Towards Fully Automated News Reporting in Brazilian Portuguese

João Gabriel Moura Campos[1], André Luiz Rosa Teixeira[2], Thiago Castro Ferreira[2], Fabio Gagliardi Cozman[1], Adriana Silvina Pagano[2]

[1] Escola Politécnica, Universidade de São Paulo
São Paulo, Brasil

[2]Laboratório Experimental de Tradução, Universidade Federal de Minas Gerais
Belo Horizonte, Brasil

{joaogcampos,fgcozman}@usp.br,
{andrelrt,thiagocf05,apagano}@ufmg.br

Abstract. *We introduce robot journalists that cover two pressing topics in Brazilian society: COVID-19 spread and Legal Amazon deforestation. Our approach is able to automatically analyse structured domain data, select relevant content, generate news texts and publish them on the Web. We provide a thorough description of our system architecture, report on the results of automatic evaluation, discuss some of the advantages of robot-journalism in society, and point out further steps in our work. Corpus and code are publicly available*[1].

1. Introduction

Data-to-text Natural Language Generation (NLG) seeks to develop computational models for converting non-linguistic data into natural language in the format of text and speech [Reiter and Dale 2000, Gatt and Krahmer 2018]. Data-to-text applications have been proposed to automatically generate, for instance, weather forecasts [Mei et al. 2016], neonatal intensive care reports [Portet et al. 2009], car driver feedback [Braun et al. 2018] and texts from RDF triples [Moussallem et al. 2018]. Within this range of NLG applications, robot-journalism is one of the most prominent endeavors.

Robot-journalists are data-to-text applications that allow generation of news reports from non-linguistic data. A driving force behind automated-generated news is the availability of structured and machine-readable data [Graefe 2016], from sensors, organizations, and even social networks. Although data is newsworthy, the format in which it is made available is not reader-friendly. To overcome this limitation, automated-journalism has flourished in recent years and successful examples can be found both in academia and industry [Theune et al. 2001, van der Lee et al. 2017, Leppänen et al. 2017].

Automated-journalism increases the speed and scale of news coverage [Graefe 2016]; moreover, in comparison with human-generated news, automatic-generated ones are rated as more descriptive, informative, trustworthy and objective, even if more tedious and less pleasant to read [Clerwall 2014]. There are gains to be made in automated journalism as well as challenges yet to be met.

Although robot-journalism is a reality around the world and current technological resources are ideal for its development, only a few initiatives have explored automatic

[1]https://github.com/BotsDoBem/DEMO_INPE_COVID

generation of news in Brazilian Portuguese [DalBen 2019], a still low-resource language in NLG field [Campos and Cozman 2019]. We purport to fill this gap by introducing the first Brazilian robot-journalists developed in academia, as described in this paper. Our system enables multi-domain application, as illustrated with two highly-sensitive domains gathering interest from both local and international audiences: COVID-19 in the country and deforestation in Brazil's Legal Amazon area. The use of robot journalists to report primary data on such domains ensures data fidelity and fast updates while allowing human journalists to devote more time to investigative tasks.

Our robot-journalist automatically analyses structured domain data, selects relevant content, generates text news and publishes them on the Twitter platform. The system is based on a pipeline architecture for NLG [Reiter and Dale 2000], where non-linguistic data is converted into natural language through several explicit intermediate representations. End-to-end neural methods are currently favored in the NLG field; however, recent empirical studies have shown that texts produced by pipeline methods are more adequate than the former [Moryossef et al. 2019, Mille et al. 2019, Ferreira et al. 2019], which often "hallucinate" content not supported by the semantic input. For the particular task of journalism, reporting inaccurate data and hallucination would seriously undermine a robot's credibility. Besides that, a modular model allows for auditing, as compared with neural end-to-end approaches, which behave as black-boxes.

For the two selected domains, a corpus of verbalizations of non-linguistic data was created based on syntactical and lexical patterning abstracted from posts extracted from Twitter. Intermediate representations were annotated for each entry in order to develop our robot-journalist pipeline. An automatic evaluation experiment was then carried out to measure the fluency and lexical variety of the generated texts.

2. Related Work

Robot-journalism finds itself in a world of ever increasing data availability, with explosive growth in research and applications. A major obstacle, though, is how to render data readable to the lay audience. In academic research, Refs. [Theune et al. 2001] and [van der Lee et al. 2017] explore the generation of sportscasting news, whereas Ref. [Leppänen et al. 2017] proposes an NLG system to automatize the generation of news texts about elections in Finland. In industry, one of the first examples dates back to 2014, when the *Los Angeles Times* newspaper broke the news of an earthquake with a text report fully written by a robot-journalist called Quakebot. This robot-journalist was able to monitor seismological sensory data, detect an earthquake as well as automatically write and post news about it.[2] A short while later, an NLG company *Automated Insights*, in partnership with the news agency *Associated Press*, created a robot-journalist able to automatically generate and publish news reports about the quarterly earnings of US corporations.[3]

More recently, the British NLG company ARIA, in partnership with BBC news, developed a robot-journalist that automatically generated nearly 700 articles covering the results of the 2019 elections in the United Kingdom. It was the first time that BBC news was able to publish overnight a news story for every constituency that declared election

[2] https://www.bbc.com/news/technology-26614051

[3] https://blog.ap.org/announcements/a-leap-forward-in-quarterly-earnings-stories

results.[4] A further example is British company Radar AI (UK) has developed a data-to-text application converting periodic-release public data into textual reports.

In Brazil, although there have been robot-journalism initiatives, there is a wealth of data which is largely under-explored, particularly in human-readable format. The present paper addresses this issue and describes a data-to-text robot-journalist that automatically generates news articles on publicly-released Brazilian data about COVID-19 and deforestation in Brazil's Legal Amazon, as explained in the following sections.

3. System Architecture

Traditionally, data-to-text NLG systems are developed using two sequential modules: Content Selection and Surface Realization [Castro Ferreira 2018]. We account for our decisions concerning both modules in the following subsections, as well as detailing our implementation scheme.

Most, if not all, literature on NLG does not account for decisions taken at each step in the pipeline, merely describing systems in very cryptic terms. In this section we attempt to guide the reader throughout the pipeline architecture, describing the input and output formats of each module. In particular, we present transcripts of the messages exchanged amongst various modules in the system.

3.1. Content Selection

In a data-to-text system, the content selection module chooses *what* to say, i.e., it selects the communicative messages to be expressed in the text. Based on the ARRIA NLG Engine[5], our model breaks this step in 3 sub-tasks: data ingestion, data analysis and data interpretation. Data Ingestion automatically collects raw data via a range of data sources of interest. Data Analysis processes the raw data collected in the previous step and extracts key facts. Rule-based and data-driven methods may be used to do this kind of analysis. Finally, Data Interpretation filters and groups extracted key facts into chunks of relevant information to the audience, referred to as *messages*.

Because content selection is strictly related to the domain covered by the model, we separately explain the data ingestion, analysis and interpretation steps for the COVID-19 and deforestation domains in what follows.

Consider first the COVID-19 spread domain. The system starts by running the data ingestion step. From 9am to 8pm Brasilia time, it performs hourly extraction of the current number of cases, deaths and recovered patients of COVID-19 in Brazil. Our data ingestor scraps this information from the Worldometers website[6], which updates information based on official sources with low latency.

Comparing current information obtained by scraping data with the one from the previous day, the data analysis step extracts the number of daily new cases, daily new deaths and the daily variation of both numbers. Moreover, it also extracts current active cases (i.e., the number of total cases minus the total number of deaths plus recovered patients) and its variation in comparison with the previous day. At the end of data analysis,

[4] https://www.bbc.com/news/technology-50779761

[5] https://en.wikipedia.org/wiki/Arria_NLG

[6] https://www.worldometers.info/coronavirus/country/brazil/

8 key facts are returned in this domain: number of total cases, daily cases variation, total deaths, daily deaths variation, active cases, daily active cases variation, daily new cases and daily new deaths.

The key facts extracted in the data analysis are fed into the data interpretation module, which extracts the relevant messages based on rules. Figure 1 depicts the decision flow adopted by this module for the COVID-19 domain, which first checks if the number of active cases decreased after the pre-defined time setting of 6:50pm Brasilia time zone. If positive, this key fact and its daily variation are selected as messages to be reported, and are structured in the following format:

`ACTIVE_CASES(active_cases)`

`ACTIVE_CASES_VARIATION_LAST_DAY(variation,trend=low).`

Otherwise, the number of total cases is selected as a message to be reported, followed by the total number of deaths. Both messages are structured as follows:

`TOTAL_CASES(cases)`

`TOTAL_DEATHS(deaths).`

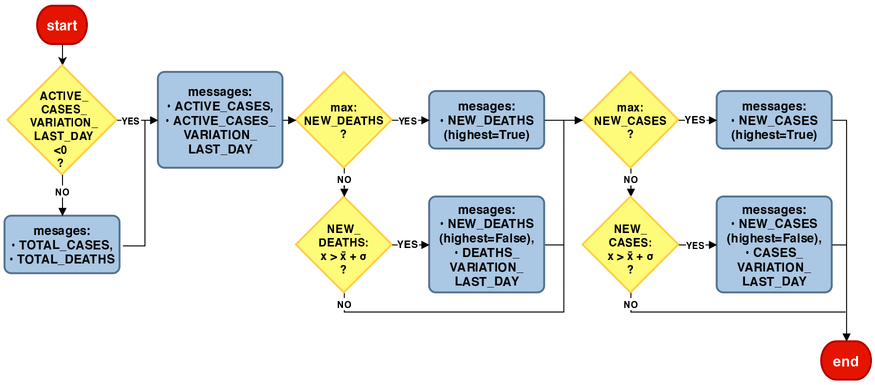

Figure 1. Content Selection data interpretation decision flowchart for the COVID-19 domain.

Once the number of total cases and deaths of COVID-19 in Brazil is processed, the data interpretation module handles the daily new cases. As Figure 1 shows, the module checks whether this indicator is the highest in the time series. If this is the case, it builds a message with the attribute “highest” set to `True`: `NEW_CASES(cases,highest=True).`

Otherwise, the module checks whether the indicator is greater than the average plus the standard deviation of the time series. If that is the case, it selects as messages the number of daily new cases and its variation in comparison with the last reported ones:

`NEW_CASES(cases,highest=False)`

`CASES_VARIATION_LAST_DAY(variation,trend=high).`

Finally, the same procedure performed for daily new cases is followed for daily new deaths. If this indicator is the highest in the time series, the model builds a message with the attribute “highest” set to `True`:

`NEW_DEATHS(deaths,highest=True).`

Otherwise, if the daily death cases is greater than the average plus the standard deviation of the time series, the model selects the following messages:

`NEW_DEATHS(deaths,highest=False)`

DEATHS_VARIATION_LAST_DAY(variation,trend=high).

Regarding Brazil's Legal Amazon deforestation, unlike hourly extraction for COVID19, our robot-journalist extracts information for monthly alerts: it accesses the database every day and checks for availability of data from the previous month. If such is the case, the robot posts news firsthand. Our data ingestor extracts deforestation raw data by an API[7] from DETER, a system developed by INPE to report alerts of deforestation in Brazil's Legal Amazon and the Cerrado ecosystems [Diniz and et al. 2015].

Once the raw data on deforestation is obtained from DETER, the data analyzer algorithm extracts as key-facts the total number of square kilometers of deforested land in the target month and variations of deforested area in comparison with the preceding month and the same month of the preceding year. Moreover, for the target month, the module also extracts the total amount of deforestation in square kilometers for each deforestation cause as well as for each state, city and conservation unit which are part of the Legal Amazon area.

Every key-fact extracted during this data analysis is then processed by the data interpretation model. Figure 2 depicts the decision flow of this module in this domain. First, the total deforested land in the target month is selected together with the total variation in deforestation in comparison with the previous month. Both facts are structured in the following *intent-attribute-value* messages:

TOTAL_DEFORESTATION(area,month,year)

LAST_MONTH_VARIATION_DEFORESTATION(variation,month,year).

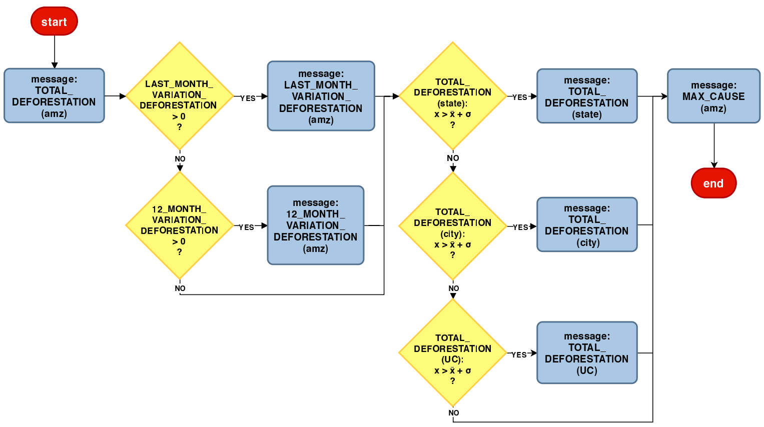

Figure 2. Content Selection data interpretation decision flowchart for the Amazon Deforestation domain.

If variation in deforested area in the previous 12 months is different from the variation in the target month, this fact is also selected and structured as the message:

12_MONTH_VARIATION_DEFORESTATION(variation,month,year).

Regarding the states' deforestation key-fact, the data interpreter selects the one with the highest deforested area if this area is greater than the states' deforestation average for the target month. The selected fact is structured as a message:

TOTAL_DEFORESTATION(area,state,month,year).

[7] http://terrabrasilis.dpi.inpe.br/homologation/file-delivery/download/deter-amz/daily

For the city and conservation area key-facts, the data interpreter follows the same process. The only difference is that, for each dimension (e.g., city and conservation area), it selects the key-fact with highest deforested area only if it is greater than the average deforested area plus the standard deviation in its respective dimension. If selected, city and conservation area key-facts are structured in the following intent messages:
`TOTAL_DEFORESTATION(area,state,city,month,year)`
`TOTAL_DEFORESTATION(area,state,city,uc,month,year)`.

Finally, the cause for the highest deforested area is structured into the message:
`CAUSE(area,month,year)`.

3.2. Surface Realization

Once the content selection task is over, the selected set of intent messages is fed into a surface realizer. Surface Realization is the task responsible for *how* to textually realize the selected information, i.e., make the most appropriate syntactic and lexical choices to convert the selected content into a grammatical and coherent text [Castro Ferreira 2018].

As stated, surface realization is traditionally run within a pipeline architecture [Reiter and Dale 2000], in which the selected content is converted into natural language through several explicit intermediate representations. Although this is a classical method, recent empirical studies have shown that texts produced by pipeline methods are more adequate than the outputs produced by novel end-to-end data-to-text methods [Moryossef et al. 2019, Mille et al. 2019, Ferreira et al. 2019], which suffer from hallucination (i.e. verbalization not supported by the selected content), with serious implications for information credibility and trustworthiness.

Our surface realizer was developed based on the pipeline architecture depicted in [Ferreira et al. 2019], which converts a set of intent messages into text in 5 steps: Discourse Ordering, Text Structuring, Lexicalization, Referring Expression Generation and Textual Realization. To explain those steps as a running example, take the following set of alphabetically-sorted intent messages about COVID-19 spread:

```
DEATHS_VARIATION_LAST_DAY(trend="high",variation="0.06")
NEW_CASES(cases="15305",highest="True")
NEW_DEATHS(deaths="824",highest="False")
TOTAL_CASES(cases="218223")
TOTAL_DEATHS(deaths="14817")
```

Discourse Ordering is the surface realization step that decides the order in which the communicative goals should be verbalized in the target text. For our example, a likely output might be:

```
TOTAL_CASES(cases="218223")
NEW_CASES(cases="15305",highest="True")
NEW_DEATHS(deaths="824",highest="False")
TOTAL_DEATHS(deaths="14817")
DEATHS_VARIATION_LAST_DAY(trend="high",variation="0.06")
```

Text Structuring aims to structure the ordered intent messages in paragraphs and sentences [Ferreira et al. 2019]. For our running example, this step might return:

```
<PARAGRAPH>
    <SENTENCE>
        TOTAL_CASES(cases="218223")
```

```
        NEW_CASES(cases="15305",highest="True")
    </SENTENCE>
    <SENTENCE>
        NEW_DEATHS(deaths="824",highest="False")
        TOTAL_DEATHS(deaths="14817")
        DEATHS_VARIATION_LAST_DAY(trend="high",variation="0.06")
    </SENTENCE>
</PARAGRAPH>
```

Lexicalization is responsible for finding the proper phrases and words to verbalize each structured sentence [Reiter and Dale 2000]. Similarly to work reported by Ref. [Ferreira et al. 2019], we adopted a template-based lexicalization approach to verbalize the intents of the messages. Besides the lexical choices made to verbalize intents, the generated lexicalized template uses tags to indicate (e.g., `COUNTRY`, `DEATH`, etc) where the attributes of the messages should be verbalized as well as their impact on the surrounding lexicon. The result of this step might be:

```
<PARAGRAPH>
    <SENTENCE>
        COUNTRY_1 VP[aspect=simple,tense=present,voice=active,
        person=COUNTRY_1,number=COUNTRY_1] totalizar CASES_1
        NN[number=CASES_1,gender=male] caso de #COVID-19 ,
        CASES_2 NN[number=CASES_2,gender=male] caso a mais
        de o que em o dia anterior .
    </SENTENCE>
    <SENTENCE>
        Desde ontem , VP[aspect=simple,tense=past,voice=passive,
        person=DEATHS_1,number=DEATHS_1,gender=female] registrar
        em COUNTRY_1 DEATHS_1 ADJ[number=DEATHS_1,gender=female]
        novo NN[number=DEATHS_1,gender=female] morte , que agora
        VP[aspect=simple,tense=present,voice=active,
        person=DEATHS_2,number=DEATHS_2] somar DEATHS_2 ,
        representando DT[number=singular,gender=TREND_1] um TREND_1
        ADJ[number=singular,gender=TREND_1] diário de VARIATION_1 .
    </SENTENCE>
</PARAGRAPH>
```

Referring Expression Generation is responsible for generating appropriate references to discourse entities [Krahmer and van Deemter 2012]. This consists in replacing the tags throughout the template with appropriate referring expressions to the entities in context. Moreover, this step is also responsible for replacing person, number and gender features in the lexicon based on the information of the produced referring expressions.

For our running example, a lexicalized template with appropriate referring expressions to intent attributes would be:

```
<PARAGRAPH>
    <SENTENCE>
        O Brasil VP[aspect=simple,tense=present,voice=active,
        person=3rd,number=singular] totalizar 218,223
        NN[number=plural,gender=male] caso de #COVID-19 ,
        15,305 NN[number=plural,gender=male] caso a mais
        de o que em o dia anterior .
    </SENTENCE>
    <SENTENCE>
        Desde ontem , VP[aspect=simple,tense=past,voice=passive,
```

```
        person=3rd,number=plural,gender=female] registrar
        em o país 824 ADJ[number=plural,gender=female]
        novo NN[number=plural,gender=female] morte , que agora
        VP[aspect=simple,tense=present,voice=active,
        person=3rd,number=plural] somar 14,817 ,
        representando DT[number=singular,gender=female] um alta
        ADJ[number=singular,gender=female] diário de 6% .
    </SENTENCE>
</PARAGRAPH>
```

Textual Realization is responsible for the last decisions to convert non-linguistic inputs into text. These consist in producing lexicon in its proper form (for instance, `VP[aspect=simple, tense=past, voice=passive, person=3rd, number=plural, gender=female] registrar` → `foram registradas`), outputting Brazilian Portuguese contractions (`em + o` → `no`) and de-tokenizing the text. In our example, the final verbalization based on the preceding choices would be:

> *O Brasil totaliza 218.223 casos de #COVID-19, 15.305 casos a mais do que no dia anterior. Desde ontem, foram registradas no país 824 novas mortes, que agora somam 14.817, representando uma alta diária de 6%.*

The generated texts related to COVID-19 spread are published on the Twitter account @*CoronaReporter*,[8] while texts about Amazon Deforestation are published on the Twitter account @*DaMataReporter*.[9]

4. Experiment

We ran an automatic evaluation experiment to measure fluency and lexical diversity of the outputs of the proposed model.

4.1. Data

For surface realization training, we collected a corpus of verbalizations for both domains. First, we performed content selection for past time-series about COVID-19 spread in Brazil and deforestation numbers in the Legal Amazon area. Based on the selected content, we grouped the sets with the same combination of intent messages and randomly chose 2 sets for each group. In total, 28 distinct sets of intent messages were selected for the COVID-19 domain and 13 for Legal Amazon deforestation.

After selecting distinct sets of intent messages for each domain, two of the authors in this paper verbalized each of them in Brazilian Portuguese. Verbalizations were made based on a small size sample of 200 texts (100 text per domain) extracted from the Twitter platform. Syntactic and lexical patterns in the samples were used to produce a variety of target intent verbalizations.

Finally, intermediate representations in the pipeline steps (e.g., discourse ordering, text structuring, lexicalization, referring expression generation and surface realization) were annotated.

---

[8] `https://twitter.com/CoronaReporter`

[9] `https://twitter.com/DaMataReporter`

4.2. Model Set-Up

We contrasted the performance of two generation implementation models, *random* and *majority*. Given all the options for a given context (e.g., all the templates available to lexicalize the intent `TOTAL_CASES`), the *random* model operates by randomly selecting one of the options, while the *majority* model selects the most frequent option.

4.3. Method

We automatically evaluated the fluency of the texts produced by both versions of our model in comparison with the gold-standard texts in the collected corpus using BLEU [Papineni et al. 2002] and chrF++ [Popović 2017], two popular metrics in data-to-text evaluation.

Besides fluency, we also aimed to measure how well our approach could generate lexically diverse texts to communicate a set of intents. As Ref. [Castro Ferreira 2018] shows, lack of variation is one of the reasons why automatically-generated texts are rated *"tedious"* by humans. To automatically perform this evaluation, we assessed lexical diversity of the produced texts as compared with the compiled tweet samples, using MTLD (Measure of Textual Lexical Diversity) [McCarthy and Jarvis 2010].

4.4. Results

Table 1 depicts BLEU and chrF++ fluency scores of our model's versions for both domains. In the COVID-19 domain, the *majority* model outperforms the *random* one according to both scores. In the Amazon Deforestation domain, the *majority* approach was also the best in terms of BLEU, but showed a slightly lower chrF++ score in comparison with the one reported for the *random* approach.

	COVID-19		**Deforestation**	
Model	**BLEU**	**chrF++**	**BLEU**	**chrF++**
Random	0.28	0.49	0.38	0.61
Majority	0.37	0.54	0.4	0.60

Table 1. Results of the random and majority version of the proposed model in the domain of COVID-19 and Deforestation of Legal Amazon.

Table 2 shows lexical diversity scores of the *majority* and *random* approaches in comparison with the compiled samples for both domains. A closer analysis reveals that in the COVID-19 domain, the output texts presented greater lexical diversity in the *random* model than the *majority* model. On the other hand, the Amazon Deforestation domain showed an opposite trend, in which the *majority* model yielded slightly higher lexical diversity than the *random* one. With a low margin, the differences between the MTLD scores by the model and the human-generated compiled samples suggest that, although the compiled samples present higher lexical diversity, our output texts show comparable scores. Finally, an inter-domain comparison shows that scores of lexical diversity are higher for Amazon Deforestation texts when compared to COVID-19 texts. This is expected, since the Amazon Deforestation domain is associated with a wider range of communicative intents and entities.

Model	COVID-19 Lex. Diversity(MTLD)	Amazon Deforestation Lex. Diversity(MTLD)
Compiled	46.5	63.3
Random	40.5	52.4
Majority	36.0	53.3

Table 2. MTLD scores of the generated texts in random and majority models and Twitter compiled samples

5. Conclusion

This is the first initiative which describes the development of a Brazilian robot-journalist in academic research. Our paper provides a thorough description of the steps followed to implement our robots. The proposed approach is able to analyse structured data, select relevant content, generate news texts in Brazilian Portuguese and publish them on the Twitter platform. The selected domains – COVID-19 spread in Brazil and Brazil's Legal Amazon deforestation – are highly relevant topics for civil society, which benefits from a real-time coverage by reliable sources of structured data.

Pipeline Architecture Because fact-based news and commitment to the truth are among the most important principles of journalism, our approach was developed based on a NLG pipeline architecture, which, unlike novel neural end-to-end approaches, ensures consistent adequacy of the output texts to the input content. Moreover, being modular, the decisions of our model are easier to be accounted for and audited, as compared with black-boxes in neural end-to-end approaches. On top of that, a modular design allows for domain specific customization with minimum effort.

Unlike studies that fail to provide thorough descriptions of their models' decisions, our manuscript carefully guides the reader along our pipeline, describing the input-output intermediate representations of each single step. This contributes to a deeper understanding of the architecture, offering a methodology for robot-journalists to be developed in other domains, expected to foster initiatives for the growth of robot-journalism in Brazil.

Social Relevance The two selected domains are highly-sensitive and are targets of interest for local and international audiences. Since the onset of the COVID-19 pandemic outbreak, accurate case and death counts are an important aspect to successfully contain the infection spread, as these metrics are fundamental to drive public awareness and guide public health policies.[10] Concerning deforestation in the Legal Amazon area, this is a topic that systematically draws national and international attention, in particular since 2019, when a new record deforestation was reported within an eleven-year span.[11]

There are two main reasons in favor of robot-journalists in popular and highly-sensitive domains as the ones chosen in this study. First, robot-journalists can be as fact-accurate as human journalists, with the added advantage that they can find and publish news based on raw data much faster than the latter. Second, assigning robot-journalists to cover primitive information, structured in a machine-readable format, allows human

[10] `https://www.nature.com/articles/d41586-020-01008-1`
[11] `https://www.bbc.com/news/world-latin-america-50459602`

journalists to devote more time to investigative tasks, which robots are not envisaged to do.

Fluency and Lexical Diversity assessment Fluency assessment indicated that our *majority* version tended to perform better than *random* for both domains of interest. Regarding lexical diversity, a closer analysis of the results for the COVID-19 domain revealed a higher lexical diversity by the output of the *random* model. Conversely, in the Amazon Deforestation domain, scores showed a negligible difference, which may reflect less lexical diversity of verbalizations in the training set.

Despite a better performance by the *majority* model in terms of the fluency BLEU score in both domains, the texts generated by the *random* model scored within adequate levels. Moreover, the latter yielded higher lexical diversity scores in the COVID-19 domain. In order to avoid lack of variation in text verbalization, assumed to be the reason why automatically-generated texts are *tedious* for human raters [Castro Ferreira 2018], we chose the *random* model for generating news on Twitter, our official publication platform.

Future Work A further step in our work is to improve our approach to allow for multilingual output drawing on multilingual grammars and to expand coverage of new domains. Concerning the architecture, we intend to collect new training corpora and to implement data-driven tools to improve the performance of the pipeline modules.

Acknowledgments

Research funded by the Coordination for the Improvement of Higher Education Personnel (CAPES) under grants 88887.488096/2020-00, 88882.349188/2019-01 and 88887.367980/2019-00; the National Council for Scientific and Technological Development (CNPq), grant 312180/2018-7; the São Paulo Research Foundation (FAPESP), grant 2019/07665-4; and the State Funding Agency of Minas Gerais (FAPEMIG), grant APQ-01129-17.

References

Braun, D., Reiter, E., and Siddharthan, A. (2018). Saferdrive: An NLG-based behaviour change support system for drivers. *Natural Language Engineering*, 24(4):551–588.

Campos, J. G. M. and Cozman, F. G. (2019). A review of Natural Language Generation: A corpus in Brazilian Portuguese. In *VIII WPGEC - Workshop de Pós-Graduação de Engenharia Da Computação*, São Paulo, Brasil.

Castro Ferreira, T. (2018). *Advances in Natural Language Generation: Generating Varied Outputs from Semantic Inputs*. PhD thesis, Tilburg University. Series: TiCC Ph.D. Series Volume: 64.

Clerwall, C. (2014). Enter the Robot Journalist: Users' perceptions of automated content. *Journalism Practice*, 8(5):519–531.

DalBen, S. (2019). Robots in Brazilian journalism: Three case studies. In *VI Seminário de Pesquisa Em Jornalismo Investigativo - ABRAJI*, São Paulo, Brasil.

Diniz, C. G. and et al. (2015). Deter-b: The new amazon near real-time deforestation detection system. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(7):3619–3628.

Ferreira, T. C., van der Lee, C., van Miltenburg, E., and Krahmer, E. (2019). Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. In *EMNLP/IJCNLP*.

Gatt, A. and Krahmer, E. (2018). Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170.

Graefe, A. (2016). Guide to Automated Journalism. Technical report, Tow Center for Digital Journalism, Columbia University, New York.

Krahmer, E. and van Deemter, K. (2012). Computational generation of referring expressions: A survey. *Comput. Linguist.*, 38(1):173–218.

Leppänen, L., Munezero, M., Granroth-Wilding, M., and Toivonen, H. (2017). Data-driven news generation for automated journalism. In *Proceedings of INLG*.

McCarthy, P. M. and Jarvis, S. (2010). MTLD, vocd-D, and HD-D: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior Research Methods*, 42(2):381–392.

Mei, H., Bansal, M., and Walter, M. R. (2016). What to talk about and how? selective generation using LSTMs with coarse-to-fine alignment. In *Proceedings of NAACL*, San Diego, California.

Mille, S., Dasiopoulou, S., and Wanner, L. (2019). A portable grammar-based NLG system for verbalization of structured data. In *Proceedings of the 34th ACM/SIGAPP*.

Moryossef, A., Goldberg, Y., and Dagan, I. (2019). Step-by-step: Separating planning from realization in neural data-to-text generation. In *Proceedings of NAACL*, Minneapolis, Minnesota.

Moussallem, D., Ferreira, T., Zampieri, M., Cavalcanti, M. C., Xexéo, G., Neves, M., and Ngonga Ngomo, A.-C. (2018). RDF2PT: Generating Brazilian Portuguese texts from RDF data. In *Proceedings of LREC*, Miyazaki, Japan.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of ACL*, Philadelphia, USA.

Popović, M. (2017). chrF++: Words helping character n-grams. In *Proceedings of WMT*, Copenhagen, Denmark.

Portet, F., Reiter, E., Gatt, A., Hunter, J., Sripada, S., Freer, Y., and Sykes, C. (2009). Automatic generation of textual summaries from neonatal intensive care data. *Artificial Intelligence*, 173(7–8):789 – 816.

Reiter, E. and Dale, R. (2000). *Building Natural Language Generation Systems*. Studies in Natural Language Processing. Cambridge University Press, Casmbridge, U.K.

Theune, M., Klabbers, E., De Pijper, J. R., Krahmer, E., and Odijk, J. (2001). From data to speech: a general approach. *Natural Language Engineering*, 7(1):47–86.

van der Lee, C., Krahmer, E., and Wubben, S. (2017). PASS: A Dutch data-to-text system for soccer, targeted towards specific audiences. In *Proceedings of INLG*, INLG’2017, Santiago de Compostela, Spain.