INLG 2023

The 16th International Natural Language Generation Conference: System Demonstrations

Proceedings of the System Demonstrations

September 11 - 15, 2023

Order copies of this and other ACL proceedings from:

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

ISBN 979-8-89176-002-8

Preface

We are excited to present the Proceedings of the 16th International Natural Language Generation Conference (INLG 2023). This year is the first time since the Covid-19 pandemic that the event will run mainly in-person again, from 11 to 15 September 2023 in Prague, Czech Republic. A novel aspect of this year's INLG is that, for the first time in its history, it is held jointly with the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDial 2023). INLG-SIGDIAL 2023 was locally organized by Charles University, thanks to the tireless efforts of the local chair Ondřej Dušek and his team.

The INLG conference is the main international venue for presentation of novel research and discussion of the computational task of Natural Language Generation (NLG) and its broad range of applications, including mainly data-to-text, text-to-text, and image-to-text approaches. Also this year, INLG consisted of several events.

The conference took place from 13 to 15 September. For the main track, we received a total of 98 conference submissions, 4 ARR submissions, and 4 demo paper submissions. After review by at least three program committee members and a meta review from the area chairs, 19 were accepted as long papers, 17 as short papers, and 4 as demo papers.

INLG, jointly with SIGDIAL, featured four keynote speakers, being:

- Barbara Di Eugenio, University of Illinois, Chicago, USA
- Emmanuel Dupoux, Ecole des Hautes Etudes en Sciences Sociales, France
- Ryan Lowe, OpenAI, USA
- Elena Simperl, King's College London, UK

The Generation Challenge, i.e., a set of shared tasks, was a track of the main conference also this year. It was chaired by Simon Mille. Details about the challenge and the proceedings will appear in a companion proceedings volume.

The main event was preceded by two days of workshops held jointly with SIGDIAL2023, of which two focussed on NLG, being the workshop on "Multimodal, Multilingual Natural Language Generation and Multilingual WebNLG Challenge" and a hackathon on practical "LLM-assisted data-to-text generation".

The event received sponsorship from: Liveperson and Luxai (Platinum), Apple (Gold), Furhat (Silver), and Bloomberg and Ax Semantics (Bronze).

It is also important to mention that the 16th INLG would not be possible without the help of the Area Chairs and Program Committee members for their reviewing contributions for whom we express our gratitude, and the expertise of SIGGEN representatives Raquel Hervás and Emiel van Miltenburg.

C. Maria Keet
Hung-yi Lee
Sina Zarrieß
INLG 2023 Program Chairs

Organizing Committee

Program Chairs

C. Maria Keet (University of Cape Town, South Africa)
Hung-yi Lee (National Taiwan University, Taiwan)
Sina Zarrieß (University of Bielefeld, Germany)

Generation Challenge Chair

Simon Mille (ADAPT Research Centre, Dublin City University, Ireland)

Local Organization Committee

Ondřej Dušek (Charles University, Czech Republic)

SIGGEN Exec

Raquel Hervás (University Complutense of Madrid, Spain)
Emiel van Miltenburg (Tilburg University, the Netherlands)

Publication Chair

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

Sponsor Chair

Ramesh Manuvinakurike (Intel Labs)

Invited Speakers

Barbara Di Eugenio (University of Illinois Chicago, USA)
Emmanuel Dupoux (Ecole des Hautes Etudes en Sciences Sociales, France)
Ryan Lowe (OpenAI, USA)
Elena Simperl (King's College London, UK)

Organizing Committee

Area Chairs

Suma Bhat (University of Illinois at Urbana-Champaign)
Joan Byamugisha (IBM Research)
Brian Davis (Dublin City University)
Albert Gatt (Utrecht University)
Yufang Hou (IBM Research)
Wei-Yun Ma (Academia Sinica)
Lara Martin (University of Maryland)
Samira Shaikh (University of North Carolina)
Kees van Deemter (Utrecht University)
Chris van der Lee (Tilburg University)

Program Committee

Manex Agirrezabal (University of Copenhagen)
Mary-Jane Antia (University of Cape Town)
Vinayshekhar Bannihatti Kumar (AWS AI)
Anya Belz (ADAPT Research Centre, Dublin City University)
Raffaella Bernardi (University of Trento)
Jennifer Biggs (U.S. Naval Research Laboratory)
Nadjet Bouayad-Agha (Universitat Oberta de Catalunya)
Daniel Braun (University of Twente)
Gordon Briggs (U.S. Naval Research Laboratory)
Alberto Bugarín-Diz (Universidad Santiago de Compostela)
Jan Buys (University of Cape Town)
Michele Cafagna (University of Malta)
Deng Cai (Tencent AI Lab)
Eduardo Calò (Utrecht University)
Thiago Castro Ferreira (Federal University of Minas Gerais)
Khyathi Raghavi Chandu (Allen Institute of AI)
Cheng-Han Chiang (National Taiwan University)
Yagmur Gizem Cinar (Amazon)
Elizabeth Clark (Google Research)
Nina Dethlefs (University of Hull)
Simon Dobnik (University of Gothenburg)
Farhood Farahnak (Concordia University)
Nicolas Garneau (Universite Laval)
Pablo Gervás (Universidad Complutense de Madrid)
Martijn Goudbeek (Tilburg University)
Mika Hämäläinen (Rootroo Ltd)
Ting Han (Smartnews Inc.)
Aki Harma (Philips Research)
Hiroaki Hayashi (Salesforce Research)
Philipp Heinisch (Bielefeld University)
Po-chun Hsu (National Taiwan University)
Nikolai Ilinykh (University of Gothenburg)
Takumi Ito (Tohoku University / Langsmith Inc. / Utrecht University)
Mihir Kale (Google)
Zdeněk Kasner (Charles University)
Natthawut Kertkeidkachorn (Japan Advanced Institute of Science and Technology)
Emiel Krahmer (Tilburg University)

Cyril Labbe (Université Grenoble Alpes)
Luc Lamontagne (Laval University)
Maurice Langner (Ruhr-Universität Bochum)
Yucheng Li (University of Surrey)
Yizhi Li (University of Sheffield)
Michela Lorandi (Dublin City University)
Saad Mahamood (Trivago N.V)
Zola Mahlaza (University of Cape Town)
Aleksandre Maskharashvili (Ohio State University)
Kathleen McCoy (University of Delaware)
David McDonald (Smart Information Flow Technologies)
Antonio Valerio Miceli Barone (The University of Edinburgh)
Fatemehsadat Mireshghallah (UC San Diego)
Ryo Nagata (Konan University)
Christina Niklaus (University of St. Gallen)
Avinesh P.V.S (Apple Inc.)
Patrizia Paggio (University of Copenhagen, University of Malta)
Daniel Paiva (Arria NLG)
Suraj Pandey (Open University UK)
Steffen Pauws (Philips Research)
Pablo Perez De Angelis (tuQuejaSuma.com)
Paul Piwek (Open University UK)
François Portet (Université Grenoble Alpes)
Toky Raboanary (University of Cape Town)
Philipp Sadler (University of Potsdam)
Daniel Sanchez (University of Granada)
Sashank Santhanam (University of North Carolina at Charlotte, Apple)
David Schlangen (University of Potsdam)
Simeon Schüz (Bielefeld University)
Balaji Vasan Srinivasan (Adobe Research, India)
Somayajulu Sripada (Arria NLG Plc, University of Aberdeen)
Symon Stevens-Guille (Ohio State University)
Kristina Striegnitz (Union College)
Hsuan Su (National Taiwan University)
Hiroya Takamura (The National Institute of Advanced Industrial Science and Technology)
Ece Takmaz (University of Amsterdam)
Xiangru Tang (Yale University)
Marc Tanti (University of Malta)
Mariët Theune (University of Twente)
Ross Turner (Arria NLG)
Henrik Voigt (Friedrich-Schiller-University)
Di Wang (ContextLogic Inc)
Qingyun Wang (University of Illinois at Urbana-Champaign)
Robert Weißgraeber (CTO @ AX Semantics)
Michael White (Ohio State University)
Yuan-Kuei Wu (National Taiwan University)
Juncheng Xie (National Taiwan University)
Xinnuo Xu (University of Edinburgh)
Bohao Yang (University of Sheffield)
Ziheng Zeng (University of Illinois at Urbana-Champaign)
Zhirui Zhang (Tencent AI Lab)

Zaixiang Zheng (ByteDance AI Lab)
Yinhe Zheng (miHoYo)
Wanzheng Zhu (Google)

Table of Contents

Overview of MiReportor: Generating Reports for Multimodal Medical Images

Xuwen Wang, Hetong Ma, Zhen Guo and Jiao Li
Institute of Medical Information and Library,
Chinese Academy of Medical Sciences and Peking Union Medical College, Beijing, China
li.jiao@imicams.ac.cn

Abstract

This demo paper presents a brief introduction of MiReportor, a computer-aided medical imaging report generator, which leverages a unified framework of medical image understanding and generation to predict readable descriptions for medical images, and assists radiologists in imaging reports writing.

1 Introduction

In the intelligent-assisted diagnosis scenario, computers are required to present reliable interpretation of medical imaging findings. Medical Imaging Report Generation (MIRG) integrates advanced technologies such as computer vision and natural language processing for identifying critical information from medical images and giving reasonable explanations (Messina, 2022). This demo paper presents a brief overview of MiReportor (**M**edical **i**maging **Report** generat**or**), a prototype system designed for computer-aided imaging report writing, open accessed by
`http://mireportor.com`

MiReportor generates fluent imaging reports in both Chinese and English and provides human-computer interaction service for radiomics researchers on image reading, report reviewing and editing.

2 System Overview

The initial design of MiReportor was derived from the unified framework of medical image understanding and generation that we proposed earlier (Wang, 2019). It takes multimodal medical images, such as CT, X-ray, Ultrasound, etc. as input, and predicts related semantic labels as well as brief readable descriptions of radiology findings. In the recent work, we refer to the latest progress on vision-language representation (Feng, 2022) and update our workflow (see Figure 1).

2.1 Medical Image-Text Representation

Well pre-trained medical image-text representation is the basis for generating good descriptions. We selected the visual-language model BLIP (Li, 2022) as our backbone network to build a joint representation model of medical images and texts. We collected open source datasets containing parallel medical image and texts such as ROCO (Pelka, 2018) [1] and MedICaT (Subramanian, 2020)[2], nearly 299K medical image -text pairs for pre-training. We also extracted millions of medical image-text pairs from biomedical literatures and utilized the data bootstrapping strategy suggested by BLIP to filter noisy image-text pairs for optimizing the representation model.

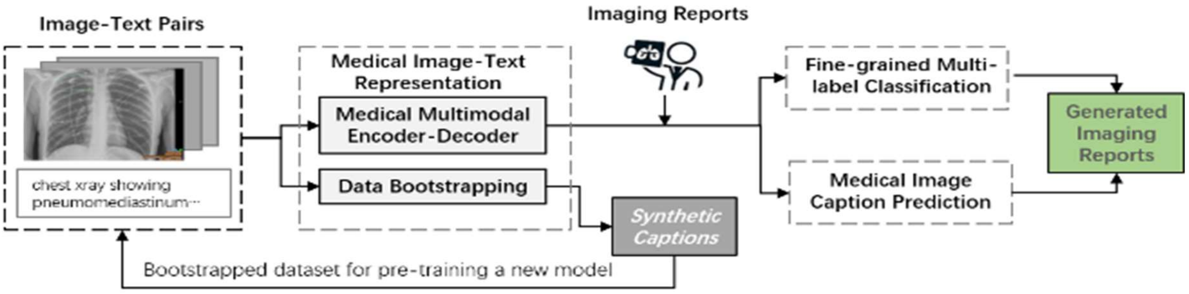


Image-Text Pairs
chest xray showing pneumomediastinum···
Medical Image-Text Representation
Medical Multimodal Encoder-Decoder
Data Bootstrapping
Imaging Reports
Synthetic Captions
Bootstrapped dataset for pre-training a new model
Fine-grained Multi-label Classification
Medical Image Caption Prediction
Generated Imaging Reports

Figure1: Workflow of MiReportor.

[1] https://github.com/razorx89/roco-dataset

[2] https://github.com/allenai/medicat

Proceedings of the 16th International Natural Language Generation Conference: System Demonstrations, pages 1–3
September 11–15, 2023. ©2023 Association for Computational Linguistics

2.2 Fine-grained Multi-label Classification

We measured the potential semantic association between medical images and reports by computing their cross-modal semantic similarities. By referring to medical knowledge systems such as UMLS and MeSH, we performed secondary data annotation on medical image datasets according to medical terms and their semantic types. Then we constructed a transfer learning-based fine-grained multi-label classification model to identify key semantic concepts related to medical images (Wang, 2021).

2.3 Medical Multimodal Encoder-Decoder

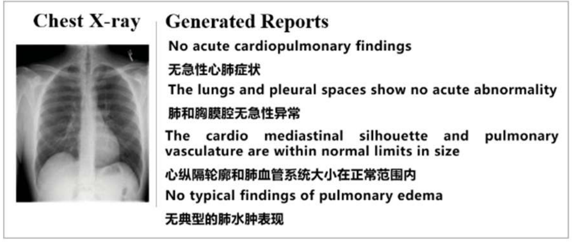
Since general vision-language model is too large to be applied under low resources, we refer to Liu (2021)'s work on Multi-stage Pre-training and proposed an improved Medical Multimodal Encoder-Decoder (MMED) adapted to the medical scenarios. To capture the alignment of medical images and multi-grained texts, MMED was pre-trained in multiple stages with different training tasks and optimizing objectives. More details about MMED are under review for publication, and we will update this module in the future version.

2.4 Medical Image Caption Prediction

Interpretable descriptions of medical images are the basic composition of semi-structured imaging reports. We developed multiple caption prediction models that generate hierarchical texts for multi-modal medical images, including semantic labels, image sentence topics and coherent sentence descriptions. To obtain accurate reports for specific anatomical parts and imaging types, we fine-tuned caption models based on different open-sourced datasets of real medical imaging reports, such as MIMIC-CXR (Johnson, 2019), Chest X-ray (Demner-Fushman, 2016), etc. One of them is TMRGM (Wang, 2021)[3], a chest X-ray report generation model. Further, by connecting the efficient Aliyun translation interface service, multilingual reports can be output.

3 Evaluation

Considering the various linguistic and visual characteristics of different groups of people, we manually annotated a sentence template library of chest X-ray image reports. We used TMRGM to generate image reports for healthy and patient groups respectively. We validated the performance of chest X-ray caption prediction based on the IU Chest X-ray Dataset, see Table 1. An example of X-ray report generated by our system is illustrated in Figure 2. The current demo deployed both TMRGM and a BLIP-based Chest X-ray report generation model. Users can compare and choose the one suitable to their data. More report generation models will be integrated for other imaging types and body parts in the future.



No acute cardiopulmonary findings

无急性心肺症状

The lungs and pleural spaces show no acute abnormality

肺和胸膜腔无急性异常

The cardio mediastinal silhouette and pulmonary vasculature are within normal limits in size

心纵隔轮廓和肺血管系统大小在正常范围内

No typical findings of pulmonary edema

无典型的肺水肿表现

Figure2: An example of imaging report generated by MiReportor.

Method	B1	B2	B3	B4	MT	RG	CD
TieNet (Wang, 2018)	0.286	0.160	0.104	0.074	0.108	0.226	--
CoAtt (Jing, 2018)	0.303	0.181	0.121	0.084	0.132	0.249	0.175
Adapt-att	0.378	0.255	0.185	0.138	0.162	**0.316**	**0.387**
BLIP	0.394	0.232	0.154	0.109	0.167	0.315	0.257
TMRGM	**0.419**	**0.281**	**0.201**	**0.145**	**0.183**	0.280	0.359

Table 1 : Preliminary results of Chest X-ray Report Generation, in which B1 to B4 refer to BLEU score, MT refers to METEOR, RG refers to ROUGE, and CD refers to CIDEr.

4 Conclusions

This paper briefly introduces MiReportor, a prototype system for interactive generation of medical imaging reports in both Chinese and English. Experiments based on public chest X-rays revealed the ability of computers on understanding and interpreting medical images. It facilitates the human-computer collaboration practice of imaging

[3] https://github.com/zhangyudoc/TMRGM

diagnosis, which may contribute to the efficient communication between radiologist, clinicians, and patients.

Acknowledgments

This work has been supported by the National Natural Science Foundation of China (Grant No. 61906214), the Beijing Natural Science Foundation (Grant No. Z200016), the CAMS Innovation Fund for Medical Sciences (CIFMS, Grant No.2021-I2M-1-056）

References

Pablo Messina, Pablo Pino, Denis Parra, Alvaro Soto, Cecilia Besa, Sergio Uribe, Marcelo andía, Cristian Tejos, Claudia Prieto, Daniel Capurro (2022). A survey on deep learning and explainability for automatic report generation from medical images. ACM Computing Surveys (CSUR), 54(10s), 1-40.

Xuwen Wang, Yu Zhang, Zhen Guo, and Jiao Li. 2019. A Computational Framework Towards Medical Image Explanation. In Artificial Intelligence in Medicine: Knowledge Representation and Transparent and Explainable Systems: AIME 2019 International Workshops, KR4HC/ProHealth and TEAAM, Poznan, Poland, June 26–29, 2019. Springer-Verlag, Berlin, Heidelberg, 120–131.

Li, Feng, Hao Zhang, Yi-Fan Zhang, Shi Tong Liu, Jian Guo, Lionel Ming-shuan Ni, Pengchuan Zhang and Lei Zhang. Vision-Language Intelligence: Tasks, Representation Learning, and Large Models. https://doi.org/10.48550/arXiv.2203.01922

LI, Junnan, LI, Dongxu, XIONG, Caiming, et al. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In : International Conference on Machine Learning. PMLR, 2022. p. 12888-12900.

Sanjay Subramanian, Lucy Lu Wang, Sachin Mehta, Ben Bogin, Madeleine van Zuylen, Sravanthi Parasa, Sameer Singh, Matt Gardner, Hannaneh Hajishirzi. MedICaT: A Dataset of Medical Images, Captions, and Textual References,2020

O. Pelka, S. Koitka, J. Rückert, F. Nensa und C. M. Friedrich. Radiology Objects in COntext (ROCO): A Multimodal Image Dataset, Proceedings of the MICCAI Workshop on Large-scale Annotation of Biomedical data and Expert Label Synthesis (MICCAI LABELS 2018), Granada, Spain, September 16, 2018, Lecture Notes in Computer Science (LNCS) Volume 11043, Page 180-189.

Xuwen Wang, Zhen Guo, Chunyuan Xu, Lianglong Sun and Jiao Li. ImageSem Group at ImageCLEFmed Caption 2021 Task: Exploring the Clinical Significance of the Textual Descriptions Derived from Medical Images. CEUR Workshop Proceedings (CEUR-WS.org), CLEF 2021 Conference and Labs of the Evaluation Forum, September 21–24, 2021, Bucharest, Romania

Tongtong Liu, Fangxiang Feng, and Xiaojie Wang. 2021. Multi-stage Pre-training over Simplified Multimodal Pre-training Models. 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 2556–2565, Online. Association for Computational Linguistics

Johnson AE, Pollard TJ, Berkowitz SJ, Greenbaum NR, Lungren MP, Deng CY, Mark RG, Horng S. MIMIC-CXR, a de-identified publicly available database of chest radiographs with free-text reports. Scientific Data. 2019;6.

Demner-Fushman D, Kohli MD, Rosenman MB, Shooshan SE, Rodriguez L, Antani S, Thoma GR, McDonald CJ. Preparing a collection of radiology examinations for distribution and retrieval. J Am Med Inform Assoc. 2016 Mar;23(2):304-10.

Xuwen Wang, Yu Zhang, Zhen Guo, Jiao Li. TMRGM: A Template-Based Multi-Attention Model for X-Ray Imaging Report Generation. Journal of Artificial Intelligence for Medical Sciences, Volume 2, Issue 1-2, June 2021, Pages 21 - 32

Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu and Ronald M. Summers, TieNet: Text-Image Embedding Network for Common Thorax Disease Classification and Reporting in Chest X-Rays, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 9049-9058, doi: 10.1109/CVPR.2018.00943

Baoyu Jing, Pengtao Xie, Eric Xing, 2018. On the automatic generation of medical imaging reports, In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2577–2586, Melbourne, Australia. Association for Computational Linguistics.

`enunlg`: a Python library for reproducible neural data-to-text experimentation

David M. Howcroft and **Dimitra Gkatzia**
Edinburgh Napier University
{D.Howcroft,D.Gkatzia}@napier.ac.uk

Abstract

Over the past decade, a variety of neural architectures for data-to-text generation (NLG) have been proposed. However, each system typically has its own approach to pre- and post-processing and other implementation details. Diversity in implementations is desirable, but it also confounds attempts to compare model performance: are the differences due to the proposed architectures or are they a byproduct of the libraries used or a result of pre- and post-processing decisions made? To improve reproducibility, we re-implement several pre-Transformer neural models for data-to-text NLG within a single framework to facilitate direct comparisons of the models themselves and better understand the contributions of other design choices. We release our library at https://github.com/NapierNLP/enunlg to serve as a baseline for ongoing work in this area including research on NLG for low-resource languages where transformers might not be optimal.

1 Introduction

Dozens of different models for neural data-to-text generation have been proposed in the last decade, before we even consider recent efforts to repurpose large language models for data-to-text natural language generation (NLG). However, these models vary greatly with respect to both low-level and high-level design choices, requiring different kinds of *delexicalisation* and normalisation processes, different ways of encoding and tracking meaning, and using a variety of neural network libraries, among other differences. While we can use the outputs of individual models released by their authors to assess the relative performance of these implementations, there is little work aiming to explore which performance differences are due to the proposed architectures themselves as opposed to other implementation details. In order to explore these differences, encourage reproduction experiments, and provide tools for teaching data-to-text NLG, we developed a Python library implementing several of these models in a common framework.

Datasets	Models
WEN	SCLSTM (described)
E2E Challenge	SCLSTM (released)
Cleaned E2E	TGEN
WEBNLG	CHECKLIST *
NMETHODIUS	CHARSCLSTM *

Table 1: Datasets & models implemented in `enunlg`. *Indicates a model whose implementation is in-progress.

2 `enunlg`: extensible NLG library

Our `enunlg` library is developed for Python 3.9 with PyTorch 1.9.1. In addition to implementing the models themselves, we provide a variety of file readers & writers to consume different corpora and convert them into appropriate representations for each model. At present, we have tools in place to work with the WEN datasets (dialog system responses for restaurant, hotel, laptop, and TV descriptions: Wen et al., 2016), cleaned data from the E2E Challenge (restaurant descriptions: Novikova et al., 2017; Dušek et al., 2019), the NMETHODIUS corpus (museum exhibits: Stevens-Guille et al., 2020), and WEBNLG (Gardent et al., 2017).

Meaning representation (MR) parsers are included for CUED dialogue acts, E2E slot-value pairs, and RDF triples. Supported neural representations for these MR types include bit-vectors, flattened trees, and unbracketed sequences of triples. Word embeddings can be randomly initialised or loaded from existing vectors.

We reimplement the SCLSTM model proposed by (Wen et al., 2015), originally implemented using Theano and Python 2. During reimplementation, we found that the codebase released with the paper implemented a different architecture from what was described in the paper, so we provide both versions

Proceedings of the 16th International Natural Language Generation Conference: System Demonstrations, pages 4–5
September 11–15, 2023.

in our library. We also provide a reimplementation of TGEN (Dušek and Jurčíček, 2016), originally implemented using Tensorflow 0.6. Kiddon et al. (2016) implemented their CHECKLIST model in Lua with Torch and Deriu and Cieliebak (2018) used Tensorflow 1.10.0 for their CHARSCLSTM.

3 Planned uses

Our goals in developing `enunlg` fall into three broad categories: reproducibility, pedagogy, and easy experimentation. By enabling the use of a single framework with consistent reference implementations of multiple models, the library promotes reproducibility and facilitates fair comparisons, controlling for differences in, e.g., delexicalisation, tokenisation, neural network libraries, etc. A small, consistent codebase that addresses the different elements of implementing neural data-to-text systems also serves a pedagogical function, providing a starting point for student projects. Finally, our design choices aim to make engineering experiments trivial (e.g. hyperparameter search, changing tokenisation, etc) and scientific experiments easy (e.g. developing new end-to-end and pipeline systems for neural NLG) and promote work in low-resource NLG (Howcroft and Gkatzia, 2022).

4 Conclusions

We present `enunlg`, a library for reproducible experimentation in neural data-to-text generation. The code is available from `https://github.com/NapierNLP/enunlg`. We hope that the availability of an extensible library for neural NLG will improve reproducibility in our research community and provide a new set of reference implementations for baseline models.

Acknowledgements

The research was supported under the EPSRC project ’NLG for low-resource domains’ (EP/T024917/1).

References

Jan Milan Deriu and Mark Cieliebak. 2018. End-to-end trainable system for enhancing diversity in natural language generation. In *Proc. of the E2E NLG Challenge System Descriptions*.

Ondřej Dušek, David M. Howcroft, and Verena Rieser. 2019. Semantic noise matters for neural natural language generation. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 421–426, Tokyo, Japan. Association for Computational Linguistics.

Ondřej Dušek and Filip Jurčíček. 2016. Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 45–51, Berlin, Germany. Association for Computational Linguistics.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for NLG micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–188, Vancouver, Canada. Association for Computational Linguistics.

David M. Howcroft and Dimitra Gkatzia. 2022. Most NLG is low-resource: here’s what we can do about it. In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 336–350, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Chloé Kiddon, Luke Zettlemoyer, and Yejin Choi. 2016. Globally coherent text generation with neural checklist models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 329–339, Austin, Texas. Association for Computational Linguistics.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 201–206, Saarbrücken, Germany. Association for Computational Linguistics.

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

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, and Steve Young. 2016. Multi-domain neural network language generation for spoken dialogue systems. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 120–129, San Diego, California. Association for Computational Linguistics.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721, Lisbon, Portugal. Association for Computational Linguistics.

VisuaLLM: Easy Web-based Visualization for Neural Language Generation

František Trebuňa and **Ondřej Dušek**
Charles University, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Prague, Czechia
ferotre@gmail.com, odusek@ufal.mff.cuni.cz

Abstract

VisuaLLM is a Python library that enables interactive visualization of common tasks in natural language generation with pretrained language models (using HuggingFace's model API), with tight integration of benchmark datasets and fine-grained generation control. The system runs as a local generation backend server and features a web-based frontend, allowing simple interface configuration by minimal Python code. The currently implemented views include data visualization, next-token prediction with probability distributions, and decoding parameter control, with simple extension to additional tasks.

```
ntp = NextTokenPredictionComponent(
    model=model,    # HuggingFace model object
    dataset=dataset # Huggingface dataset object
)
gen = InteractiveGenerationComponent(
    model=model,
    dataset=dataset,
    selectors={"num_beams": (1, 20)},
    metrics={"perplexity": Perplexity}
)
vis = DatasetVisualizationComponent(
    dataset=dataset
)
app = Server(
    __name__,
    components=[ntp, gen, vis]
).app

>> flask run
```

Figure 1: Example setup code for VisuaLLM.

1 Introduction

While pretrained language models (PLMs) reached state-of-the-art performance on many natural language generation (NLG) benchmarks (Kale and Rastogi, 2020; Ribeiro et al., 2020; Xiang et al., 2022), they are hard to control directly and are often used as a black-box architecture, where the developer feeds linearized training data and retrieves outputs in a batch-wise manner, without access to fine-grained model behavior. Interfaces for interactive PLM testing (such as the OpenAI playground[1] or the HuggingFace website)[2] typically only allow very basic operation, do not show any information beyond inputs/prompts and outputs, and are not connected to benchmark datasets.

This paper presents VisuaLLM, a simple, extensible library for visualization of PLM generation processes, built as a web-based frontend on top of the HuggingFace Transformers and Datasets frameworks (Wolf et al., 2020), taking inspiration from generic analysis tools such as TensorBoard[3] or WandB.[4] The current version allows to visualize tabular NLG datasets and the processes of low-level next-token prediction and high-level generation, with easily adjustable settings and tight benchmark dataset and metrics integration. The user can easily explore and evaluate PLMs in an interactive way, thus gaining insight into model behavior and being able to tune models more effectively. VisuaLLM is designed to be easily modified for different NLG tasks, where the user only picks a choice of datasets, models, adjustable decoding parameters and metrics. The whole interface is easily customizable via minimal Python code and can be extended to other NLG tasks. VisuaLLM can be installed by running `pip install visuallm`.[5]

2 System Architecture

We build on HuggingFace Transformers (Wolf et al., 2020) as the most commonly used PLM framework. We expect the programmer to load HuggingFace model and dataset objects and pass them to our framework as shown in Figure 1. VisuaLLM is used as a local server running on the user's machine, similar to e.g. TensorBoard. The code is divided into a web-based frontend (written

[1] https://platform.openai.com/

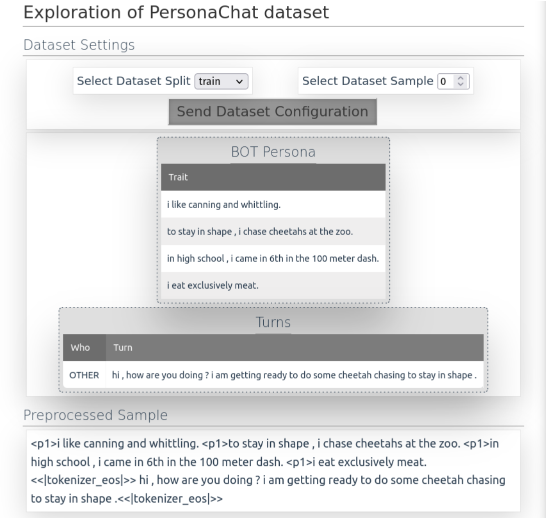
[2] https://huggingface.co/models

[3] https://www.tensorflow.org/tensorboard

[4] https://wandb.ai/

[5] Source code is available on GitHub at https://github.com/gortibaldik/visuallm. A demonstration screencast is shown at https://youtu.be/RMFEEW-Iu-4.

Proceedings of the 16th International Natural Language Generation Conference: System Demonstrations, pages 6–8
September 11–15, 2023. ©2023 Association for Computational Linguistics



Exploration of PersonaChat dataset

Dataset Settings

Select Dataset Split: train

Select Dataset Sample: 0

BOT Persona

Trait
i like canning and whittling.
to stay in shape , i chase cheetahs at the zoo.
in high school , i came in 6th in the 100 meter dash.
i eat exclusively meat.

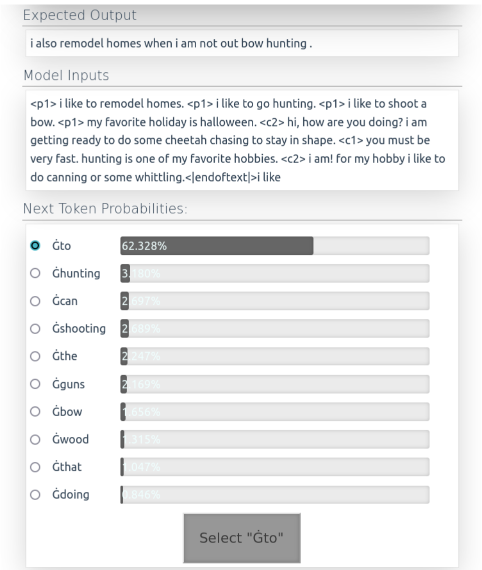
Turns

Who	Turn
OTHER	hi , how are you doing ? i am getting ready to do some cheetah chasing to stay in shape .

Preprocessed Sample

<p1>i like canning and whittling. <p1>to stay in shape , i chase cheetahs at the zoo. <p1>in high school , i came in 6th in the 100 meter dash. <p1>i eat exclusively meat. <<|tokenizer_eos|>> hi , how are you doing ? i am getting ready to do some cheetah chasing to stay in shape .<<|tokenizer_eos|>>

Figure 2: A view of a data sample from PersonaChat.



Expected Output

i also remodel homes when i am not out bow hunting .

Model Inputs

<p1> i like to remodel homes. <p1> i like to go hunting. <p1> i like to shoot a bow. <p1> my favorite holiday is halloween. <c2> hi, how are you doing? i am getting ready to do some cheetah chasing to stay in shape. <c1> you must be very fast. hunting is one of my favorite hobbies. <c2> i am! for my hobby i like to do canning or some whittling.<|endoftext|>i like

Next Token Probabilities:

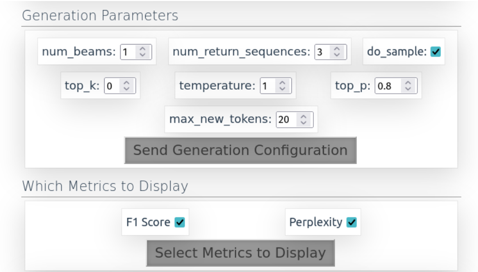
●	Ġto	62.328%
○	Ġhunting	3.180%
○	Ġcan	2.697%
○	Ġshooting	2.689%
○	Ġthe	2.247%
○	Ġguns	2.169%
○	Ġbow	1.656%
○	Ġwood	1.315%
○	Ġthat	1.047%
○	Ġdoing	0.846%

Figure 3: Next-token prediction visualization.

in Vue.js) and a Python backend (based on Flask). The frontend is built in a modular fashion, 100% configurable from Python code on the backend. Any frontend-backend communication is therefore abstracted away from the user. The whole setup is designed to use as little code as possible; customizing for any new HuggingFace-based model or task takes typically just a few dozen lines of code.

3 Usecases

We demonstrate VisuaLLM by visualizing dialogue generation on the PersonaChat benchmark (Zhang et al., 2018). The presented views can easily be extended or modified for other NLG tasks. For instance, we are currently working on an extension to allow interactive user input.



Generation Parameters

num_beams: 1

num_return_sequences: 3

do_sample: ☑

top_k: 0

temperature: 1

top_p: 0.8

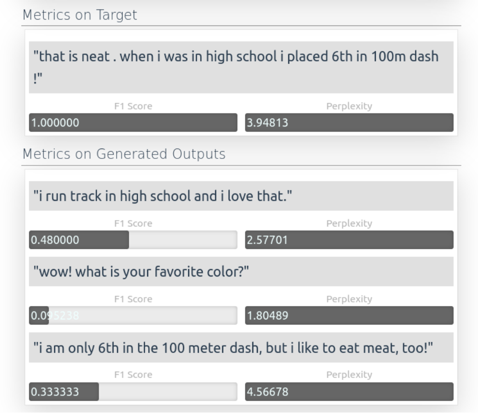
max_new_tokens: 20

Which Metrics to Display

F1 Score ☑

Perplexity ☑

Figure 4: Generation parameters and metrics control.



Metrics on Target

"that is neat . when i was in high school i placed 6th in 100m dash !"

F1 Score: 1.000000

Perplexity: 3.94813

Metrics on Generated Outputs

"i run track in high school and i love that."

F1 Score: 0.480000

Perplexity: 2.57701

"wow! what is your favorite color?"

F1 Score: 0.095238

Perplexity: 1.80489

"i am only 6th in the 100 meter dash, but i like to eat meat, too!"

F1 Score: 0.333333

Perplexity: 4.56678

Figure 5: Outputs visualization with metrics.

Dataset Visualization Figure 2 shows a single data instance from PersonaChat. The dataset assumes a short persona description and preceding dialogue context as inputs for response generation. The interface shows both a tabular human-readable representation and a low-level linearized model input (configured for a specific trained model).

Next Token Prediction Visualization Visualizing next-token probability distributions is another common debugging task for PLMs. In Figure 3, we show that VisuaLLM allows the user to interactively step through sequence generation and control which token is selected, showing next-token probability distributions at each step. This view is also configured to include low-level model inputs and human reference outputs.

Generation Visualization This view is more high-level than the previous, exploring outputs generated with different decoder settings and their automatic metric scores. Controls in Figure 4 directly translate to HuggingFace's `generate()` method parameters and allow any callable Python metrics to be used, with configurable display of the metric values. Multiple generated outputs (using different settings) can then be compared to the reference, as shown in Figure 5.

Acknowledgements

This work was supported by the project TL05000236 *AI asistent pro žáky a učitele* co-financed by the Technological Agency of the Czech Republic within the ÉTA 5 Programme, and by the European Research Council (Grant agreement No. 101039303 NG-NLG). It used resources provided by the LINDAT/CLARIAH-CZ Research Infrastructure (Czech Ministry of Education, Youth and Sports project No. LM2018101).

References

Mihir Kale and Abhinav Rastogi. 2020. Text-to-Text Pre-Training for Data-to-Text Tasks. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 97–102, Dublin, Ireland. Association for Computational Linguistics.

Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2020. Investigating Pretrained Language Models for Graph-to-Text Generation. *arXiv:2007.08426 [cs]*. ArXiv: 2007.08426.

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. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv:1910.03771 [cs]*. ArXiv: 1910.03771.

Jiannan Xiang, Zhengzhong Liu, Yucheng Zhou, Eric P. Xing, and Zhiting Hu. 2022. ASDOT: Any-Shot Data-to-Text Generation with Pretrained Language Models. In *Findings of EMNLP*, Abu Dhabi, UAE. arXiv. ArXiv:2210.04325 [cs].

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.

Audio Commentary System for Real-Time Racing Game Play

Tatsuya Ishigaki[†] **Goran Topić**[†] **Yumi Hamazono**[†] **Ichiro Kobayashi**[†°]
Yusuke Miyao[†‡] **Hiroya Takamura**[†]
[†]National Institute of Advanced Industrial Science and Technology, Japan,
[°]Ochanomizu University, [‡]The University of Tokyo,
{ishigaki.tatsuya, goran.topic, hamazono-yumi, takamura.hiroya}@aist.go.jp,
koba@is.ocha.ac.jp, yusuke@is.s.u-tokyo.ac.jp

Abstract

We introduce a live audio commentator system designed specifically for a racing game, driven by the high demand in the e-sports field. While a player is playing a racing game, our system tracks real-time user play data including speed and steer rotations, and generates commentary to accompany the live stream. The human evaluation suggested that generated commentary enhances enjoyment and understanding of races compared to streams without commentary. Incorporating additional modules to improve diversity and detect irregular events, such as course-outs and collisions, further increases the preference for output commentaries.

1 Introduction

Live commentary enriches the spectator's experience in sports events and e-sports streams, but it is often unavailable for online videos or recordings of amateur sports due to a lack of skilled commentators. Automatic generation techniques offer a potential solution to this problem.

Previous works focus on generating text-based commentaries using pre-stored tracked data (Puduppully and Lapata, 2021). In contrast, we present a real-time automatic system for generating commentaries, specifically targeting race games inspired by the growing e-sports industry. Live commentary generation typically involves tweet extraction (Kubo et al., 2013), rule-based and keyword extraction from videos (Kim and Choi, 2020), and neural network–based data-to-text approaches (Ishigaki et al., 2021; Taniguchi et al., 2019). Our system combines utterance extraction and neural network–based methods.

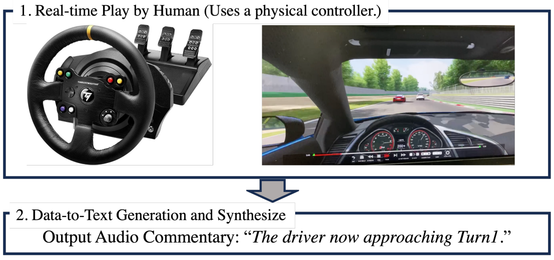

1. Real-time Play by Human (Uses a physical controller.)
2. Data-to-Text Generation and Synthesize
Output Audio Commentary: “*The driver now approaching Turn1*.”

Figure 1: The workflow of our demo. a) user plays a racing game using a physical controller; b) our system generates commentary by analyzing the race situation.

Our system works with a physical controller for real-time gameplay in a racing game (Assetto Corsa). During gameplay, the system tracks the user's data, such as speed, steering rotation, and lap progress. Then, the system generates candidates by a neural network–based generator and ranks them in terms of diversity. We also address the problem of limited coverage of rare events in the existing commentary generator. To mitigate these problems, our system detects course-outs and collisions and selects appropriate expressions from a predefined list of utterances. The current generation model is trained on an open Japanese dataset (Ishigaki et al., 2021), although it can be replaced with an English one.[1]

We conducted an evaluation by human judges in terms of enjoyment. The results suggest that: 1) commentary usage enhances user immersion, 2) diversity is important for improving the overall quality of synthesized commentaries, and 3) identifying irregular events further gains the quality.

2 Architecture

Our system consists of five modules in a pipeline. **1: Real-time Data Tracker** Assetto Corsa allows custom functionality to be added to the game via plugins. Thus, we develop a plugin which captures the relevant play data from the game's API. The data is sent to our data processing server. The

[1] An English dataset is also ready. We present our demo in English at the venue.

Proceedings of the 16th International Natural Language Generation Conference: System Demonstrations, pages 9–10
September 11–15, 2023. ©2023 Association for Computational Linguistics

server samples several features i.e., speed or other metrics, every 0.1 s and keeps the samples in a 10 s moving window. The collected features are sent to the candidate generator as a set of 100-element vectors.

2: Candidate Generator We extend an existing model (Ishigaki et al., 2021), a multi-modal generator for textual and numerical inputs. For textual input, we consider the previous L utterances (we use $L = 5$), while for numerical input, we use vectors produced from the real-time data tracker. Textual input is encoded with BART's encoder (*stockmark/bart-base-japanese-news*), while the original used LSTM. We use an MLP-based encoder for numerical inputs to obtain 728-sized vectors. Another BART encoder processes two types of embeddings: 1) embeddings of the initial token in textual input, and 2) encoded tracked data. BART's decoder generates an utterance to get k candidates by a beam search.

3: Utterance Selection In our preliminary experiments, repetition of the same or similar utterances led to a decline in naturalness. Thus, our system selects the candidate u_{cand} least similar to the previous L utterances u_i. The similarity is calculated as the exponentially weighted sum: $\sum_{i=1}^{L}(1-\alpha)^{L-i}\mathrm{Sim}(u_i, u_{cand})$, where $\alpha = 0.2$. We use BLEU as $\mathrm{Sim}()$.

4: Detection of Irregular Events To generate utterances about irregular events, such as course-outs or collisions, we use an extraction-based approach. The car is assigned a road position value, with 1.0 and -1.0 indicating the center point is at the right and left edge respectively. A course-out is indicated when this value falls beyond a threshold (set here at ± 0.9) Collisions are identified based on the distance to the nearest car, with a distance of less than five meters indicating a collision. When an irregular event is detected, we randomly select an utterance from a predefined list manually created from the training dataset.

5: Text-to-Speech (TTS) The utterance text is sent to VOICEVOX for TTS synthesis[2]. The obtained audio clip is finally played back to the user.

3 Experiments

Training: The candidate generator is trained with 34,897 gold utterance tuples, including previous utterances and tracked numerical data. Validation is performed using 12,295 tuples. The batch size is 5, and AdamW optimizer is used with a learning rate of 10^{-5}. The best model is selected based on cross entropy loss on the validation set.

Evaluation: We assess spectator immersion in synthesized commentary and compare different commentaries. Three models are compared: 1) using the best utterance in beam search, 2) incorporating a diversity module, and 3) combining the diversity and irregular event detection modules. Four human evaluators rank these models in terms of enjoyment as an audience.

4 Results

All the human judges agree that synthesized commentaries enhance immersion. In terms of enjoyment, the commentaries generated by the model with both diversity and irregular detection modules are ten times out of twelve judged better than the model that outputs the best utterance in beam search. This result suggests that these two modules are effective. The model that uses both diversity and irregular event detection modules was eight times out of twelve judged better than the model with only diversity models. Thus, the irregular event detection helps to improve quality.

5 Conclusion

We introduced a commentator for racing games, which generates real-time commentary based on tracked metrics. Future possibilities include utilizing dialogue-styled commentary or enhancing live streams with explanatory graphics. [3]

References

Ishigaki, T., Topic, G., Hamazono, Y., Noji, H., Kobayashi, I., Miyao, Y., and Takamura, H. 2021. Generating racing game commentary from vision, language, and structured data. *INLG*, pp. 103–113.

Kim, B. J. and Choi, Y. 2020. Automatic baseball commentary generation using deep learning. *ACM Sympo. on Applied Computing*, pp. 1056–1065.

Kubo, M., Sasano, R., Takamura, H., and Okumura, M. 2013. Generating live sports updates from twitter by finding good reporters. *WI-IAT*, vol. 1, pp. 527–534.

Puduppully, R. and Lapata, M. 2021. Data-to-text generation with macro planning. *TACL*, 9:510–527.

Taniguchi, Y., Feng, Y., Takamura, H., and Okumura, M. 2019. Generating live soccer-match commentary from play data. *AAAI*, vol. 33, pp. 7096–7103.

[2] https://github.com/VOICEVOX/

[3] This paper is based on results obtained from a project JPNP20006, commissioned by NEDO.

Author Index