INLG 2022

International Natural Language Generation Conference (INLG 2022)

Proceedings of the 15th Conference on Natural Language Generation: System Demonstrations

July 18-22, 2022

The INLG organizers gratefully acknowledge the support from the following sponsors.

Gold

ARRIA
NATURAL LANGUAGE GENERATION

Google

Silver

Bronze

aiXplain

Order copies of this and other ACL proceedings from:

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

ISBN 978-1-955917-60-5

Organizing Committee

Program Chair

Samira Shaikh
Thiago Castro Ferreira

Local Chair

Amanda Stent

Invited Speakers

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

SIGGEN Representatives

Ehud Reiter
Emiel Van Miltenburg

Tutorial and Hackathon Chair

Joshua Maynez

Publication Chair

Miruna Clinciu

Sponsor Chair

Dave Howcroft

Social Media Chair

Luou (Lilly) Wen

Area Chairs

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

Tirthankar Ghosal
Yufang Hou

Local Organizing Team

Amy Poulin
Charlotte Buswick
Jake Rogers

Program Committee

Program Committee

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

Table of Contents

BLAB Reporter: Automated journalism covering the Blue Amazon

Yan Vianna Sym
Escola Politécnica
Universidade de São Paulo
São Paulo, Brazil
yan.sym@usp.br

João Gabriel Moura Campos
Escola Politécnica
Universidade de São Paulo
São Paulo, Brazil
joaogcampos@usp.br

Fabio Gagliardi Cozman
Escola Politécnica
Universidade de São Paulo
São Paulo, Brazil
fgcozman@usp.br

Abstract

This demo paper introduces BLAB Reporter, a robot-journalist system covering the Brazilian Blue Amazon. The application is based on a pipeline architecture for Natural Language Generation, which offers daily reports, news summaries and curious facts in Brazilian Portuguese. By collecting, storing and analysing structured data from publicly available sources, the robot-journalist uses domain knowledge to generate, validate and publish texts in Twitter. Code and corpus are publicly available [1].

1 Introduction

Data-to-text Natural Language Generation (NLG) is the computational process of generating meaningful and coherent natural language in the form of text or speech to describe non-linguistic input data (Reiter and Dale, 2000). Successful examples of data-to-text systems can be found in both academia and industry, with applications in weather forecasting (Belz, 2008), image captions and chatbots (Adamopoulou and Moussiades, 2020). Within the range of NLG applications, robot-journalism is one of the most prominent endeavors thanks to the high volume of structured data streams available, which enables automated systems to report recurrent information with high-fidelity and lexical variety (Teixeira et al., 2020).

An interesting domain for data-to-text generation is ocean monitoring. For instance, global attention was drawn in 2021 to a container ship that obstructed the Suez Canal for six consecutive days. The result was a global shortage of essential commodities, including medical supplies and medicines, which were essential during the coronavirus pandemic (COVID-19). Accurate and low latency information reports can be very helpful in these situations, but communicating to general audiences in a accessible way usually demands coverage by specialized human journalists. To address this issue, we present our robot-journalist named *BLAB Reporter*, a NLG system based on a pipeline architecture that generates daily reports, news, content summarization and curious facts about the Blue Amazon and publishes them on Twitter in Brazilian Portuguese [2]. The Blue Amazon is the exclusive economic zone (EEZ) of Brazil, with an offshore area of 3.6 million square kilometers along the Brazilian coast, an area rich in marine biodiversity and energy resources (Wiesebron, 2013). The BLue Amazon Brain (BLAB) is a project aiming to address complex questions about the marine ecosystem, and integrates a number of services aimed at disseminating information about the Blue Amazon region and its importance.

2 System overview

Our system follows a pipeline architecture that converts non-linguistic data into text in 6 steps: *Content Selection, Discourse Ordering, Text Structuring, Lexicalization, Referring Expression Generation and Textual Realization* (Ferreira et al., 2019). Our system also comprises two additional steps: *Data Acquisition*, responsible for extracting and storing information from multiple data streams in a structured format, and *Summarization*, responsible for summarizing news in the form of small consecutive tweets. This kind of architecture, depicted in Figure 1, allows for trustworthy output as well as easy access to and maintenance of sub-modules.

The grammar used by the model was built by first running the content selection step in previous data and generating 30 non-linguistic reports. These non-linguistic reports were then manually verbalized and the input and output representations for each pipeline module were manually annotated. When deployed, each module draws on the selected combination of templates using rule-based approaches. Because we deal with a sensitive do-

[1] https://github.com/C4AI/blab-reporter

[2] https://twitter.com/BLAB_Reporter

Proceedings of the 15th International Conference on Natural Language Generation: System Demonstrations, pages 1 - 3

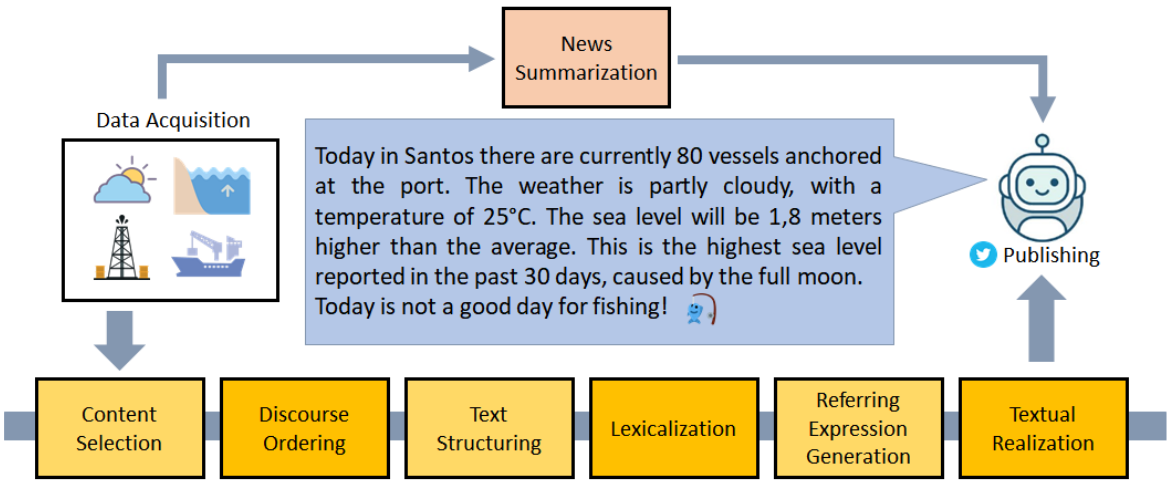

Figure 1: Robot-journalist system architecture

main, we opted to use the pipeline architecture instead of the novel end-to-end systems, which sometimes hallucinates content (Ji et al., 2022). The following sections describe each module.

Data Acquisition The first step of our system performs data gathering, filtering and cleaning before it is put in a data warehouse. In our application, this module consists of a web scraping framework for extracting information from public websites and storing it on a structured format. Our system currently collects information about weather, tide charts, marine vessel traffic and eventual earthquakes on the Brazilian coast, and stores data using MongoDB, a source-available cross-platform document-oriented database program (Győrödi et al., 2015).

Content Selection This module decides which relevant information should be verbalized in the text. The content selection process generally consists of applying domain specific knowledge to create a rule-based approach. The following text is an example of the content selection module output:

```
CURRENT WEATHER AND
TEMPERATURE (weather="partly
cloudy",temperature="25°C",city="Santos",
timestamp="May 22, 2022); FISHING
CONDITION (condition="good",event="sea
level is high";height of the sea:"1.8
meters";days since last peak="30");
CAUSE(earthquake:"no",moon
calendar:"yes");
```

Discourse Ordering and **Text Structuring** Once the relevant content has been selected, our application constructs a logical timeline of events and sorts the intent messages in sentences and paragraphs in order to enhance reader comprehension (Heilbron et al., 2019). This combined module is based on a list of possible intent orderings collected from the corpus and decides what is the most optimal way to sort the sentences, bearing in mind the 280 character limit on Twitter. For example, for the messages related to weather conditions, a possible outcome order would be:

```
WEATHER ALERT → CAUSES → DAYS
SINCE LAST PEAK
```

Lexicalization At this step of the pipeline, lexical choices are made in order to verbalize the intents, finding the proper words to generate proper sentences. We applied a template-based lexicalization with plenty of options to choose from, providing for more inflections and variety of text in comparison to the fill-template approach (Stede, 1994). The templates provide for gender and number inflection, for example: *"No Rio de Janeiro foi registrada a maior temperatura da última semana"* vs. *"Em São Paulo foi registrado o maior vento dos últimos 10 dias"*.

Referring Expression Generation In order to replace entity tags throughout the template, this module generates the appropriate references using a list of possible expressions for each entity (Krahmer and Van Deemter, 2012). For the first reference to an entity in the text, a full description is used (e.g., INSTITUTE → "The Seismological Center at the University of São Paulo (USP)"), whereas for subsequent references a random referring expression to the entity is chosen (e.g., INSTITUTE →

"The Seismological Center at USP"; "the Institute"; "sismoUSP"; "it"; etc.).

Textual Realization The last step of the pipeline is responsible for transforming intermediate representations into human-readable Brazilian Portuguese text. After the content is generated, this step applies a final rule based transformation to the text with the goal to make the texts look more natural, for example adding greetings message and emojis. We also added a validation layer in this step, to ensure there is no offensive content within the text. The output of this module is published using Twitter's API. An example of generated text is shown in Figure 1.

Summarization An extra module was implemented in our system in order to outline public news about the Blue Amazon while also splitting text into small consecutive tweets. Because data hallucination is less critical in this step, this module was implemented using PPT5, a T5 model pretrained in a large collection of web pages in Portuguese, which uses state of the art transformer architecture (Carmo et al., 2020). Key challenges of this approach are interpretation and evaluation of the generated texts (Rao and Gudivada, 2018).

The generated texts are scheduled to be published on specific periods of time. We noticed that weather related content has more user engagement during mornings, while news and curious facts content are usually more viewed in the evenings. More critical messages, for example information related to earthquakes in the Blue Amazon region, are published as soon as the data is collected and stored in the database.

3 Conclusions

This paper presents a data-to-text system based on a pipeline architecture for NLG. Our system applies robot-journalism techniques to generate and publish reports, news and curious facts in Brazilian Portuguese about the Blue Amazon. Due to its rule-base nature, our system provides high-fidelity content by applying a pipeline methodology and obtains lexical variety by drawing from a list of multiple available template options for the same intent. In the future we plan to add more sources of information to the pipeline, for example statistics about oil exploration and reporting of illegal fishing activities in real time. We also plan to utilize user engagement data and apply artificial neural network techniques to improve our system's performance.

References

Eleni Adamopoulou and Lefteris Moussiades. 2020. An overview of chatbot technology. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 373–383. Springer.

Anja Belz. 2008. Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Natural Language Engineering*, 14(4):431–455.

Diedre Carmo, Marcos Piau, Israel Campiotti, Rodrigo Nogueira, and Roberto Lotufo. 2020. Ptt5: Pretraining and validating the t5 model on brazilian portuguese data. *arXiv preprint arXiv:2008.09144*.

Thiago Castro Ferreira, Chris van der Lee, Emiel Van Miltenburg, and Emiel Krahmer. 2019. Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. *arXiv preprint arXiv:1908.09022*.

Cornelia Győrödi, Robert Győrödi, George Pecherle, and Andrada Olah. 2015. A comparative study: Mongodb vs. mysql. In *2015 13th International Conference on Engineering of Modern Electric Systems (EMES)*, pages 1–6. IEEE.

Micha Heilbron, Benedikt Ehinger, Peter Hagoort, and Floris P De Lange. 2019. Tracking naturalistic linguistic predictions with deep neural language models. *arXiv preprint arXiv:1909.04400*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2022. Survey of hallucination in natural language generation. *arXiv preprint arXiv:2202.03629*.

Emiel Krahmer and Kees Van Deemter. 2012. Computational generation of referring expressions: A survey. *Computational Linguistics*, 38(1):173–218.

CR Rao and Venkat N Gudivada. 2018. *Computational analysis and understanding of natural languages: principles, methods and applications*. Elsevier.

Ehud Reiter and Robert Dale. 2000. Building applied natural language generation systems. *Natural Language Engineering*.

Manfred Stede. 1994. Lexicalization in natural language generation: A survey. *Artificial Intelligence Review*, 8(4):309–336.

André Luiz Rosa Teixeira, João Campos, Rossana Cunha, Thiago Castro Ferreira, Adriana Pagano, and Fabio Cozman. 2020. DaMata: A robot-journalist covering the brazilian amazon deforestation. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 103–106.

Marianne Wiesebron. 2013. Blue Amazon: thinking about the defence of the maritime territory. *Austral: Brazilian Journal of Strategy & International Relations*, 2(3):107–132.

Generating Quizzes to Support Training on Quality Management and Assurance in Space Science and Engineering

Andres Garcia-Silva, Cristian Berrio, Jose Manuel Gomez-Perez
Expert.ai / Madrid, Spain
agarcia@expert.ai, cberrio@expert.ai, jmgomez@expert.ai

Jose Antonio Martinez-Heras, Patrick Fleith
Solenix / Darmstadt, Germany
jose.martinez@solenix.ch

Stefano Scaglioni
ESA / Darmstadt, Germany
stefano.scaglioni@esa.int

Abstract

Quality management and assurance is key for space agencies to guarantee the success of space missions, which are high-risk and extremely costly. In this paper, we present a system to generate quizzes, a common resource to evaluate the effectiveness of training sessions, from documents about quality assurance procedures in the Space domain. Our system leverages state of the art auto-regressive models like T5 and BART to generate questions, and a RoBERTa model to extract answers for such questions, thus verifying their suitability.

1 Introduction

The complexity, cost, and risk of space missions involving public or private investment and even human lives make quality management a critical requirement to guarantee their success. The European Space Agency (ESA) makes a continuous effort to train their staff in quality procedures and standards. Trainees are evaluated to determine the effectiveness of the training sessions, with quizzes as one of the main tools used in such evaluations.

We present SpaceQQuiz (Space Quality Quiz), a system designed to help trainers to generate quizzes from documents describing quality procedures. Such documents cover topics like *Anomaly and Problem Identification, Reporting and Resolution* or *Configuration Management*, and include stakeholder responsibilities, activities, performance indicators and outputs, among others.

To design SpaceQQuiz we use state-of-the-art models based on transformers for Question Generation (QG) and Question Answering (QA). Since we could not find specialized models for the space or quality management domains, we reuse models already pre-trained on general-purpose document corpora and fine-tuned on SQuAD[1].

[1]The Stanford Question Answering Dataset https://rajpurkar.github.io/SQuAD-explorer/

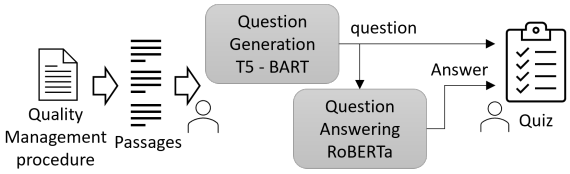

Figure 1: SpaceQQuizz - Proposed architecture.

2 Quiz generation system

Figure 1 shows the high-level architecture of SpaceQQuiz[2]. A question generation model is run on each passage extracted from the document. The generated questions and the corresponding passages are fed to a question answering model that extracts the answer from the passage. Only questions with answers are included in the candidate list that then is refined by the trainer to generate the quiz.

The process starts when the trainer uploads a quality procedure document. The system extracts the text from the PDF document using Apache PDFBox[3] and uses regular expressions to identify sections, subsections and paragraphs while removing non relevant text such as headers and footers. The trainer is presented with a list of candidate sections so that she can choose the most interesting ones for the quiz.

2.1 Question generation

To generate the questions we use a T5 model (Raffel et al., 2020) and a BART model (Lewis et al., 2020) fine-tuned on question generation. We use two models[4] in order to increase the number and variety of questions for each text passage. Both T5 and BART have excelled in sequence generation

[2]Demo: https://esatde.expertcustomers.ai/SpaceQQuiz/ user/pass demoINLG/demoINLG2022!

[3]Apache PDFBox https://pdfbox.apache.org

[4]Models withdrawn from HuggingFace by their authors. Description available at https://github.com/patil-suraj/question_generation

Proceedings of the 15th International Conference on Natural Language Generation: System Demonstrations, pages 4 - 6

tasks, such as abstractive summarization and abstractive question answering. The models we reuse were fine-tuned using SQuAD1.1, which consists of 100,000 questions created from Wikipedia articles where answers are segments in text passages.

T5 is fine-tuned using an answer-aware approach where the model is presented with the answer and a passage to generate the question. T5 is trained on a multitask objective to i) extract answers, ii) generate questions for answers using passages as context, and iii) extract answers for the generated questions. Finally the answer for the generated question is compared with the answer used to generate the questions. BART is fine-tuned following an answer-agnostic approach where the model is trained to generate questions from passages without information about the answers.

During generation, we use beam search as decoding method, with 5 as number of beams. Beam search keeps the most likely sequence of words at each time step and chooses the final sequence that has the overall highest probability. To avoid duplicity of questions in the final list, we compare them using cosine similarity between the question encoding generated through sentence transformers.[5] We discard questions similar to a previous one above an empirically defined threshold set at 0.8.

2.2 Question answering

Once the questions have been generated we use a RoBERTa model (Liu et al., 2019), fine-tuned for question answering in SQuAD2.0 to extract answers from the passages. SQuAD2.0 adds 50,000 unanswerable questions to SQuAD1.1. Thus, the fine-tuned RoBERTa is able to generate answers or not depending on the question. If RoBERTa fails to generate an answer for a generated question we remove it from the candidate list of questions presented to the trainer.

2.3 Quiz generation

The system displays the list of generated questions, answers, and the passages. The trainer can select specific questions to include in the quiz. Finally the system generates the quiz with a section containing only the questions to be handed to the trainee and another section reserved for the trainer with questions, answers and passages.

Table 1 shows some example questions generated from quality procedure documents. However, a human agent is still necessary since some of the questions generated by the models are too general (e.g. Who is the owner of the system?), syntactically incorrect, or not completely related to the context. More insights are presented in the following section.

Table 1: Questions generated by SpaceQQuiz

What is the first source for raising a spacecraft Anomaly Report?
the spacecraft log is the first source for raising ...
What does the ARB have to do in case of an anomaly detected in a shared infrastructure?
notify the relevant infrastructure team
Who can issue a supplier waiver?
OPS Project Manager or Service Manager
What does the leader of the operator's team do with the raised Anomaly Reports?
performs a preliminary review
Who chairs the Software Review Board?
the owner of the software,
What is mandatory for the closure of a Problem Report?
Root cause identification
What are minor non-conformances?
by definition, cannot be classified as major.

3 Evaluation

To evaluate the quiz generation system, we generate a quiz with 50 questions and answers pairs out of a quality procedure titled *OPS Procedure for Configuration Management*. Then a quality management expert evaluates the generated questions using relevance and correctness as evaluation criteria. The result of this manual evaluation is reported in table 2.

Table 2: Expert evaluation of question generation and question answering modules. (*) Only answers with a valid question according to the quality expert are evaluated.

	Accuracy
Generated questions	0.660
Extracted Answers	0.600
Extracted Answers*	0.818

In total 66% of questions are considered relevant and grammatically correct, and 60% of answers are also regarded as correct by the evaluator. If we focus only on the answers of relevant and correct questions then the percentage of accurate answers

[5] https://www.sbert.net/

Table 3: Example questions rated as non correct in the evaluation. In bold the answers extracted by the question answering module.

1	Context	In the process of **configuration identification** the team shall be aware on what is needed to be put under configuration control.
	Question	What shall the team know on what is needed to be put under configuration control?
2	Context	In a **continuous service** there is the concept of living baseline over a dynamic scope.
	Question	What is the concept of living baseline over a dynamic scope?
3	Context	Item configuration, in terms of implemented functions (e.g. **software version 2.0)**
	Question	What is item configuration in terms of implemented functions?
4	Context	The system under configuration includes also **the items received as Customer Furnished Item.**
	Question	What does the system under configuration include?

rises to 81.8%. The level of accuracy for the question generation makes the human-in-the-loop necessary to guarantee the quality of the questions in the quiz. However, note that the domain expert is also necessary to select the subset of questions to be included in the quiz.

By analysing incorrect question and answer pairs, we realize that despite being grammatically correct and relevant, some questions are just not possible to answer from the context used to generate them (see for example questions 1 and 2 in the table 3). This is consequence of a failure in the question answering module that produces an answer for such questions. Another example of wrong functioning of the question answering module is shown in question 3 in table 3, where the answer for the given question is extracted from the example in round brackets. A possible solution for this case is to discard examples in the text before feeding the question generation and the question answering modules.

For some correct and relevant questions (see for example question 4 in table 3) the question answering module just return partial answers. In this case the word *also* means that the answer in this context complements the answer already provided in another text excerpt. This a limitation of the extractive question answering module since it only extracts consecutive sequence of tokens from text passages as answers.

Finally the evaluator as domain expert reported that in some cases the problem is the underlying text used to generate the question that is not clear enough to formulate appropriate questions. Thus, wrong questions might indicate text excerpts that need to be reviewed by the author to convey the messages clearly.

4 Conclusions

We describe SpaceQQuiz, a system to help quality management trainers in the space domain to speed up the generation of evaluation material. SpaceQQuiz uses autoregresive models such as T5 and BART to generate the questions, and RoBERTa autoencoder to extract answers that are used as indicators of the validity of the questions.

Acknowledgements

This work is funded by ESA under contract AO/1-10291/20/D/AH - “Text and Data Mining to Support Design, Testing and Operations”.

References

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. ACL.

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.

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.

Automated Ad Creative Generation

Vishakha Kadam[1], **Yiping Jin**[2]*, **Bao-Dai Nguyen-Hoang**[1]
[1]Knorex, #04-01 21 Merchant Road, Singapore
[2]Department of Mathematics & Computer Science, Chulalongkorn University, Thailand
{vishakha.kadam, jinyiping, dai.nguyen}@knorex.com

Abstract

Ad creatives are ads served to users on a webpage, app, or other digital environments. The demand for compelling ad creatives surges drastically with the ever-increasing popularity of digital marketing. The two most essential elements of (display) ad creatives are the advertising message, such as headlines and description texts, and the visual component, such as images and videos. Traditionally, ad creatives are composed by professional copywriters and creative designers. The process requires significant human effort, limiting the scalability and efficiency of digital ad campaigns. This work introduces AutoCreative, a novel system to automatically generate ad creatives relying on natural language generation and computer vision techniques. The system generates multiple ad copies (ad headlines/description texts) using a sequence-to-sequence model and selects images most suitable to the generated ad copies based on heuristic-based visual appeal metrics and a text-image retrieval pipeline.

1 Introduction

Visually appealing ads with a compelling message will promote the brand image and lead to a better click-through rate. However, the ad composition process is time and labor-intensive, severely limiting the number of unique ads for each campaign. The ads' effectiveness deteriorates as the users are repeatedly exposed to the same ads, referred to as ad fatigue (Abrams and Vee, 2007). It underlines the importance of generating ad creatives automatically at scale.

This demo paper presents AutoCreative, a novel system for ad creative design which combines ad copy generation using a sequence-to-sequence Transformer model and ad image selection using object/scene detection and aesthetic appeal scoring. Figure 1 presents example ad creatives generated by our system.

*Work done while at Knorex.

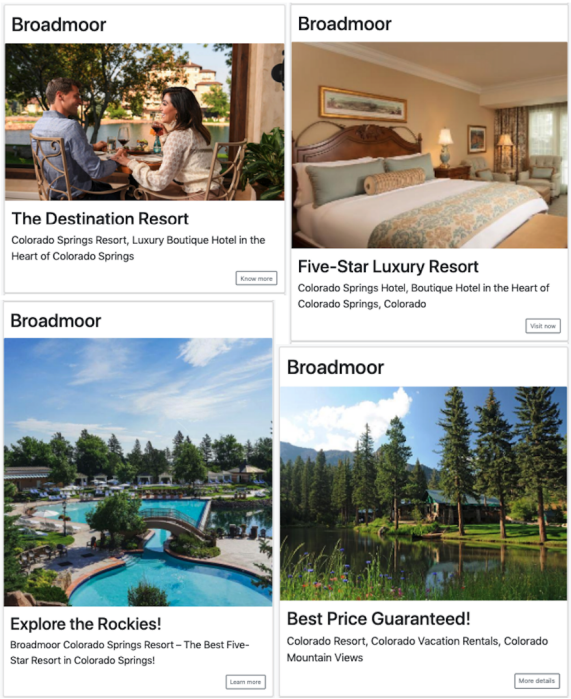

Figure 1: Examples ad creatives generated for the URL - https://www.broadmoor.com/.

2 System Overview

AutoCreative takes only the advertiser's URL as input. It first crawls the textual content and images from the advertiser's website. The ad copy generation module generates ad headlines and description texts conditioned on the content of the advertiser's website. The candidate images first go through a visual quality filter, then match against the generated ad copy using a text-image retrieval pipeline. Figure 2 overviews the framework.

2.1 Ad Copy Generation

We generate ad headlines and description texts conditioned on the textual content crawled from the advertiser's URL using a BART encoder-decoder model (Lewis et al., 2020). We use a relatively

Proceedings of the 15th International Conference on Natural Language Generation: System Demonstrations, pages 7 - 9

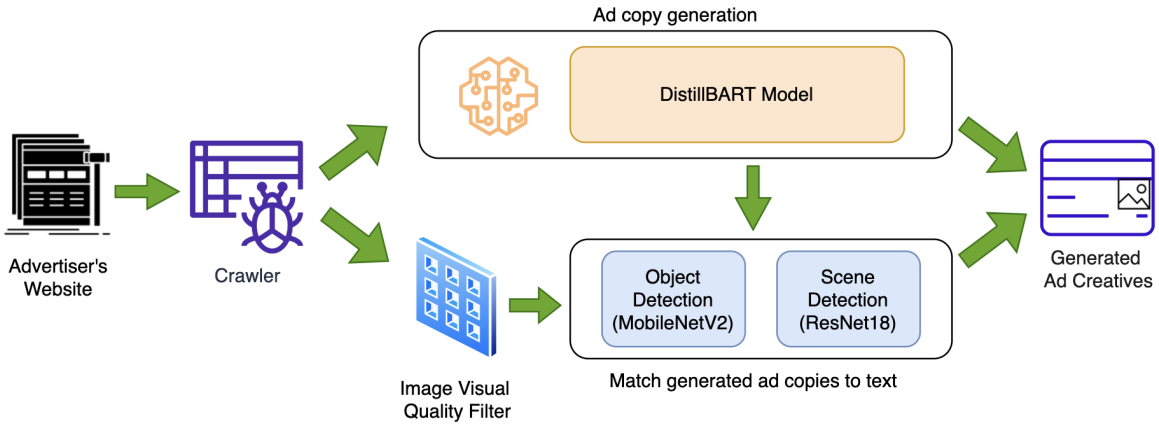

Figure 2: Overview of AUTOCREATIVE, an automated creative generation framework.

small DistilBART model [1] and fine-tuned it on a proprietary dataset of 300k (company description, advertising message) pairs (Jin et al., In press) .

Different message types (headline/description) and ad channels have different requirements for the ad message length. Therefore, we bucket the training data based on the ad message length and achieve fine-grained length control using conditional training. Similarly, we extract the POS tag of the first word in the ad message and use it as an additional control code to generate syntactically-diverse messages. The final generation probability is $P(message|description, len, pos)$, where len specifies the message length and pos specifies the POS tag of the first generated word. During inference, we choose the length based on the constraints and randomly sample POS tags to input to the model.

2.2 Image Selection

We first filter out noisy images like social media icons, background, or footer images, then measure the images' visual appeal based on factors like image colorfulness, contrast ratio, and lighting. Images that meet a pre-determined visual appeal threshold are input to a text-image matching module, where we utilize pre-trained object and scene detection models to match images with the generated headlines.

We use a MobileNet V2 (Sandler et al., 2018) model trained on Open Images Dataset V4 [2] for object detection and a ResNet-18 model trained on Places-365 Dataset [3] for scene detection (Zhou et al., 2017). We represent each image with the average *word2vec* embedding for the detected objects and scenes, and we calculate the average word2vec embeddings of the ad headlines. Finally, we return the images with the highest cosine similarity with the ad headline embeddings, as illustrated in Figure 3.

Ad headline	Buy Italian Food Online!	
Crawled Images		
Objects detected	Clothing Man Table Woman	Pasta Food
Scenes detected	Dining Hall Cafeteria	Pizzeria
Cosine similarity	0.5172	0.6971
Selected image	✗	✓

Figure 3: Matching image to a given ad headline.

3 Feedback from Account Team and Clients

We deployed AUTOCREATIVE to production in March 2022 and integrated it to Knorex XPO [4], a self-serve cloud marketing automation platform. The system has enabled over 30 advertisers to generate ad creatives across various campaigns. The feedback we received from our internal account team and various clients was overwhelmingly pos-

[1] https://huggingface.co/sshleifer/distilbart-cnn-6-6

[2] https://tfhub.dev/google/openimages_v4/ssd/mobilenet_v2/1

[3] https://github.com/CSAILVision/places365

[4] https://www.knorex.com/

itive. The technology was especially appreciated by small advertisers, who cannot afford an internal creative design team. Before the system was deployed, our creative designer team used to take more than a day to produce a set of ad creatives (including the communication overhead and the overhead of working on multiple creative design tasks simultaneously). Our system drastically reduced the turn-around time to a few minutes. Advertisers can now generate a set of appealing and diverse ad creatives from the UI without any creative design or copywriting knowledge. Critical feedback mostly relates to the cases where crawling is blocked on the advertiser landing page or no suitable image is available.

4 Conclusion

We introduced AutoCreative, a novel framework to automatically generate ad creatives. It has been deployed to production and used by clients of a global digital advertising company.

Acknowledgements

Yiping was supported by the scholarship from 'The 100^{th} Anniversary Chulalongkorn University Fund for Doctoral Scholarship' and also 'The 90^{th} Anniversary Chulalongkorn University Fund (Ratchadaphiseksomphot Endowment Fund)'.

References

Zoë Abrams and Erik Vee. 2007. Personalized ad delivery when ads fatigue: An approximation algorithm. In *Proceedings of the International Workshop on Web and Internet Economics*, pages 535–540, Bangalore, India. Springer.

Yiping Jin, Akshay Bhatia, Dittaya Wanvarie, and Phu T. V. Le. In press. Toward improving coherence and diversity of slogan generation. *Natural Language Engineering*, pages 1–33. Cambridge University Press.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves 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*, page 7871–7880.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, Salt Lake City, Utah, USA. IEEE.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2017. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464.

THEaiTRobot: An Interactive Tool for Generating Theatre Play Scripts

Rudolf Rosa[μ], **Patrícia Schmidtová**[μ], **Alisa Zakhtarenko**[μ],
Ondřej Dušek[μ], **Tomáš Musil**[μ], **David Mareček**[μ], **Saad Obaid**[μ],
Marie Nováková[σμ], **Klára Vosecká**[δ], **Daniel Hrbek**[σδ] and **David Košťák**[σ]
[μ]Charles University, Faculty of Mathematics and Physics, Prague, Czechia
[σ]The Švanda Theatre in Smíchov, Prague, Czechia
[δ]The Academy of Performing Arts in Prague, Theatre Faculty (DAMU), Prague, Czechia
rosa@ufal.mff.cuni.cz

Abstract

We present a free online demo of THEaiTRobot, an open-source bilingual tool for interactively generating theatre play scripts, in two versions. THEaiTRobot 1.0 uses the GPT-2 language model with minimal adjustments. THEaiTRobot 2.0 uses two models created by fine-tuning GPT-2 on purposefully collected and processed datasets and several other components, generating play scripts in a hierarchical fashion (title → synopsis → script). The underlying tool is used in the THEaiTRE project to generate scripts for plays, which are then performed on stage by a professional theatre.

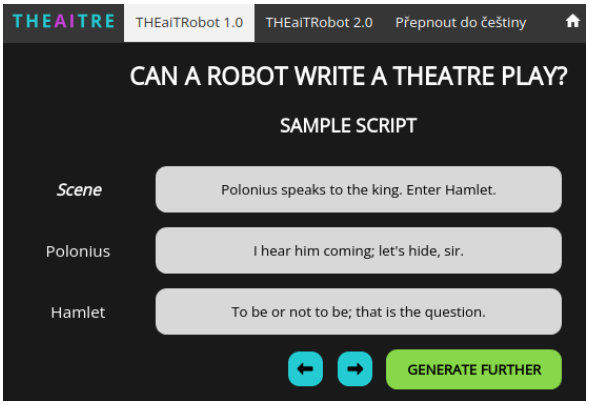

Figure 1: THEaiTRobot 1.0 input screen.

1 Introduction

We present a demo version of THEaiTRobot, a tool for interactively generating theatre play scripts. THEaiTRobot 1.0 is a vanilla GPT-2 model (Radford et al., 2019) with several adjustments for the theatrical domain (Rosa et al., 2021a); THEaiTRobot 2.0 features two fine-tuned GPT-2 models operating in a two-step hierarchical fashion. Machine translation allows the tool to operate both in English and in Czech.[1]

The tool was used within the THEaiTRE project to generate scripts of theatre plays, which were then staged by a professional theatre. The script of the 60-minute-long first play, *AI: When a Robot Writes a Play*, consists from 90% of automatically generated texts, with only 10% human contributions and edits (THEaiTRobot 1.0 et al., 2021), unprecedented for such a long play. The online premiere of the play was viewed by thousands of spectators worldwide (Moutinho, 2021).

One of our goals in the THEaiTRE project is making the public more familiar with the actual state and operation of artificial intelligence and natural language generation in particular. In addition to theatrical shows complemented by follow-up discussions with the audience, we want to further support this goal by making the demo of the THEaiTRobot tool freely available online for anyone to experiment with.[2] A short video showing the usage of the demo is available on YouTube.[3]

2 Related Work

A number of GPT-based language generation tools is available online, such as news generators (Zellers et al., 2019; Geitgey, 2019),[4,5] text adventure games,[6] code completion tools,[7,8] or chatbots.[9] However, to the best of our knowledge, no script generation tool has been released so far.

Script generation has been explored in several other projects, none of which however make their tools publicly available, often not even sharing quite enough details about the design of the tool, and also either using extensive human curation

[1]Adapting to a different language is very simple, as the translation system is external to the generator. It can easily be replaced by changing the respective call to the external API.

[2]https://theaitre.com/demo
[3]https://youtu.be/B3U38UgeZ9w
[4]https://rowanzellers.com/grover/
[5]https://newsyoucantuse.com/
[6]https://play.aidungeon.io/
[7]https://copilot.github.com/
[8]https://www.tabnine.com/
[9]https://projectdecember.net/

Proceedings of the 15th International Conference on Natural Language Generation: System Demonstrations, pages 10 - 13

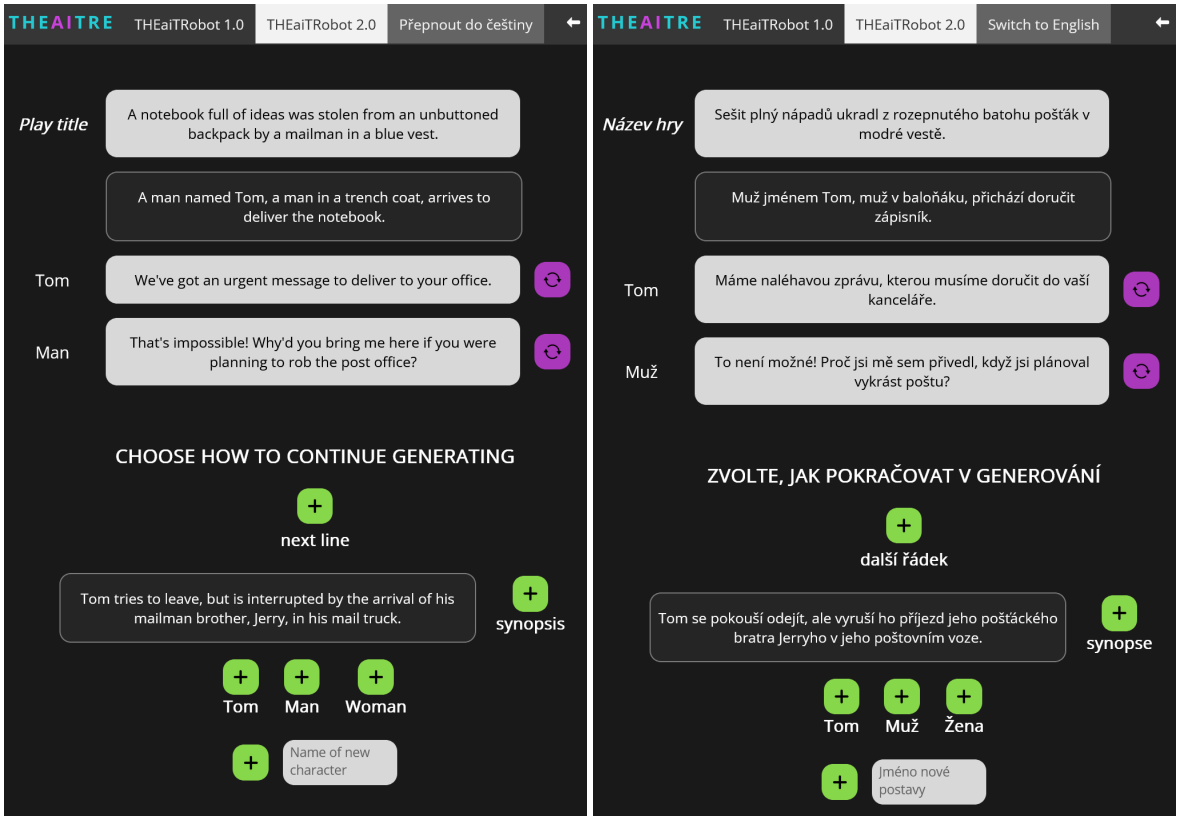

Figure 2: THEaiTRobot 2.0 synopsis → script interface (left: English, right: Czech with the same script).

and/or generating only very short scripts (Colton et al., 2016; Benjamin et al., 2016; Helper and Gillies, 2018; Mathewson and Mirowski, 2017).

3 THEaiTRobot 1.0

The first version of our tool is based on a vanilla GPT-2 XL language model with several adjustments, mainly employing TextRank-based (Mihalcea and Tarau, 2004) extractive text summarization to deal with GPT-2's limited window of 1,024 tokens, so longer scripts can be generated without losing context (Rosa et al., 2021b). The tool uses CUBBITT (Popel et al., 2020) for on-the-fly machine translation of the outputs into Czech.

In the demo version of the tool, the user can input a scene setting, character names and their first lines, or use one of the predefined inputs (see Figure 1), from which we construct the input prompt for the GPT-2 model in the following format:

```
Scene setting.

Character Name: Character line.

Character Name: Character line.
```

The tool then generates a continuation of the script line by line. At each step, the user can choose to continue generating or to regenerate a previously generated line (i.e. generate a different continuation from that position onward).

An early version of this demo was presented to the public in an exhibition at Goethe-Institut in Prague,[10] where about 100 users interacted with it, mostly finding it amusing and intriguing. We also gained feedback that helped us improve the demo.

4 THEaiTRobot 2.0

The second version of the tool uses a two-step hierarchical generation approach, first generating a play synopsis and then expanding that synopsis into a full play script, with specific models trained on our datasets for each of the steps.

The input for the synopsis generation step is the play title, in a fashion similar to script generation in THEaiTRobot 1.0 but with the underlying GPT-2 model fine-tuned on synopsis data. For the fine-tuning, we used a dataset consisting of ca. 65k synopses of theatre plays (scraped by us from Wikipedia), movies (Robischon, 2018; Kar et al., 2018), TV series (scraped by us from fan wiki pages) and books (Bamman and Smith, 2017).

[10] https://www.goethe.de/ins/cz/cs/ver.cfm?event_id=22345514

Once the user is happy with the generated synopsis, the synopsis is used as input for the second step.

In the second step, the play script is generated from the synopsis using a GPT-2 model fine-tuned for generating script sections based on synopsis sections. Here we use a near-domain ScriptBase corpus (Gorinski and Lapata, 2018), which contains movie synopses and scripts. We split the synopses into sentences and align these in a monotonic one-to-many fashion to script lines.

The user now has more options when generating (see Figure 2): generating a character line, also choosing which character should speak, or moving on to the next part of the generated synopsis.

5 Conclusion

We release an online demo of THEaiTRobot, a tool for interactive generation of theatre play scripts. The tool is free for non-commercial use, and its source code is released under the MIT licence.[11]

Acknowledgements

The project *TL03000348* is co-financed with the state support of Technological Agency of the Czech Republic within the ETA 3 Programme. It used services provided by the LINDAT/CLARIAH-CZ Research Infrastructure (https://lindat.cz), supported by the Ministry of Education, Youth and Sports of the Czech Republic (Project No. LM2018101).

References

David Bamman and Noah Smith. 2017. CMU book summary dataset. Released on Kaggle.

AI Benjamin, Oscar Sharp, and Ross Goodwin. 2016. Sunspring, a sci-fi short film starring Thomas Middleditch. Released on YouTube.

Simon Colton, Maria Teresa Llano, Rose Hepworth, John Charnley, Catherine V. Gale, Archie Baron, François Pachet, Pierre Roy, Pablo Gervás, Nick Collins, Bob Sturm, Tillman Weyde, Daniel Wolff, and James Robert Lloyd. 2016. The Beyond the Fence musical and Computer Says Show documentary. In *Proceedings of the Seventh International Conference on Computational Creativity*.

Adam Geitgey. 2019. *Machine Learning is Fun!* Self-published.

Philip John Gorinski and Mirella Lapata. 2018. What's this movie about? A joint neural network architecture for movie content analysis. In *Proceedings of NAACL-HLT*, pages 1770–1781, New Orleans, Louisiana.

Roslyn Helper and Harriet Gillies. 2018. Lifestyle of the Richard and family. Theatre play.

Sudipta Kar, Suraj Maharjan, A. Pastor López-Monroy, and Thamar Solorio. 2018. MPST: A corpus of movie plot synopses with tags. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France.

Kory W Mathewson and Piotr Mirowski. 2017. Improvised theatre alongside artificial intelligences. In *Thirteenth Artificial Intelligence and Interactive Digital Entertainment Conference*.

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.

Sofia Moutinho. 2021. Kinky and absurd: The first AI-written play isn't shakespeare—but it has its moments. *Science*.

Martin Popel, Marketa Tomková, Jakub Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondřej Bojar, and Zdeněk Žabokrtský. 2020. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature Communications*, 11(4381):1–15.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. Technical report, OpenAI.

Justin Robischon. 2018. Wikipedia movie plots. Released on Kaggle.

Rudolf Rosa, Tomáš Musil, Ondřej Dušek, Dominik Jurko, Patrícia Schmidtová, David Mareček, Ondřej Bojar, Tom Kocmi, Daniel Hrbek, David Košťák, et al. 2021a. When a robot writes a play: Automatically generating a theatre play script. In *ALIFE 2021: The 2021 Conference on Artificial Life*. MIT Press.

Rudolf Rosa, Tomáš Musil, Ondřej Dušek, Dominik Jurko, Patrícia Schmidtová, David Mareček, Ondřej Bojar, Tom Kocmi, Daniel Hrbek, David Košťák, Martina Kinská, Marie Nováková, Josef Doležal, Klára Vosecká, Tomáš Studeník, and Petr Žabka. 2021b. THEaiTRE 1.0: Interactive generation of theatre play scripts. In *Proceedings of the Text2Story'21 Workshop*, volume 2860 of *CEUR Workshop Proceedings*, pages 71–76, Aachen, Germany. RWTH Aachen University.

THEaiTRobot 1.0, David Košťák, Daniel Hrbek, Rudolf Rosa, and Ondřej Dušek. 2021. AI: When a robot writes a play. Technical Report ÚFAL TR-2021-67, ÚFAL MFF UK, Praha, Czechia.

[11] https://github.com/ufal/theaitrobot

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 9054–9065. Curran Associates, Inc.

Author Index