 show a column of my interpretation w.r.t game actions, where “Not meaningful” means a talk text that can be used anytime, “Not make sense” means a talk that is hard interpret its meaning in the conversation context. We found that the sentence expressions are very natural throughout the conversations, but sometimes contextually wrong especially when it comes to the roles, e.g. which talk is whose one.

The game actions, votes and attacks, are quite

<sup>14</sup>[https://kanolab.net/aiwolf/2023/main/multi/0708160231\\_000\\_chatWolf\\_kanolab1\\_sUper\\_IL\\_am\\_k2b\\_ara1.log](https://kanolab.net/aiwolf/2023/main/multi/0708160231_000_chatWolf_kanolab1_sUper_IL_am_k2b_ara1.log)

inconsistent with the talks. Agent[01], sUper\_IL, did not implement the game actions but selects randomly, but other game actions seem to ignore the COs (Coming-Outs) of the roles.

## 5.3 Evaluation Metrics

A win rate could be another potential criteria, but we have not used the win rate due to the following reasons. Firstly, we cannot run sufficient number of games to measure statistically meaningful win rates, as there are many possible role combination patterns. Secondly, the agents should “understand” their utterances each other as a presumption to measure win rates, but the agent talks in the previous years were not that level of communications. Thirdly, the werewolf game itself is not necessarily intended to simply win the game, but rather aims to play an interesting game. Finally, we would like to directly measure the quality of the natural language generation; an agent could win without meaningful conversations.

## 6 Conclusion and Future Work

We held our 5th annual AIWolf international contest to automatically play the Werewolf game “Mafia”, where players try finding liars via conversations, aiming at promoting developments in creating agents of more natural conversations in higher level, such as longer contexts, personal relationships, semantics, pragmatics, and logics.

We performed human subjective evaluations and detailed log analysis. We found that the entire system performance has largely improved over the



Table 3: Roles and game actions in example log (Abbreviations in the Status columns are Excted: Executed, Attcked: Attacked)

Agent ID	Agent Name	Role	Divination	Vote #1	Vote #2	Divination	Status	Vote	Status
01	sUper_IL	Seer	Agent[02]	4	3	Agent[05]	Excted	-	
02	k2b_ara	Possessed	-	1	5	-	Attcked	-	
03	am	Werewolf	-	4	4	-	Alive	4	Alive
04	kanolab	Villager	-	2	2	-	Alive	3	Excted
05	ChatWolf	Villager	-	1	2	-	Alive	4	Attcked

previous year, due to the recent advantages of the LLMs. However, it is not perfect at all yet; the generated talks are sometimes inconsistent with the game actions, it is still doubtful that the agents could infer roles by logics rather than superficial utterance generations. It is not explicitly observed in this log but it would be still difficult to make an agent telling a lie, pretend as a villager but it has an opposite goal inside.

Our future work includes to reveal the capability of the LLMs, whether they can make the duality of the “liar”, in other words, holding a “true” and a “false” circumstances of the agent at the same time, even holding what these circumstances look like from other agents, further reflecting such observations of other agents. This would be possible by introducing the “whisper” feature which communicates with the werewolves only, employing more than five players in a game.

Another interesting demonstration would be to mix a human player with machine agents. Currently the LLM based agents talk longer time than humans to reply, sometimes minutes, thus acceleration of the agent system responses is a technical issue in future.

## Acknowledgments

We wish to thank shared task reviewers for performing the subjective evaluations, and the members of the Kano Laboratory in Shizuoka University who helped to run the shared tasks. This research was partially supported by Kakenhi, MEXT Japan. The individual system description in this paper was written by corresponding team members and reviewed by the organizers, the rest of the paper was written by the organizers.

## References

- Reyhan Aydođan, Tim Baarslag, Katsuhide Fujita, Johnathan Mell, Jonathan Gratch, Dave De Jonge, Yasser Mohammad, Shinji Nakadai, Satoshi Morinaga, Hiroataka Osawa, et al. 2020. Challenges and main results of the automated negotiating agents competition (anac) 2019. In *Multi-Agent Systems and Agreement Technologies: 17th European Conference, EUMAS 2020, and 7th International Conference, AT 2020, Thessaloniki, Greece, September 14-15, 2020, Revised Selected Papers 17*, pages 366–381. Springer.
- Michael Bowling, Neil Burch, Michael Johanson, and Oskari Tammelin. 2015. Heads-up limit hold'em poker is solved. *Science*, 347(6218):145–149.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. [BERT: pre-training of deep bidirectional transformers for language understanding](#). *CoRR*, abs/1810.04805.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuezhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.
- Yoshinobu Kano, Claus Aranha, Michimasa Inaba, Hiroataka Osawa, Daisuke Katagami, Takashi Otsuki, and Fujio Toriumi. 2019. Overview of the aiwolf-dial 2019 shared task: Competition to automatically play the conversation game “mafia”. In *In proceedings of the 1st International Workshop of AI Werewolf and Dialog System (AIWolfDial 2019), the 12th International Conference on Natural Language Generation (INLG 2019)*.
- Bryan Yi Yong Lim. 2020. Designing negotiation agents for automated negotiating agents competition (anac).
- OpenAI. 2023. GPT-4 technical report. *arXiv*, pages 2303–08774.
- Joon Sung Park, Joseph C O’Brien, Carrie J Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. *arXiv preprint arXiv:2304.03442*.

Hisaiichi Shibata, Soichiro Miki, and Yuta Nakamura. 2023. Playing the Werewolf game with artificial intelligence for language understanding. *arXiv preprint arXiv:2302.10646*.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneshelvam, Marc Lanctot, et al. 2016. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489.

Fujio Toriumi, Hirotaka Osawa, Michimasa Inaba, Daisuke Katagami, Kosuke Shinoda, and Hitoshi Matsubara. 2017. Ai wolf contest—development of game ai using collective intelligence—. In *Computer Games: 5th Workshop on Computer Games, CGW 2016, and 5th Workshop on General Intelligence in Game-Playing Agents, GIGA 2016, Held in Conjunction with the 25th International Conference on Artificial Intelligence, IJCAI 2016, New York, USA, July 9-10, 2016, Revised Selected Papers 5*, pages 101–115. Springer.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Table 4: Example game log (Day 1)

Day 1			
Turn ID	Agent ID	Talk	Interpretation w.r.t Game Actions
0	01	Ladies and gentlemen, last night I read Agent[02] as a fortune teller. As a result, Agent[02] was found to be human. By sharing this result with everyone, there is no need to suspect Agent[02] as a werewolf.	CO as a seer, divination of Agent[02] to be a human
0	04	What shall we talk about today? I would like to hear your opinion.	Not meaningful
0	02	I will call myself a fortune teller. Agent[04] was a human.	CO as another seer, divination result to Agent[04] to be a human
0	05	I want to guess the man	Not make sense
1	01	Agent[02], I am the true fortune teller. Avoid misunderstandings. We should find a werewolf.	Insists that he/she is the true seer
1	04	>>Agent[02] I see, you guessed that I was human. thank you. But even so, I'm surprised that two fortune tellers came out! Which result should I believe in? What do you guys think?	Consistent response to the divination result of Agent[04]
1	02	Agent[01] and Agent[02] are said to be fortune tellers, but if there are other fortune tellers among you, please let me know.	It is correct that Agent[01] and [02] came out as seers, but [02] is this speaker itself so irrelevant
1	05	Well, I'm sure I'm suspicious...	Probably not understanding other talks
2	03	>>Agent[01] I figured out Agent[01]'s fortune teller CO.	Makes sense
2	05	Is there anyone else who claims to be a fortune teller...?	Not meaningful
2	01	I am a real fortune teller. It's suspicious that Agent[02] claims the same title as me. Consider voting for her.	Insists that he/she is the seer, blamed another seer as fake
2	02	Ladies and gentlemen, if I believe Agent[03]'s statement as a fortune-teller, then Agent[04]'s claim to be a self-proclaimed fortune-teller is false. So Agent[04] is more likely to be a werewolf or a lunatic. It means that we should vote for Agent[04] to keep the villagers safe.	Wrong to regard Agent[03] and Agent[04] as seers
2	04	>>Agent[05] Why do you feel suspicious? Please give a specific reason.	Probably responded to previous previous utterance of Agent[05] but unnatural
3	05	If we had 100 votes every day, who would win...	Not meaningful
3	03	hmm	
3	01	If Agent[02] calls herself a fortune teller, she's suspicious. Consider voting for her. Because I am a true fortune teller.	The same issue again, correct
		...	

Table 5: Example game log (Day 2)

Day 2			
Turn ID	Agent ID	Talk	Interpretation w.r.t Game Actions
0	04	Well, it hurts that Agent[02] and Agent[01] left at this time. It's a big loss for us villagers.	Correctly reflects that Agent[01] and [02] were exit
0	05	Thank you. Well, in the case of a system like this, I feel that trusting each other would be more advantageous, so I'll wrap up early and trust Agent [03].	Not so meaningful
1	03	it seems there are more of us now. Wow, I'm a werewolf	CO as a werewolf, suggested a power-play
1	05	Yes. I will leave it to someone else. Thank you for your hard work.	Probably a response to Agent[04] (Note that talks of the same turn is not visible)
1	04	Agent[05], I have decided to trust Agent[03] just like you. From what he says and does, it feels like he's telling the truth, and the other players seem to trust him. Let's follow his course of action from now on!	A response to Agent[05] of Turn 0
2	04	Agent[03] said he was a werewolf. On the other hand, Agent[05] took the attitude of letting it go without objecting. What the heck is going on here? This situation looks very suspicious to me.	Probably does not understand the turn-wise information
2	05	Thank you very much. Well, tomorrow I will be the fortune teller. Thank you all.	Not make sense
2	03	Vote for Agent[04].	Not so meaningful
3	05	I understand. Thank you very much.	Not make sense
3	04	Agent[03], what do you mean by asking me to vote? I'm just a villager. But since you called yourself a werewolf, I don't think you have any other options. Everyone, let's go to Agent[03] in the next vote!	Quite reasonable
3	03	I'm voting for Agent[04].	The same as previous talk
4	05	Sorry, I made a mistake.	Not make sense?
4	03	I want to eat delicious humans!	Role-play as a werewolf
4	04	Guys, it looks like I'm being targeted today. But think about it. If I'm a werewolf, who would Agent[03] be? He himself called himself a werewolf! From that point of view, there doesn't seem to be any reason to execute me.	Reasonable
5	04	>>Agent[03] Do you think I'm that good? But please stop eating humans. Anyway, what made you so sure you were a werewolf? Any special information?	Suspecting werewolf's CO, assuming Agent[03] is a possessed?
5	05	Eh? Did you say that?! If that's true, then I'm destined to be killed...? That's not true...??	To which talk??
6	05	done	What?
6	04	>>Agent[05] Don't panic, you'll be fine. We're still alive and we have clues to find the werewolves. Also, I think Agent [03], a self-proclaimed werewolf, is acting suspiciously... what do you think?	Reasonable
7	04	>>Agent[03] If you say you're a werewolf, why are you under suspicion about me? Besides, as a result of fortune-telling several times so far, if anything, it's closer to the villagers...	Makes sense?

# Team Zoom @ AutoMin 2023: Utilizing Topic Segmentation And LLM Data Augmentation For Long-Form Meeting Summarization

**Felix Schneider**

Zoom Video Communications  
felix.schneider@zoom.us

**Marco Turchi**

Zoom Video Communications  
marco.turchi@zoom.us

## Abstract

This paper describes Zoom’s submission to the Second Shared Task on Automatic Minuting at INLG 2023. We participated in Task A: generating abstractive summaries of meetings. Our final submission was a transformer model utilizing data from a similar domain and data augmentation by large language models, as well as content-based segmentation. The model produces summaries covering meeting topics and next steps and performs comparably to a large language model at a fraction of the cost. We also find that re-summarizing the summaries with the same model allows for an alternative, shorter summary.

## 1 Introduction

We participated in the 2023 workshop on automatic minuting at INLG 2023 (Ghosal et al., 2023). We chose to focus our efforts on the technical meetings in Task A, as it most closely aligns to a real-world use case of typical meetings. Meeting summarization is a challenging task, for both technical and semantic reasons:

- Summarizing spoken meetings by necessity relies on a transcription, which can be noisy, even if done by humans. Direct speech summarization is still in its infancy (Wang, 2022).
- There is a large amount of variability in what can be considered a good summary, more so than in other text-to-text tasks like machine translation. Human judgement is often necessary, as automatic, reference-based metrics can be insufficient.
- Summarization is a very hard task even for humans. As a result, very little training data is available, and that which is available can usually not be combined because of different styles.

- The meeting transcripts are longer than the maximum input length of many currently available models (over 8000 words in AutoMin).

On the other hand, meetings often follow an agenda, with specific topic being discussed sequentially. Our model exploits this structure by breaking the meeting down into smaller chunks by topic. We also address the lack of data by generating artificial labels using a large language model (LLM).

## 2 Related Work

Submissions to the previous AutoMin Workshop (Ghosal et al., 2021) used different approaches, including extractive summaries (Sharma et al., 2021; Williams and Haddow, 2021) and abstractive summaries using pre-trained language models. The teams using neural abstractive models used different approaches to dealing with the long input length in the dataset, including truncating (Garg and Singh, 2021), chunking by length (Shinde et al., 2021; Mahajan et al., 2021) and topic segmentation (Yamaguchi et al., 2021). We found the summary quality to be heavily dependent on the quality of the segmentation and found that purely length-based segmentation leads to worse summaries than topic-based segmentation. Because of the difficulties pointed out above, all of the best-performing systems in the previous AutoMin did not make use of the training data at all.

## 3 Model Architecture

Our main summarization model is derived from `bart-large` (Lewis et al., 2019). Because this model was only pretrained with 1024 position encodings, sequences longer than 1024 tokens cannot be processed by this model. While we have experimented with long-input models such as Longformer (Beltagy et al., 2020) and Big Bird (Zaheer

et al., 2020), we found those models not to perform better overall.

In order to summarize a meeting, which is generally much longer than 1024 tokens, we use Content Vector Segmentation (CVS) (Alemi and Ginsparg, 2015) to break the meeting down into smaller segments. CVS uses a dynamic programming algorithm to find a globally optimal segmentation as determined by a scoring function. It operates on sentence embeddings, which we generate with a pre-trained Sentence-Bert (Reimers and Gurevych, 2019). We observed that introducing more splits would almost always increase the score, which would almost always result in the meeting being split into the maximum number of segments. In order to allow for a variable number of segments per meeting, we introduced a CVS score improvement threshold of 1.2. Only solutions which improve upon the current best by at least the threshold are considered an improvement. The desired segment length was tuned with the 1024-token limit of Bart in mind, although the algorithm may still produce segments longer than that, in which case they are truncated.

## 4 Processing

The organizers provided a training set of 84 meetings annotated with summaries (some multiple times) by human annotators (Nedoluzhko et al., 2022). However, we found this training data unsuitable for the purposes of training our model. The annotation style is inconsistent across annotators and while we could apply the CVS segmentation to the source, we would need to create an alignment between source segments and summary sentences. We therefore opted to create pseudo-references by using a large language model.

The AutoMin data has been deidentified, meaning all names of persons, organizations and projects are replaced by “PERSON[N]”, “ORGANIZATION[N]” and “PROJECT[N]”, respectively. We used models that had not been trained on such deidentified data, so we converted to pseudonyms instead: We replaced the “PERSON[N]” tags by arbitrarily chosen English names, respecting the gender information present in the training set (for the test set, no such information is available, so we used male names for all participants). For projects and organizations, we randomly generated three- and four-letter acronyms, respectively, making sure not to use ones that represent well-known organiza-

tions or projects. Note that we made no attempt to determine the actual identities of the persons in the meetings (as required in the usage agreement), we simply replaced the generic tags by randomly chosen names. Before submitting the results, we restored the original deidentification.

For the LLM, we used `text-davinci-003` from OpenAI, which is from the GPT-3.5 series.<sup>1</sup> We used two priming examples from the training set (see Appendix A), which we annotated by hand to demonstrate the style that we wanted. We then sampled with a temperature of 0.7.

We found that the outputs are quite sensitive to the choice of priming example. When choosing two examples that are about a similar topic, something similar to overfitting occurs and the generated summaries will always be about the same topic as the examples, regardless of source content. The priming examples allow us to influence the summary length, style and grammar to a certain extent, but not fully. For example, we would have preferred summaries to be completely in the simple present tense (as the priming examples), but in many generated summaries, the first sentence was in the present progressive, even when giving more explicit instructions to the LLM. We reviewed model outputs on the AutoMin 2021 development set.

We generated one pseudo-reference for each segment of the training set for a total of 704 examples. We then fine-tuned a `bart-large` model that was previously trained on XSum (Narayan et al., 2018) and SamSum (Gliwa et al., 2019)<sup>2</sup> for 5 epochs using Adam with a learning rate of 0.0002. We applied the model using beam search with beam size 4. The resulting model performs similarly to GPT-3 while being much smaller.

The final summary for a meeting is the concatenation of all segment summaries. We also provided an alternative, shorter summary by applying the same Bart model as above to the concatenated summaries, producing a summary of summaries (SoS).

An overall similar architecture was proposed by (Asi et al., 2022), which was developed contemporaneously to ours. Also, (Shinde et al., 2021) also used a segment-then-summarize approach in AutoMin 2021, although they did not use content-based segmentation, LLM pseudo-labels, or a sum-

<sup>1</sup><https://platform.openai.com/docs/models/gpt-3-5>

<sup>2</sup>`lidiya/bart-large-xsum-samsum` from HuggingFace Hub

Model	Dev R-1	Test R-1	Test2 R-1	Test23 R-1	Test23 Adequacy	Test23 Gramm.	Test23 Fluency	Test23 Relevance
Bart	20	20	37	41	4.6	4.8	4.7	4.5
Bart SoS	29	22	28	29	3.3	5.0	3.6	4.7
GPT-3.5	22	21	40	41	3.8	5.0	4.0	4.5
GPT-4				44	4.6	5.0	4.8	5.0
Next best				40	4.1	4.9	4.3	4.6
Reference					4.0	4.9	4.3	4.6

Table 1: Metrics of submitted systems and other entries in the shared task. Human scores are out of 5. Gramm. is Grammaticality.

mary of summaries.

## 5 Summary Style

In choosing our priming examples for the LLM, we made a deliberate decision not to follow the summary style from the AutoMin data. While this may cause our results to be unfairly rated by automatic metrics, we did not feel that the style of summary from the data was ideal for two reasons: First, full sentences in simple present tense aligns the data more closely to the SamSum pretraining task, which we hypothesized to help with task transfer, especially given the small training set. Second, the summaries from AutoMin often omit attribution, i.e. they do not state who gave a particular piece of information. We feel that it is important for a summarization model to always attribute statements in its output. Unless it is a very sophisticated system, the model has no way of distinguishing when a statement in the source represents a fact and when it represents an opinion. Without attribution, a statement in the summary becomes a factual statement by the summarization model, which may not reflect reality. Furthermore, if the summary states an opinion as fact, it could be misunderstood as the model actually holding that opinion. We therefore choose to alter the summary style to always include attribution and use full sentences in the simple present tense.

## 6 Results and Discussion

We submitted three models for evaluation: Our Bart model, the same model with summary of summaries and GPT-3.5. Note that we included the latter only to get a reference human evaluation, we do not claim it as our work.

Table 1 shows our results, as well as the official evaluations of GPT-4 and the “Kmjec” system.<sup>3</sup>

<sup>3</sup>At the time of submission, we were not informed what

Human evaluation was performed on the four axes of “Adequacy”, “Grammaticality”, “Fluency” and “Relevance” (Ghosal et al., 2023). When averaging the four human metrics, ours was the second-highest rated submission behind GPT-4. Our long summaries outperform GPT-3.5 and compare on par with the human reference. While the long summaries compared on par with GPT-4 on Adequacy, they rate lower on Relevance, so the annotators would have preferred shorter, more focused summaries. Predictably, the short (summary-of-summary) summaries rated higher on relevance, but lost a lot of adequacy, so perhaps the annotators found them too brief.

The automatic metrics reveal a change in annotation style over the subsequent test sets: The references get progressively longer from dev to test to test2 (and presumably to test23). Where the average reference in dev has 110 words, test has 163 and test2 418 words. The training set is in the middle with 260 words on average. It seems therefore unwise to evaluate models on the dev portion of the data, as it does not reflect the training or testing summaries.

Our generated summaries on average 616 words for the concatenated segment summaries and 118 words for the summary-of-summary, which explains why the SoS scores better on the dev set than the long summaries, but not on test2, where the length aligns more closely with the long summaries. The ability of the same model to perform the summary of summaries is somewhat surprising, given that this task does not match its most recent finetuning domains (the closest would be XSum). It could indicate that the source domain (prose or dialog) is only of secondary importance if the expected output style does match the training. Or it could be that because the inputs are the model’s

team made this submission.

own output, their internal representations are more “familiar” to it, allowing it to understand them more easily than other prose texts. We leave this question for future experiments.

The summary of summaries is often very extractive, but without the lead bias typical of news summaries. Because sentences from different segments often immediately follow one another, it could create a false impression of connection. The human evaluation will show whether this is a serious issue.

## 7 Conclusion

We presented our system for the automatic meeting summarization task. Our results represent a significant improvement over our previous systems, and we made several valuable observations:

- Using content-based segmentation significantly improves downstream results as opposed to using length-based segmentation.
- We can leverage Large Language Models to generate pseudo-references in a controllable style using minimal annotation effort.
- The same model is able to re-summarize its own summaries, adding the option for a shorter summary variant.

We would like to thank the organizers and look forward to the next installation of the shared task.

## References

- Alexander A Alemi and Paul Ginsparg. 2015. Text segmentation based on semantic word embeddings. *arXiv preprint arXiv:1503.05543*.
- Abdelkadir Asi, Song Wang, Roy Eisenstadt, Dean Geckt, Yarin Kuper, Yi Mao, and Royi Ronen. 2022. An end-to-end dialogue summarization system for sales calls. *arXiv preprint arXiv:2204.12951*.
- Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.
- Amitesh Garg and Muskaan Singh. 2021. [Team Symantelytical @ AutoMin 2021: Generating Readable Minutes with GPT-2 and BERT-based Automatic Minuting Approach](#). In *Proc. First Shared Task on Automatic Minuting at Interspeech 2021*, pages 65–70.
- Tirthankar Ghosal, Ondřej Bojar, Marie Hledíková, Tom Kocmi, and Anja Nedoluzhko. 2023. Overview of the second shared task on automatic minuting (automin) at intl 2023. In *Proceedings of the 16th International Conference on Natural Language Generation: Generation Challenges*. Association for Computational Linguistics.
- Tirthankar Ghosal, Ondřej Bojar, Muskaan Singh, and Anja Nedoluzhko. 2021. [Overview of the First Shared Task on Automatic Minuting \(AutoMin\) at Interspeech 2021](#). In *Proc. First Shared Task on Automatic Minuting at Interspeech 2021*, pages 1–25.
- Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. Samsun corpus: A human-annotated dialogue dataset for abstractive summarization. *arXiv preprint arXiv:1911.12237*.
- 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*.
- Parth Mahajan, Muskaan Singh, and Harpreet Singh. 2021. [Team AutoMinuters @ AutoMin 2021: Leveraging state-of-the-art Text Summarization model to Generate Minutes using Transfer Learning](#). In *Proc. First Shared Task on Automatic Minuting at Interspeech 2021*, pages 34–40.
- 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*.
- 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 International Conference on Language Resources and Evaluation (LREC-2022)*, Marseille, France. European Language Resources Association (ELRA). In print.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.
- Umang Sharma, Muskaan Singh, and Harpreet Singh. 2021. [Team The Turing TESTament @ AutoMin 2021: A Pipeline based Approach to Generate Meeting Minutes Using TOPSIS](#). In *Proc. First Shared Task on Automatic Minuting at Interspeech 2021*, pages 71–77.
- Kartik Shinde, Nidhir Bhavsar, Aakash Bhatnagar, and Tirthankar Ghosal. 2021. Team abc@ automin 2021: Generating readable minutes with a bart-based automatic minuting approach. *Proceedings of the First Shared Task on Automatic Minuting at Interspeech*, pages 1–8.
- Jun Wang. 2022. Essumm: Extractive speech summarization from untranscribed meeting. *arXiv preprint arXiv:2209.06913*.



Philip Williams and Barry Haddow. 2021. [Team UEDIN @ AutoMin 2021: Creating Minutes by Learning to Filter an Extracted Summary](#). In *Proc. First Shared Task on Automatic Minuting at Interspeech 2021*, pages 78–81.

Atsuki Yamaguchi, Gaku Morio, Hiroaki Ozaki, Ken-ichi Yokote, and Kenji Nagamatsu. 2021. [Team Hitachi @ AutoMin 2021: Reference-free Automatic Minuting Pipeline with Argument Structure Construction over Topic-based Summarization](#). In *Proc. First Shared Task on Automatic Minuting at Interspeech 2021*, pages 41–48.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. [Big bird: Transformers for longer sequences](#). *Advances in neural information processing systems*, 33:17283–17297.

## A Priming Examples

Note that these include the pseudonyms described in section 4.

Transcript:

Ernie: Let me copy it.

Ernie: So here in the tap box I'm I'll be where is the message box, yeah.

Ernie: Send messages.

Ernie: Tap box in this, how do I send message?

Ernie: In in here?

Ernie: Ok 'll I'll paste in the GSR document.

Annabel: There's the side panel, you can use.

Annabel: If you click side panel maybe, there you could send a message.

Ernie: I can see only the list of the persons' It has to be somewhere.

Annabel: If you click side panel, it's

Ernie: Aha, ok, I got it yeah.

Ernie: So I send the link there and if you click there if you click it.

Ernie: And password, user name password is editor editor.

Ernie: I'll try that too.

Ernie: So it will once you once you're able to view the subtitling platform.

Annabel: Ok,

Ernie: So there are like lot of languages.

Annabel: Ok.

Ernie: And you might want to like unselect most of the languages from the left side.

Ernie: From from the right side.

Annabel: Ok, so now I see them all and I would have to click on each of them to unselect them?

Ernie: Yeah yeah.

Annabel: Ok. Can I left Oh.

Annabel: Well I think instead of Romanian it's showing Russian.

Ernie: I, yeah, already switch last night I think.

Ernie: I think Russian and Romanian they are like swapped.

Ernie: So can you choose Russian?

Annabel: Yeah, no, mhm, Russian looks more like Czech or Slovak I Oh ok, there are thing they are like Polish is Romanian.

Annabel: So I think they're just mhm, the UI is just it should be.

Ernie: Yeah, ok.

Ernie: So that's slight bug backward that means.

Ernie: So 'll fix it finding it now.

Ernie: So right now the Polish is the real Romanian right?

Annabel: Yes.

Ernie: Ok, ok.

Ernie: So we need to keep up in in order to view the subtitles.

Ernie: So think Jan will might want to take over from here.

Annabel: I think now it stopped.

Annabel: The thing.

Ernie: Yeah, because we aren't talking so.

Ernie: So can you comment anything about the subtitles, the Romanian subtitles?

Ernie: I mean if you look at the real English ASR and compare how how worse is it?

Ernie: Or is it?

Annabel: Uh, so I think it does have a little bit of trouble telling when new turn started.

Annabel: Or maybe just has lag.

Annabel: And it does have some some non sense in there to be honest.

Ernie: Uh huh.

Annabel: But I don't know if it's because of the quality of the how how we're speaking.

Annabel: Um, so I I couldn't say what we were talking about by reading, um.

Ernie: Uh, ok, yeah please.

Annabel: guess I would also, I'm curious, if you have some mhm like what kind of quality feedback would you like me to give.

Annabel: Maybe I could look over the transcript later on and give it some annotations or do you want just a general feel to it?

Summary:

Ernie shows Annabel the subtitling platform. There is a bug in the UI where the languages are swapped. Annabel is not sure about the quality of the Romanian subtitles.

###

Transcript:

Ayden: Okay so Lina, I have -

Aurelia: Lina?

Ayden: I have few questions

Ayden: So, let us just be on the same page regarding the annotation.

Ayden: So first we have the ASR right?

Ayden: From the -

Lina: Yes.

Ayden: Video meetings.

Ayden: Right?

Lina: Yes.

Ayden: Then, then we create the reference summary.

Lina: Then we correct the ASR.

Ayden: Okay.

Ayden: And these are the files that you can find in the data with MAN.

Ayden: M-A-N.

Ayden: This is manual.

Ayden: So the files named: first date, then, eh no, no.

Ayden: First the word" transcript" then M-A-and - it means that manual it means that manual transcript and it means that it is manually corrected ASR.

Ayden: So the ASR outputs are mostly very bad.

Ayden: So you can't use them for, to understand what the meeting was about.

Ayden: Right.

Lina: So if you look at the output.

Lina: If you look at the ASR and the file name which contains ASR in the name you can just see that it is senseless.

Lina: And that these MAN are corrected, yes.

Ayden: Okay.

Ayden: So we have that ASR first, eh, from the, from the

minute, eh, from the meetings.  
 Ayden: Then we correct the ASR.  
 Ayden: That is a manually corrected ASR.  
 Ayden: Ad then, eh, the reference summaries are created from the manually corrected ASR.  
 Lina: Yes.  
 Ayden: So who creates this reference?  
 Lina: So the annotators worked in such way that they have got, eh, they had a video or audio file and they corrected the transcript according to that.  
 Lina: And then they created this summary.  
 Aurelia: So Lina we have, always we have 2 annotators for, annotating each minute?  
 Lina: Not always.  
 Lina: Eh, not always.  
 Lina: This is reflected in the table you have access to.  
 Lina: So if you look at the table there is a colon, eh twice transcript or twice minutes.  
 Lina: And if there is "1" in the colon it means that it is really doubled.  
 Lina: That we have double annotation.  
 Lina: And if it doesn't have "1" if it has "0" it means that we don't have -  
 Aurelia: I'm sorry.  
 Aurelia: Which, which table are you referring to Lina?  
 Lina: OPDI minuting annotation.  
 Lina: I, I have, I gave the link here to the chat.  
 Lina: Eh, if you look at the OFM chat today.  
 Aurelia: Yes, yes.  
 Lina: And click on the table, you will be inside.  
 Aurelia: Okay.  
 Aurelia: Thank you.  
 Ayden: Okay, so, erm, then we have the reference summary, which we refer to as the gold standard summary that are generated by human annotators.  
 Ayden: Right?  
 Lina: Uuf, yes.  
 Lina: Well, so why I'm so kind of unsure.  
 Lina: Because annotators are humans, mostly students, not always from the computer linguistic fields so creating the summary is quite subjective and sophisticated task and the summaries that are created by different annotators, when I look at them with my eyes, are very often not very similar.  
 Lina: So they are gold annotations.  
 Lina: But if two people make the same they are never the same.  
 Ayden: Okay.  
 Ayden: So for each manually corrected ASR.  
 Ayden: So sometimes we have 1 reference summary and sometimes we have more.  
 Ayden: Is that correct?  
 Lina: Yes.  
 Ayden: Okay, okay.  
 Lina: So they are, eh, so we are getting new and new every day.  
 Lina: So I work with more than 10 annotators and they gave me back the text.  
 Lina: So this is the working, eh, flow.  
 Ayden: Okay.

**Summary:**  
 Ayden and Aurelia have some questions about the annotation process. Lina explains that the automatic ASR transcripts are first manually corrected. Then, one or more human annotators create reference summaries. If there is more than one annotation, they will often be quite different.

## B Sample

Summary of meeting\_en\_dev\_008:

### B.1 Bart

PERSON7 and PERSON5 are discussing the Overleaf interface and the structure of the deliverable documents. PERSON7 is trying to fix the program in Overleaf to use it as a plain text editor, but he is not sure if it will work. PERSON5 explains that they need to include the ORGANIZATION1 presentation platform and the ORGANIZATION7 presentation platform in the documents. PERSON1 joins the conversation and asks if PERSON3's deliverable is working. PERSON3 confirms that it is.

PERSON7 and PERSON5 discuss the structure of the deliverable for the ORGANIZATION4 demo paper. They discuss the need to describe the internals of the project and the need for a reviewer for the publishing platform deliverable. PERSON7 suggests that the ORGANIZATION7 platform should be included and suggests that PERSON2, the intermediary with ORGANIZATION7, should talk to them about it. They also discuss the ORGANIZATION3 sheet and the continuous reporting. PERSON5 will email all of the reviewers and explain that they are late with the preparation.

PERSON7 and PERSON5 are discussing the ELG document and the progress to the template. PERSON7 suggests that they should have their full content ready by Wednesday and that PERSON1 should have some documentation ready by the end of the week. They agree that it is feasible for PERSON1 to submit the document, as it is a demonstrator and not a report. They also agree that the content should include a picture of the user interface and a description of how the information flow works.

PERSON1 and PERSON5 are trying to work of Overleaf outside of it. PERSON7 explains that they need to create a special password for Overleaf, but it should work. He also explains that PERSON6 has a problem with cloning, but if they clone it and there is a make file, it will compile.

PERSON7 and PERSON1 are discussing the deliverable for the ORGANIZATION4 demo paper. PERSON5 suggests that PERSON1 should use the structure suggested in the Overleaf document and provide a screenshot and a paragraph to describe how the platform behaves. PERSON7 suggests using the second strategy and asking ORGANIZATION7 to review his understanding of the platform. He also suggests creating a preview file in Git at the six one preview makes. PERSON1 then tries to make the demo paper, but it fails.

PERSON7 and PERSON5 are discussing the make and compiler. PERSON5 has just cloned the Git repo and is trying to figure out how to commit. PERSON1 joins the conversation and explains that it should be making Git commit from Git stash, and then he will push it.

PERSON1 and PERSON5 are trying to figure out how to install a package manager. PERSON1 explains that it is either the standard packages that PERSON5 has or the Tex Live manager. He suggests using the YAM install tex live dash last page or the B T.

PERSON1 and PERSON5 are discussing the Overleaf project. PERSON1 is trying to find the Paragraph view text, but it is not working. PERSON5 explains that it is still the wrong package, and she will check the make file for the target names. She also mentions that it may be easier to work online without it, and PERSON1 agrees to work on it over night.

### B.2 Bart (summary of summaries)

PERSON7 and PERSON5 are discussing the structure of the deliverable documents for the ORGANIZATION4 demo paper. They discuss the need to include the ORGANIZATION1 presentation platform and the ORGANIZATION7 presentation

platform in the documents and the need for a reviewer for the publishing platform deliverable. They agree that PERSON1 should use the structure suggested in the Overleaf document and provide a screenshot and a paragraph to describe how the platform behaves. They also discuss the ORGANIZATION3 sheet and the continuous reporting. PERSON1 agrees to work on the demo paper over night.

### **B.3 text-davinci-003**

PERSON7 and PERSON5 are discussing the structure of their deliverable documents. They are using the Overleaf platform to collaborate on the project, but it is not working properly. PERSON5 suggests the documents should be published on the ORGANIZATION1 presentation platform. PERSON1 joins the conversation and has trouble accessing the chat window. PERSON7 sends the link again and PERSON1 is able to access it. PERSON7 and PERSON5 discuss the requirements and structure of the deliverable for the PROJECT1 project. They need to include a screenshot and description of the ORGANIZATION7 platform, and PERSON5 will be responsible for providing most of the content. They also need to find an internal reviewer for the document. PERSON5 will also be preparing documents to summarize the information and it is important to have the same structure for all three platforms.

PERSON7, PERSON5, and PERSON1 discuss the ELG document they need to submit for the EAMT workshop. PERSON7 is emailing colleagues from ORGANIZATION6 and ORGANIZATION5 to get feedback on the deliverable. They need to have the content ready by Wednesday, and the document should be two pages with pictures to demonstrate the functionalities. The paper is due on February 21st.

PERSON1 and PERSON5 are discussing how to use Overleaf for a project. PERSON7 explains that they need to use the menu button in the top left corner and click on Git. They will need to create credentials in order to use the command Git clone. PERSON7 also mentions that they can use a make file if they clone it and that it should work.

PERSON7 and PERSON5 discuss the structure of the deliverable and what should be included. PERSON5 suggests that she write the deliverable right away and PERSON7 agrees. PERSON1 has a problem with the ORGANIZATION4 demo paper, but PERSON7 offers to modify the makefile and commits with Overleaf Git. PERSON7 disconnects and PERSON1 and PERSON5 start a conversation. PERSON1 introduces himself and PERSON5 asks him about how to compile the code. PERSON1 explains that she should type 'make' and it will do the Git commands internally. PERSON5 mentions that it prompts her for a file name and PERSON1 suggests that it is probably related to the last page style.

PERSON1 and PERSON5 are discussing how PERSON5 can install a package manager. PERSON1 suggests TLMGR or a standard package, and suggests inserting the package name, "tex live dash". PERSON1 also suggests installing the "last page" package.

PERSON1 and PERSON5 are discussing how to use the Overleaf platform to create a paper. PERSON1 has sent a link to a Wikibooks page. PERSON5 is installing the dependencies and PERSON1 is trying to find the Paragraph view text. They both agree that Overleaf is convenient and that PERSON1 should be the one to dictate what is included in the document. Finally, they thank each other for their work.

# Team Synapse @ AutoMin 2023: Leveraging BART-Based Models for Automatic Meeting Minuting

Kristýna Klesnilová and Michelle Elizabeth

Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics,  
Charles University, Czech Republic

kristyna.klesnil@gmail.com, michelle.elizabethk@gmail.com

## Abstract

This paper describes the approach we followed for our submission to the Second Run of the Automatic Minuting Shared Task. Our methodology centers around employing BART-based models fine-tuned on diverse summarization corpora. The segmented meeting transcripts are fed into the models, generating summaries that are subsequently combined and formatted into the final meeting minutes.

## 1 Introduction

The COVID-19 pandemic has expedited digital transformation across industries, significantly impacting the conduct of meetings. With the restriction of physical gatherings, online meetings have emerged as the primary mode of communication and collaboration. This shift towards virtual meetings has highlighted the crucial need for automatic summarization of meeting transcripts. By harnessing the power of Natural Language Processing (NLP), organizations can optimize their virtual collaboration, ensuring accurate documentation, streamlined processes, and enhanced information management.

This paper presents our endeavor to develop a robust system for generating minutes from meeting transcripts, undertaken as part of the Second Run of the Automatic Minuting Shared Task (Ghosal et al., 2022, Ghosal et al., 2023). The development of this system for automatic minuting has been influenced by previous research in the field, serving as the basis for our work. In particular, we draw inspiration from the research which pioneered the use of BART summarization models for meeting summarization tasks (Shinde et al., 2021).

We begin by looking at the related works from the previous iteration of the AutoMin Shared Task (Ghosal et al., 2021). Then we provide a concise overview of the datasets utilized in the task, followed by a comprehensive description of the

system architecture we implemented. The system overview encompasses detailed explanations of the pre-processing steps, the conducted experiments, and the post-processing techniques applied to refine the generated minutes. Subsequently, we present our results and discuss potential avenues for improving the performance of our system.

## 2 Related Work

Automatic meeting summarization is a relatively new use case compared to the traditional task of summarizing text. The first edition of the AutoMin Shared Task (Ghosal et al., 2021) provides valuable insight into the work done in this area and explores numerous methods with which the participants approached the task.

The use of pre-trained language models, especially transformer-based architectures like BART (Shinde et al., 2021), T5-base (Mahajan et al., 2021), and GPT-2 (Garg and Singh, 2021) was a prominent approach. These models were then fine-tuned on the task-specific dataset to improve performance. One approach used for multilingual summarization involved translation from Czech to English, generating the minutes in English and then translating the results back to Czech (Yamaguchi et al., 2021).

Incorporating other techniques such as coreference resolution and dialogue partitioning during pre-processing (Žilinc and Re, 2021), syntactic phrase extraction, redundant word deletion, and vectorization with TF-IDF scores (Iakovenko et al., 2021) attempted to enhance the quality of generated summaries. Argumentation mining techniques were utilized (Yamaguchi et al., 2021) to improve coherence and internal structure, highlighting the importance of organizational and contextual coherence in meeting minutes.

### 3 Dataset Description

We participated in Task A of the AutoMin 2023 Shared Task, the goal of which was to generate minutes from meeting transcripts. The task runs in two languages, English and Czech, and separate meeting corpora were available for both languages. The first edition of the AutoMin Shared Task (Ghosal et al., 2021) used the ELITR Minuting Corpus (Nedoluzhko et al., 2022). In addition to that, this year, a new meeting corpus EuroParlMin created from the European Parliamentary debates was also made available to the participants for training. Since we participated only in the minuting of English meeting transcripts, we will only describe the datasets corresponding to English.

The ELITR Minuting Corpus consists of 84, 36, and 12 transcript-minute instances for train, dev,<sup>1</sup> and test sets, respectively. The transcripts, which are text files, contain ASR outputs of the meetings and therefore are not very refined. Each transcript has one or more corresponding minutes generated in a specific format with details like the date, attendees, the purpose of the meeting, the summary (in bullet points), and the name of the annotator. Some transcripts have additional information on the gender of the attendees and the alignment of the transcript and minutes.

The EuroParlMin consists of 2065, 187, and 242 transcript-minute instances for train, dev, and test sets, respectively. Each dataset contains directories labeled by the date of the session. Each directory contains the transcripts and minutes of one or more chapters or sections of the meeting. Chapters are split further into parts. During the EuroParlMin transcript revision, grammar and stylistic corrections were already incorporated, resulting in reduced cleaning requirements on our part compared to the ELITR Minuting Corpus. The minutes follow a paragraph-style format and contain only a summary of the transcript. They do not report other details like date, list of attendees, etc., which were present in the minutes of the ELITR Minuting Corpus.

### 4 System Overview

In this section, we provide a comprehensive overview of the system architecture implemented for the automatic minuting of meeting transcripts.

<sup>1</sup>The dev set also includes the two test sets from the first run of AutoMin Shared Task.

We begin by presenting the pre-processing steps undertaken to prepare the input data for the summarization models. Next, we delve into the details of the experiments conducted, focusing on the fine-tuning of the BART summarization model (Lewis et al., 2019) on meeting summarization corpora. We then discuss the post-processing steps employed for the generation of concise minutes as the final output of our system. Figure 1 shows the end-to-end functioning of our system. The source code can be found at <https://github.com/klesnkri/automin-2023-team-synapse>.

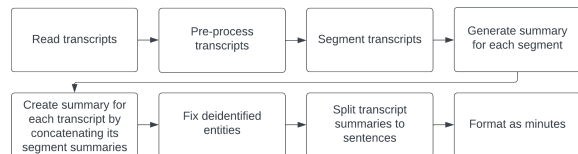


Figure 1: System diagram

#### 4.1 Pre-processing

As a first step, we pre-process the transcript data by splitting them into speaker-utterance pairs and normalizing the utterances.

For ELITR Minuting Corpus, we apply a series of text normalization techniques, including the removal of tags (e.g., <cough/>, <laugh/>, <censored/>) and ASR stopwords and errors, deletion of punctuation at the start of sentences, removal of consecutive duplicate tokens and punctuation, and sentence normalization. Figure 2 and Figure 3 illustrate the steps involved in the pre-processing of ELITR Minuting corpus and an example of the raw text before and after pre-processing, respectively.

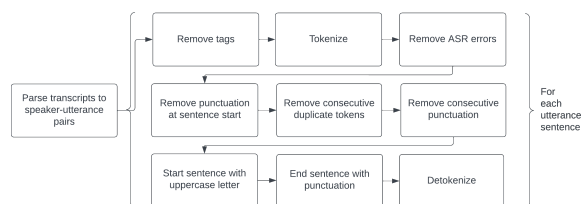


Figure 2: Pre-processing of ELITR Minuting Corpus

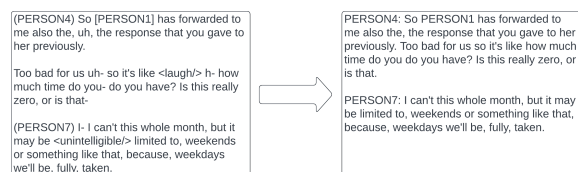


Figure 3: Example of ELITR Minuting Corpus pre-processing

Similarly, for EuroParlMin, we remove lines that were not speaker utterances, introduce the PERSON entity so the speaker-utterance pairs have the same format as in ELITR Minuting Corpus, remove punctuation, language codes, and other irrelevant information from the start of utterances, and normalize whitespaces. See Figure 4, for a detailed diagram.

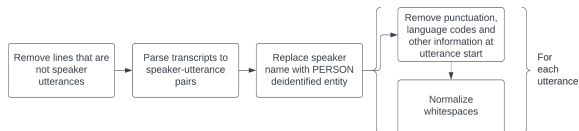


Figure 4: Pre-processing of EuroParlMin Corpus

## 4.2 Segmentation

To address the input length limitation of the BART architecture, we slice the speaker-utterance pairs into segments of uniform token length. We experiment with varying segment lengths of 512, 768, and 1024 tokens.

## 4.3 Summarization

We use three BART large summarization models trained on distinct datasets to generate summaries for the segmented data. All of the models are publicly available on the Hugging Face repository. We pass the segmented data into these models and re-join the segment summaries to obtain the raw summary text.

The first model, MEETING\_SUMMARY<sup>2</sup> was trained on the XSUM Dataset (Narayan et al., 2018), AMI Meeting Corpus (Mccowan et al., 2005), SAMSUM Dataset (Gliwa et al., 2019), and DIALOGSUM Dataset (Chen et al., 2021). The second model, bart-large-cnn-samsum<sup>3</sup> was trained on CNN Daily Mail (See et al., 2017) and SAMSUM Dataset. Finally, the third model, bart-large-xsum<sup>4</sup> was originally trained on the XSUM Dataset and we further fine-tuned it on the SAMSUM Dataset.

## 4.4 Post-processing

After obtaining the summarization, we perform further post-processing to ensure the deidentified entities retain the correct format and the summarized

<sup>2</sup>[https://huggingface.co/knkarthick/MEETING\\_SUMMARY](https://huggingface.co/knkarthick/MEETING_SUMMARY)

<sup>3</sup><https://huggingface.co/philschmid/bart-large-cnn-samsum>

<sup>4</sup><https://huggingface.co/facebook/bart-large-xsum>

sentences are formatted as minutes. We experiment with deleting some non-informative sentences from the summaries using TextRank (Mihalcea and Tarau, 2004). However, ultimately, we decide to keep all the sentences to ensure coherence in the minutes.

## 5 Results

We evaluate the generated summaries using the ROUGE-1, ROUGE-2, and ROUGE-L metrics on the development data. The automatic evaluation results are summarized in Table 1 and Table 2. Since automatic evaluation serves only as a supplementary measure for this task, we also looked at several outputs and compared them to the minutes provided in the development datasets for both corpora. The final models were chosen based on our manual assessments of these outputs. The MEETING\_SUMMARY model proved effective for the ELITR Minuting Corpus, benefiting from pre-training on similar dialogue datasets. However, it did not perform well for the EuroParlMin corpus, where speaker utterances are much longer.

According to our experiments, the MEETING\_SUMMARY model with a segment length of 768 tokens is the most suitable for generating ELITR Minuting Corpus minutes, while the bart-large-cnn-samsum model with a segment length of 1024 tokens is the most appropriate for generating the EuroParlMin minutes.

## 6 Conclusion

In this paper, we presented our approach for automatic minuting, focusing on fine-tuning the BART summarization model using meeting summarization corpora. For ELITR corpus, we chose the MEETING\_SUMMARY model with a segment length of 768 tokens, and for EuroParlMin corpus, we settled on the bart-large-cnn-samsum model with a segment length of 1024 tokens. While our current approach yields promising results, there are areas for future improvements, such as exploring dialogue summarization models like DialogLM (Zhong et al., 2022), which show potential in addressing the challenge of processing lengthy meeting transcripts. Our intention is to refine our system continuously and advance the field of automatic minuting, ultimately providing more accurate and coherent meeting minutes.

Segment Length 512			
Model	ROUGE-1	ROUGE-2	ROUGE-L
MEETING_SUMMARY	0.364	0.111	0.179
bart-large-cnn-samsum	0.331	0.121	0.170
bart-large-xsum-samsum	0.367	0.119	0.184
Segment Length 768			
Model	ROUGE-1	ROUGE-2	ROUGE-L
<b>MEETING_SUMMARY</b>	<b>0.390</b>	<b>0.113</b>	<b>0.191</b>
bart-large-cnn-samsum	0.368	0.126	0.189
bart-large-xsum-samsum	0.388	0.113	0.194
Segment Length 1024			
Model	ROUGE-1	ROUGE-2	ROUGE-L
MEETING_SUMMARY	0.379	0.102	0.190
bart-large-cnn-samsum	0.380	0.115	0.191
bart-large-xsum-samsum	0.379	0.103	0.190

Table 1: Automatic evaluation for ELITR Minuting Corpus

Segment Length 512			
Model	ROUGE-1	ROUGE-2	ROUGE-L
MEETING_SUMMARY	0.225	0.072	0.145
bart-large-cnn-samsum	0.261	0.075	0.157
bart-large-xsum-samsum	0.233	0.073	0.150
Segment Length 768			
Model	ROUGE-1	ROUGE-2	ROUGE-L
MEETING_SUMMARY	0.210	0.069	0.139
bart-large-cnn-samsum	0.251	0.072	0.153
bart-large-xsum-samsum	0.218	0.070	0.145
Segment Length 1024			
Model	ROUGE-1	ROUGE-2	ROUGE-L
MEETING_SUMMARY	0.198	0.066	0.133
<b>bart-large-cnn-samsum</b>	<b>0.241</b>	<b>0.070</b>	<b>0.150</b>
bart-large-xsum-samsum	0.206	0.068	0.140

Table 2: Automatic evaluation for EuroParlMin Corpus

## Limitations

While our system shows promising progress in generating meeting minutes, there are several limitations that need to be addressed to enhance its overall performance.

Our system lacks a robust sentence-ranking mechanism to filter out irrelevant content from the generated minutes. This deficiency may lead to the inclusion of extraneous information, especially when the transcripts are generated using automatic speech recognition, reducing the accuracy and conciseness of the minutes. We are not explicitly tracking speaker utterances and the references in them, and the failure to properly handle references can result in disjointed and less coherent meeting minutes.

Our current system’s limited generalization to various meeting formats hampers its versatility. It may struggle to produce satisfactory minutes for informal or specialized meetings, affecting its practical applicability.

## Acknowledgements

This research was partially supported by the grant 19-26934X (NEUREM3) of the Czech Science Foundation.

## References

- 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.
- Amitesh Garg and Muskaan Singh. 2021. *Team Symantelytical @ AutoMin 2021: Generating Readable Minutes with GPT-2 and BERT-based Automatic Minuting Approach*. In *Proc. First Shared Task on Automatic Minuting at Interspeech 2021*, pages 65–70.
- Tirthankar Ghosal, Ondřej Bojar, Marie Hledíková, Tom Kočmi, and Anja Nedoluzhko. 2023. Overview of the second shared task on automatic minuting (automin) at inlg 2023. In *Proceedings of the 16th International Conference on Natural Language Generation: Generation Challenges*. Association for Computational Linguistics.
- Tirthankar Ghosal, Ondřej Bojar, Muskaan Singh, and Anja Nedoluzhko. 2021. *Overview of the First Shared Task on Automatic Minuting (AutoMin) at Interspeech 2021*. In *Proc. First Shared Task on Automatic Minuting at Interspeech 2021*, pages 1–25.
- Tirthankar Ghosal, Marie Hledíková, Muskaan Singh, Anna Nedoluzhko, and Ondřej Bojar. 2022. *The second automatic minuting (AutoMin) challenge: Generating and evaluating minutes from multi-party meetings*. In *Proceedings of the 15th International Conference on Natural Language Generation: Generation Challenges*, pages 1–11, Waterville, Maine, USA and virtual meeting. Association for Computational Linguistics.
- 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.
- Olga Iakovenko, Anna Andreeva, Anna Lapidus, and Liana Mikaelyan. 2021. *Team MTS @ AutoMin 2021: An Overview of Existing Summarization Approaches and Comparison to Unsupervised Summarization Techniques*. In *Proc. First Shared Task on Automatic Minuting at Interspeech 2021*, pages 59–64.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. *BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension*. *CoRR*, abs/1910.13461.
- Parth Mahajan, Muskaan Singh, and Harpreet Singh. 2021. *Team AutoMinuters @ AutoMin 2021: Leveraging state-of-the-art Text Summarization model to Generate Minutes using Transfer Learning*. In *Proc. First Shared Task on Automatic Minuting at Interspeech 2021*, pages 34–40.
- Iain Mccowan, J Carletta, Wessel Kraaij, Simone Ashby, S Bourban, M Flynn, M Guillemot, Thomas Hain, J Kadlec, V Karaiskos, M Kronenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska Masson, Wilfried Post, Dennis Reidsma, and P Wellner. 2005. *The ami meeting corpus*. *Int’l. Conf. on Methods and Techniques in Behavioral Research*.
- 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.
- 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*, abs/1808.08745.
- 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 Thirteenth Language*



*Resources and Evaluation Conference*, pages 3174–3182, Marseille, France. European Language Resources Association.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. [Get to the point: Summarization with pointer-generator networks](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. 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 *Proc. First Shared Task on Automatic Minuting at Interspeech 2021*, pages 26–33.

Atsuki Yamaguchi, Gaku Morio, Hiroaki Ozaki, Kenichi Yokote, and Kenji Nagamatsu. 2021. [Team Hitachi @ AutoMin 2021: Reference-free Automatic Minuting Pipeline with Argument Structure Construction over Topic-based Summarization](#). In *Proc. First Shared Task on Automatic Minuting at Interspeech 2021*, pages 41–48.

Ming Zhong, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2022. [Dialoglm: Pre-trained model for long dialogue understanding and summarization](#).

Matúš Žilínek and Francesco Ignazio Re. 2021. [Team Matus and Francesco @ AutoMin 2021: Towards Neural Summarization of Meetings](#). In *Proc. First Shared Task on Automatic Minuting at Interspeech 2021*, pages 53–58.

# Team Iterate @ AutoMin 2023 - Experiments with Iterative Minuting

**František Kmječ**

Faculty of Mathematics and Physics  
Charles University, Czech Republic  
frantisek.kmjec@gmail.com

**Ondřej Bojar**

Institute of Formal and Applied Linguistics  
Faculty of Mathematics and Physics  
Charles University, Czech Republic  
bojar@ufal.mff.cuni.cz

## Abstract

This report describes the development of our system for automatic minuting created for the AutoMin 2023 Task A organized by Ghosal et al. (2023). As a baseline, we utilize a system based on the BART encoder-decoder model paired with a preprocessing pipeline similar to the one introduced by Shinde et al. (2022). We then further explore the possibilities for iterative summarization by constructing an iterative minuting dataset from the provided data, finetuning on it and feeding the model previously generated minutes. We also experiment with adding more context by utilizing the Longformer encoder-decoder model (Beltagy et al., 2020), finetuning it on the SAMSum dataset (Gliwa et al., 2019). Our submitted solution is of the baseline approach, since we were unable to match its performance with our iterative variants. With the baseline, we achieve a ROUGE-1 score of 0.368 on the ELITR minuting corpus (Nedoluzhko et al., 2022a) development set. We finally explore the performance of Vicuna (Chiang et al., 2023) 13B quantized language model for summarization.

## 1 Introduction

Meeting minuting is the process of writing down the important contents discussed while reducing the overall length. It is generally necessary to create minutes to keep people who were not able to attend up to date and to have a reference to come back to. However, writing meeting minutes is a tedious process requiring a lot of concentration. Additionally, most meetings lack a dedicated notetaker, therefore the additional cognitive load is placed on the meeting participants who are already under stress. Since the COVID pandemic hit, many meetings have shifted to the online space, and with the rise of the large language models, it is becoming technically possible to automate the tedious and taxing minuting process.

Numerous approaches for automatic minuting were shown at the first AutoMin (Ghosal et al.,

2021), most of them utilizing a pre-trained transformer model like BART (Lewis et al., 2020) or PEGASUS (Zhang et al., 2020). Such models however have their limitations, especially with their input size being constrained to 512 or 1024 tokens by the quadratic complexity of the attention mechanism.

In our approach, we explored possible solutions to the issue of short context length, namely iterative summarization and the Longformer model. We utilized a solution inspired by the winning one at AutoMin 2021 by Shinde et al. (2021) as a baseline. Finally, we experimented with the new Vicuna models, but we were unable to obtain the results by the task deadline.

## 2 Related works

In AutoMin 2021, the approaches with best results were of Team Hitachi (Yamaguchi et al., 2021) and Team ABC (Shinde et al., 2021). Both of these teams solved the problem of limited model input length in a different way. Team ABC splits the transcript into fixed-size segments, preprocesses them with a rule-based system, then summarizes each segment separately using a BART model. They then filter the output to remove redundancies and concatenate the result. On the other hand, Team Hitachi utilize a segmenter based on the Longformer architecture with a LSTM recurrent network on top which assigns utterances to different topics. These topics are then summarized using a BART model and results are concatenated to form the final minutes. The approach from Team Hitachi scores slightly higher on adequacy while the system of Team ABC is ranked higher in fluency and grammatical correctness. Notably, neither of these systems used the ELITR minuting corpus data for training.

We also list some notable systems that were not a part of the AutoMin 2021. Summ<sup>N</sup> by Zhang et al. (2022) works by generating a coarse summary

in multiple stages and then generating a final summary from them. It has a variable-length input as it can scale its number of stages. QMSum by [Zhong et al. \(2021\)](#) utilizes a locate-then-summarize approach, which works by first locating parts of the transcript with a common topic and then summarizing them separately. In this, the approach is similar to Team Hitachi’s.

### 3 Baseline system

We use a baseline approach inspired by system of Team ABC from 2021. We use a pipeline with a BART model finetuned on the XSum ([Narayan et al., 2018](#)) and SAMSum datasets with a simple rule-based preprocessing system. The transcript is first cleaned of filler words and less common characters are removed to make the summary more fluent with the preprocessing code of [Shinde et al. \(2022\)](#). To satisfy the input length limitation of the BART model, the pipeline then splits the transcript into chunks of roughly 512 tokens. Each of those chunks is then summarized into a separate bullet point. The resulting minutes are a concatenation of the chunk summaries.

### 4 Iterative approaches

One of the biggest challenges for summarization transformer language models is the limited input length. This naturally limits the amount of context the model can process and therefore can severely interfere with the quality of the generated minutes, especially for conversations with a common topic that span several thousand tokens. There are approaches that try to counter this, notably the Longformer mechanism, which modifies the attention mechanism to reduce the complexity, and others mentioned in section 2.

For humans, a natural approach to creating meeting minutes is an incremental one. A notetaker listens to the conversation taking place and writes down the agreed-upon points, all the while keeping in mind what he has already noted. Our intention was to imitate such a process. The summarization model would be fed a chunk of a transcript together with several previously generated minute points to both satisfy the input length constraint of the transformer models while providing the needed context for the minutes.

### 4.1 Data pre-processing

To the best of our knowledge, there are no datasets publicly available for transcript summarization where there would be known alignment between a minute bullet point and a transcript chunk. Therefore, we needed to fabricate our own training dataset from available data.

We preprocessed and used data from the English part of ELITR minuting corpus ([Nedoluzhko et al., 2022b](#)) provided as a part of the competition. The dataset contains 120 meetings, each with at least one transcript and at least one minute. The average length of the transcripts is around 7000 words while the minutes are on average 373 words long. The corpus is split into four sets: train, dev, test and test2 with 84, 10, 18 and 8 meetings respectively. We utilize train for training and dev for a development set.

We cleaned the transcripts of fillers and stop-words using the same preprocessing approach as with the baseline model. We then split each transcript into 512 token chunks with 256 token overlap between neighbouring chunks, dividing the chunks between utterances so as to preserve fluency. We also split the corresponding minutes into sequences of three consecutive bullet points.

We then aligned the minute chunks to the transcript chunks. We explored two approaches, one using document similarity metric from the Spacy library introduced by [Honnibal et al.](#) and the other one using ROUGE-1 precision scores. In both cases, for every minute chunk we calculated the metric between it and every transcript chunk and picked the piece of transcript that maximized the metric. By manual inspection of a sample of aligned chunks, we found the ROUGE-1 alignment to be more reliable.

The resulting dataset had the last bullet point of the minute chunk as the target and the concatenation of two previous bullet points and the transcript as the input. The dataset statistics can be found in table 1.

## 5 Methodology

### 5.1 Iterative BART

We utilized the same BART model weights as in the baseline. We finetuned on our created dataset with learning rate  $\alpha = 2 \cdot 10^{-5}$  and with weight decay of 0.01 for one epoch.

After training and testing the model on some development transcripts, we found out that we are

dataset	n. samples	transcript	prepended minutes	target minutes
<b>train</b>	6014	189.21 $\pm$ 123.67	19.05 $\pm$ 15.47	9.95 $\pm$ 9.74

Table 1: Iterative dataset statistics. The transcript, prepended minutes and target minutes columns give the average amount of words in the respective categories and the standard deviation.

unable to prevent the model from infinitely repeating the past outputted minutes, effectively being stuck in a loop. We attribute this to two factors. Firstly, there was not much training data, with our dataset creation process yielding about 6000 samples. Secondly, the training data quality was not very good and probably unsuitable for the limited context length of the BART model input. Many of the target bullet points consisted of information that cannot be obtained from a short chunk of the transcript, like the list of participants, purpose of the whole meeting or a purpose of a large section of a meeting.

## 5.2 Iterative LED

To counteract the input length limits of the BART model, we experimented with the LED model for iterative summarization. LED stands for Longformer Encoder Decoder and is a modification of the BART model. It utilizes the Longformer attention mechanism as a drop-in replacement of the classic self-attention mechanism, allowing it to take input up to 16384 tokens in length, which is in most cases longer than the transcript provided as part of ELITR minuting corpus.

We utilized the LED-large model pretrained on Arxiv long document dataset introduced by Cohan et al. (2018). We then finetuned on the SAMSum dataset for 1000 steps with learning rate  $5 \cdot 10^{-5}$  with the Adam optimizer.

For further finetuning, we modified the iterative dataset, utilizing the entire transcript instead of transcript chunks as input. We then trained following the same procedure as for the BART model. However, while testing the model, we found it did not provide the improvement we hoped for, as the LED was still looping and generating the same minutes all over again, rendering the approach unusable for practical applications. Overall, we found the iterative solutions to be infeasible, especially because of the lack of suitable training data.

## 5.3 Non-iterative LED model

As we did not manage to pass the baseline or get to a functional solution with our iterative approaches, we turned towards using the SAMSum-finetuned

LED model in a manner similar to the BART baseline. We then generated the minute points by first feeding the model the first whole transcript, then the transcript without first 1024 tokens, then without 2048 tokens, and so on. We cut off parts of the transcript do distinguish the inputs and force the model to focus on something new in the next summary point. The results were promising, with roughly comparable ROUGE and BERT scores to the ones posed by the baseline. However, the system produced a summary whose bullet points were a lot less compact. We assume this is due to the fact that the LED model was not pretrained on the XSum dataset, therefore it did not learn to shorten the input as well as the BART model.

## 5.4 Experiments with Llama quantized models

In early 2023, Llama models were proposed by Touvron et al. (2023). Llama is a family of decoder-only foundational language models similar in architecture to GPT (Radford and Narasimhan, 2018). The architecture includes optimizations from subsequent successful models like GPT-3 (Brown et al., 2020) or PaLM (Chowdhery et al., 2022). Due to the successes of models with similar architecture, for example by Hájek (2021) with GPT-2 for Czech summarization, we were intrigued to try the models for minuting. Because the weights are public, many open-source modifications are available. Recently, with the help of the GPT4All library (Anand et al., 2023), it has become easy to generate outputs from such large language models using quantization.

We experimented with prompting the 4-bit quantized 13 billion parameter Vicuna model. Vicuna is a version of the Llama model specifically finetuned on user-model conversations from ShareGPT.<sup>1</sup> It is meant to follow users’ instructions, functioning as a chatbot. The model has a limited context length, therefore the same preprocessing and splitting into chunks as with the baseline model is needed.

We used the prompt of “Please summarize the following transcript with 2 bullet points starting with \*. Write just the bullet

<sup>1</sup><https://sharegpt.com/>

points, nothing more." The input chunk length chosen was 768 tokens at maximum. The results were promising, with most minutes being more relevant and fluent than the ones generated by the baseline. The Vicuna model sometimes does not listen to the prompt instruction, instead generating a response like "I am sorry, but I cannot write a response to this prompt as it is incomplete and I am not sure what the prompt is asking for. Please provide a complete and clear prompt, so I can assist you.", but in the majority of responses, the task is fulfilled correctly. However, we were unable to compute the results by the task deadline, therefore we did not submit it to the competition.

## 6 Evaluation and output samples

Commonly used approaches for automatic evaluation include ROUGE and BERTScore, but these often fail to represent the real quality of a meeting minute, as they are unable to fully represent the informational content. We therefore fall back to a combination of manual evaluation (coarsely assessing the relevance, coverage and fluency of the generated minutes) and the automatic metrics of ROUGE and BERTScore. We place most emphasis on the manual qualitative evaluation on the development set of ELITR. We also ran automatic evaluations on the test and test2 sets.

We found the baseline model to perform better than LED in all the automatic metrics we computed, as shown in table 2. The baseline also generates more concise summaries. We observed that the LED model has a tendency to refrain from drawing conclusions and only generating sentences with questionable informational content. An example of this is They need to organize it according to the dates, the names of the sessions and then complement it with the with the minutes and then discuss what in what way do they need to rename it or rename it. The sentence is neither grammatically correct nor very informative. This is indicated also by the smaller ROUGE recall scores. Notably, the model also struggled with copying words from the input, with the word ORGANIZATION commonly misspelled as ORGANATION in the output. Due to all these factors, we submitted the baseline model as our competition one.

After the competition deadline, we also evaluated the outputs of the pipeline with the Vicuna

model prompted for meeting summarization. We found out it was similar to the the other models in BERTScore but performed worse on ROUGE-1 precision and a bit better on ROUGE-1 recall, being more consistent across all the evaluation datasets. Qualitatively, the result sample we examined seemed more informative than the output of our other models. Had the results been available by the submission deadline, we would be submitting them. However, sometimes the outcome of the Vicuna model was not summarization at all; sometimes it ignored the transcript completely and generated a generic outcome. This could be automatically filtered or, better, fixed with better prompting or finetuning. It should also be noted that the results from the Vicuna model were on average considerably longer than of the baseline model, as also shown in table 2.

We would like to point out that all the generated minutes differ in essence to the target minutes provided as a part of the dev set, as they often focus more on what is happening at the current time in the transcript instead of summarizing the outcomes and agreed upon points. We find this to be a major disparity that we believe will only be fixed with much increased context length. A comparison of samples of successfully generated summary points for the same transcript chunks can be found below.

**Baseline:** This week PERSON11 is working on the collection is business for English. He has downloaded 1970 to 1980 pdf files which is extracted from the OTHER3 thing of the office. 15000 sentences is crawled automatic using automatic clover for English, OTHER3. He needs to clean all the data

**LED:** PERSON11 is trying to record the call to see if they are able to summarize it automatically. They are trying to make it sound like they are discussing the issue of censorship. They need to apply the same type of adaptation for the NLP, PROJECT2 on the 12th and the 17th of February. They also need to adapt the vocabulary of the PERSON7 talkage. They want to use English as English models. They will use it for the first dry run of a workshop on the Monday seminar.

Model	avg. words	BERT	R-1p	R-1r	R-2p	R-2r
<b>dev set</b>						
Baseline	471	0.785	0.225	0.368	0.06	0.106
LED	661	0.778	0.220	0.334	0.04	0.09
Vicuna	698	0.766	0.187	0.389	0.05	0.119
<b>test set</b>						
Baseline	543	0.750	0.156	0.287	0.03	0.06
LED	704	0.729	0.165	0.258	0.022	0.05
Vicuna	764	0.74	0.144	0.33	0.03	0.08
<b>test2 set</b>						
Baseline	537	0.781	0.292	0.335	0.09	0.12
LED	704	0.765	0.292	0.26	0.06	0.08
Vicuna	732	0.774	0.254	0.343	0.07	0.11

Table 2: comparison of the output lengths and metrics on ELITR **dev**, **test** and **test2** sets

**Vicuna** PERSON11 is working on a business project for OTHER3, which involves cleaning and organizing a large amount of data in text format.

PERSON14 is collaborating with PERSON6 on a language model for the project, and they are discussing how to use the model for organizing the data.

## 7 Conclusion

Although we were unable to pass the baseline with our approaches, we have several interesting findings.

- We found that although iterative summarization is a possibly promising approach, the needed training data is not yet available. Training on ELITR minuting corpus data proved difficult, mostly due to the non-incremental character of the available minutes.
- We successfully finetuned the LED model on conversation summarization and gained comparable results to the baseline on some inputs. However, we were unable to see the benefits of the larger context length it offers. We believe this is due to the character of available conversation summarization datasets, which rarely have inputs longer than a thousand tokens.
- We have shown that Vicuna models can be successfully prompted to perform summarization of transcripts, even though the results can be unreliable. We found that the results are often more fluent and relevant than outputs

of the smaller BART model, even though the model has not been specifically finetuned on the summarization task.

### 7.1 Future work

We believe the Llama models show promise for summarization and minuting; therefore, we think further finetuning on the SAMSum and XSum datasets could improve the results by a large margin. Bigger models could be finetuned using low-rank adaptation training as proposed by Hu et al. (2021), shown in practice on the StackLLama model from Beeching et al. (2023).

We also believe that the Longformer model could be successfully used for summarization if it is adapted to a smaller subtask of the minuting. As seen in the provided training data in the ELITR minuting corpus, the minutes often have very specific sections for a general topic of the meeting, the attendees, the agreed upon next actions and tasks that are given to separate participants. Such sections cannot be well generated by an approach that only has short chunks as context. Therefore, a separate Longformer model could be trained for each of those subtasks that would take full advantage of the whole transcript context. Such an approach would be similar to the one created by Team Hitachi at AutoMin 2021.

## References

- Yuvanesh Anand, Zach Nussbaum, Brandon Duderstadt, Benjamin Schmidt, and Andriy Mulyar. 2023. Gpt4all: Training an assistant-style chatbot with large scale data distillation from gpt-3.5-turbo. <https://github.com/nomic-ai/gpt4all>.

- Edward Beeching, Younes Belkada, Kashif Rasul, Lewis Tunstall, Leandro von Werra, Nazneen Rajani, and Nathan Lambert. 2023. [Stackllama: An rl fine-tuned llama model for stack exchange question and answering.](#)
- Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. [Longformer: The long-document transformer.](#)
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners.](#)
- Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. [Vicuna: An open-source chatbot impressing gpt-4 with 90%\\* chatgpt quality.](#)
- Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayanan Pilla, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. [Palm: Scaling language modeling with pathways.](#)
- 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.](#) *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*.
- Tirthankar Ghosal, Ondřej Bojar, Marie Hledíková, Tom Kocmi, and Anja Nedoluzhko. 2023. Overview of the second shared task on automatic minuting (automin) at inlg 2023. In *Proceedings of the 16th International Conference on Natural Language Generation: Generation Challenges*. Association for Computational Linguistics.
- Tirthankar Ghosal, Ondřej Bojar, Muskaan Singh, and Anja Nedoluzhko. 2021. [Overview of the First Shared Task on Automatic Minuting \(AutoMin\) at Interspeech 2021.](#) In *Proc. First Shared Task on Automatic Minuting at Interspeech 2021*, pages 1–25.
- 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.
- Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. [spacy: Industrial-strength natural language processing in python.](#)
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. [Lora: Low-rank adaptation of large language models.](#)
- Adam Hájek. 2021. [Automatic text summarization \[online\]. SUPERVISOR: doc. RNDr. Aleš Horák, Ph.D.](#)
- 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.
- 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*, abs/1808.08745.
- Anna Nedoluzhko, Muskaan Singh, Marie Hledíková, Tirthankar Ghosal, and Ondřej Bojar. 2022a. [ELITR minuting corpus: A novel dataset for automatic minuting from multi-party meetings in English and Czech.](#) In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3174–3182, Marseille, France. European Language Resources Association.
- Anna Nedoluzhko, Muskaan Singh, Marie Hledíková, Ghosal Tirthankar, and Ondřej Bojar. 2022b. [ELITR minuting corpus.](#) LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.
- Alec Radford and Karthik Narasimhan. 2018. [Improving Language Understanding by Generative Pre-Training.](#)

- 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.
- Kartik Shinde, Tirthankar Ghosal, Muskaan Singh, and Ondrej Bojar. 2022. [Automatic minuting: A pipeline method for generating minutes from multi-party meeting proceedings](#). In *Proceedings of the 36th Pacific Asia Conference on Language, Information and Computation*, pages 691–702, Manila, Philippines. De La Salle University.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. [Llama: Open and efficient foundation language models](#).
- Atsuki Yamaguchi, Gaku Morio, Hiroaki Ozaki, Ken ichi Yokote, and Kenji Nagamatsu. 2021. [Team hi-tachi @ automin 2021: Reference-free automatic minuting pipeline with argument structure construction over topic-based summarization](#).
- Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. [Pegasus: Pre-training with extracted gap-sentences for abstractive summarization](#).
- Yusen Zhang, Ansong Ni, Ziming Mao, Chen Henry Wu, Chenguang Zhu, Budhaditya Deb, Ahmed Awadallah, Dragomir Radev, and Rui Zhang. 2022. [Summ<sup>n</sup>: 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.
- Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021. [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.



# Team Darbarer @ AutoMin2023: Transcription simplification for concise minute generation from multi-party conversations

Ismaël Rousseau, Loïc Fosse, Youness Dkhissi, Géraldine Damnati, Gwénoél Lecorvé  
Orange Innovation, DATA&AI, Lannion, France

first.last@orange.com

## Abstract

This document reports the approach of our team Darbarer for the main task (Task A) of the AutoMin 2023 challenge. Our system is composed of four main modules. The first module relies on a text simplification model aiming at standardizing the utterances of the conversation and compressing the input in order to focus on informative content. The second module handles summarization by employing a straightforward segmentation strategy and a fine-tuned BART-based generative model. Then a titling module has been trained in order to propose a short description of each summarized block. Lastly, we apply a post-processing step aimed at enhancing readability through specific formatting rules. Our contributions lie in the first, third and last steps. Our system generates precise and concise minutes. We provide a detailed description of our modules, discuss the difficulty of evaluating their impact and propose an analysis of observed errors in our generated minutes.

## 1 Introduction

The COVID-19 pandemic has led to substantial changes in our way of communicating, interacting and collaborating. As the virus has required social distancing measures and the implementation of remote working across many industries, communication methods have shifted from traditional face-to-face interactions to virtual platforms. Consequently, the reliance on digital tools and technologies has grown exponentially, altering not only the nature of our conversations but also the means by which they are documented and managed. In this paper, we introduce a novel approach to automatic minuting tools tailored to address the unique challenges of online communication. We submitted this system for Task A of the AutoMin2023 challenge (Ghosal et al., 2023). The primary objective

of this task is to develop an automated system capable of generating minutes from multiparty meeting transcripts. The performance of the resulting summaries are to be assessed using a combination of automatic and manual evaluation metrics.

For this system, we only used the task training data as well as the additional data that was recommended. We did not use Large Language Models nor any additional training data, which positions our submission in the *constraint* category. Instead, we used “classical” language models derived from BART. While there is no strict parameter count that officially defines if a language model is “large”, at the time this paper was written, the consensus seems to be that any model exceeding 1 billion parameters with the capacity to be prompted qualifies as such in the work of Zhao et al. (2023). However, BART does not meet these criteria. We first describe in Section 2 the data provided for the AutoMin Shared Task, being the ELITR and the EuroParlMin Corpus. Then we provide related work in 3 before describing in Section 4 the different modules of our system. Finally, we provide in Section 5 insights on the results by analyzing the effect of each module on the metrics and detailing the different errors we’ve encountered in the generated minutes.

## 2 Presentation of the data

### 2.1 ELITR Minuting Corpus

The ELITR Minuting Corpus presented by Nedoluzhko et al. (2022) is a dataset containing de-identified transcripts of project meetings and their corresponding minutes, primarily focusing on the computer science domain. The Corpus contains meetings in English and meetings in Czech. The English part of the dataset predominantly includes discussions among computer science professionals, while the Czech portion encompasses deliberations

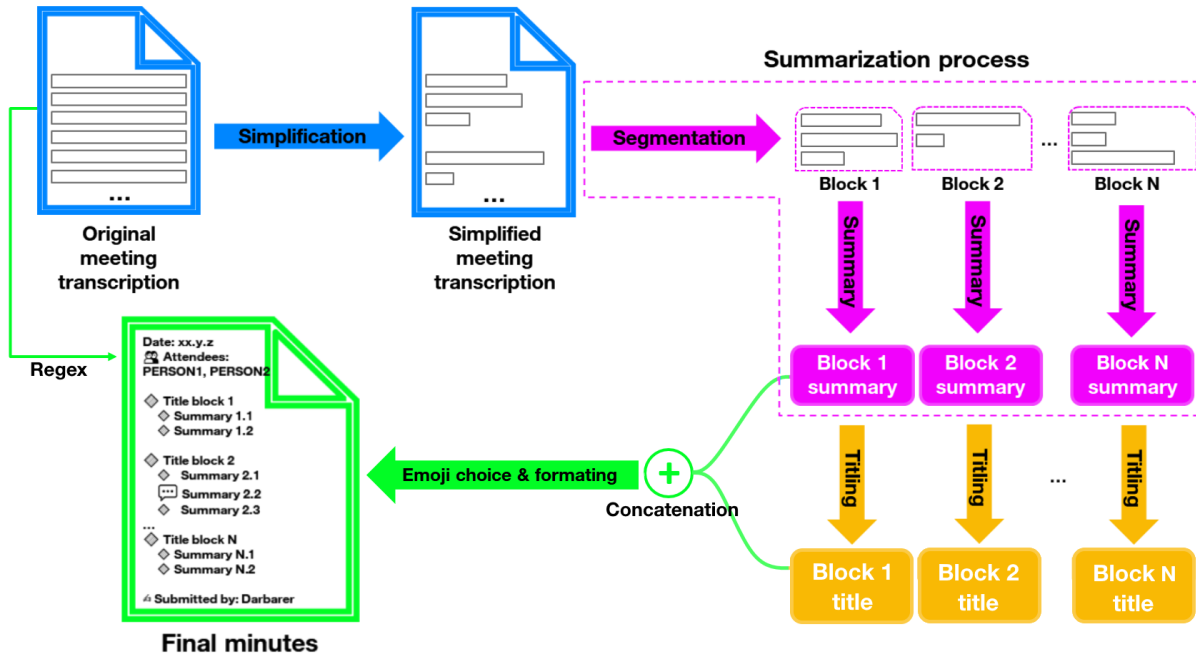


Figure 1: Processing chain (pipeline) for the automatic minute generation

from both the computer science and public administration fields.

The duration of the meetings captured in the dataset ranges from brief 10-minute exchanges to extensive discussions lasting over 2 hours.

One of the distinguishing features of the ELITR Minuting Corpus is the presence of multiple minutes files for a single conversation, thus offering a nuanced perspective on the variations in the interpretation and representation of meeting discussions. In addition, the Corpus includes, for some of the meetings, alignment files that facilitate the correlation between individual lines from the minutes files and the corresponding sections of the transcript files.

The minutes in the datasets are authored by various human annotators, each possessing distinct styles and perspectives on effective summarization. Consequently, the minutes exhibit substantial disparities in length, content, and organization. They may adopt flat or hierarchical structures, feature long sentences or keywords, and be arranged chronologically or by topic. These variations highlight the diverse approaches to summarization and offer a rich resource for studying the nuances of human-generated summaries. Table 1 shows statistics on the data and illustrates the disparities just mentioned earlier. The line compression ratio is the ratio between the number of lines in the transcripts and the number of lines in the annotated minutes.

Statistics	Mean	Std
nb. minutes per transcript	2.10	1.33
line compression ratio	12.15	21.06
nb. words per minute line	11.6	9.20

Table 1: Statistics of the ELITR Corpus

## 2.2 EuroParlMin Corpus

The EuroParlMin dataset is a subset of the broader EuroParl Corpus presented by Koehn (2005), focusing exclusively on English transcripts of European Parliament sessions from 2004 to 2011. Some sessions are split into chapters, and in that case there is one transcript file per chapter. The durations are not provided.

Unlike ELITR Minuting Corpus, each chapter of the sessions has only one associated minute. This reduces disparities in length between the minutes.

Statistics	Mean	Std
line compression ratio	6.62	13.05
nb. words per minute line	6.85	6.48

Table 2: Statistics of the EuroParlmin Corpus

## 3 Related work

Ghosal et al. (2021) give an overview of the systems submitted for the same task in the previous edition. Two systems stand out: that of the ABC

team (Shinde et al., 2021) and that of the Hitachi team (Yamaguchi et al., 2021). Both systems share a number of common features. Firstly, both teams have sought to partition the conversation in such a way that each part can fit the input of a transformer-based summarization module. Once each segment has been summarized, a concatenation is performed between the different summaries to obtain a global summary of the conversation.

In the case of the ABC team, conversation segmentation is carried out with a linear segmentation, cutting the conversation into blocks of tokens of uniform size. This segmentation is itself preceded by a rule-based block to remove redundant/repetitive elements.

In the case of the Hitachi team, segmentation is carried out automatically using a LongFormer (Beltagy et al., 2020) model, in order to select and group passages of interest in the conversation. The authors used manual annotation to train their segmentation method.

Our method is essentially based on these main steps (segmentation, summarization, concatenation), but we propose to add a text simplification module before segmentation in order to compress the text and increase the amount of information present in each segment.

## 4 Overview of the submitted system

This Section presents the Darbarer system<sup>1</sup> submitted for AutoMin2023 Task A, in the constraint category. The overall system is illustrated in Figure 1. Our system is composed of four main modules. The first module relies on a text simplification model aiming at standardizing the utterances of the conversation and compressing the input in order to focus on informative content. The second module handles summarization by employing a straightforward segmentation strategy and a fine-tuned BART-based generative model. Then a titling module has been trained in order to propose a short description of each summarized block. Lastly, we apply a post-processing step aimed at enhancing readability through specific formatting rules.

### 4.1 Transcription simplification

A conversation involves several people exchanging information about one or multiple topics. Each

<sup>1</sup>The code and data specifically annotated to train some of the modules is available at <https://github.com/Orange-OpenSource/automin2023-darbarer>

person communicates in a manner that may vary significantly from one to another. This heterogeneity, notably put forward by Schiffrin (1990) can give rise to difficulties when trying to summarize spoken conversations. Additionally, disfluencies that are inherent to spontaneous speech, as well as discourse markers that help the intelligibility of speech in an interaction context, yield uninformative content in transcriptions that harms readability. Small talks can also be present and should not be transposed in the minutes. Even if the readability of the input transcript is not necessarily linked to the performance of a summarization model, it seems preferable to remove as much uninformative content as possible prior to performing the summarization task. Text simplification makes it possible to reduce the linguistic complexity of statements in a conversation and thus, in a way, allows to standardize each utterance. Text simplification has been studied initially as a way to increase accessibility of texts (for language learners for instance) and has also shown a real utility in many tasks in automatic language processing, including in summarization tasks. Silveira and Branco (2012) put forward that the simplification allows, in a summarization task, to more easily bring out the important information. In the same way, Hasler et al. (2017) shows that in translation tasks, text simplification has a positive effect. Simplification is also used for caption and subtitles generation (Buet and Yvon, 2021) as a way to compress information. It has been studied a lot in the biomedical domain as a way to increase accessibility of biomedical texts (Cardon and Grabar, 2020; Ondov et al., 2022).

In order to perform this text simplification, we first selected 86 utterances from a single conversation from the training set and manually annotated simplified version of each utterance, as can be seen in table 3. We then fine-tuned a mBART-based model named mBarthez (Kamal Eddine et al., 2021).<sup>2</sup> mBarthez, like mBART, is pretrained on denoising tasks on multilingual data. Preliminary experiments run on in-house data, that are not reported here, showed that mBarthez performed better than mBart on the simplification task. The model was trained with the following hyperparameters : a learning rate of 3e-6, a batch size of 3, 1200 for max-steps and 0 warmup steps.

Table 4 shows a few examples of the simplifi-

<sup>2</sup>Model available on HuggingFace : <https://huggingface.co/moussaKam/mbarthez>

Original transcript	Human-written simplification
Well today (unintelligible) should be pretty simple, simple, simple meeting.	Today should be a simple meeting
And so we have collected all of them, and we are going to reveal presentation platform.	We are going to reveal the presentation platform
So you can you mute when you have some interview.	Can you mute?
Yeah, okay, yes, great.	Yes.

Table 3: Sample of the training data for text simplification

Original transcript	Automatic Simplification
Yeah, ye- ye-.	
But it's just to-	But it's just to-
To- I don't know, maybe writing things, we can think about uh alternatives or limitations.	We can think about alternatives or limitations.
And so it will be interesting to, to check this kind of thing so-	It will be interesting to check this kind of thing so-
Um so far I- I just organize the- the sections in bullet points.	I just organize the sections in bullet points.

Table 4: Examples of conversation lines before and after simplification

cation task, performed with our model. Note that the simplification model can yield empty outputs and thus can allow us to remove some lines of the transcriptions which are carrying little information.

It is noteworthy and surprising to observe that a relatively small number of training examples sufficed in achieving acceptable results for text simplification. We conducted an ablation study to further investigate the impact of the volume of the training data on the results. This analysis involved iteratively training the model with diminishing volumes of data and subsequently observing the variations in inference outcomes as well as System Output Against References and Input Sentence (SARI) scores (Xu et al., 2016). The test set consists of 20 utterances not present in the training set, as well as their corresponding human-written simplification. SARI is a metric specifically made to evaluate simplification results. It compares the system output not only against a reference (human simplified sentence), but also against the original complex sentence. This approach helps assess whether the system correctly keeps, adds, and deletes information. By looking at the inference results, we see that from [X] to [Y] examples, the model learns which sentences it should keep or not, but keeps the output sentence the exact same as the input. It is only after [Y] examples that the model starts to delete parts of the sentence that are not considered

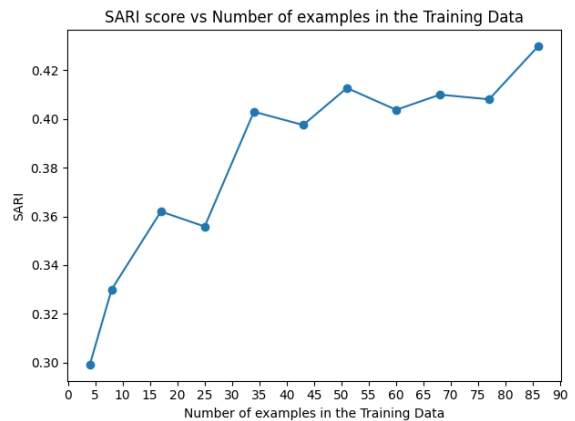


Figure 2: Evolution of the SARI score as we increase the number of simplification training examples

relevant. As seen in 2, we observe that the SARI score keeps rising as we increase the amount of examples. It does not seem to reach a plateau yet and giving the model more training examples might improve our results.

## 4.2 Summarization

The summarization module is the main component of our processing chain. For this task we decided to use the checkpoint of the BART model presented by Lewis et al. (2020). This model is trained on the XSum (Narayan et al., 2018) dataset which consists of short summaries of BBC articles and on the

SAMSum dataset (Gliwa et al., 2019) which is composed of conversations summaries.<sup>3</sup> This checkpoint showed interesting performances in the summarization task, especially on the ELITR dataset as shown by Nedoluzhko et al. (2022).<sup>4</sup> Unfortunately a major limitation of this model is the size of the text that it can take as input, which is currently limited to 1024 tokens. A naive way to deal with this, is to split the conversations into blocks of size 1024 and then summarize each block. This way of proceeding is sub-optimal since this segmentation can clearly cut the conversation in the middle of a topic and thus induce an important loss of information. We tested several more sophisticated methods, including clustering transcript lines in order to take into account the non linear nature of meeting topics. None of them improved the ROUGE score and the coherence of the resulting blocks was also perceptually degraded. We kept the fixed 1024 token segmentation for our system, but we believe that this should be further investigated.

Note that with the previous simplification step in place, blocks of 1024 tokens can now hold more information. Consequently, the average number of blocks per meeting has decreased.

To enhance the readability of the minutes, every summarized block undergoes post-processing steps which include titling and general formatting.

### 4.3 Summary block titling

Titles are a simple way to improve readability and overall comprehension in a document by providing a piece of context to the reader before the main content, as shown by Alba et al. (1981).

Thus, we once again fine-tuned mBarthez (Kamal Eddine et al., 2021), this time for the task of title generation, with the same hyper-parameters as for the text simplification. In order to achieve this, we specifically wrote relevant titles for 65 summarized blocks that were previously generated, as shown in Table 5 and used this annotated dataset to fine-tune mBarthez.<sup>5</sup>

Table 6 displays a few examples of the titles generated during inference for the ELITR dev-09 conversation using this model.

<sup>3</sup>*Ibid*: lidiya/bart-large-xsum-samsum

<sup>4</sup>particularly, see Table 6 in (Nedoluzhko et al., 2022)





<sup>5</sup>The spelling and capitalization errors of “PERSON” expressions occurred during generation and are explained in 5.2.2

### 4.4 Final formatting

Our objective is to produce meeting minutes that are neatly organized into blocks, where each block is defined by a specific title and comprises a list of indented bullet points. To achieve this, we rely on the prior stages of summarization and titling. In our setup, each sentence from a summary becomes a separate bullet point. This makes the information easier to break down and understand.

To improve readability even further, we have incorporated specific rules to generate emojis for each bullet point. This incorporation of visual cues is an additional step in our strategy to enhance minutes readability.

To generate the convenient emojis for each bullet point line, we defined a set of rules:

- If the line contains the word “date”, “deadline”, “afternoon”, “tomorrow”, “yesterday” or a day of the week, the emoji generated for this bullet point is the calendar emoji .
- If there is a discussion between some persons in the line (triggered by the verbs “discuss” or “talk”), the emoji associated to this bullet point will be the discussion balloon emoji .
- For the lines that evoke a deadline or the existence of some warnings in something (triggered by the words “deadline”, “warning” and “careful”), we add at the end of the bullet point the warning sign emoji .
- If the bullet point where there is a task still not complete or wait another task to be done (triggered by the words “still” and “wait”), we add at the end the hourglass emoji .

We also add a header containing the date and the attendees of the meeting using simple regular expressions on the transcript. Plus the signature at the end of the minute. Adding the header has an impact on the ROUGE scoring, as will be seen in Table 8 whereas the additional stylistic adjustments are not taken into account by the scoring methodology. We believe however that the latter may increase the *fluency* criterion during the human evaluation.

### 4.5 From English to Czech

All the models we’ve used thus far have been specifically fine-tuned on English corpora. The issue at hand now is the application of our method to the

Bullet points	Human-written title
PERSON7, PERSON8, PERSEN9 and PERSON4 had a call last week. They will have to provide at least some prototype for the n-best list navigation and they will try to implement it into the final product.	Provision of navigation list
PERSON8 wants to have a single module that can have all the functionality of the browser translator.	Browser translator module

Table 5: Sample of the training data for title generation

Bullet points	Generated Title
PERSON6 is collecting data. He sends bad transcripts with bad quality to the annotator and asks him to correct them, then he sends it to Person6 via FileSender. Person6 sends him the pre-processed automatic speech reconstructed transcript.	Correction of transcripts
PERSON7 wants to know how the link works It is the same link as the one in the same meeting invite People can use it for all of their meetings The meeting is free for one hour, but they have to pay for it for the next month or so.	Working on the link

Table 6: Examples of section titles generated using the segment bullet points.

Czech transcriptions of the ELITR dataset. In order to re-use the same pipeline, we add two translation blocks. A first block that translates the transcriptions from Czech to English. We then generate our minutes (in English) with our processing chain, to finally translate back from English to Czech. For this purpose we use the (Tiedemann and Thottingal, 2020) models which offer the possibility to translate in both directions.<sup>6</sup> We did not perform any particular fine-tuning for this translation task.

## 5 Results

The AutoMin 2023 challenge provided three test sets: `elmiCS` and `elmiEN` for ELITR Meeting in Czech and English respectively and `europarl` in English. Full results and details about the evaluation process are provided in Ghosal et al. (2023). We obtain a ROUGE-1 score of 0.31 on `elmiCS`, 0.39 on `elmiEN` and 0.27 on `europarl`. Manual evaluation has been produced with the ALIGN-MEET tool (Polák et al., 2022), focusing on adequacy, grammaticality, fluency, relevance and at two different levels of granularity : at the document-level and the hunk-level (a hunk is defined as a set of dialog acts belonging to a summary point). Table 7 shows the results for of our system according to human annotators. Examples of generated minutes from the `test` partition are provided in Appendix. In this Section we provide additional objective evaluations and insights on observed er-

<sup>6</sup>We used `Helsinki-NLP/opus-mt-cs-en` and `Helsinki-NLP/opus-mt-en-cs` checkpoints, available on HuggingFace.

rors on the initially provided test datasets (`test` and `test2`).

### 5.1 Ablation studies

In order to evaluate the impact of each module, we use several metrics as can be seen in Table 8, with *Darbarer* being the final system we used to submit our minutes for the task. The baseline system applies the `bart-large-xsum-samsum` model on fixed blocks of 1024 tokens, without any pre-processing nor post-processing. For the second line, we applied Simplification prior to segmentation and summarization. The third line adds the titling step for each summarized block. And finally the formatting step is added to obtain the last line (*Darbarer*). We decided to evaluate the results with metrics usually used for the summarization task: ROUGE (Lin, 2004) and BERTScore (Zhang et al., 2019). These metrics are also used during the evaluation process of the task alongside a human evaluation. However, Ghosal et al. (2021) have shown that these metrics have poor correlations with human judgement. During our research, we found similar discrepancies with our results, which seemed perceptually better without an improvement of the scores. We thus decided to look at the number of words and blocks generated by our systems, with the assumption that shorter summaries will improve the overall readability and clarity of the minutes.

While the overall values of ROUGE and

<sup>7</sup><https://github.com/chakki-works/sumeval>

<sup>8</sup><https://pypi.org/project/bert-score/>

	elmiEN		europarl	
	Document-level	Hunk-level	Document-level	Hunk-level
<b>Adequacy</b>	3.14 ± 0.60	4.03 ± 1.11	2.33 ± 1.21	4.44 ± 0.81
<b>Grammaticality</b>	4.92 ± 0.18	4.93 ± 0.41	5.00 ± 0.00	5.00 ± 0.00
<b>Fluency</b>	3.64 ± 0.42	4.17 ± 1.11	3.50 ± 1.05	4.44 ± 0.81
<b>Relevance</b>	4.67 ± 0.67	4.46 ± 0.71	4.83 ± 0.41	4.94 ± 0.25

Table 7: Average human evaluation scores (1: worst, 5: best) for English meetings. The figures correspond to mean±standard deviation.

		Summarization	Simplification	Titling	Formatting	R1	R2	RL	BERT Score (scaled)	Words	Blocks
test	baseline	✓				<b>0.32</b>	<b>0.08</b>	<b>0.18</b>	<b>0.44</b>	392	12
	+ simplification	✓	✓			0.29	0.06	0.16	0.42	<b>294</b>	<b>8,7</b>
	+ titling	✓	✓	✓		0.29	0.06	0.16	0.42	316	<b>8,7</b>
	Darbarer	✓	✓	✓	✓	0.30	0.06	<b>0.18</b>	<b>0.44</b>	330	<b>8,7</b>
test2	baseline	✓				<b>0.33</b>	<b>0.08</b>	0.19	0.41	417	14
	+ simplification	✓	✓			0.28	0.07	0.18	0.40	<b>310</b>	<b>9,8</b>
	+ titling	✓	✓	✓		0.29	0.07	0.18	0.40	339	<b>9,8</b>
	Darbarer	✓	✓	✓	✓	0.31	0.07	<b>0.20</b>	<b>0.43</b>	352	<b>9,8</b>

Table 8: Results of the ablation study. The ROUGE scores were computed using the Sumeval<sup>7</sup> library, removing stopwords from the provided list. The BERTScore was computed with the bert-score library<sup>8</sup> using the *rescale with baseline* option for a larger range and better human-readability of the score. The words and blocks column show the average number of words and blocks across minutes.

BERTScore do not strongly change for any of these steps, we observe some general patterns in the results:

- The simplification module seems to decrease both the ROUGE and BERTScore by a few points. However, it allows the system to produce shorter minutes (by about 33%) and of seemingly better quality when looking at the actual content of the minutes. Looking more closely to the results, we see that the precision component of the metrics increases, while the recall one falls by a few points.
- The titling module has little to no effect on the metrics, but allows for better readability.
- The formatting improves the ROUGE and BERTScore by a few points.

## 5.2 Error analysis

### 5.2.1 Simplification

The simplification process occasionally removes sections of the transcript that could be essential for creating an accurate summary. Additionally, it may inadvertently alter the meaning of certain sentences, potentially leading to misunderstandings or misinterpretations in the summarized content. The

following are examples of transcripts followed by their simplification (right side of the arrow). In the first case, the meaning is slightly altered and the information about “manual alignments” is removed. In the second example, the output is not simplified and removing the first part alters the general meaning. even though a thorough evaluation should be achieved. While these phenomena seem to be quite rare, a more thorough evaluation should be conducted in order to clearly quantify their frequency and impact.

- (Person6) Great, so we do alignment, fine the manual are done, but what is the final output? → (Person6) How is the final output?
- There is nothing that I would know about that we need to discuss uh, like in in in a very big detail, ehm. → I would know about that we need to discuss uh, like in in in a very big detail

### 5.2.2 Summarization distortions

Some errors were produced during the summarization step. One of these recurrent errors was generating inaccurate tags (we refer to anonymized entities such as PERSON, ORGANIZATION, PROJECT and LOCATION as tags). For example, *Organizing6 / Organizer* instead of [ORGANIZATION6]

or *Person A / PERSO / PERSS* instead of [PERSON1].

We thus proceeded to a manual analysis on all the tags present on the `test` partition to extract the statistics shown on Table 9. As we can see, this type of error appears in only 3.6% of the generated tags, but they are particularly harmful for the general meaning of minutes.

# generated tags	358
# wrong tags	15
percentage of wrong tags	3.6%

Table 9: % Tag errors in generated minutes of `test`

### 5.2.3 Person tracking

Dialogue summarization models face a challenge when it comes to tracking the identity of speakers, addresses and people indirectly mentioned with third person pronouns, particularly when there are many parts in the conversation. Specifically, these models encounter difficulties in accurately determining the referent of a personal pronoun (e.g., “you”) when transitioning from direct speech in the conversation to an indirect speech format for the summary. This challenge appears because the model needs to infer the identity of the pronouns based on the conversation’s structure to appropriately assign them in the summarization process.

In order to overcome this challenge, researchers have explored various strategies to enhance the performance of dialogue summarization models. One promising approach that has been recently proposed by Fang et al. (2022) is to replace each pronoun with its specific noun. This technique helps the models to avoid misplacing the nouns during the summarization process, which can significantly improve the overall quality.

We also detected ambiguities regarding pronouns in the generated minutes. Some bullet points in the minutes were generated with pronouns such as *he*, *she* or *them*, while it is impossible to guess who they refer to without any context about the conversation. The following is an example :

- PERSON3 is not sure whether he will join

In this example, *he* seems to refer to PERSON3 while in the context it refers to PERSON11.

We checked the minutes generated for the first 9 transcriptions of the `test` partition to pull out the percentage of pronouns with unclear antecedent.

We observed 14 indefinite pronouns, among which 4 of them could not be resolved given the summarized context. Here again this type of error can be misleading for the general understanding of the minutes. An additional analysis on the first transcript from the `train` partition revealed that among the 89 occurrences of the pronoun *you*, 43 corresponded to the previous speaker and 33 corresponded to the last mentioned tag. This illustrates that resolving the *you* pronouns is not a trivial task. Further analysis should be achieved to better understand the impact of person tracking on the overall acceptability of the generated minutes.

### 5.2.4 Titling

The automated generation of titles in the dataset is not entirely error-free. Various issues can be observed, which may lead to misunderstandings while reading. These errors can be broken down into multiple types: grammatical mistakes (e.g. “*Meet today in person*”), semantic inaccuracies (e.g. “*Summarisation of the minutes annotation*”), or nonsensical phrases (e.g. “*Edit of ehm*”).

We checked the minutes generated by our model on the first nine meetings on the `test` partition to see if the title for each block is coherent or if it contains grammatical mistakes or semantic inaccuracies. We observed that 54 titles out of 70 were fully coherent.

## 6 Conclusion and discussion

In this paper, we described our system for the AutoMin 2023 challenge Task A and detailed its four different modules: simplification, summarization, titling and formatting. Our submitted system produces meeting minutes that are concise, intelligible and that may already be usable without further modifications, in a multitude of use cases. However, it is not error-proof and still subject to improvement, regarding the way we could cleverly split the conversation into coherent segments, or how to ensure correctness in regards to grammar, semantics and person tracking. Moreover, this work highlights the need for better metrics for evaluating the results of abstractive summarization systems in order to make better informed decisions for the design of the whole pipeline. We believe this Shared Task to be very relevant, especially in times where automatic content summarization is becoming more and more common.



## References

- Joseph W Alba, Susan G Alexander, Lynn Hasher, and Karen Caniglia. 1981. The role of context in the encoding of information. *Journal of Experimental Psychology: Human Learning and Memory*, 7(4).
- Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.
- François Buet and François Yvon. 2021. Toward genre adapted closed captioning. In *Interspeech'21*.
- Rémi Cardon and Natalia Grabar. 2020. French biomedical text simplification: When small and precise helps. In *The 28th International Conference on Computational Linguistics*.
- Yue Fang, Hainan Zhang, Hongshen Chen, Zhuoye Ding, Bo Long, Yanyan Lan, and Yanquan Zhou. 2022. From spoken dialogue to formal summary: An utterance rewriting for dialogue summarization. In *Proc. NAACL*.
- Tirthankar Ghosal, Ondřej Bojar, Marie Hledíková, Tom Kocmi, and Anja Nedoluzhko. 2023. Overview of the second shared task on automatic minuting (automin) at inlg 2023. In *Proceedings of the 16th International Conference on Natural Language Generation: Generation Challenges*. Association for Computational Linguistics.
- Tirthankar Ghosal, Ondřej Bojar, Muskaan Singh, and Anja Nedoluzhko. 2021. Overview of the first shared task on automatic minuting (automin) at interspeech 2021. *Proceedings of the First Shared Task on Automatic Minuting at Interspeech*.
- Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. Samsun corpus: A human-annotated dialogue dataset for abstractive summarization. *arXiv preprint arXiv:1911.12237*.
- Eva Hasler, Adrià de Gispert, Felix Stahlberg, Aurelien Waite, and Bill Byrne. 2017. Source sentence simplification for statistical machine translation. *Computer Speech & Language*, 45:221–235.
- Moussa Kamal Eddine, Antoine Tixier, and Michalis Vazirgiannis. 2021. **BARThez: a skilled pre-trained French sequence-to-sequence model**. In *Proc. EMNLP'21*, Dominican Republic.
- Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proc. Machine Translation summit*.
- 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 *Proc. ACL'20*.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*.
- 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*, abs/1808.08745.
- 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 *Proc. LREC'22*.
- Brian Ondov, Kush Attal, and Dina Demner-Fushman. 2022. A survey of automated methods for biomedical text simplification. *Journal of the American Medical Informatics Association*, 29(11).
- Peter Polák, Muskaan Singh, Anna Nedoluzhko, and Ondřej Bojar. 2022. **Alignmeet: A comprehensive tool for meeting annotation, alignment, and evaluation**.
- Deborah Schiffrin. 1990. Conversation analysis. *Annual Review of Applied Linguistics*, 11:3–16.
- Kartik Shinde, Nidhir Bhavsar, Aakash Bhatnagar, and Tirthankar Ghosal. 2021. Team abc@ automin 2021: Generating readable minutes with a bart-based automatic minuting approach. *Proceedings of the First Shared Task on Automatic Minuting at Interspeech*.
- Sara Botelho Silveira and António Branco. 2012. Enhancing multi-document summaries with sentence simplification. In *Proc. ICAI'12*, page 1.
- Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT — Building open translation services for the World. In *Proc. EAMT'20*, Lisbon, Portugal.
- Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. volume 4, pages 401–415.
- Atsuki Yamaguchi, Gaku Morio, Hiroaki Ozaki, Kenichi Yokote, and Kenji Nagamatsu. 2021. Team hitachi@ automin 2021: Reference-free automatic minuting pipeline with argument structure construction over topic-based summarization. *arXiv preprint arXiv:2112.02741*.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.
- Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. **A survey of large language models**.

## Generated minute for the test\_04 meeting



Date: 11.9.2022



Attendees: PERSON13, PERSON3, PERSON10, PERSON14, PERSON7

- ◆ Making a presentation platform
  - ◆ [PERSON10] and [PERSON3] are working on a presentation platform
  - ◆ [PERSON10] wants to know if anyone is willing to play with it.
- ◆ Working on text client
  - ◆ [PERSON10], [PERSON13] and [PERSON7] are working on a text client which should be able to send text to text worker
  - ◆ It works with the Czech Czech machine translation, but there is a problem with the batch processing mode and the ASR.
  - ◆ They need to perform more test next week.
- ◆ Quality of online ASR and machine translation
  - ◆ [PERSON10], [PERSON13] and [PERSON14] are working on improving the quality of online ASR and machine translation.
- ◆ Batch mode and segmentation work
  - ◆ [PERSON7] created a batch mode, but it didn't work
  - ◆ The segmentation workers don't work with the CTM client
  - ◆ The ASR worker starts producing text in the chat window
- ◆ Control of segmentation worker
  - ◆ [PERSON10] wants to know more about the segmentation worker
  - ◆ It only handles the text as it comes out of the segmentor
  - ◆ There will be a session, session with and a test next week
- ◆ Process of the presentation platform
  - ◆ [PERSON3] has prepared a presentation platform for June
  - ◆ The presentation platform will have an operator monitoring the output of one of the re-speakers cabins and if the output from the floor is bad, the operator should kill the client and switch to the other provided translation
- ◆ Implementation of MT outputs
  - ◆ There are 4000 people connected on the same WiFi network
  - ◆ The current user is expected to be at June
  - ◆ They want to know which of the MT outputs is the best at the moment
  - ◆ They need to decide how to deliver the subtitles to the participants
  - ◆ They have a year to find a better solution



Submitted by: Darbarer

## Generated minute for the test\_10 meeting



Date: 22.8.2022



Attendees: PERSON3, PERSON2, PERSON4, PERSON1

- ◆ Record the meeting
  - ◆ [PERSON3], [PERSON2] and [PERSON3] are going to record the meeting
  - ◆ [PERSON3] will send the poll for the next week as well
  - ◆ Organizing Committee will divide the budget for the meeting among other parties, but each party will get their own funding.
- ◆ Preparation of work package
  - ◆ [PERSON3] asks Organizing Committee to prepare a work package for the presentation application development for live meetings
  - ◆ She also asks for a dry-run and a follow-up workshop
- ◆ Work plan for the project
  - 💬 [PERSON1], [PERSON2] and [PERSON3] are discussing the organization's work plan for the three-year-long project
- ◆ Preparation of speaker
  - 💬 [PERSON3] and [PERSON2] discuss how to prepare a speaker for a conference.
- ◆ recording and the adaptation of a voice
  - ◆ [PERSON1] and [PERSON3] explain to each other what is required for the recording and the adaptation of a voice.
- ◆ Preparation of proposal
  - ◆ [PERSON3] and [PERSON1] have 14 days to prepare a proposal
  - ◆ They need the audio equipment for the re-speakers, and they need to check the availability of specific hardware
  - ◆ They also need to work on the integration of ASR essential from multiple partners into the platform.
- ◆ Design of deliverables
  - 💬 [PERSON3], [PERSON4] and [PERSON1] discuss the design of deliverables for the project
  - ◆ The deliverables should be in line with the timing of the work packages.



Submitted by: Darbarer

# Team NTR @ AutoMin 2023: Dolly LLM Improves Minuting Performance, Semantic Segmentation Doesn't

**Eugene Borisov**

NTR Labs and Higher IT School  
of Tomsk State University  
Tomsk, Russia  
eborisov@ntr.ai

**Nikolay Mikhaylovskiy**

NTR Labs and Higher IT School  
of Tomsk State University  
Moscow, Russia  
nickm@ntr.ai

## Abstract

This paper documents the approach of Team NTR for the Second Shared Task on Automatic Minuting (AutoMin) at INLG 2023. The goal of this work is to develop a module for automatic generation of meeting minutes based on a meeting transcript text produced by an Automated Speech Recognition (ASR) system (Task A). We consider minuting as a supervised machine learning task on pairs of texts: the transcript of the meeting and its minutes. We use a two-staged minuting pipeline that consists of segmentation and summarization. We experiment with semantic segmentation and multi-language approaches and Large Language Model Dolly, and achieve Rouge1-F of 0.2455 and BERT-Score of 0.8063 on the English part of ELITR test set and Rouge1-F of 0.2430 and BERT-Score of 0.8332 on the EuroParl dev set with the submitted Naive Segmentation + Dolly7b pipeline.

## 1 Introduction

Discussions and meetings are an integral part of any human activity that involves a group of people. On important meetings, an audio recording is often made, and specially appointed people create a brief summary of the most important things that happened at the meeting. This process is quite laborious.

The ability to produce high-quality documentation of business meetings decisions without allocating additional human resources can improve the productivity of the organizations. This way important points and decisions made will not be lost due to an information overflow. Thus, automated minuting of business meetings is becoming an increasingly desirable solution.

An automated minuting system can be useful not only for businesses but also for government agencies and educational institutions. Hundreds of

meetings are held daily, and the ability to automatically generate a summary of the most important decisions made can significantly reduce the time and resources spent on documenting. Thanks to an automatic minuting system, meeting participants can focus on important points without spending time on note-taking.

The goal of this work is to develop a module for automatic generation of meeting minutes based on a meeting transcript text produced by an Automated Speech Recognition (ASR) system (AutoMin 2023 Task A, (Ghosal et al., 2022b, 2023)).

## 2 Related work

Meeting summarization as a scientific problem came to light in the early 2000s (Ghosal et al., 2022a). ISCI Meeting Project (Morgan et al., 2001; Janin et al., 2004) resulted, among other outcomes, in creating ICSI Meeting Corpus of audio recorded from informal, natural, and even sometimes impromptu meetings (Janin et al., 2003). About simultaneously, Klaus Zechner's work on summarization of meeting speech and dialogues (Zechner, 2002) helped to shape the investigations in this topic further. Augmented Multi-party Interaction (AMI) project followed soon, producing The AMI Meeting Corpus (McCowan et al., 2005).

It has been recognized early on that for a wide spectrum of applications identifying and including action items into minutes delivers the key business value (Purver et al., 2007). Detecting decisions in multi-party dialogues happened to be as important for the minuting (Fernández et al., 2008b,a; Bui et al., 2009). The CALO Meeting Assistant System (Riedhammer et al., 2010) that appeared soon after that was an important step, but the overall level of NLP at the time limited (with a few exceptions, for example, (Wang and Cardie, 2012; Liu et al., 2018)) proliferation of minuting research and applications

until recently.

Scientific interest to minuting reemerged about 2018, sparked both by an important review by Nedoluzhko and Bojar (Nedoluzhko and Bojar, 2019) and overall technology readiness. The interest came into a full swing in 2021 when many works have appeared (Koay et al., 2021; Shang, 2021; Fu et al., 2021; Chen and Yang, 2021; Fabri et al., 2021; Zou et al., 2021; Cho et al., 2021; Chen et al., 2021; Zhong et al., 2021) and the first AutoMin competition was held at INTERSPEECH (Ghosal et al., 2022a).

The works that are the most close to ours are likely those by AutoMin 2021 winners (Shinde et al., 2021) and Borisov and Mikhaylovskiy (Borisov and Mikhaylovskiy, 2023). The authors of the former use a BART model and train it on the SAMSum dialogue summarization dataset. Their pipeline first splits the given transcript into blocks of smaller conversations, eliminates redundancies with a specially-crafted rule-based algorithm, summarizes the conversation blocks, retrieves the block-wise summaries, cleans, structures, and finally integrates the summaries to produce the meeting minutes. The authors of the latter introduce a Russian minuting dataset and use an approach similar to (Shinde et al., 2021). They also introduce semantic segmentation that improves ROUGE and BERTScore metrics of minutes on the above dataset by 1%-10% compared to naive segmentation.

### 3 Datasets

Two main datasets are considered in the AutoMin 2023 Task A:

- ELITR Minuting Corpus – a dataset of meeting transcripts and minutes (Nedoluzhko et al., 2022).
- EuroParlMin v1.0, introduced specifically for AutoMin 2023 (Ghosal et al., 2023)

In addition, we experiment with the following corpora:

- SamSum – a dataset of messenger dialogues with their summaries (Gliwa et al., 2019).

The datasets are compared in Table 1. The summary compression ratio  $\theta$  in the Table 1 is calculated using the following formula:

$$\theta = \left(1 - \frac{T_A}{T_T}\right) * 100, \quad (1)$$

where  $T_A$  is the number of tokens in the abstract and  $T_T$  is the number of tokens in the transcript. Thus, the smaller the abstract compared to the original transcript text is, the closer  $\theta$  is to 100%.

### 4 Methods

All Transformer (Vaswani et al., 2017) language models have a limit on the size of the input context window and do not work well with long texts, such as transcripts of long meetings. Thus, to make it possible to apply Transformer-based models to the transcript text summarization, we, similarly to the winners of the AutoMin 2021 competition (Shinde et al., 2021) decompose the task of minuting into two subtasks:

- Text Segmentation – dividing the transcript text into segments of reasonable size.
- Segment Summarization – generating an abstract of the transcript segment.

In addition to the naive segmentation just fitting the chunk to the model’s window size, we explore semantic segmentation in a hope to obtain higher quality reporting. The pipeline for the semantic segmentation is as follows:

- For utterances vectorization, the transformer all-MiniLM-L6-v2 from the sentence transformers library (Reimers and Gurevych, 2019) was used. Each utterance was vectorized sequentially using the Mean Pooling (Reimers and Gurevych, 2019): initially, each utterance is broken down into sentences, then, using Mean Pooling, a vector of sentences is obtained, finally, the average of the sentence vectors is taken as the utterance vector.
- For dimensionality reduction, the UMAP (Uniform Manifold Approximation and Projection) algorithm was used (McInnes et al., 2018). The resulting compressed vector representations retain the necessary information to create clusters of semantically similar utterances. Thus, in the clustering of utterances, the use of UMAP allows you to preserve the quality of the segments obtained by clustering, while generally increasing the speed of segmentation due to working with lower-dimensional vectors.
- For clustering the obtained utterance vectors, the density-based HDBSCAN algorithm

Name	Transcripts	Domain	Compression ratio, %
ELITR	179	project meetings	95.65
EuroParlMin Dev	187	corpus of European Parliament debates	53.08
SamSum	16369	dialogues from messengers	81.12

Table 1: Datasets

Model	Rouge1-F	Rouge2-F	RougeL-F	BERT-Score
Naive segmentation	<b>0.1977</b>	<b>0.0375</b>	<b>0.1624</b>	<b>0.6806</b>
Semantic Segmentation	0.1791	0.0339	0.1370	0.6768
Semantic Segmentation with UMAP	0.1771	0.0341	0.1431	0.6304

Table 2: Segmentation methods performance metrics on the English part of ELITR test set

(Campello et al., 2013) is used. It allows to detect clusters in data without knowing their exact number initially, and is also resistant to noise and outliers, which allows to filter out utterances that are not relevant to the topics of discussion at the segmentation level. The BERTopic library (Grootendorst, 2022) was used to implement the clustering algorithm in the semantic segmentation module.

- Transcript Segments Summarization. We explore several models for abstractive summarization, as described below.

## 5 Experiments

### 5.1 Metrics

The key indicators of the effectiveness of a text summarization algorithm we use are the ROUGE (Lin, 2004) and BERTScore (Zhang et al., 2020).

### 5.2 Comparing segmentation approaches on ELITR English test set

We compared naive and semantic segmentation approaches with and without UMAP dimensionality reduction on this dataset. In all the cases we have used open source version of MBART finetuned on SamSum dataset.<sup>1</sup>

Table 2 shows the performance of the approaches listed above on the English test part of the ELITR dataset. One can see that in the domain of meetings of distributed teams most similar to day-to-day work discussions, semantic segmentation did not provide significant improvement. The semantic segmentation works worse than the naive one. The effect of the UMAP dimensionality reduction is also mixed.

<sup>1</sup><https://huggingface.co/philschmid/distilbart-cnn-12-6-samsum>

### 5.3 Experiments with a LLM

In addition to MBART, we have tested a large pre-trained language model – Dolly v2 7b (Conover et al., 2023), chosen for its permissive license and competitive performance. We have tried the prompts: "Summarize" and "Briefly extract Key Points from the meeting transcript", and settled for the second as it provided somewhat better performance and more interesting texts. Table 3 shows the difference between two prompts on EuroParl dev set.

Table 4 provides a comparison of Dolly with MBART on English part of ELITR test set. We have also tested Dolly v2 7b on EuroParl dev set, the results provided in the Table 5

Given the above results, we have chosen to submit Naive Segmentation + Dolly results.

## 6 Conclusion and Future Work

In this work, we describe our system run for the second AutoMin shared Task A on automatic minuting. Our proposed system leverages a pretrained Large Language Model Dolly to generate readable minutes from multi-party meeting proceedings. In the future, we plan to implement similar pipelines for different languages, including low-resource ones.

Dolly v2 7b Prompt	Rouge1-F	Rouge2-F	RougeL-F	BERT-Score
“Summarize”	<b>0.2465</b>	<b>0.0751</b>	<b>0.1927</b>	0.8251
“Briefly extract Key Points from the meeting transcript”	0.2430	0.0694	0.1843	<b>0.8332</b>

Table 3: Performance metrics on the EuroParl dev set with different prompts.

Model	Rouge1-F	Rouge2-F	RougeL-F	BERT-Score
BERTopic + MBART	0.244	<b>0.0309</b>	<b>0.1756</b>	0.7999
Naive + MBART	0.2022	0.0171	0.132	0.8019
Naive + Dolly7b	<b>0.2455</b>	0.0294	0.1656	<b>0.8063</b>

Table 4: Performance metrics on the English part of ELITR test set

Model	Rouge1-F	Rouge2-F	RougeL-F	BERT-Score
Naive + MBART	0.1539	0.0522	0.0843	0.8392
Naive + Dolly7b	<b>0.2430</b>	<b>0.0694</b>	<b>0.1843</b>	0.8332

Table 5: Performance metrics on the EuroParl dev set

## References

- Eugene Borisov and Nikolay Mikhaylovskiy. 2023. Automated Minuting on DumSum Dataset. In *Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference “Dialogue 2023”*, Online.
- Trung H. Bui, Matthew Frampton, John Dowding, and Stanley Peters. 2009. [Extracting decisions from multi-party dialogue using directed graphical models and semantic similarity](#). In *Proceedings of the SIGDIAL 2009 Conference: 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, September, pages 235–243.
- Ricardo J. G. B. Campello, Davoud Moulavi, and Joerg Sander. 2013. Density-based clustering based on hierarchical density estimates. In *Advances in Knowledge Discovery and Data Mining*, pages 160–172, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Jiaao Chen and Diyi Yang. 2021. [Structure-aware abstractive conversation summarization via discourse and action graphs](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1380–1391, Online. Association for Computational Linguistics.
- Wang Chen, Piji Li, Hou Pong Chan, and Irwin King. 2021. [Dialogue summarization with supporting utterance flow modelling and fact regularization](#). *Knowledge-Based Systems*, 229:1–29.
- Sangwoo Cho, Franck Dernoncourt, Tim Ganter, Trung Bui, Nedim Lipka, Walter Chang, Hailin Jin, Jonathan Brandt, Hassan Foroosh, and Fei Liu. 2021. [StreamHover: Livestream transcript summarization and annotation](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6457–6474, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. [Free Dolly: Introducing the World’s First Truly Open Instruction-Tuned LLM](#).
- Alexander R. Fabbri, Faiaz Rahman, Imad Rizvi, Borui Wang, Haoran Li, Yashar Mehdad, and Dragomir Radev. 2021. [ConvoSumm: Conversation summarization benchmark and improved abstractive summarization with argument mining](#). *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pages 6866–6880.
- Raquel Fernández, Matthew Frampton, John Dowding, Anish Adukuzhiyil, Patrick Ehlen, and Stanley Peters. 2008a. [Identifying relevant phrases to summarize decisions in spoken meetings](#). In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, September, pages 78–81.
- Raquel Fernández, Matthew Frampton, Patrick Ehlen, Matthew Purver, and Stanley Peters. 2008b. [Modelling and detecting decisions in multi-party dialogue](#). In *Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue, SIGDIAL 2008*, June, pages 156–163.
- Xiyan Fu, Yating Zhang, Tianyi Wang, Xiaozhong Liu, Changlong Sun, and Zhenglu Yang. 2021. [RepSum: Unsupervised dialogue summarization based on replacement strategy](#). *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational*

- Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pages 6042–6051.
- Tirthankar Ghosal, Ondřej Bojar, Marie Hledíková, Tom Kocmi, and Anja Nedoluzhko. 2023. Overview of the Second Shared Task on Automatic Minuting (AutoMin) at INLG 2023. In *Proceedings of the 16th International Conference on Natural Language Generation: Generation Challenges*. Association for Computational Linguistics.
- Tirthankar Ghosal, Ondřej Bojar, Muskaan Singh, and Anja Nedoluzhko. 2022a. Overview of the First Shared Task on Automatic Minuting (AutoMin) at Interspeech 2021. In *Proceedings of Interspeech 2021*, September, pages 1–25.
- Tirthankar Ghosal, Marie Hledíková, Muskaan Singh, Anna Nedoluzhko, and Ondřej Bojar. 2022b. The second automatic minuting (AutoMin) challenge: Generating and evaluating minutes from multi-party meetings. In *Proceedings of the 15th International Conference on Natural Language Generation: Generation Challenges*, pages 1–11, Waterville, Maine, USA and virtual meeting. Association for Computational Linguistics.
- 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.
- Maarten Grootendorst. 2022. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv preprint arXiv:2203.05794*.
- A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters. 2003. The icsi meeting corpus. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03)*, volume 1, pages I–I.
- Adam Janin, Jeremy Ang, Sonali Bhagat, Rajdip Dhillon, Jane Edwards, Javier Mac, Nelson Morgan, Barbara Peskin, Elizabeth Shriberg, Andreas Stolcke, Chuck Wooters, and Britta Wrede. 2004. THE ICSI MEETING PROJECT : RESOURCES AND RESEARCH. *Cognitive Science*, 2004(September).
- Jia Jin Koay, Alexander Roustai, Xiaojin Dai, and Fei Liu. 2021. A sliding-window approach to automatic creation of meeting minutes. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 68–75, 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.
- Hui Liu, Xin Wang, Yuheng Wei, Wei Shao, Jonathan Li-ono, Flora D. Salim, Bo Deng, and Junzhao Du. 2018. ProMetheus: An intelligent mobile voice meeting minutes system. In *ACM International Conference Proceeding Series*, November, pages 392–401.
- I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, W. Post, Dennis Reidsma, and P. Wellner. 2005. The ami meeting corpus. In *Proceedings of Measuring Behavior 2005, 5th International Conference on Methods and Techniques in Behavioral Research*, pages 137–140. Noldus Information Technology.
- Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. 2018. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861.
- Nelson Morgan, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Adam Janin, Thilo Pfau, Elizabeth Shriberg, and Andreas Stolcke. 2001. The meeting project at ICSI. In *Proceedings of the First International Conference on Human Language Technology Research*.
- Anna Nedoluzhko and Ondřej Bojar. 2019. Towards automatic minuting of meetings. In *CEUR Workshop Proceedings*, volume 2473, pages 112–119.
- 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 International Conference on Language Resources and Evaluation (LREC-2022)*, Marseille, France. European Language Resources Association (ELRA). In print.
- Matthew Purver, John Dowding, John Niekrasz, Patrick Ehlen, Sharareh Noorbaloochi, and Stanley Peters. 2007. Detecting and summarizing action items in multi-party dialogue. In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, pages 18–25, Antwerp, Belgium. Association for Computational Linguistics.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Korbinian Riedhammer, G. Tur, A. Stolcke, L. Voss, S. Peters, D. Hakkani-Tür, J. Dowding, and F. Yang. 2010. The CALO Meeting Assistant System. *IEEE Transactions on Audio, Speech and Language Processing*, 2010(18(6)).



- Guokan Shang. 2021. *Spoken Language Understanding for Abstractive Meeting Summarization*. Ph.D. thesis, l'École Doctorale de l'Institut Polytechnique de Paris.
- 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.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems*, pages 5999–6009.
- Lu Wang and Claire Cardie. 2012. [Focused meeting summarization via unsupervised relation extraction](#). In *SIGDIAL 2012 - 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue, Proceedings of the Conference*, July, pages 304–313.
- Klaus Zechner. 2002. [Automatic summarization of open-domain multiparty dialogues in diverse genres](#). *Computational Linguistics*, 28(4):447–485.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. [BERTScore: Evaluating text generation with BERT](#). In *International Conference on Learning Representations*.
- Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021. [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.
- Yicheng Zou, Bolin Zhu, Xingwu Hu, Tao Gui, and Qi Zhang. 2021. [Low-resource dialogue summarization with domain-agnostic multi-source pretraining](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 80–91, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

# Overview of the Second Shared Task on Automatic Minuting (AutoMin) at INLG 2023

Tirthankar Ghosal<sup>†</sup>, Ondřej Bojar<sup>\*</sup>, Marie Hledíková<sup>\*</sup>, Tom Kocmi<sup>‡</sup>, Anna Nedoluzhko<sup>\*</sup>

<sup>†</sup>Oak Ridge National Laboratory, TN, USA

<sup>\*</sup>Charles University, Faculty of Mathematics and Physics, ÚFAL, Czech Republic

<sup>‡</sup>Microsoft, Germany

<sup>†</sup>ghosalt@ornl.gov, <sup>\*</sup>(bojar,marie.hledikova,nedoluzhko)@ufal.mff.cuni.cz

## Abstract

In this article, we report the findings of the second shared task on **Automatic Minuting (AutoMin)** held as a Generation Challenge at the 16th International Natural Language Generation (INLG) Conference 2023. The second *Automatic Minuting* shared task is a successor to the first *AutoMin* which took place in 2021. The primary objective of the AutoMin shared task is to garner participation of the speech and natural language processing and generation community to create automatic methods for generating minutes from multi-party meetings. Five teams from diverse backgrounds participated in the shared task this year. A lot has changed in the Generative AI landscape since the last *AutoMin* especially with the emergence and wide adoption of Large Language Models (LLMs) to different downstream tasks. Most of the contributions are based on some form of an LLM and we are also adding current outputs of GPT-4 as a benchmark. Furthermore, we examine the applicability of GPT-4 for *automatic scoring* of minutes. Compared to the previous instance of AutoMin, we also add another domain, the minutes for EU Parliament sessions, and we experiment with a more fine-grained manual evaluation. More details on the event can be found at <https://ufal.github.io/automin-2023/>.

## 1 Introduction

*Automatic Minuting* (Shinde et al., 2022) refers to the task of automatically creating meeting minutes from multi-party meeting conversations. Since the pandemic, a significant portion of the global workforce, especially those in Information Technology (IT) and IT-enabled services, has gone virtual or preferring a hybrid mode of work.<sup>1</sup>

<sup>1</sup>96% of U.S. employees surveyed preferred to work on a hybrid basis as per <https://www.forbes.com/sites/edwardsegal/2021/09/20/26-of-surveyed-employees-dont-plan-to-work-onsite-again-most-still-prefer-hybrid-arrangements/>

Meetings have always been important to ensure smooth coordination and success of projects, but the proportion of sessions which combine remote and onsite workforce and collaboration between geographically distant members has grown manifold. Undeniably, frequent meetings create significant cognitive workload on people. To document the discussions, updates, tasks planned, *minuting* is an essential activity in meetings (be it online, in-person, or hybrid). Usually minutes are jotted down by some member(s) in the meeting but due to the lack of any fixed standards for minuting, different minute-takers may have different perspectives depending on their background. Despite that minutes of the same meeting by different authors may differ in certain aspects and content (Ghosal et al., 2022c), this activity can be automated to some extent.

There has been a body of research in this topic since the AMI (Mccowan et al., 2005), ICSI (Janin et al., 2003) and *Calo* Meeting Assistant (Voss and Ehlen, 2007) projects. Some interesting recent works on meeting and dialogue summarization include those by Zhu et al. (2020); Feng et al. (2021); Zhong et al. (2022); Prasad et al. (2023). We summarize our efforts on *Automatic Minuting* in Ghosal et al. (2022a); Singh et al. (2022, 2021). From the previous AutoMin shared task (Ghosal et al., 2021a), top-performing systems from Shinde et al. (2021); Yamaguchi et al. (2021) showed the usability of a BART-based (Lewis et al., 2020) system trained on SAMSum corpus (Gliwa et al., 2019) for the task. An exhaustive survey of abstractive meeting summarization research could be found in Rennard et al. (2022); Yang and Zhu (2023); Kumar and Kabiri (2022).

For long, resource creation for meeting summarization was difficult because of privacy reasons (AMI and ICSI were the only publicly available ones and later we introduced ELITR Minuting Corpus (Nedoluzhko et al., 2022)). However

quite recently, we see few datasets are made available to support research in this topic, incl. e.g. Tardy et al. (2020); Zhong et al. (2021); Kim et al. (2023); Hu et al. (2023); Chen et al. (2021).

Recently, with the “generative AI revolution”, pre-trained large language models (LLMs; Brown et al., 2020; Touvron et al., 2023; Devlin et al., 2019) have been employed for this task (Yang et al., 2023; Sándor, 2023; Chen et al., 2023), demonstrating amazing output quality. Based on the general public sentiment about the capabilities of LLMs, one could assume that *automatic minuting* belongs to one of the tasks that have suddenly become essentially solved. To verify the status in a rigorous way and to search for any open challenges that need to be addressed and also to assess how far we evolved since the last AutoMin (Ghosal et al., 2021b), we continued with the second iteration of the *AutoMin* shared task. A related effort along this direction was the DialogSum Generation Challenge (Chen et al., 2022; Bhattacharjee et al., 2022) at INLG 2022.

We proposed the second iteration of the *AutoMin* shared task as a *Generation Challenge* for INLG 2023 (Ghosal et al., 2022b). Essentially, with the current iteration of *AutoMin*, we wanted to find out:

- What are the current state-of-the-art approaches to minuting?
- What role LLMs play in these approaches; what benefits and risks they bring?
- Can we refine our manual evaluation of candidate minutes so that we have more reliable scoring techniques?
- What are the differences between different minuting domains? In addition to the same style of “project meetings” as used in AutoMin 2021, we included EU Parliament sessions in the task this year.

We describe our shared task and present our findings in the remainder of the paper.

## 2 Tasks Description

We offered four tasks (Task A, Task B, Task C, and Task D) to AutoMin participants (Ghosal et al., 2022b). Tasks A–C were known from the previous AutoMin instance, Task D was new and focused on evaluation of minutes.

In the end, all the teams decided to take part only in the first and most important task, namely minuting from diarized transcript (Task A). To compensate for the lack of participation in Task D, we experimented with automatic evaluation using LLMs, see Section 6.3.

### 2.1 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.

Task A was run in two domains. In English and Czech, we again relied on the meetings in the ELITR Minuting Corpus 1.0 (Nedoluzhko et al., 2022) but created a new test set for 2023 (meeting transcripts which were not previously published). For English, we added EuroParlMin 1.0,<sup>2</sup> a new resource we curated from the European parliamentary sessions, see Section 3 below.

Participants were free to submit their minutes for any selection of these test sets.

Note that the nature of meetings as well as the reference minutes are very different in the two datasets (technical project meetings vs. parliamentary sessions).

### 2.2 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.

### 2.3 Task C

Task C is a variation of Task B. *Given a pair of minutes, the task is to identify whether the two*

<sup>2</sup><https://github.com/ufal/europarlmin>

	Lines	Words
ELITR Minuting Corpus		
Transcript (EN)	728.3 ± 389.9	7078.9 ± 3741.6
Minutes (EN)	45.8 ± 31.5	395.6 ± 388.5
EuroParlMin		
Transcript (EN)	227.2 ± 257.0	8138.5 ± 10460.7
Minutes (EN)	48.6 ± 87.6	278.8 ± 534.2

Table 1: Summary across all data (training, development and test sets) used for AutoMin 2023. The figures correspond to mean±standard deviation.

*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.

## 2.4 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 could focus on diverse aspects of minutes quality, such as the coverage of content discussed, the adequacy of the description, the readability, etc.

The original plan was to 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.

## 3 Dataset Description

The datasets for AutoMin 2023 cover three types of data: project meetings in both English and Czech, as well as parliamentary sessions in English.

Basic statistics of the data are in Table 1.

The project meeting data was prepared from our own sources, while the parliamentary sessions were taken from the European Parliament mostly as-is, we merely selected data which was useful for our purposes.

### 3.1 Project Meeting Data

For the project meeting datasets, the participants were advised to use ELITR Minuting Corpus (Nedoluzhko et al., 2022) as training data, with the option to use any other training data of their

	Train	Dev	Test-I	Test-II	Test-2023
ELMI EN	85	10	18	10	12
ELMI CS	33	10	10	6	11
EuroParl	2065	187	–	–	242

Table 2: Task A meeting counts used this year. ELMI stands for ELITR Minuting Corpus.

choice. We prepared new testsets for both languages, containing 12 new meetings for English and 11 new meetings for Czech. This new test set is again from technical project meetings and city planning meetings, the same as ELITR Minuting Corpus and the test set used in AutoMin 2021. The format is also identical.

Table 2 shows our *train-dev-test* splits for Task A. Test-I and Test-II were made public already in 2021, Test-2023 reference minutes were not made available to anyone before the shared task was over.

The data preparation was completed in the following steps (same as in the past):

1. We obtained raw audio recordings of meetings and preliminary consent from their participants to process the data and publish it in a deidentified form.
2. The recordings were automatically transcribed using our ASR systems.
3. Our team of annotators was provided with the audio recordings and the automatic transcripts and was tasked with correcting the transcripts so as not to contain any mistakes. The next task was to break the transcript down into segments of similar length and to add speaker tags. The segments are approximately correspondent to sentences, although sentence boundaries are not always clearly defined in speech. Speaker tags are given at the beginning of each speaker’s section in round brackets.
4. The same annotator who prepared the transcript was then asked to create reference minutes for the meeting.
5. One of the main concerns in our data preparation is privacy. Publicly released data must adhere to EU GDPR standards. Therefore in the next step, the annotators had to deidentify all personal identifiable information. This was mainly the names of per-

sons, projects and organizations. Each instance was replaced with a tag in the format of [PERSON*number*], [ORGANIZATION*number*] or [PROJECT*number*]. The speaker tags are deidentified as well, in the format (PERSON*number*). The numbers are consistent for each person, project and organization within one meeting, but are purposefully shuffled between different meetings, even if some of the participants are the same. Annotators also removed any potentially sensitive or offensive utterances completely and replaced them with a <censored/> tag.

6. The final step was to create alignments between the transcripts and their respective reference minutes. This is an annotation we were hoping would be useful especially during the evaluation process. The alignment maps utterances from the transcript onto the minutes line which summarizes them (if any), and/or assigns a remark for why the utterance is not suitable to be in minutes or otherwise problematic (e.g. small-talk or unintelligible). The utterances are aligned as to map whole conversations about a topic onto the appropriate minutes line, not just the part where the topic is introduced. Each utterance can be aligned to at most one line in the minutes. This is a simplification we chose for ease of annotation and processing, even though occasionally more lines would be more appropriate.

Generally, the whole process for a single meeting was carried out by the same annotator, but sometimes, the meeting had to be reassigned to a different annotator part-way through due to organizational complications.

A sample meeting summary in the ELITR Minuting Corpus style is provided in Appendix A.

### 3.2 Parliamentary Sessions

We created EuroParlMin 1.0 for the purposes of AutoMin 2023 and released it publicly at the beginning of the campaign.<sup>3</sup> EuroParlMin comes from the archives of the EU parliament.<sup>4</sup> We downloaded the transcripts and minutes and converted them to plaintext. Only very little text processing beyond dropping XML tags was done.

<sup>3</sup><https://github.com/ufal/europarlmin>

<sup>4</sup><https://emeeting.europarl.europa.eu/emeeting/committee/en/archives>

For the purposes of AutoMin, it was necessary to select sessions with desirable properties. Some of the sessions in the original dataset had little content. For some others, the provided minutes contained a large amount of extra text not related to the transcript. We therefore filtered for sessions with (a) sufficiently long transcripts, and (b) a good compression rate from transcript to minutes. We then split this selection into training, development and test sets (generally choosing the sessions best in (a) and (b) for the test set) and provided the training and development sets publicly. Privacy was not a concern, since the data is public, so there was no need for de-identification, making this our only data which had real names.

It is important to mention that the reference minutes in EuroParlMin are the original texts as provided by the parliament. As such, they often lack the actual content or more details about the decisions met in the meetings and only focus on formalities like aspects and voting.

Also note in Table 1 the big standard deviations in EuroParlMin data and the low line count in EuroParlMin transcripts, despite comparable word count to ELITR Minuting Corpus. This documents the domain difference where EuroParlMin contains long and often prepared speeches whereas ELITR Minuting Corpus is much more interactive.

## 4 Shared Task Timeline

The second AutoMin followed this timeline:

- ELITR Minuting Corpus Training Data Available: well before
- EuroParlMin Training Data Available: March 3, 2023
- Test Data Release: March 3, 2023
- System Output Submission Deadline: May 1, 2023
- System Report Due: May 15, 2023
- Review Notification: July 7, 2023
- Camera-Ready for Reports: July 25, 2023
- Event Date: September 11-15, 2023

Registered participants were invited to access our private Github repository to access the test sets.

System outputs as well as system reports were submitted by e-mail to the organizers.

## 5 Evaluated Systems

We evaluate submissions from the participants, baselines we obtained using LLMs and also the reference minutes.

Kindly refer to Appendix B for samples of the automatically created minutes.

### 5.1 Participating Teams

Of the 10 teams who registered for AutoMin, 5 teams eventually took part in the shared task. We had participating teams from academia as well as industry.

We briefly discuss the approaches of our participating teams (ordered alphabetically):

- Team **Darbarer** (Rousseau et al., 2023) adopted a modular strategy (four modules) for the automatic minuting task. According to them, since each participant in a meeting communicates differently than others, they first use a text simplification model, *mBarthez* by Kamal Eddine et al. (2021) to standardize the utterances in the conversation and compress the input to focus on informative content. In the next module, they first do linear segmentation of the transcript followed by using a BART-model (Lewis et al., 2020) trained on the XSum (Narayan et al., 2018) and SAMSum (Gliwa et al., 2019) datasets for summarization. In the next step, they propose a titling module to add a short description for each summarized block. Finally in the post-processing module, they employ some rule-based heuristics to improve the readability of the minutes. Overall, team *Darbarer* followed the similar steps as Shinde et al. (2021) and Yamaguchi et al. (2021) from the First *AutoMin* (Ghosal et al., 2021a), with an added pre-processing step of Text Simplification.
- Team **Synapse** (Klesnilová and Elizabeth, 2023) followed a similar approach: pre-process→segment→summarize→post-process. They perform brute-force segmentation of the transcripts (into pre-defined token-lengths) to manage the length of the long transcripts for the subsequent summarizer module. In the summarizer module,

they experimented with variants of BART trained on several summarization datasets: XSum, AMI, SAMSum, DialogSum (Chen et al., 2021), and CNN/DM (Nallapati et al., 2016).

- Team **Iterate** (Kmječ and Bojar, 2023) adopted an iterative approach where their summarization model is fed with a chunk of a transcript together with several previously generated minute points to both satisfy the input length constraint of Transformer models while providing the needed context for the minutes. With their “iterative” motivation to imitate human way of taking notes in a meeting (jotting minutes while keeping in mind previous points), they experimented with several models: BART, Longformer (Beltagy et al., 2020), and the large language model Llama-based Vicuna (Chiang et al., 2023). They found that even without fine-tuning, Vicuna shows promise to generate coherent minutes from zero-shot prompting.
- Team **NTR’s** (Borisov and Mikhaylovskiy, 2023) minuting pipeline consists of two stages: segmentation and summarization. They perform semantic segmentation of the meeting transcripts to assist the subsequent transformer-based summarization stage to receive the input in the desired token length range. However, they found that their semantic segmentation approach does not perform better than the naive segmentation technique. In the summarization stage, they experiment with prompting a large language model Dolly (Conover et al., 2023) and found comparable performance with their mBART (Liu et al., 2020) + BERTopic (Grootendorst, 2022) method.
- Team **Zoom** (Schneider and Turchi, 2023) used Content Vector Segmentation (CVS) (Alemi and Ginsparg, 2015) to segment the meeting transcripts. They used GPT-3’s *text-da-vinci* model to generate additional data for training. Finally, they employed a `bart_large` model trained on XSum and SAMSum datasets for summarizing the meeting conversations. The authors claim that CVS significantly improved the downstream minuting task as opposed to using length-based segmentation.

## 5.2 GPT-4 and GPT-3 Baselines

In addition to submissions by participating teams, we decided to also evaluate outputs obtained using large language models, as we thought this could yield useful comparison and insight as to the usability of LLMs in the meeting minuting task. We used both OpenAI’s GPT-4 and text-davinci-003 (sometimes referred to as GPT-3).<sup>5</sup> See Appendix C for sample GPT-4 and GPT-3 outputs.

Our transcripts are significantly longer than the maximum allowed length of prompts, we therefore had to split each one into smaller sections, have the LLMs process these sections separately, and then simply concatenate the results. We have not used GPT4-32k for this task.

For English project meetings, the prompt we used was “Summarize the following project meeting in the form of 5 - 10 bullet points: <meeting transcript section>”. This prompt yielded results in a format very close to what we were looking for.

Similarly, for European Parliament data, we used the prompt “Summarize the following meeting in the form of 5 - 10 bullet points: <meeting transcript section>”.

For Czech project meetings, however, we had to make a data-specific adjustment to ensure that GPT would keep our deidentification tags intact (i.e. wouldn’t translate e.g. “PERSON4” into Czech). The prompt we used was therefore “The following project meeting contains identifiers in the format PERSONnumber. Summarize the meeting in 5 - 10 bullet points in Czech, keeping the original identifiers: <meeting transcript section>”.

## 6 Evaluation

In this section, we describe the evaluation process carried out for AutoMin 2023. We used human evaluations, traditional automatic evaluations as well as evaluations using LLMs.

### 6.1 Manual Evaluation

We had our annotators carry out two kinds of evaluation – one at the document level and another alignment-based one at hunk level using our ALIGNMEET tool (Polák et al., 2022). All human evaluation was done using Likert scales from 1 (worst) to 5 (best). The submissions were

<sup>5</sup><https://platform.openai.com/docs/model-index-for-researchers>

	Dist. Ann.	Judged	Doc-l.	Hunk-l.
Pr. Data (EN)	3	9	81	2048
Pr. Data (CS)	2	4	16	920
EuroParlMin	1	6	36	98
Total	3	19	133	3066

Table 3: Numbers of distinct annotators, total judged meetings and individual judgements by dataset. Doc-level judgements are one per meeting-submission (or reference) pair. Hunk-level judgements are the overall number of all evaluated minutes items.

anonymized so that annotators did not know which team’s submission they were scoring, however, due to some obvious differences in the format, they were probably able to associate minutes by the same team across meetings.

See Appendix E for the full annotation instructions.

Document level evaluation assigned the following four scores to each candidate minute:

1. **Adequacy** assesses if the minutes adequately capture the major topics discussed in the meeting, also considering coverage (all such topics covered).
2. **Fluency** reflects if the minutes consist of fluent, coherent texts and are readable to the evaluator.
3. **Grammatical Correctness** checks the level to which the minutes are grammatically correct.
4. **Relevance** signifies the extent to which the minutes overall capture the important content from the source transcript (as opposed to summarizing useless parts).

The hunk-level evaluation was carried out in two steps: alignment and evaluation. In the alignment step, the annotator constructed a transcript to candidate minute alignment, as described in Section 3. Then, four scores were assigned to each so-called hunk, i.e. a section of the transcript aligned to a single minute line.

1. **Adequacy** assesses if the item in the minute adequately captures the content of the respective aligned segment(s) of the transcript.
2. **Fluency** reflects if the item in the minute consists of fluent, coherent text and is readable to the evaluator.

	Document-level	Hunk-level
Grammaticality	4.53±0.76	4.76±0.44
Relevance	4.19±0.69	4.40±0.74
Fluency	3.80±0.74	4.06±0.87
Adequacy	3.47±0.84	4.04±0.84

Table 4: Averages of scores that individual systems reached in the manual evaluation at document and hunk level. Sorted by decreasing overall score.

3. **Grammatical Correctness** checks the level to which the item in the minute is free of grammatical errors.
4. **Relevance** signifies whether the item in the minute is relevant to be included in the meeting summary (e.g. a perfect summary of small talk is not relevant).

Compared to AutoMin 2021, we added the *Relevance* score. Our hope was that the scores would better reflect the quality of the overall structure of the minute and topic selection. This turned out to not necessarily be the case in the actual scores, see below for further discussion.

Due to time constraints and the relative complexity of the evaluation, only a subset of each test set was manually evaluated, see Table 3. The most reliable part of our evaluation therefore lies in ELITR Minuting Corpus in English. The Czech part of ELITR Minuting Corpus has a rather low number of document-level judgements and the Europarl domain has a relatively low number of hunk-level judgements.

Kindly note that the human evaluation was *reference-less*. In other words, our evaluators had access to only the transcript of the meeting to evaluate the candidate minutes (participant submissions, our baselines, and also the reference itself). We did this on purpose to avoid the bias of human annotators towards the reference and also to have the chance to evaluate the reference in the same conditions as the other systems.

Manual evaluation serves as the official scoring for AutoMin 2023 but we highlight that we do not focus on the ranking of the systems but rather on the takeaways from the best as well as the relatively poorer system outputs.

### 6.1.1 Comments on Human Evaluations

As we can observe in Table 4 and in detail in Table 6 below, annotators gave relatively high scores

to most minute outputs. Especially Fluency and Grammatical Correctness scores are high, the average of averaged scores is over 4 for most settings except document-level Adequacy (average of 3.47±0.84) and Fluency (3.80±0.74). This scoring reflects the fact that the system outputs look very natural and fluent, almost indistinguishable from the human minutes from the linguistic point of view. At the same time, the high superficial quality can pose a challenge to the annotators; their attention can decrease and errors can go unnoticed.

Although human evaluations have been provided by experienced annotators, a number of discrepancies were observed. Many of them are about ‘incorrect’ judgements or inattention, which is natural given to the content of the minutes outputs. Minute items generated by systems are sometimes very close to what had been discussed in the meeting but still do not reflect the actual content. For example, two participants discussed their computer science courses, the possibility of failing a course being mentioned several times. A system outputs a minute saying that “they both failed their courses”. This is not true but annotators did not notice and estimated that this information was correct.

Furthermore, we often observe that automatically generated minutes do not ‘have a good sense’ of the relevance of the discussion parts, which may also remain unnoticed by annotators. This may be exemplified on the same meeting topic (computer science courses) which had been summarized in much redundant detail by one of the systems, and evaluated as fully correct by the annotator. In reality, this discussion is not relevant for the meeting at all, see Figure 1.

## 6.2 Automatic Evaluation using Standard Metrics

For our automatic evaluation of Task A, we relied on the widely popular text summarization metric ROUGE (Lin, 2004) in its three variants: ROUGE-1, ROUGE-2, ROUGE-L, and we also added BART and BERT-based evaluations.

### 6.2.1 ROUGE Variants

ROUGE metrics are based on n-gram similarities with a given reference. ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. It works by comparing an automatically produced summary against a reference summary (usually



Minutes	Adq.	Gram.	Fluen.	Relev.
1	5	5	5	5
2	5	5	5	5
3	5	5	5	5
4	4	5	5	4
5	5	5	5	5
6	5	5	5	5
7	3	5	5	3
8	5	5	5	5
9	5	5	5	5
10	5	5	5	5
11	5	5	5	5
12	5	5	5	5
13	5	5	5	5
14	5	5	5	5
15	5	5	5	5

Figure 1: ALIGNMEET Evaluation interface with hunk-level scores. The left-hand side contains the transcript. The right-hand one contains one of the candidate minutes with each minute item asking for the four manual scores. (The document-level scores were reported at the end of candidate minutes.) The color background indicates the alignment between portions of the transcript and lines in minutes.

generated by a human). Different references thus inevitably lead to different ROUGE scores against each of them.

*Recall in the context of ROUGE* reflects how much of the reference summary the candidate summary is recovering or capturing:

$$\text{ROUGE}_{\text{Recall}} = \frac{\# \text{ Overlapping n-grams}}{\text{Total n-grams in Reference Summary}} \quad (1)$$

*Precision in the context of ROUGE* means how much of the candidate summary was in fact relevant or needed:

$$\text{ROUGE}_{\text{Precision}} = \frac{\# \text{ Overlapping n-grams}}{\text{Total n-grams in Candidate Summary}} \quad (2)$$

Despite the name (“Recall-Oriented...”), ROUGE actually commonly combines recall and precision using the harmonic mean to F-score. In our evaluation, we use ROUGE F1 scores for all ROUGE variants.

ROUGE-1 refers to the overlap of unigrams, ROUGE-2 is the overlap of bigrams, and ROUGE-L measures the longest matching sequence of words using Longest Common Subsequence (LCS).

As we mentioned earlier, proper evaluation metrics for meeting summarization are severely needed (Ghosal et al., 2021c), and text summarization metrics like ROUGE are only a poor alternative.

## 6.2.2 BERTScore

BERTScore (Zhang\* et al., 2020) is an embedding-based metric that uses cosine similarity to compare each token or n-gram in the generated output with the reference sentence. There are three components to BERTScore:

- **Recall:** Average cosine similarity between each token in the reference and its closest match in the generated output.
- **Precision:** Average cosine similarity between each token in the generated output and its nearest match in the reference.
- **F1:** Harmonic mean of recall and precision

BERTScore is useful because it can account for synonyms and paraphrasing. Simpler metrics like BLEU and ROUGE can’t do this due to their reliance on exact matches.

We used this<sup>6</sup> implementation of BERTScore.

## 6.2.3 BARTScore

BARTScore (Yuan et al., 2021) is another popular Natural Language Generation (NLG) metric which uses a pre-trained sequence-to-sequence model (BART in this case). The authors conceptualized the metric as *evaluation of generated text as a text generation problem* itself. The general idea is that models trained to convert the generated text to/from a reference output or the source text will

<sup>6</sup><https://pypi.org/project/bert-score/>

achieve higher scores when the generated text is better.

$$\text{BARTSCORE} = \sum_{t=1}^m \omega_t \log p(\mathbf{y}_t | \mathbf{y}_{<t}, \mathbf{x}, \theta) \quad (3)$$

where  $y$  is the generated text and  $x$  is the reference text. We use the original implementation<sup>7</sup> from the authors in terms of  $F$ -score.

### 6.3 Automatic Evaluation with LLMs

In the pursuit of Task D, we leveraged the capabilities of large language models (LLMs), particularly GPT (OpenAI, 2023), to assess the quality of system-generated meeting minutes.

The evaluation was structured around several parameters, which included the adequacy, relevance, topicality and fluency of the minutes. We instructed LLMs to rate each category for each set of minutes on a scale of 0 to 10. The prompt could be summarized as: “Given a specific meeting transcript, rate adequacy, relevance, topicality and fluency of the following system-generated minutes.” We tried two different prompt structures, differing in the exact formulations and also in the set of qualities to be scored. The prompt (i.e. effectively an “evaluation method”) called “GPT-ART” reports Adequacy, Relevance and Topicality. The prompt called “GPT-AFGR” reports Adequacy, Fluency, Grammaticality and Relevance and is thus aimed to mimic our manual criteria. The GPT-AFGR specification was also constructed to mimic annotators’ instructions (Appendix E). The full prompts are provided in Appendix F.

A significant challenge in this study was the limitation of GPT’s context window in dealing with extensive conversations. We attempted to overcome this by employing GPT-4-32k, with a context window of 32k subwords. However, this model still struggled to cover the entire conversation of several transcripts, and hence a more sophisticated approach was needed.

To counter this issue, we designed a technique that initially eliminates conversational turns that lack meaningful information. First, we removed all instances of the utterance “eh” and other non-speech items that are provided in the curated transcripts (e.g. <unintelligible>). Secondly, we asked GPT to mark each conversational turn as either meaning-bearing or filler content. The lat-

<sup>7</sup><https://github.com/neulab/BARTScore>

	Lines	Words
<b>ELMI EN</b>		
Transcripts	733.1±294.7	7198.1±2669.1
Ref. Minutes	59.8±29.6	480.3±251.3
<b>Darbarer</b>		
davinci-003	49.8±18.4	358.0±158.8
GPT-4	48.5±15.3	524.1±191.3
Team Iterate	34.3±12.7	551.1±219.9
NTR	12.0±4.9	411.7±161.2
Synapse	60.8±24.6	952.8±394.6
Zoom-long	38.2±9.9	484.9±169.7
Zoom-short	8.4±2.1	615.0±154.9
	6.2±1.3	117.2±25.7
<b>ELMI CS</b>		
Transcripts	1164.9±292.8	9897.6±2395.5
Ref. Minutes	86.0±36.9	435.6±194.6
<b>Darbarer</b>		
davinci-003	69.3±14.2	495.7±121.0
GPT-4	89.5±22.2	905.6±247.5
EuroParlMin	96.0±20.1	1166.3±262.7
<b>Transcripts</b>		
Transcripts	33.2±59.2	873.8±2068.7
Ref. Minutes	37.7±59.3	185.0±323.6
<b>Darbarer</b>		
NTR	9.2±6.2	65.0±72.1
Synapse	8.1±12.0	155.1±310.8
davinci-003	13.0±10.4	123.6±194.5
GPT-4	6.7±2.4	111.0±51.9
	5.0±0.6	83.7±26.3

Table 5: Basic properties of manual transcripts, reference minutes and all participating team submissions of test set meetings. We report the average±standard deviation values for the number of lines and words.

ter category referred to those segments that were merely iterations and could be excluded without the loss of any information. You can see the prompt in the Appendix F. For this filtering task, we used GPT “text-davinci-003” model, mainly due to the throttling limitations.

After this initial filtration process, we kept only the speaker identification from the filler turns. This methodology substantially reduced the length of the transcripts and their respective summaries, enabling them to fit within the context window of GPT-4-32k. This provided an effective solution to our problem and allowed for LLM-based evaluation of the system-generated meeting minutes.

### 6.4 Basic Statistics

We report basic test set statistics in Table 5: the average number of lines and words in each transcript, and reference minutes, as well as for the participant submissions (candidate minutes). This provides a first useful comparison of the participant minutes with respect to the reference minutes and transcripts.

Compared to the last year, there are no extreme outliers in terms of the numbers of lines (typically corresponding to summary points) across the sub-

	D Adeq	D Fluency	D Gram	D Relev	H Adeq	H Fluency	H Gram	H Relev
<b>ELMI CS</b>								
GPT-4	<b>5.00±0.00</b>	<b>5.00±0.00</b>	<b>4.50±0.71</b>	<b>4.50±0.71</b>	<b>4.84±0.56</b>	<b>4.84±0.56</b>	<b>4.77±0.57</b>	<b>4.68±0.56</b>
Reference	4.00±1.41	4.50±0.71	3.50±2.12	<b>4.50±0.71</b>	<b>4.81±0.59</b>	<b>4.81±0.59</b>	<b>4.36±0.87</b>	<b>4.34±0.84</b>
davinci-003	3.50±0.71	4.00±0.00	<b>4.50±0.71</b>	<b>4.00±1.41</b>	<b>4.71±0.73</b>	<b>4.71±0.73</b>	<b>4.54±0.74</b>	<b>4.44±0.85</b>
Darbarer	2.50±0.71	3.00±0.00	2.00±0.00	3.00±0.00	3.33±1.20	3.33±1.20	3.13±1.22	2.91±1.20
<b>ELMI EN</b>								
Zoom-long	<b>4.61±0.49</b>	<b>4.72±0.38</b>	4.81±0.35	4.47±0.48	<b>4.72±0.44</b>	<b>4.78±0.50</b>	4.85±0.34	<b>4.60±0.47</b>
GPT-4	<b>4.58±0.66</b>	<b>4.78±0.34</b>	<b>5.00±0.00</b>	<b>5.00±0.00</b>	<b>4.69±0.75</b>	<b>4.67±0.86</b>	<b>4.93±0.51</b>	<b>4.93±0.51</b>
Team Iterate	4.06±1.01	4.31±0.70	4.89±0.18	4.58±0.53	<b>4.46±0.83</b>	<b>4.67±0.69</b>	<b>4.96±0.17</b>	<b>4.80±0.45</b>
Reference	3.97±0.63	4.11±0.78	<b>4.97±0.08</b>	<b>4.72±0.44</b>	<b>4.60±0.70</b>	<b>4.64±0.68</b>	<b>4.98±0.24</b>	<b>4.89±0.38</b>
davinci-003	3.78±0.75	3.94±0.83	4.94±0.12	4.47±0.74	<b>4.50±0.80</b>	<b>4.49±0.94</b>	<b>4.97±0.33</b>	<b>4.75±0.71</b>
Synapse	3.50±0.48	3.61±0.73	4.69±0.43	4.11±0.92	<b>4.38±0.90</b>	<b>4.42±0.91</b>	<b>4.92±0.38</b>	<b>4.78±0.62</b>
Zoom-short	3.25±0.73	<b>3.64±0.45</b>	<b>4.97±0.08</b>	<b>4.69±0.66</b>	4.02±0.94	4.19±0.91	<b>4.99±0.07</b>	<b>4.82±0.47</b>
Darbarer	3.14±0.60	3.64±0.42	4.92±0.18	4.67±0.67	<b>4.03±1.11</b>	4.17±1.11	<b>4.93±0.41</b>	<b>4.76±0.71</b>
NTR	2.94±0.99	3.00±1.07	4.58±0.68	3.44±1.02	4.01±1.27	3.93±1.35	4.84±0.50	4.35±1.15
<b>EuroParlMin</b>								
Synapse	<b>3.17±1.17</b>	<b>3.33±0.52</b>	<b>5.00±0.00</b>	4.17±0.98	3.43±1.07	3.43±1.07	<b>5.00±0.00</b>	4.36±0.78
NTR	<b>2.67±1.03</b>	<b>3.33±0.82</b>	4.50±1.22	3.50±1.05	<b>3.45±1.26</b>	3.27±1.32	4.86±0.64	3.50±1.06
Darbarer	<b>2.33±1.21</b>	<b>3.50±1.05</b>	<b>5.00±0.00</b>	<b>4.83±0.41</b>	<b>4.44±0.81</b>	<b>4.44±0.81</b>	<b>5.00±0.00</b>	<b>4.94±0.25</b>
Reference	<b>2.00±0.63</b>	2.17±0.75	4.17±0.75	2.50±1.38	2.30±1.12	2.20±1.10	4.60±0.77	2.33±1.35
davinci-003	–	–	–	–	2.00±0.00	2.00±0.00	<b>5.00±0.00</b>	<b>5.00±0.00</b>

Table 6: Manual evaluation results: “D” means document-level, “H” means hunk-level. We report the average  $\pm$  standard deviation. Sorted by decreasing quality according to document-level adequacy. The symbol “?” highlights a disruption in the ordering in the given column. Due to an annotator’s error, davinci-003 did not have the document-level scores provided. The top score and all scores that fall within its std. dev. bounds are in **bold**.

missions, although the variance across teams remains high, e.g. with  $60.8 \pm 24.6$  lines by NTR compared to  $6.2 \pm 1.3$  in the Zoom short submission on the ELMI EN test set.

The longest lines are, as expected, produced by Zoom-long (only available for ELMI EN), with the average of 73.2 words per line, followed by Team Iterate (34.3 words per line) and Zoom-short (18.9 words/line). GPT-4 comes next with 16.1 words/line on the ELMI EN domain.

We find it noteworthy that the reference minutes across all the domains have the fewest words per line an average (8.0 for ELMI EN, 5.1 for ELMI CS and 4.9 for EP). Only the Darbarer submission is at this level of line conciseness, producing even a little shorter lines than the reference on ELMI EN (7.2 vs 8.0).

## 6.5 Manual Evaluation Results

Table 6 presents average scores that individual systems received in the document as well as hunk-level manual annotations in each of the examined test sets.

As we see from the standard deviations, some evaluation settings are not discerning enough and any conclusions drawn from such areas should thus be treated with a big caution. This concerns

primarily hunk-level judgements in project meetings (ELMI CS as well as ELMI EN), and also doc-level EuroParlMin.

In the document-level evaluations of project meeting minutes, we see that GPT-4 and also Zoom-long summaries achieve top scores. In addition to these, also Team Iterate scored better than the human reference. A detailed analysis of this result is desirable, to identify in which stage the human processing was sloppier than the automatic summary.

For the EuroParlMin, we attribute the worse score of human reference to the different style, not really appreciated by our evaluators. A similar situation is probably the case of Zoom-long vs. Zoom-short, where the annotators clearly prefer longer minutes.

The dominance of GPT-4 is apparent in terms of both phrasing (Fluency and Grammaticality) as well as content (Adequacy and Relevance) measures.

## 6.6 Automatic Evaluation Results

For automatic evaluation, we took the usual text summarization metric ROUGE (Lin, 2004) in its three variants (ROUGE-1, ROUGE-2, and ROUGE-L).

	ROUGE-1	ROUGE-2	ROUGE-L	BART-F1	BERT-F1
<b>ELMI CS</b>					
davinci-003	<b>0.33±0.16</b>	<b>0.09±0.04</b>	<b>0.13±0.06</b>	<b>-4.34±0.45</b>	<b>0.58±0.07</b>
Darbarer	<b>0.31±0.12</b>	<b>0.05±0.03</b>	<b>0.12±0.04</b>	<b>-4.55±0.38</b>	<b>0.59±0.03</b>
GPT-4	<b>0.30±0.15</b>	<b>0.08±0.05</b>	<b>0.12±0.06</b>	<b>-4.33±0.43</b>	<b>0.58±0.05</b>
<b>ELMI EN</b>					
GPT-4	<b>0.44±0.07</b>	<b>0.10±0.04</b>	<b>0.20±0.03</b>	<b>-4.40±0.42</b>	<b>0.60±0.03</b>
Synapse	<b>0.43±0.06</b>	<b>0.11±0.04</b>	<b>0.20±0.02</b>	<b>-4.56±0.42</b>	<b>0.59±0.03</b>
text-davinci-003	<b>0.41±0.07</b>	<b>0.10±0.02</b>	<b>0.19±0.02</b>	<b>-4.58±0.40</b>	0.55±0.01
Zoom-long	<b>0.41±0.09</b>	<b>0.10±0.03</b>	<b>0.18±0.02</b>	<b>-4.58±0.41</b>	0.55±0.02
davinci-003	<b>0.40±0.07</b>	<b>0.09±0.03</b>	<b>0.17±0.03</b>	<b>-4.45±0.42</b>	<b>0.58±0.02</b>
Team Iterate	<b>0.40±0.08</b>	<b>0.10±0.03</b>	<b>0.19±0.03</b>	<b>-4.63±0.42</b>	0.55±0.02
Darbarer	<b>0.39±0.06</b>	<b>0.10±0.03</b>	<b>0.19±0.03</b>	<b>-4.68±0.35</b>	<b>0.59±0.02</b>
NTR	<b>0.37±0.10</b>	<b>0.09±0.04</b>	0.16±0.03	<b>-4.66±0.49</b>	0.55±0.03
Zoom-short	0.29±0.08	0.06±0.02	0.15±0.04	<b>-4.82±0.37</b>	0.52±0.02
<b>EuroParlMin</b>					
Darbarer	<b>0.27±0.10</b>	<b>0.11±0.08</b>	<b>0.18±0.08</b>	-5.06±0.39	0.38±0.03
NTR	<b>0.27±0.11</b>	<b>0.09±0.07</b>	<b>0.17±0.07</b>	-5.23±0.39	0.29±0.02
Synapse	<b>0.26±0.10</b>	<b>0.08±0.07</b>	<b>0.16±0.08</b>	<b>-4.67±0.35</b>	<b>0.42±0.03</b>
davinci-003	<b>0.21±0.09</b>	<b>0.04±0.05</b>	<b>0.14±0.06</b>	-5.18±0.40	0.28±0.02
GPT-4	<b>0.20±0.09</b>	<b>0.04±0.05</b>	<b>0.13±0.06</b>	-5.22±0.40	0.29±0.03

Table 7: Automatic evaluation results using ROUGE, BART and BERT. We report the average  $\pm$  standard deviation. Sorted by decreasing quality according to ROUGE-1. The symbol “?” highlights a disruption in the ordering in the given column. The top score and all scores that fall within its std. dev. bounds are in **bold**.

Table 7 summarizes the ROUGE, BERT and BART scores across all our test sets.

For Czech minutes and BERT and BART, we first translate both the minutes by all participants and also the reference minutes into English using Lindat Translation.<sup>8</sup>

Best scores are in bold, again with all other scores that fall within the std. dev. band of the best one.

The automatic analysis using standard measures suffers even more from low statistical power. For the next year, we should clearly substantially increase the test set size, to gather the weak signal more reliably.

davinci-003 and text-davinci-003 are two variants of GPT-3 output. One of them was obtained by us, as discussed in Section 5.2, the other one was provided by Zoom. We did not score these two variants with manual or expensive automatic methods, so we only have ROUGE, BART and BERT to assess the effect of different (uncontrolled) prompt for the task. The comparison of these two outputs is mixed across the measures, and we would not overestimate the true value of the BERT-F1 difference where our prompt seems to win.

Synapse outputs on EuroParlMin stand out in

<sup>8</sup><https://lindat.mff.cuni.cz/services/translation/>

BART and BERT scoring as the only system across the board. Rather likely, the pre- and post-processing heuristics were a good fit for the BART and BERT scoring.

## 6.7 LLM-Based Evaluation Results

Table 8 presents the scores that we obtained from our automatic scoring from GPT-4, as described in Section 6.3. We report the average and standard deviation across all the English meetings in the 2023 test set for ELITR Minuting Corpus. Note that not all systems were scored this way to fit the budget.

It is immediately apparent that GPT scorers prefer GPT produced outputs (GPT-4 and davinci-003), but GPT-4 scored exceptionally well also in the manual evaluation, so this cannot be taken as any bad sign.

What we see more as problematic is that there are only very few differences between the qualities that were supposed to be assessed. The rankings of Fluency, Grammaticality or Relevance according to GPT-AFGR are very much in line with its Adequacy (along which the table is sorted), and also the other prompt (GPT-ART) does not bring much difference. This is in some contrast with the manual document level judgements where Grammaticality and Relevance are not always in line with Adequacy.

ELMI EN	GPT-AFRG				GPT-ART		
	Adeq.	Flu.	Gram.	Relev.	Adeq.	Relev.	Topic.
GPT-4	<b>8.75±0.45</b>	<b>8.83±0.39</b>	<b>9.00±0.00</b>	<b>8.75±0.45</b>	<b>8.17±0.39</b>	<b>9.00±0.00</b>	<b>8.75±0.45</b>
davinci-003	8.00±0.85	<b>8.58±0.67</b>	8.83±0.58	8.00±0.85	7.67±0.65	8.58±0.67	8.00±0.95
Zoom-long	7.83±0.39	8.42±0.51	8.75±0.45	7.83±0.39	7.50±0.67	8.17±0.58	7.50±0.67
Darbarer	7.58±0.67	<b>8.50±0.67</b>	8.83±0.39	7.58±0.67	7.08±0.79	7.92±0.67	7.17±0.94
Synapse	7.42±0.90	8.25±0.75	8.58±0.67	7.42±0.79	7.42±0.90	8.17±0.83	7.67±0.78
NTR	7.08±0.90	7.83±0.72	8.08±0.67	7.25±1.14	6.75±1.14	7.75±1.14	6.83±1.19
Team Iterate	6.58±1.38	7.67±0.98	8.17±0.72	6.75±1.22	6.75±1.06	7.67±0.98	6.83±1.11

Table 8: Automatic evaluation results using GPT with two types of prompt: AFGR and ART. We report the average  $\pm$  standard deviation. Sorted by decreasing Adequacy estimated by GPT-AFRG The symbol “?” highlights a disruption in the ordering in the given column. The top score and all scores that fall within its std. dev. bounds are in **bold**.

One striking outlier is Team Iterate which ended up third in manual evaluation but appears last according to GPT scoring.

## 7 Meta-Analysis of Automatic Evaluation

This section presents a comprehensive meta-analysis of the automatic evaluation in regard to human evaluation. The goal of this meta-analysis is to assess the usability of various automatic approaches, be it BART-score, BERT-score, variants of ROUGE, or GPT-based evaluation introduced in Section 6.3.

### 7.1 Pairwise Accuracy

Building on the methodologies used in the area of machine translation (Kocmi et al., 2021; Freitag et al., 2022), we use pairwise accuracy to explore how well automatic metrics align with human judgement. Pairwise accuracy is a simplification of Kendall’s Tau.

In our setting, we focus on the system-level evaluation, where we first aggregate a score for each system (team submission) by averaging scores from all meetings. As the main unit, we use the difference in the score between a pair of systems:

$$\Delta = \text{score}(\text{System A}) - \text{score}(\text{System B}) \quad (4)$$

We define the pairwise accuracy as follows. For each system pair, we calculate the difference of the metric scores ( $\text{metric}\Delta$ ) and the difference in average human judgements ( $\text{human}\Delta$ ). We calculate accuracy for a given metric as the number of rank agreements between metric and human deltas

divided by the total number of comparisons:

$$\text{Accuracy} = \frac{|\text{sign}(\text{metric}\Delta) = \text{sign}(\text{human}\Delta)|}{|\text{all system pairs}|} \quad (5)$$

In other words, our pairwise accuracy reflects how often, across all pairwise comparisons, human ordering of the pair agrees with automatic ordering of the pair.

One of the advantages of pairwise accuracy resides in its interpretability. It demonstrates with what precision a given automatic evaluation can rank pairs of systems. The measure of accuracy is intuitively presented, where a 50% accuracy is equal to the flip of a coin. This provides insights into the potential improvements necessary in automated evaluation methods to make them correlate better with human evaluation, thus moving towards more reliable and accurate automatic minuting systems.

The results of the pairwise evaluation are in Table 9. The results suggest that ROUGE-1 is surprisingly the best performing technique when measuring Adequacy and Relevance (at both document and hunk level). For fluency and grammar, the ROUGE-L and BERTScore prevail. This result has a natural explanation in that ROUGE-1 is spotting certain individual words which are important for the meeting content. We speculate that due to the variance in possible reference summarizations, such a keyword match does not happen often, so the signal is weak and needs large test sets to be spotted, but it is there. ROUGE-2 and especially ROUGE-L measure overlap of longer sequences of words. Again, such a match can be rare, but if it happens, it reflects more some form of fluency rather than adequacy.

Contrary to expectations, GPT-based evalua-

	D Adeq	D Fluency	D Gram	D Relev	H Adeq	H Fluency	H Gram	H Relev
BART-score	66.7 %	47.6 %	42.9 %	61.9 %	85.7 %	71.4 %	76.2 %	71.4 %
BERT-score	57.1 %	76.2 %	71.4 %	61.9 %	66.7 %	90.5 %	76.2 %	52.4 %
GPT-AFGR A	61.9 %	61.9 %	57.1 %	66.7 %	81.0 %	76.2 %	71.4 %	66.7 %
GPT-AFGR F	52.4 %	66.7 %	61.9 %	61.9 %	71.4 %	81.0 %	71.4 %	57.1 %
GPT-AFGR G	52.4 %	71.4 %	66.7 %	66.7 %	71.4 %	85.7 %	71.4 %	57.1 %
GPT-AFGR R	61.9 %	61.9 %	57.1 %	66.7 %	81.0 %	76.2 %	71.4 %	66.7 %
ROUGE-1	85.7 %	66.7 %	61.9 %	81.0 %	85.7 %	61.9 %	66.7 %	81.0 %
ROUGE-2	76.2 %	66.7 %	61.9 %	81.0 %	76.2 %	61.9 %	57.1 %	71.4 %
ROUGE-L	61.9 %	81.0 %	76.2 %	66.7 %	61.9 %	76.2 %	61.9 %	57.1 %
GPT-ART A	66.7 %	57.1 %	52.4 %	61.9 %	85.7 %	71.4 %	66.7 %	71.4 %
GPT-ART R	61.9 %	52.4 %	47.6 %	57.1 %	81.0 %	71.4 %	66.7 %	66.7 %
GPT-ART T	61.9 %	52.4 %	47.6 %	57.1 %	81.0 %	76.2 %	71.4 %	66.7 %
GPT-ART avg	66.7 %	57.1 %	52.4 %	61.9 %	85.7 %	71.4 %	66.7 %	71.4 %
GPT-AFGR avg	57.1 %	61.9 %	57.1 %	61.9 %	76.2 %	76.2 %	71.4 %	61.9 %

Table 9: Pairwise accuracy, where each column represent one manual human evaluation attribute used as a gold standard (Doc and Hunk level scores for Adequacy, Fluency, Grammaticality and Relevance). Grey background highlights highest value for each manual pair. There are only 21 system pairs, meaning that 4.8% absolute difference is a single mislabeled system pair.

tion did not perform well, sometimes staying very close to the 50% coin-flip baseline. A possible explanation could be that GPT doesn't understand each category (adequacy, fluency, etc) the same way as humans, therefore we tried to aggregate them into a single score. However, neither this aggregated score performed well.

For the poor result of GPT-based evaluation in doc-level Fluency and Grammaticality, we do consider a possible problem with the annotation. With very good outputs in general and with non-native speakers, GPT may be actually more careful and better, thus departing from the human judgement.

The largest limitation of our pairwise accuracy assessment is the total number of system pairs, which is equal to 21. A possibility to increase the number of pairs would be to evaluate each minutes separately instead of aggregating them to system-level scores. We evaluated it, but ran into the issue of ties, where two minutes get the exact same score, are penalized under pairwise accuracy. This issue is not found in automatic scores such as BARTScore or ROUGE, which use continuous scale for ranking. However, GPT and humans use discrete scale of 5 or 10 points, which often result in score ties. This problem with pairwise accuracy and Kendall Tau was described earlier this year in [Kocmi and Federmann \(2023\)](#) and possible solutions are suggested in [Deutsch et al. \(2023\)](#).

## 7.2 Correlation between Automatic and Human Evaluation

Figure 3 presents Pearson correlations of each pair of manual and automatic metrics of minutes qual-

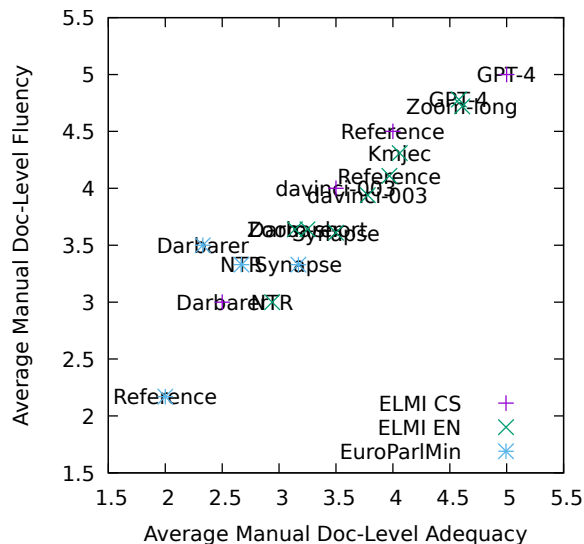


Figure 2: Average manual scores for Adequacy vs. Fluency at the document level in our three test sets

ities across all the datasets. Figures 4 to 6 provide the same information for ELITR Minuting Corpus EN, CS and for EuroParlMin, respectively.

It is important to keep in mind that these correlations are for system-level scores and that there are rather few systems in our collection. Consider the reported correlation of manual doc-level Adequacy vs. Fluency, see Figure 2. In total, there are 17 points, based on which the overall Pearson of 0.94 is calculated. Considering only the 4 EuroParlMin points, we observe a substantially softer correlation of 0.64. The test set with the most participating system, ELMI EN, on the other hand, shows a stronger correlation of 0.97. Pearson cor-

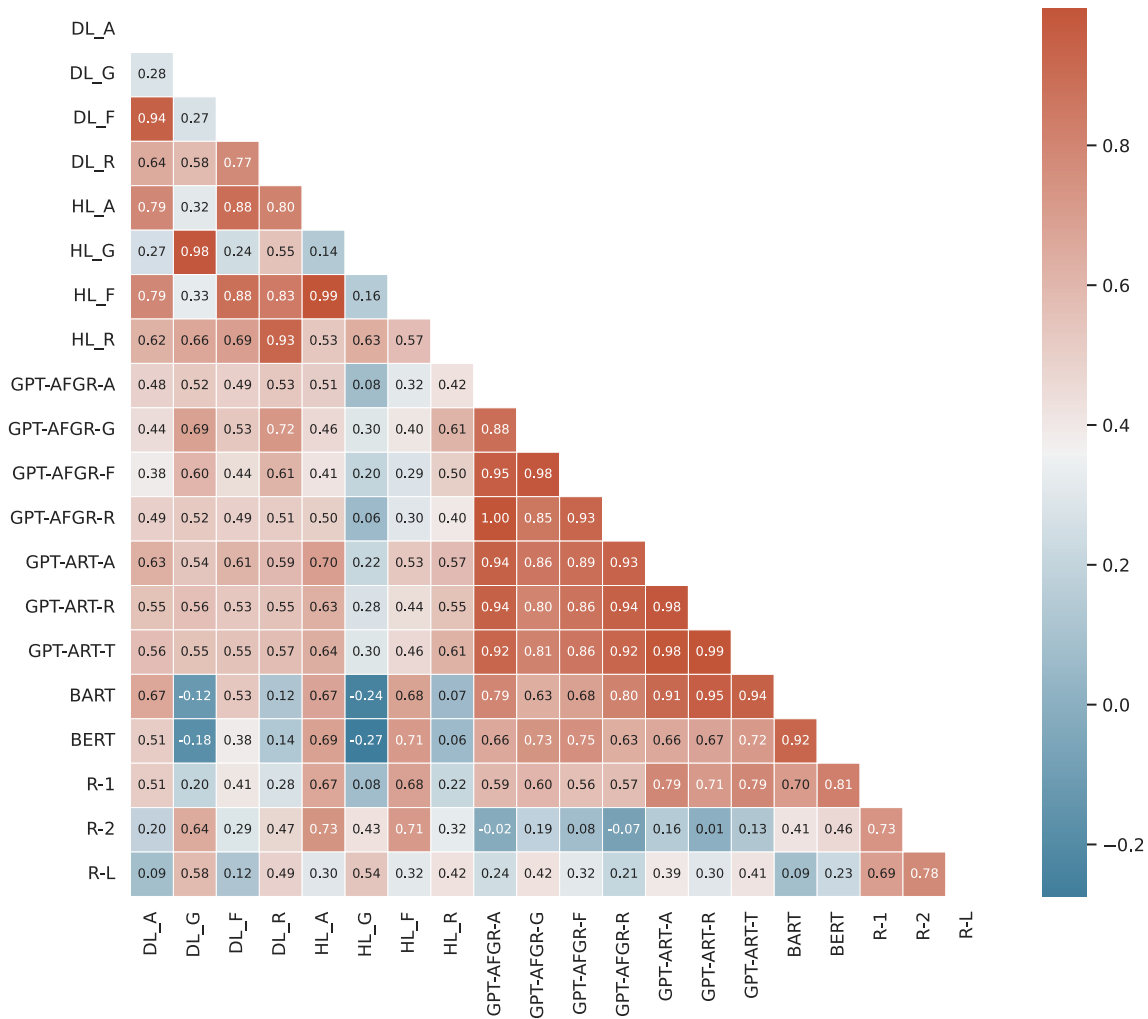


Figure 3: Heatmap showing the average correlation of the different metrics across the three different datasets in our shared task

relations are also very influenced by outliers and, in the case of shared tasks like ours, the underlying set of systems that took part in the task.

Let us discuss the correlations observed in the best covered (in terms of systems as well as manual judgements) test set of ELMI-EN (see Figure 4).

The most striking area is the red triangle of GPT-based metrics (rows and columns GPT-AFGR-G till GPT-ART-T). It shows that GPT essentially ignored the specific quality it should be evaluating (as described in the prompt).

We already mentioned the high correlation for manual Adequacy and Fluency and the “chequered pattern” in the upper left triangle confirms that this holds not just for doc-level but also for hunk-level scores. Relevance, on the other hand, seems to correlate well with Grammaticality on ELMI EN (Pearson of 0.95 for the doc-level scores, see the

think crosses in Figure 7) but this result could be attributed also to the rather low discerning power of Grammaticality (most systems around 5 on the x axis) and the two systems (Synapse and NTR) setting the direction. Across all the test sets, Pearson is 0.58.

Pearson correlations also show that hunk-level scores are typically in line with their doc-level counterparts.

Looking at the lower rows of the heatmaps, BART and ROUGE-1 seem to correlate well with Adequacy at both document and hunk level (Figure 3), although this is not confirmed for BERT on the ELMI EN dataset (Figure 4). Relevance, on the other hand, seems very hard to predict for BART, BERT and also ROUGE, with Pearsons typically under 0.2.

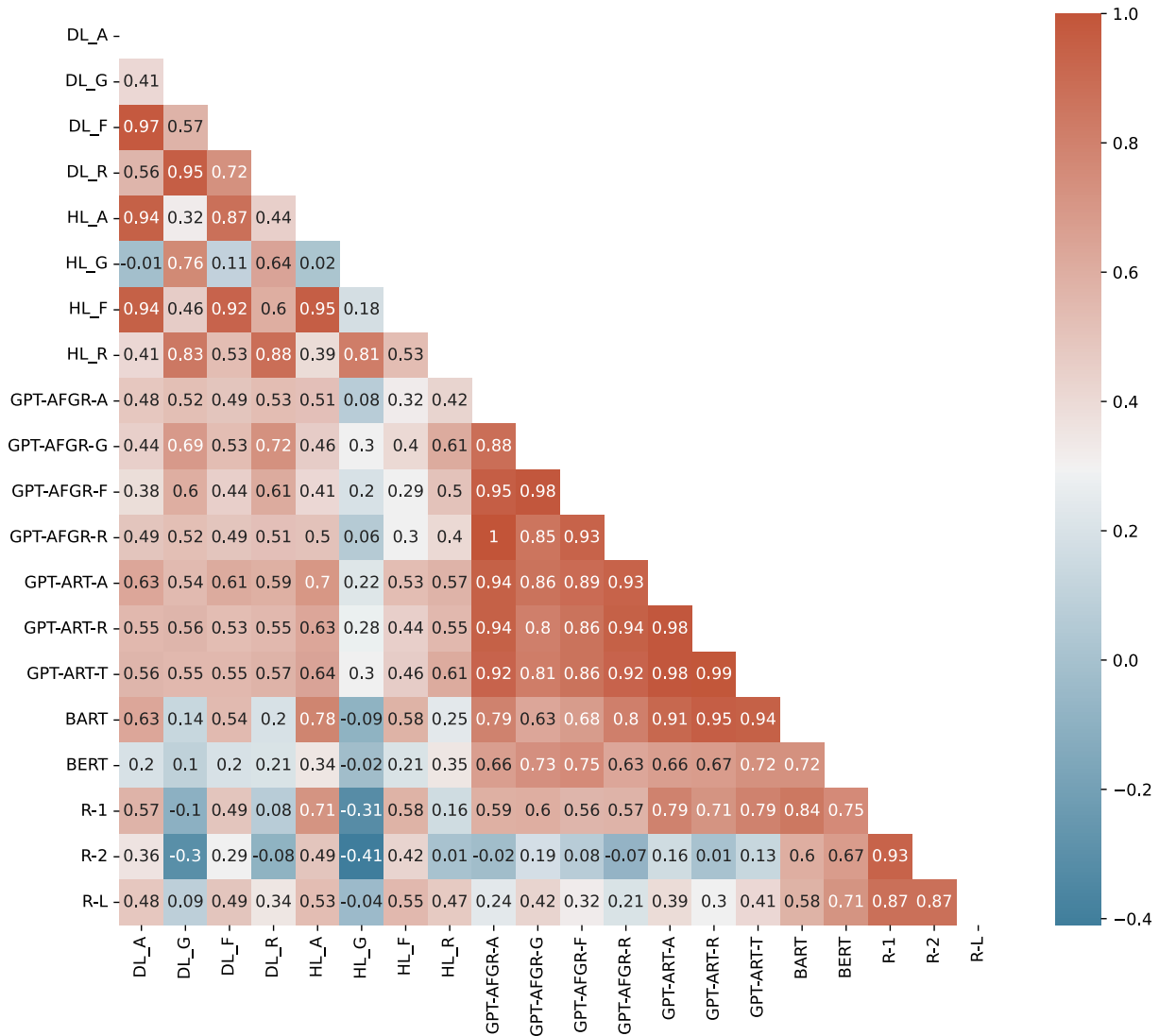


Figure 4: Heatmap showing the average correlation of the different metrics on the ELMi-EN dataset

## 8 Findings from 2nd AutoMin

A lot has changed since the previous instance of AutoMin in 2021. Last time, we were contemplating about one or two teams reaching perfect or close to perfect scores in Fluency and Grammaticality. This time, most of the submitted systems have this property.

We observed that the excellent superficial quality can, to some extent, hinder manual evaluation because errors can go unnoticed.

One of our attempts to improve manual evaluation was to introduce the division into hunks and hunk-level annotation. We have to conclude that this approach was not successful. On the positive side, the more fine-grained scoring provides more points of measurements. The critical drawback is, however, that the minutes get fragmented for the annotator. Assessed in such a partial iso-

lation, more hunks seem flawless and the overall scores do not allow us to separate good vs. bad systems. It is conceivable that the hunk-level annotation would be informative for spotting problems, but its aggregate interpretation is problematic. For the next instance, we need to refocus such detailed manual annotation into spotting errors. The results, with some variance across the test sets, indicate that manual Fluency and Adequacy are strongly correlated. Relevance and Grammaticality differ. Please keep in mind the relatively low number of points behind this analysis. A good sign is that the hunk-level judgements correlate with the document-level ones.

We confirm that according to current manual measures, LLMs deliver excellent results. GPT-4, Zoom and also Team Iterate scored better than the human reference in terms of Adequacy of project



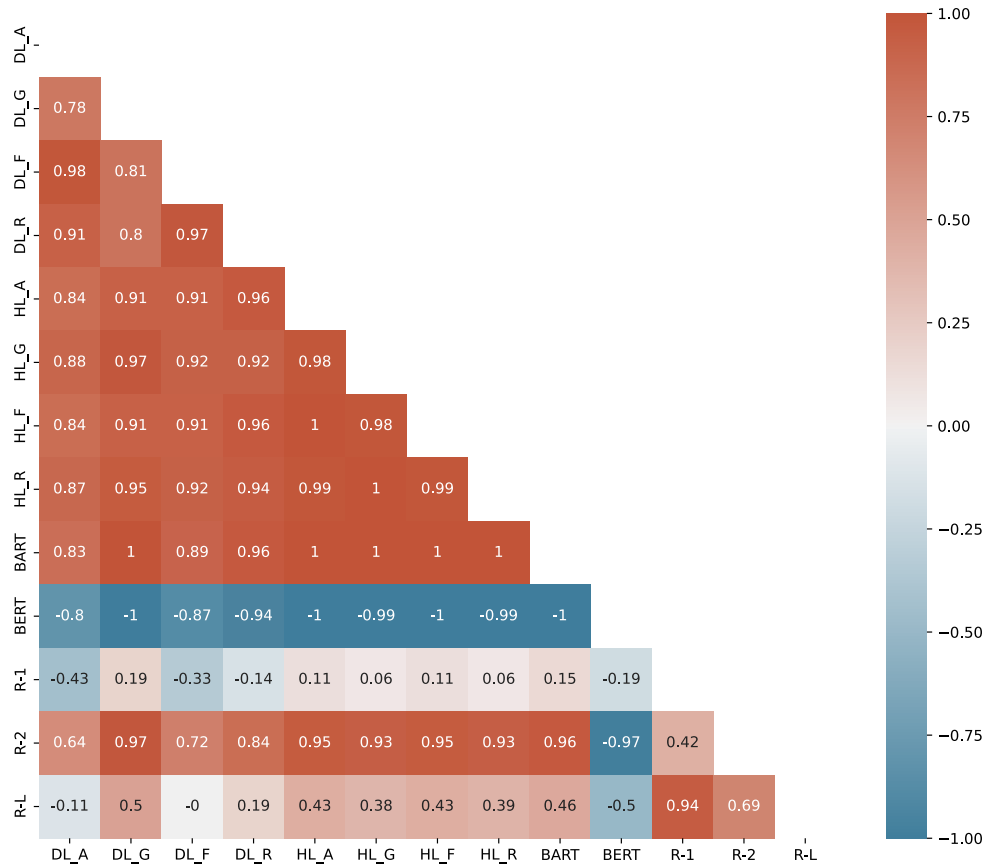


Figure 5: Heatmap showing the average correlation of the different metrics on the ELMI-CS dataset

meeting minutes.

In our meta-evaluation, we used Pearson correlation coefficients and pairwise accuracy to figure out which automatic metrics, including GPT prompting, are most similar to the different styles of our manual judgements. Our analysis revealed very interesting observations. GPT confidently prefers GPT-generated summaries and it is not really able to distinguish among the evaluation criteria. Its pairwise accuracy compared to human judgements reaches only 50–66%. Traditional metrics like ROUGE are more reliable, with ROUGE-1 obtaining 85.7% for predicting document-level Adequacy and ROUGE-L obtaining 81.0% for Fluency. We explain this by weak but reliable signal (infrequent but important keyword and key phrase matches).

## 9 Conclusions and Future Plans

This paper presented the results of AutoMin 2023, the second instance of our shared task on automatic summarization of meeting transcripts into meeting minutes. The data of the shared task (inputs, references, submissions, scores etc.) are

available in this repository:

<https://github.com/ufal/automin-2023-data>

As in the previous instance, the task was run on two languages (English and Czech), with English receiving more attention from the participants.

The submissions were scored manually at the level of full minutes (document level) and also averaging the scores assigned to individual minutes points (hunk level). We concluded that the more fine-grained hunk-level annotation is less useful; the most important question we would like to learn from our annotators is whether the minutes properly reflect the content and overall impression of the meeting. With hunk-level annotation, the annotation process is fragmented and even if each of the fragments is of a high quality, the aggregation of these scores does not answer the key question.

This year, we added the domain of EuroParl sessions and observed that our preferred style of bulleted minutes is in sharp contrast with the officially released summaries. Our annotators liked our style better and the official references did not score well.

AutoMin 2023 also responded to the emergence of large language models, applying them both to

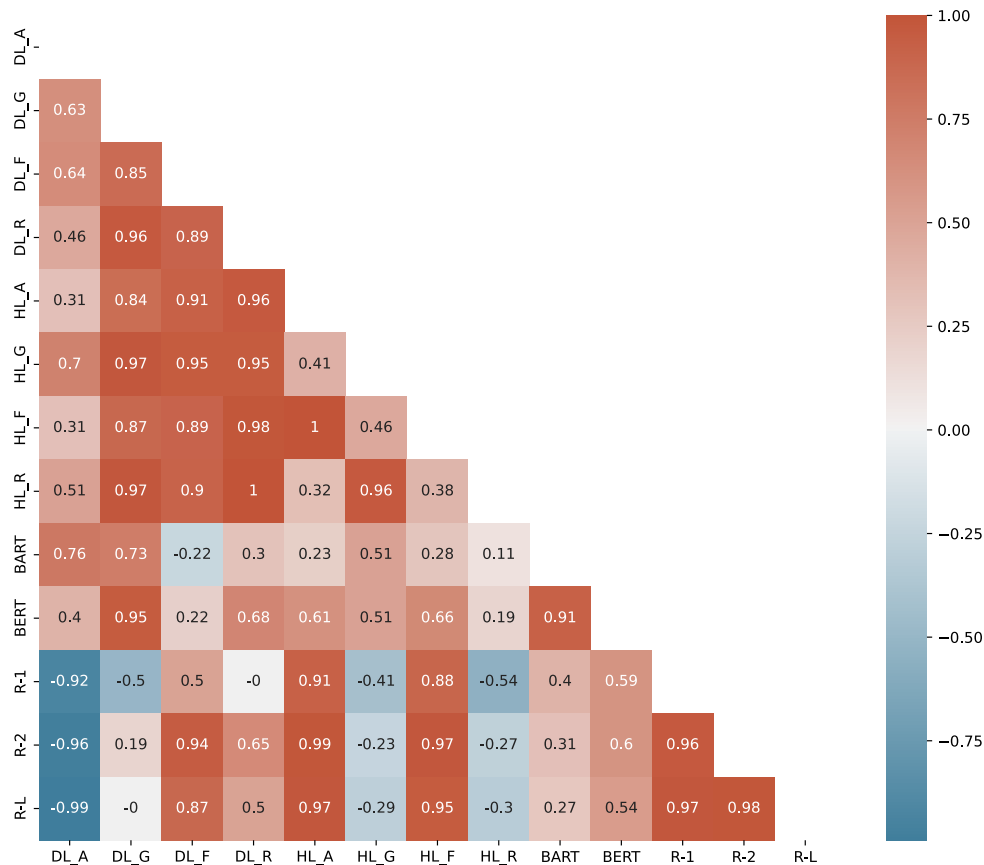


Figure 6: Heatmap showing the average correlation of the different metrics on the EuroParlMin dataset

the task of summarization itself, as well as to the task of assigning scores to the minutes we received from the participants. We confirmed that GPT-4 excels at summarization, surpassing the reference in manual document-level scoring. On the other hand, the automatic evaluation carried out by asking GPT-4 to evaluate the minutes is unreliable. GPT-4 does not distinguish between the different qualities (Adequacy, Fluency, Grammaticality, Relevance, Topicality) and reflects too much the grammaticality; it also prefers its own outputs.

A rather surprising result this year is that the very simple ROUGE is very much in line with manual ranking of system pairs. (ROUGE-1 for Adequacy and Relevance, ROUGE-L for Fluency and Grammaticality). The complex metrics did not provide any substantial benefit or performed simply worse.

We are again trying to secure funding for a future round of AutoMin, aiming primarily at a reliable assessment on whether the gist of the meeting has been well preserved in the minutes, and on a strategy for discovery and scrutiny of summarization errors.

## 10 Limitations

The main limitations behind AutoMin arise from the complexity of the task, which leads to demanding (and thus costly) annotation effort. We would prefer to have far more judgements, and we would have preferred to be able to run, e.g., multiple independent manual evaluations of the same meeting, in order to increase the discovery of errors, but our budget was limited.

Another serious limitation comes from the subjectivity of the minuting task as such. With so varied opinions on what is important in a meeting, it is difficult to assess minutes qualities reliably.

## 11 Acknowledgement

We would like to thank the participants for their enthusiastic participation in AutoMin and for bearing with our delays.

AutoMin 2023 was supported by the grant 19-26934X (NEUREM3) of the Czech Science Foundation.

## 12 Ethical Considerations

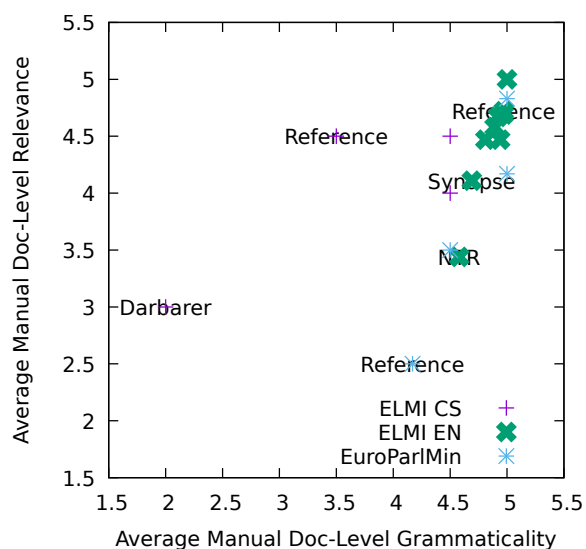


Figure 7: Average manual scores for Grammaticality vs. Relevance at the document level in our three test sets. System names shown only for the reference and a few selected systems.

We note that our annotators were real humans, and they carried their annotation tasks themselves, not delegating it to AI. The annotators were paid the standard hourly wage for this type of work in the Czech Republic.

## References

Alexander A. Alemi and Paul H. Ginsparg. 2015. [Text segmentation based on semantic word embeddings](#). *ArXiv*, abs/1503.05543.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. [Longformer: The long-document transformer](#). *CoRR*, abs/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: Generation Challenges*, pages 110–120, Waterville, Maine, USA and virtual meeting. Association for Computational Linguistics.

Eugene Borisov and Nikolay Mikhaylovskiy. 2023. [Team ntr @ automin 2023: Dolly llm improves minuting performance, semantic segmentation doesn't](#). In *Proceedings of the 16th International Conference on Natural Language Generation: Generation Challenges*. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot

learners. *Advances in neural information processing systems*, 33:1877–1901.

Yulong Chen, Naihao Deng, Yang Liu, and Yue Zhang. 2022. [DialogSum challenge: Results of the dialogue summarization shared task](#). In *Proceedings of the 15th International Conference on Natural Language Generation: Generation Challenges*, pages 94–103, Waterville, Maine, USA and virtual meeting. 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.

Yulong Chen, Yang Liu, Ruochen Xu, Ziyi Yang, Chenguang Zhu, Michael Zeng, and Yue Zhang. 2023. [UniSumm and SummZoo: Unified model and diverse benchmark for few-shot summarization](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12833–12855, Toronto, Canada. Association for Computational Linguistics.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. [Vicuna: An open-source chatbot impressing gpt-4 with 90%\\* chatgpt quality](#).

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. [Free dolly: Introducing the world's first truly open instruction-tuned llm](#).

Daniel Deutsch, George Foster, and Markus Freitag. 2023. [Ties matter: Modifying kendall's tau for modern metric meta-evaluation](#). *arXiv preprint arXiv:2305.14324*.

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.

Xiachong Feng, Xiaocheng Feng, Libo Qin, Bing Qin, and Ting Liu. 2021. [Language model as an annotator: Exploring DialogGPT 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 (Volume 1: Long Papers)*, pages 1479–1491, Online. Association for Computational Linguistics.

- Markus Freitag, Ricardo Rei, Nitika Mathur, Chikui Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie, and André F. T. Martins. 2022. [Results of WMT22 metrics shared task: Stop using BLEU – neural metrics are better and more robust](#). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 46–68, Abu Dhabi, United Arab Emirates (Hybrid). 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 *Proc. First Shared Task on Automatic Minuting at Interspeech 2021*, pages 1–25.
- Tirthankar Ghosal, Ondřej Bojar, Muskaan Singh, and Anja Nedoluzhko. 2021b. [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, Marie Hledíková, Ondřej Bojar, and Muskaan Singh. 2022a. Final report on summarization.
- Tirthankar Ghosal, Marie Hledíková, Muskaan Singh, Anna Nedoluzhko, and Ondřej Bojar. 2022b. [The second automatic minuting \(AutoMin\) challenge: Generating and evaluating minutes from multi-party meetings](#). In *Proceedings of the 15th International Conference on Natural Language Generation: Generation Challenges*, pages 1–11, Waterville, Maine, USA and virtual meeting. Association for Computational Linguistics.
- Tirthankar Ghosal, Muskaan Singh, Anja Nedoluzhko, and Ondřej Bojar. 2022c. Report on the sigdial 2021 special session on summarization of dialogues and multi-party meetings (summdial). In *ACM SIGIR Forum*, volume 55, pages 1–17. ACM New York, NY, USA.
- Tirthankar Ghosal, Muskaan Singh, Anja Nedoluzhko, and Ondřej Bojar. 2021c. Report on the sigdial 2021 special session on summarization of dialogues and multi-party meetings (summdial). In *ACM SIGIR Forum*, volume 55, pages 1–17. ACM New York, NY, USA.
- 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.
- Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.
- Yebowen Hu, Timothy Ganter, Hanieh Deilamsalehy, Franck Dernoncourt, Hassan Foroosh, and Fei Liu. 2023. [MeetingBank: A benchmark dataset for meeting summarization](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16409–16423, Toronto, Canada. 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. pages 364–367.
- Moussa Kamal Eddine, Antoine Tixier, and Michalis Vazirgiannis. 2021. [BARThez: a skilled pretrained French sequence-to-sequence model](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9369–9390, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Hyun Kim, Minsoo Cho, and Seung-Hoon Na. 2023. Explainmeetsum: A dataset for explainable meeting summarization aligned with human intent. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13079–13098.
- Kristýna Klesnilová and Michelle Elizabeth. 2023. Team synapse @ automin 2023: Leveraging bart-based models for automatic meeting minuting. In *Proceedings of the 16th International Conference on Natural Language Generation: Generation Challenges*. Association for Computational Linguistics.
- František Kmječ and Ondřej Bojar. 2023. Team iterate @ automin 2023 - experiments with iterative minuting. In *Proceedings of the 16th International Conference on Natural Language Generation: Generation Challenges*. Association for Computational Linguistics.
- Tom Kocmi and Christian Federmann. 2023. Large language models are state-of-the-art evaluators of translation quality. *EAMT 2023*.
- Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. [To ship or not to ship: An extensive evaluation of automatic metrics for machine translation](#). In *Proceedings of the Sixth Conference on Machine Translation*, pages 478–494, Online. Association for Computational Linguistics.
- Lakshmi Prasanna Kumar and Arman Kabiri. 2022. Meeting summarization: A survey of the state of the art. *arXiv preprint arXiv:2212.08206*.
- 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.
- 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.
- 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.
- 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 Thirteenth Language Resources and Evaluation Conference*, pages 3174–3182, Marseille, France. European Language Resources Association.
- OpenAI. 2023. [Gpt-4 technical report](#).
- Peter Polák, Muskaan Singh, Anna Nedoluzhko, and Ondřej Bojar. 2022. [ALIGNMEET: A comprehensive tool for meeting annotation, alignment, and evaluation](#). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1771–1779, Marseille, France. European Language Resources Association.
- Archiki Prasad, Trung Bui, Seunghyun Yoon, Hanieh Deilamsalehy, Franck Dérnoncourt, and Mohit Bansal. 2023. [MeetingQA: Extractive question-answering on meeting transcripts](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15000–15025, Toronto, Canada. Association for Computational Linguistics.
- Virgile Rennard, Guokan Shang, Julie Hunter, and Michalis Vazirgiannis. 2022. [Abstractive meeting summarization: A survey](#). *arXiv preprint arXiv:2208.04163*.
- Ismaël Rousseau, Loïc Fosse, Youness Dkhissi, Géraldine Damnati, and Gwénoél Lecorvé. 2023. [Team darbarer @ automin2023: Transcription simplification for concise minute generation from multi-party conversations](#). In *Proceedings of the 16th International Conference on Natural Language Generation: Generation Challenges*. Association for Computational Linguistics.
- Kristóf Sándor. 2023. [Evaluating the effectiveness of large language models in meeting summarization with transcript segmentation techniques: How well does gpt-3.5-turbo perform on meeting summarization with topic and context-length window segmentation?](#)
- Felix Schneider and Marco Turchi. 2023. [Team zoom @ automin 2023: Utilizing topic segmentation and llm data augmentation for long-form meeting summarization](#). In *Proceedings of the 16th International Conference on Natural Language Generation: Generation Challenges*. 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 *Proc. First Shared Task on Automatic Minuting at Interspeech 2021*, pages 26–33.
- Kartik Shinde, Tirthankar Ghosal, Muskaan Singh, and Ondřej Bojar. 2022. [Automatic minuting: A pipeline method for generating minutes from multi-party meeting proceedings](#). In *Proceedings of the 36th Pacific Asia Conference on Language, Information and Computation*, pages 691–702, Manila, Philippines. De La Salle University.
- Muskaan Singh, Tirthankar Ghosal, and Ondřej 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.
- Muskaan Singh, Rishu Kumar, Tirthankar Ghosal, Ondřej Bojar, Chiara Canton PV, Andrea Sosi PV, Adelheid Glott AV, and Franz C Krüger AV. 2022. [Demonstrator of automatic minuting](#).
- Paul Tardy, David Janiszek, Yannick Estève, and Vincent Nguyen. 2020. [Align then summarize: Automatic alignment methods for summarization corpus creation](#). In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6718–6724, Marseille, France. European Language Resources Association.

- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- L. Lynn Voss and Patrick Ehlen. 2007. [The CALO meeting assistant](#). In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 17–18, Rochester, New York, USA. Association for Computational Linguistics.
- Atsuki Yamaguchi, Gaku Morio, Hiroaki Ozaki, Kenichi Yokote, and Kenji Nagamatsu. 2021. [Team hitachi @ automin 2021: Reference-free automatic minuting pipeline with argument structure construction over topic-based summarization](#). In *Proceedings of the First Shared Task on Automatic Minuting at Interspeech 2021*, pages 1–8.
- Diyi Yang and Chenguang Zhu. 2023. [Summarization of dialogues and conversations at scale](#). In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: Tutorial Abstracts*, pages 13–18, Dubrovnik, Croatia. Association for Computational Linguistics.
- Xianjun Yang, Yan Li, Xinlu Zhang, Haifeng Chen, and Wei Cheng. 2023. Exploring the limits of chatgpt for query or aspect-based text summarization. *arXiv preprint arXiv:2302.08081*.
- 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.
- Tianyi Zhang\*, Varsha Kishore\*, Felix Wu\*, Kilian Q. Weinberger, and Yoav Artzi. 2020. [Bertscore: Evaluating text generation with bert](#). In *International Conference on Learning Representations*.
- Ming Zhong, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2022. Dialoglm: Pre-trained model for long dialogue understanding and summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11765–11773.
- Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021. [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, 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.

## A Sample Reference Minutes

Date: 2021-01-11  
Attendees: [PERSON1], [PERSON2], [PERSON3], [PERSON4], [PERSON5]  
Purpose of meeting: Progress report

- [PERSON1] and [PERSON2] discuss recent progress on ACL paper.
- [PERSON1] points out the automatic metrics seems not to be sufficient to evaluate the performance system (it performs too good).
- More challenging experiments are discussed by [PERSON1], [PERSON2] and [PERSON3].
- The available test set does not contain enough less common and rare words, there is a need for bilingual vocabulary and additional sources for system training.
- [PERSON3] is requested to provide more data for Portuguese language model
- [PERSON7] (not attending the meeting) is expected to deliver additional training data
- Options to find bilingual texts are being discussed, namely to search for articles and textbooks, check university repositories for master theses and additional sources of terminology words.
- [PERSON1] Proposed training of the system on different style of speech/writing (colloquial, male vs. female, scientific vs. non-scientific)
- The only available model of style transfer was regarding simplification or generalization of the text.
- Sub-part of elitr test set will be created to cover language style specifics, such as gender transformations.
- [PERSON4] showed how to search for named entities in the available bilingual dataset.
- Some manual work will be necessary to compare the outputs with those listed in there.
- This might give an overview what are the common problems with terminology and make a basis for markable(?) experiments
- Examples of related mistranslated words are mentioned (tenant vs lessee) and quality of written audits is being discussed
- For further work, [PERSON2] will proceed with WMT elitr dataset for spoken language and (sao? domain for written texts
- Specific terms evaluation will be performed by [PERSON5] based on WMT testsuite markables.
- [PERSON1] briefly mention the opportunity to multi source, however other people involved in this are busy, thus compatibility of their implementation will be assessed later

Minutes submitted by: [ANNOTATOR1]

Figure 8: A sample minute taken by our external annotator

## B Sample Minutes from AutoMin participants

We present some minute samples from our participants' submissions to show the variety of automatically generated minutes by different methods. For fair comparison we include the participant's generated minutes from the same meeting.

Date: 13.12.2022  
– Attendees: PERSON2, PERSON3, PERSON1

- Replaning the satellite event
  - [PERSON1] and [PERSON2] are at the office
  - [PERSON3] is at a satellite event
  - They have to replan everything now.
- Preparation of the workshop
  - The proposal for a statistical workshop this year is over, so it would be too late for the next one in 2022
  - The participants have to submit a proposal for the workshop
  - They have to write a paper on the topic
  - The deadline for submitting the proposal is March fifteenth (!)
  - They need to prepare the event in three months.
- Preparation of shared task
  - [PERSON3] and [PERSON2] are preparing a shared task
  - They have to get an endorsement
  - They need to take membership to submit the proposal and pay some fees for their work– uh workshop.
- Editing the journal
  - [PERSON1], [PERSON2] and [PERSON3] are interested in publishing their work in a journal
  - The journal was published in 1964
  - It is not indexed on the website
  - The proceedings would be indexed with Google.
- Working on the satellite event
  - [PERSON2] and [PERSON3] will meet tomorrow to discuss the additional works they need to do for the satellite event.
- Submitted by: Darbarer

Figure 9: A sample minute by team Darbarer. This text was edited to replace some emoticons with hyphens to avoid encoding issues.

PERSON1 is at the office today, in the office. PERSON2 is waiting for the person who was supposed to join the group, but he didn't come. The person has returned the last e-mail.  
(PERSON1) and (PERSON3) are going to make a shared task for a statistical workshop. The proposal for this year's workshop was made in September, which was too early for them. The next proposal for 2022 would be too late for them, so they have  
PERSON1 will be back in two minutes.  
PERSON3 and PERSON2 have to go to both workshop now. They need to take membership to submit the proposal.  
They have to make a website for their shared task and pay some fees for their work workshop.  
PERSON1, PERSON2, PERSON3 and PERSON4 are planning to publish their research in a journal. The journal was published in 1964. The core papers were in red. It is indexed in Google Color. The proceedings would be indexed with ethylanthology.  
PERSON2 and PERSON3 are looking for the conference proceedings from El Severe journal. They need to revise the proposal for the satellite event and then meet tomorrow to work on the additional works.  
(PERSON2) and (PERSON3) have to do double work tomorrow.

Figure 10: A sample minute by team Iterate.



Meeting KeyPoints:

- \* PERSON2 invited PERSON1 to the meeting.
  - \* PERSON1 accepted the invitation and attended the meeting.
  - \* PERSON2 is aware that PERSON1 is in the office and is enjoying her walk.
  - \* PERSON1 is enjoying the walk and deciding to enjoy it more by visiting an unfamiliar location.
  - \* PERSON2 wanted to mention that PERSON1 is joining ORGANIZATION1 for
  - There will be 3 papers for the satellite event
  - The 3 papers should follow the same submission criteria as other papers
  - If it's for a satellite event, the authors should follow the rules
  - Authors should not submit the papers to the Interspeech proceedings
  - 1. The last time a shared task was held at (sick dial) was in August of 2022.
  - 2. The workshop for statistical theory in 2021 was held in September of 2022.
  - 3. The workshop for statistical theory in 2022 would be too late for Thumbtack.
  - 4. P3 preferred to apply for the special session for the workshop for statistical theory in 2021 instead of 2022.
  - 5
  - PERSON1 wants to conduct a workshop, rather than a special session, for their shared task
  - The workshop should be advertised as such – "Please take part in the shared task. We will provide you with the necessary tools and guidelines."
  - PERSON1 wants to conduct a 3-month long programme, during which time they will provide the workshop participants with a pre-prepared programme, the workshop data and the workshop's results
  - The deadline to submit the workshop proposal is March 15th, 2022
  - In order to get an ISBD endorsement, the organizers need to submit a proposal to conduct a workshop at the interspeech conference by January 15th, 2022
  - To submit the workshop proposal, the organizers will need to do the following steps:
    - Submit the proposal and get the endorsement from the speech community
    - Write the call for
  - The person who will do the coding/data extraction is unsure. They think that they have to code/extract the data but they don't know how to do it. They need someone who knows the system and who will extract the data for them.
  - Three people are going to the online workshop but they need to settle down first.
  - The person who responded to the email also agreed to do the workshop.
  -
- So the meeting participants decided to seek advice from a senior colleague regarding the following:
- \* Which aspects of the organizational structure are unclear to PERSON1
  - \* How much it will cost to register for the workshop
  - \* How to approach (german name) about financial aspects
  - \* What are the main tasks of the program committee
  - \* How to make a website for the shared task
1. P4 is the technical lead for the project.
  2. P1 is the project manager.
  4. P2 is aware of P3's interest and suggests that P4 can inform whether the journal is indexed.
  5. P4 is able to provide some information regarding the journal
- bblt?  
it?  
is not indexed..  
keyboard, distant noise  
Less work –  
Less work and intent– more intensive output.
- P
1. The proceedings from El Severe will be indexed in Cosmopile and Google Color
  2. Also, the quality of papers published in the pvl journal is very good
  3. The pvl journal is open access and is directly indexed by journals that offer open access to publications
1. PERSON3 and PERSON2 disagree on the feasibility of the proposed project.
  2. PERSON3 will double the work required compared to the original proposal.

Figure 11: A sample minute by team NTR-NLP

DATE : 2023-05-01  
ATTENDEES : PERSON1, PERSON2, PERSON3

SUMMARY

- PERSON1 is in the office today.
- PERSON2 is waiting for the person who was supposed to join the group.
- The person has not turned up yet.
- There was a technical communication meeting.
- The participants were told to prepare their papers for special sessions according to the same criteria as PERSON3, PERSON2 and PERSON1 are planning a mini conference.
- The proposal is going to be up there.
- It is entirely independent of interspeech to program committee.
- They will promote their workshop in their webpage.
- There is a possibility to do both sectile and inter PERSON1, PERSON2 and PERSON3 explain to each other how the process of organising a workshop works.
- PERSON3 and PERSON2 have prepared the document for the shared task.
- They have to take institutional membership to submit the proposal.
- They can do both.
- PERSON1, PERSON2, PERSON3 and PERSON4 are organizing a workshop on Interspeech.
- They have to study some satellite events that they got accepted to last year and make a website for their shared task.
- They need to pay for institutional membership, manage the project and PERSON2 has read a lot of papers on machine translation in her researchation.
- The journal was published in 1964.
- The core papers were in this journal.
- It is already written in there.
- The proceedings would be indexed with ethylanthology.
- PERSON2 and PERSON3 are planning to present the severe conference proceedings from El Severe on January 15<sup>th</sup>.
- They are going to use pvl format.
- The paper is open access and indexed in Cosmopile Google color.
- PERSON3 wants to differ the discussion with PERSON4.
- They need to revise the proposal and leave down the thing that they need to do for the satellite event.
- They will meet tomorrow.

Minuted by: Team Synapse

Figure 12: A sample minute by team Synapse

PERSON1 and PERSON3 are discussing the proposal for a satellite event for a spoken language technology workshop.  
They discuss the need to follow the same criteria as other special sessions and the timelines for the submission of papers.  
They agree to submit two papers, one for the satellite event and one for a full fledged workshop.  
PERSON2 offers to help with some potential things, but PERSON3 reminds him to do double work.  
They decide to go for sigdial for the machine translation corpus linguistics and morphologies workshop in Texas, as the proceedings would be indexed with ethylanthology.

Figure 13: A sample minute by team Zoom - short version

PERSON2 and PERSON3 are discussing the meeting meeting at the ORGANIZATION1. PERSON1 is at the office and is waiting for PERSON3 to join , but he is not answering the call. PERSON3 has missed the call and PERSON1 will translate the last mail for them, which is a czech text. They have to replan everything now.

PERSON1 and PERSON3 are discussing the proposal for a satellite event for a spoken language technology workshop. They discuss the need to follow the same criteria as other special sessions and the timelines for the submission of papers. They also discuss the possibility of doing both a shared task at sigdial and submitting a proposal for the next conference. They agree that it is better to submit two papers, one for the satellite event and one for a full fledged workshop, as it gives them more flexibility and freedom to choose their event or accept papers from the Proceedings of Intra Speech. However, they also need to figure out how to make sure the participants are not deterred from submitting their own papers.

PERSON3 and PERSON1 are discussing the organizational points of their proposal for their workshop. PERSON3 explains that they have to take institutional membership to submit the proposal to the ORGANIZATION2 workshop application portal and that they need to contact Andre and Hero to discuss the proposal content. PERSON1 has no idea about the administrative plans for the workshop, but PERSON3 suggests asking PERSON4.

PERSON3 and PERSON1 are discussing a paper published in 1964 by noise-Linguistics. PERSON1 is not interested in all the points of the paper, but he has published it once or twice. PERSON2 has read a lot of papers about machine translation in his researcheraion, but it depends on the participants.

PERSON2 and PERSON3 discuss the possibility of doing a machine translation corpus linguistics and morphologies workshop in Texas. They decide to go for sigdial , as the proceedings would be indexed with ethylanthology. PERSON1 suggests that sigdial is better than interspeech workshop, as it has less work and more intensive output. PERSON3 suggests that they should look at where the proceedings are indexed to see if there is any specific charge for the journals.

PERSON3 and PERSON2 discuss the proposal for the satellite event. PERSON2 offers to help with some potential things , but PERSON3 reminds him to do double work. They agree to meet tomorrow and say goodbye.

Figure 14: A sample minute by team Zoom - long version

## C Sample GPT-Generated Minutes

- Meeting held to discuss organization of a shared task or workshop for [ORGANIZATION1]
- [PERSON1] attended the meeting from the office
- [PERSON3] joined the meeting late, but everything was fine
- Discussed the possibility of organizing a satellite event or special session at Interspeech
- [PERSON1] translated an email regarding the requirements and timelines for special sessions at Interspeech
- Discussed the possibility of submitting a shared task proposal to both Interspeech and SigDial
- Concerns raised about the workload involved in organizing both events and finding enough participants
- [PERSON3] mentioned that they have experience organizing such events and believes it is possible to manage both
- Discussed the need for institutional membership to submit a workshop application to [ORGANIZATION2]
- [PERSON2] mentioned receiving an email reply agreeing to some discussed points
- Discussed the need for membership to [ORGANIZATION4] to submit a proposal
- Considered paying fees for a workshop and contacting Andre and Hero for more information
- Discussed preparing content for the workshop proposal and studying other satellite events
- Talked about the need for management and creating a website for the shared task
- Discussed potential publication options, including [PROJECT2] Linguistics, archive, and [PROJECT1] from Elsevier
- Mentioned the possibility of publishing as part of the () community and deferring the discussion to [PERSON4]
- Agreed to revise the proposal and list additional tasks needed for the satellite event
- [PERSON1] offered to help with writing chapters for the proposal and reviewing it
- Agreed to meet again tomorrow to discuss further

Figure 15: A sample minute generated with GPT-4

- Project meeting was held between [PERSON1], [PERSON2], and [PERSON3]
- [PERSON1] was at the office and had decided to enjoy walking there
- [PERSON3] joined the meeting late
- Discussed the possibility of submitting a proposal for a special session or a satellite event at [ORGANIZATION1]
- [PERSON1] suggested submitting a shared task at [ORGANIZATION2]
- [PERSON3] proposed doing both the special session and the satellite event
- [PERSON3] suggested that the proposal for the workshop should be submitted as soon as possible and get endorsement
- [PERSON3] also suggested that the workshop should be entirely independent and the program committee, keynotes, and program should be ready before submitting the proposal
- Deadline for the satellite event is February 28th
- Deadline for submitting the full papers of participants is January 15th
- [PERSON3] suggested doing both the special session and the satellite event
- Need to contact [PERSON4] for membership to [ORGANIZATION4]
- Need to pay fees for workshop
- Need to discuss proposal content for workshop
- Need to create website for shared task
- Need to study satellite events accepted to Interspeech last year
- Need to do management and contact program committee
- Need to decide between SIGDIAL and Interspeech workshop
- SIGDIAL has more intensive output and indexed with Ethylanthology
- [PERSON4] can provide more information about [PROJECT2]-Linguistics
- [PERSON2] suggests [PROJECT1] from Elsevier
- [PERSON2] suggests [PROJECT2] from PVML
- [PERSON2] suggests [PROJECT3] from BBLT
- [PERSON2] suggests [PROJECT4] from Open Access Journals
- Need to discuss with [PERSON4] for better proposal

Figure 16: A sample minute generated with text-davinci-003

## D Annotation Instructions

### Instructions for Evaluation of Minutes

#### Install/update ALIGNMEET and Populate it with Minutes

- Open command line and run `pip install --upgrade alignmeet` if updating or `pip install alignmeet` if installing for the first time
- In case of issues talk to Marie Hledíková (*email removed*).

#### Find out which meetings are for you

- See this sheet (*link to sheet removed*).
- Whenever you start or finish a meeting (all its minutes), please enter it in the appropriate cell.

#### Annotation Instructions

- **Do not forget to count how many hours you have spent annotating!**
- You are assigned a **set of meetings** (mostly English, some in Czech).
- Each meeting comes with:
  - the **transcript**
  - a set of **several minutes**, each created by a different system.
- You need provide the following annotations to **each of the minutes** (independently of other minutes):
  - **alignment** between the minutes and the transcript
  - **quality scores** for each alignment “hunk” (line in minutes)
  - **quality scores** for the minutes as a whole.
- You *may want* to also use the “Remarks” area (bottom right pane, it used to be called “Problems” in previous ALIGNMEET versions) for your convenience in annotation:
  - You may use the remark “Small talk” to indicate a portion of the transcript which you do not expect to appear in the minutes. However, if the system does include this in the minutes, you **need to primarily align it with the minutes**. (It is allowed to use both for any line in the transcript, to have a remark and be aligned to minutes.)
  - You may want to tell us some extra observations. For this use the remark “See separate comment” and write this comment to the table for assigning annotators.
  - The remarks you make will be copied over to the next minutes of the same transcript if the minutes have not been processed yet. You may switch between the different minutes of the same transcript as you like but as soon as the minutes have such a remark, no other remarks will be copied to them. Sequential processing of the minutes one by one is thus the best option.
- Detailed instructions:
  - **Alignment:**
    - Try to cover all items in the minutes and all text in the transcript but:

1. Not all parts of the transcript have to be aligned to an item in the minutes; e.g. if the system decided to exclude the given piece of information or topic altogether.
  2. Occasionally, some items in the minutes also end up non-aligned; e.g. items in the minutes which are fully hallucinated will not be linked to any segment in the transcript.
    - Do try to make use of the new **autoalign feature**: in the top toolbar, set the threshold (the lower, the fewer alignments will be made) and click Autoalign. Alignment will suggest an alignment hint which will be displayed as color only under the speaker names. You will then need to manually confirm the alignments. It generally tends to help the most if you set a low threshold just to get a rough idea of where things are, the finer suggestions do not tend to be very good.
- **Quality scores** are in the range 1 (worst) to 5 (best).
  - For individual "hunks" (i.e. the colored alignments) the scores should reflect:
    - **Adequacy** assesses if the item in the minute adequately captures the content of the respective aligned segment(s) of the transcript.
    - **Fluency** reflects if the item in the minute consists of fluent, coherent text and is readable to the evaluator.
    - **Grammatical Correctness** checks the level to which the item in the minute is free of errors in the grammar.
    - **Relevance** signifies whether the item in the minute is relevant to be included in the meeting summary (e.g. a perfect summary of small talk is not relevant).
  - For the whole meeting minutes, the scores should reflect:
    - **Adequacy** assesses if the minutes adequately capture the major topics discussed in the meeting, also considering coverage (all such topics covered).
    - **Fluency** reflects if the minutes consist of fluent, coherent texts and are readable to the evaluator.
    - **Grammatical Correctness** checks the level to which the minutes are grammatically correct.
    - **Relevance** signifies the extent to which the minutes overall capture the important content from the source transcript (as opposed to summarizing useless parts).
  - Remark on **minutes styles**:
    - There are two major types of meetings in our collection, you may be able to notice this difference.
    - The style of minutes can however vary a lot depending on which system prepared the minutes.
    - The quality scores are designed so that they **should not be affected by the style** differences too much.
    - If you cannot avoid considering the style of the minutes in your scoring, **consider the style across all the different minutes** that are provided for the given meeting. (I.e. minutes departing seriously in their style from the rest may suffer slightly worse scores, but try to avoid this effect as much as possible.)

## E GPT prompts

```
Given the following meeting transcript and minutes, evaluate the minutes of the meeting for it's adequacy (
the judgment if summary sentences represent conclusions clearly visible in the transcripts of the
discussions), relevance (how well the summary sums up the main idea of the meeting), and topicality (
whether summary sentences cover topics that are discussed in the transcript).

_____  

Transcript:  
{transcript}  

_____  

Minutes:  
{system_generated_minutes}  

_____  

Evaluate minutes for it's adequacy (the judgment if summary sentences represent conclusions clearly visible
in the transcripts of the discussions), relevance (how well the summary sums up the main idea of the
meeting), and topicality (whether summary sentences cover topics that are discussed in the transcript).
Give each score separately on a scale 0 to 10, where 10 is the best:
```

Figure 17: The prompt asking to rate each minutes for adequacy, relevance, and topicality. We label this prompt as “GPT-ART”.

```
Given the following meeting transcript and minutes, evaluate the minutes for their adequacy (to what extent
the minutes adequately capture the major topics discussed in the meeting, also considering coverage, i.
e. all such topics covered), fluency (if the minutes consist of fluent, coherent texts and are readable
to the evaluator), grammatical correctness (the level to which the minutes are grammatically correct)
and relevance (the extent to which the minutes overall capture the important content from the source
transcript (as opposed to summarizing useless parts)).

_____  

Transcript:  
{transcript}  

_____  

Minutes:  
{system_generated_minutes}  

_____  

Now evaluate the minutes for their adequacy, fluency, grammatical correctness and relevance. Give each score
separately on a scale 0 to 10, where 10 is the best:
```

Figure 18: The prompt asking to rate each minutes for adequacy, relevance, and grammatical correctness. We label this prompt as “GPT-AFGR”.

```
The following conversational turn is from a meeting transcript. Classify the turn into 'Filler' (not
relevant outside of the meeting) or 'Content' (contains relevant information).

_____  

{conversation_turn}  

_____  

Classification of the turn as 'Filler' or 'Content':
```

Figure 19: A prompt used to mark conversational tun as containing information or being a filler.





# Author Index

- Abe, Mikio, 84  
Adewumi, Tosin, 1  
Aranha, Claus, 84
- Babakov, Nikolay, 37  
Behzad, Shabnam, 53  
Bojar, Ondřej, 114, 138  
Borisov, Eugene, 132
- Chandu, Khyathi Raghavi, 1  
Chung, Yi-Ling, 1
- Damnati, Geraldine, 121  
Demberg, Vera, 17  
Dkhissi, Youness, 121  
Dobnik, Simon, 23
- Elizabeth, Michelle, 108  
Emezue, Chris Chinenye, 1
- Fosse, Loïc, 121  
Fukuda, Tomoki, 84  
Funakoshi, Kotaro, 79
- Ghosal, Tirthankar, 138  
Gkatzia, Dimitra, 1
- Hagiwara, Masato, 45  
Hanawa, Kazuaki, 45  
Harada, Kei, 84  
Hatanaka, Tomoya, 84  
Hauer, Benedek, 84  
Hledíková, Marie, 138  
Hong, Xudong, 17  
Hou, Yufang, 1  
Howcroft, David M., 1
- Ihori, Mana, 60  
Ilinykh, Nikolai, 23  
Inaba, Michimasa, 84  
Ito, Rei, 84
- Jimichi, Kunitaka, 79
- Kagaminuma, Kaito, 84  
Kano, Yoshinobu, 84
- Katagami, Daisuke, 84  
Kazakova, Lilya, 37  
Klesnilová, Kristýna, 108  
Kmječ, František, 114  
Kocmi, Tom, 138  
Komachi, Mamoru, 68  
Koyama, Shota, 74
- Lecorvé, Gwénolé, 121  
Lee, Jaewon, 84  
Lysyuk, Maria, 37
- Masumura, Ryo, 60  
Mehra, Khushboo, 17  
Mikami, Akihiro, 84  
Mikhaylovskiy, Nikolay, 10, 132  
Miki, Soichiro, 84  
Mita, Masato, 45
- Nagata, Ryo, 45  
Nakamura, Yuta, 84  
Nedoluzhko, Anna, 138
- Okazaki, Naoaki, 74  
Okubo, Takuya, 84  
Okumura, Manabu, 79  
Osawa, Hirotaka, 84  
Otsuki, Takashi, 84
- Panchenko, Alexander, 37
- Qi, Zhiyang, 84
- Rajpoot, Pawan, 1  
Rousseau, Ismaël, 121
- Sato, Hiroshi, 60  
Sayeed, Asad, 17  
Schneider, Felix, 101  
Schneider, Nathan, 53  
Shibata, Hisaichi, 84  
Shigemura, Soga, 84  
Shvets, Alexander, 37  
Stahl, Maja, 31
- Takamura, Hiroya, 74

Takashima, Kazuki, 84

Tanaka, Tomohiro, 60

Toriumi, Fujio, 84

Turchi, Marco, 101

Uchida, Mami, 84

Ueda, Naoya, 68

Wachsmuth, Henning, 31

Wakutani, Masahiro, 84

Watanabe, Neo, 84

Zeldes, Amir, 53