EMBEDDIA

Cross-Lingual Embeddings for Less-Represented Languages in European News Media

Research and Innovation Action
Call: H2020-ICT-2018-1
Call topic: ICT-29-2018 A multilingual Next generation Internet
Project start: 1 January 2019 Project duration: 39 months

D5.7: Final evaluation report on multilingual text generation technology (T5.4)

Executive summary
This report (D5.7) is the final evaluation report on multilingual text generation technology. We evaluate the three main natural language generation (NLG) components of EMBEDDIA: the multilingual natural language generation method from T5.1, the document planning and content selection methods from T5.2, and the headline generation method from T5.3. The evaluations are based on a combination of qualitative and quantitative methods and analysis of the software. The results indicate that the fundamental technology developed for natural language generation is sound and fits the design goals, and that journalists find the results useful. The approaches developed for document planning and content selection support a range of different concrete use cases. For headline generation, our results show that pre-trained multilingual NLG models are a good choice for low-resourced languages.

Partner in charge: UH

Project co-funded by the European Commission within Horizon 2020 Dissemination Level		
PU	Public	PU
PP	Restricted to other programme participants (including the Commission Services)	–
RE	Restricted to a group specified by the Consortium (including the Commission Services)	–
CO	Confidential, only for members of the Consortium (including the Commission Services)	–

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825153

Deliverable Information

Document administrative information	
Project acronym:	**EMBEDDIA**
Project number:	**825153**
Deliverable number:	**D5.7**
Deliverable full title:	**Final evaluation report on multilingual text generation technology**
Deliverable short title:	**Final evaluation report on multilingual text generation technology**
Document identifier:	**EMBEDDIA-D57-FinalEvaluationReportOnTextGeneration-T54-submitted**
Lead partner short name:	**UH**
Report version:	**submitted**
Report submission date:	**28/02/2022**
Dissemination level:	**PU**
Nature:	**R = Report**
Lead author(s):	**Leo Leppänen (UH)**
Co-author(s):	**Hannu Toivonen (UH-CS), Eliel Soisalon-Soininen (UH-CS), Matej Martinc (JSI)**
Status:	**_ draft, _ final, X submitted**

The EMBEDDIA Consortium partner responsible for this deliverable has addressed all comments received. Changes to this document are detailed in the change log table below.

Change log

Date	Version number	Author/Editor	Summary of changes made
20/01/2022	v0.1	Leo Leppänen (UH)	First draft
31/02/2022	v0.2	Matej Martinc (JSI)	Added material related to T5.3
05/02/2022	v0.3	Leo Leppänen (UH)	Complete draft
06/02/2022	v0.4	Hannu Toivonen (UH)	WP leader's check
07/02/2022	v1.0	Leo Leppänen (UH)	Submitted for internal review
16/02/2022	v1.1	Jose G Moreno (ULR)	Internal review
17/02/2022	v1.2	Saturnino Luz (UEDIN)	Internal review
18/02/2022	v1.3	Leo Leppänen (UH)	Modifications to address internal review
21/02/2022	v1.4	Jose G Moreno (ULR)	Modifications to address internal review
21/02/2022	v2.0	Leo Leppänen (UH)	Ready for quality control
21/02/2022	v2.1	Nada Lavrač (JSI)	Quality control
22/02/2022	v2.2	Hannu Toivonen (UH)	Modifications to address QC comments
23/02/2022	v2.3	Matej Martinc (JSI)	Modifications to address QC comments
24/02/2022	v2.4	Leo Leppänen (UH)	Modifications to address QC comments
25/02/2022	final	Leo Leppänen (UH)	Ready for submission
28/02/2022	submitted	Tina Anžič (JSI)	Report submitted

Table of Contents

List of abbreviations

NLG	Natural Language Generation
E2E	End-to-End
WP	Work Package
STT	Finnish News Agency, *Suomen Tietotoimisto*
NHL	National Hockey League
CNN	Convolutional Neural Network
BART	Bidirectional Autoencoder Representations from Transformers
BERT	Bidirectional Encoder Representations from Transformers
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
NLI	Natural Language Inference
SS	Semantic Similarity

1 Introduction

The main objective of the EMBEDDIA project is to develop methods and tools for effective exploration, generation and exploitation of online content across languages thereby building the foundations for the multilingual next generation internet, for the benefit of European citizens and industry using less-represented European languages. One facet of this effort is EMBEDDIA work package 5 (WP5), which is concerned with Natural Language Generation (NLG). In order to support journalists and media companies in efficiently reaching as many demographics as possible, the objective of WP5 has been to design and develop news automation systems that are transferable across languages, transferable across domains, and transparent in their NLG process.

Specifically, WP5 has aimed to (1) develop a self-explainable, flexible, accurate, and transparent NLG system architecture that can be transferred to new domains and languages with minimal human effort; (2) develop tools for creation of dynamically evolving content, incorporating narrative structure and user knowledge; and (3) develop tools for creation of figurative language and headlines. Accordingly, the work package consists of three primary tasks (T5.1–T5.3) plus the current task (T5.4) on resource gathering, benchmarking and evaluation.

Task T5.1, Multilingual text generation from structured data, has adapted NLG technology for the requirements of news generation. The task has developed mechanisms for (i) determining what is interesting or important in the given data and deciding what to report, and for (ii) rendering that information in an accurate manner (iii) in multiple languages.

Task T5.2, Multilingual storytelling and dynamic content generation, has developed a novel method for automatically organising news articles based on the domain of the article.

Task T5.3, Creative language use for multilingual news and headline generation, has made the generated texts more varied and colourful by producing creative expressions, especially in headlines. We use, for instance, context-dependent embeddings to find similar and analogous terms and metaphors.

This report (D5.7) is the final evaluation report on multilingual text generation technology. We evaluate the three main NLG components of EMBEDDIA: the multilingual natural language generation method from T5.1 (Section 2), the document planning and content selection methods from T5.2 (Section 3), and the headline generation method from T5.3 (Section 4).

In some cases, evaluations were already reported in earlier deliverables (especially in D5.6 for various creative tasks) and in the respective original articles (Alnajjar, Leppänen, & Toivonen, 2019; Alnajjar & Toivonen, 2021; Alnajjar & Hämäläinen, 2021; Wright & Purver, 2021) and are not repeated here.

2 Multilingual natural language generation

In Task T5.1 we investigated natural language generation in a multilingual setting. The most significant contribution on this front is a natural language generation method that is embodied in case study systems, most notably a multilingual natural language generation system for producing news text from Eurostat datasets. This system is described in detail in the deliverables of Task T5.1, with the high-level architecture described in Deliverable D2.4, 'Multilingual Language Generation Technology'.

We start by briefly reviewing the requirements for natural language generation in EMBEDDIA. We then assess the system against the requirements: first the technical properties of the system, then its outputs.

2.1 Requirements analysis

In Deliverable D5.2 ('Initial News Generation Technology') we identified that news generation system requirements can be thought of in terms of transparency; accuracy; modifiability and transferability;

fluency; data availability; and topicality. This analysis was based on our previous work in the news generation domain (Leppänen, Munezero, Granroth-Wilding, & Toivonen, 2017).

The requirement for *transparency* stems from a media-specific need for accountability (McBride & Rosenstiel, 2013; Stark & Diakopoulos, 2016) and strive for objectivity (Mindich, 2000), as well as the increased public scrutiny of the fairness of algorithmic decision making in general (e.g. Angwin, Larson, Mattu, & Kirchner, 2016). This need is also driven by a more concrete need to protect newsrooms from legal consequences. For example, in Finland the editor-in-chief of a newsroom is always accountable for everything that is published. It is very difficult – if not impossible – for the editor to ethically take responsibility for a black box generated text, without assigning a human to check the texts produced by the NLG automation tool. Such a system of checks, however, distinctly diminishes the potential of automation.

The requirement for *accuracy* is self-evident. A system producing untruthful content, on the one hand, exposes the newsroom employing the system to legal liability, and on the other hand, erodes the readership's trust in the news product. Consequently, any automated system must be known to be accurate in its output. In fact, automation has been classically used in news domains that are prototypically objective and have the highest accuracy requirements, such as weather reports (Goldberg, Driedger, & Kittredge, 1994) and financial news coverage (Yu, 2014).[1] The requirement for accuracy also interplays with the aforementioned requirement for transparency: establishing trust in the system's accuracy requires either very extensive testing or a transparent system. Notably, this requirement remains even when news automation is directed at journalists rather than at general audiences: in such a scenario, if the *journalists* lose trust in the system, it becomes equivalently unhelpful.

The system must also be *modifiable and transferable*. NLG systems are costly to set up. Unless the same underlying technology can be reused in multiple domains, the newsrooms will have very few domains wherein the potential profits and savings offered by the use of automation can justify a from-scratch effort to produce an automated system. As noted by an anonymous interviewee of Linden (2017): "It is difficult to create generic solutions; we have to start from scratch for each new case, and relatively little is reusable."

In terms of *fluency*, the level required is dependent on how the system is intended to be used. In cases where the output is directed at human journalists who can polish the text and add additional analytical details, the requirement is significantly lower than in cases where the text is delivered directly to the news consumer. In both cases, however, the fluency must be high enough to ensure that the information content of the text is understood correctly by the readers.

The *availability of data* is less important from an academic perspective, but is crucial from a business perspective, as it is related more to the business feasibility of a technology rather than its scientific value. Namely, to be a worthwhile expenditure for a for-profit business, any developed systems must be able to produce enough content to cover the cost of their creation. As such, the system needs to produce content from datasets where multiple stories are available. It is notable, however, that this content needs not be produced in a single go. Rather, both a constant drip of news stories (for example, a constantly updating coverage of the present state of the COVID-19 situation) and an occasional bulk production (for example, generating a multitude of stories every time new data on the economy is released) are viable options. These last factors, however, also indicate a need for *topicality* in the data: however cheap, producing automatic summaries of decades old NHL ice hockey games is unlikely to be a sound business move.

[1] This might be a consequence of most pre-existing automation approaches being unsuitable for more complex journalism, see Stray (2019).

2.2 Technical properties of the EMBEDDIA news generation system

Based on the requirements analysis, from the technical perspective it is desirable to produce a transparent and accurate system that is modifiable, transferable and of some minimal fluency level. We believe that the EMBEDDIA news generation systems developed in Work Package WP5, as well as the general text generation architecture originally described in Task T2.3 of Work Package WP2, match these requirements.

First, with a rule-based approach the system is more transparent than a comparable system based on neural processing. This allows investigating and surgically correcting any mistakes or bugs in the systems. In comparison, the only practically available correction process for a neural system is retraining, either with an expanded training set (which includes examples explicitly addressing the observed undesirable behaviour) or with modified initial parameters. Rule-based approaches, such as those used in WP5, also provide better understood 'quality floors' where it is easier to reason about the possible failure modes of the system than it is for an equivalent machine learning system.

The requirement for modifiability and transferability seems to favour systems based on machine learning, such as end-to-end neural NLG systems, over rule-based systems. It is useful, however, to distinguish here between the *theoretical* and *practical* transferability of the systems. In practice, neural end-to-end NLG system are only transferable at the cost of large amounts of training data in the form of aligned input-output pairs of structured data and human-written text. It is our understanding that aligned training data is exceedingly rare in the news world outside of some specific domains such as sports, finance and weather. Furthermore, such data cannot, by definition, exist for new domains and text types where the costly human news production is not profitable, but where automation could be useful. While some systems have been presented for unsupervised learning of an NLG model (e.g. Schmitt, Sharifzadeh, Tresp, & Schütze, 2019), they make several significant assumptions regarding the structure of the input data, effectively requiring a partially lexicalized document plan as input. For example, Schmitt et al. (2019) generate English language outputs using as input knowledge graphs defined using English language labels and relations, thus giving almost all the necessary lexical information 'for free.' This severely limits the *practical* transferability of neural approaches.

Simultaneously, the requirement for modifiability and transferability indicates that 'global' (i.e. non-modular) rule-based systems are not optimal, as transferability is maximized when large parts of the system can be reused when transferring to a new domain. As such, we construe this requirement as pointing towards modular rule-based approaches, and towards modular hybrid approaches that incorporate neural components not dependent on aligned training data, e.g., ones that can be trained solely on textual corpora.

We have applied the same general EMBEDDIA text generation approach in multiple text generation systems which share significant amounts of the modular pipelines' modules. In addition to the Eurostat and COVID19 news generation systems described in Deliverables D5.2 and D5.4, the same underlying pipeline is also used by the system that produces natural language reports from news comments (see Deliverable D3.5). We interpret this as indicating that the text generation method is indeed highly transferable and modifiable between different text domains.

Further evidence towards the modifiability of the system is provided by our experiences in trialing several different content selection and document planning methods (described in Deliverables D5.3 and D5.6 and evaluated below in Section 3) and the experiments conducted in Task T2.3 in relation to (re)lexicaliation. Furthermore, this approach allowed us to develop a system translation tool as described in Deliverable D5.5. By translating the system itself (more specifically, the templates used therein), the working of the system in the new language can be inspected, corrected and modified by humans collectively for all texts to be produced, rather than translating and fixing each individual text produced.

Our experiments using the system translation tool to produce an Italian version of the system indicated

that the tool was able to produce useful first drafts of templates. Note that the system translation approach does not compete with machine translation of documents; rather, it uses machine translation internally to produce a version of the system (i.e., its templates) that operates in a new language. Based on our experience, the draft templates produced automatically can cut down on the general localization effort especially when the technical personnel localizing the system are not native speakers of the target language. At the same time, our experiments with Finnish indicate that the system translation quality suffers when producing draft templates for morphologically more complex languages. As such, while providing further proof of the benefits of the modular approach used by the EMBEDDIA language generation technology, the translation tool is not a panacea that trivializes system language support extension efforts.

Regarding transferability between multiple languages, the Eurostat news generation system produces text in six different European languages (Finnish, Croatian, English, Estonian, Slovene, Russian). Finnish and Estonian are Finno-Ugric languages, completely unrelated to the other four which are Indo-European. This indicates that the technology developed is well-suited for generating text in different languages. At the same time, the results from human evaluations indicate that the present level of language support for some of the languages is not very good (see next section).

As for system accuracy, the language generation processes of the EMBEDDIA text generation methods were designed so as to minimize the danger of producing textual outputs that are either unsupported by or disagree with the underlying data. Most significantly, the system holds the underlying numerical information in immutable data structures for as much of the processing as possible. This limits the likelihood that any programming errors would be able to accidentally modify the underlying data. In contrast, empirical evidence suggests that the most commonly employed neural text generation methods suffer from a type of overfitting, 'hallucination', where the system produces output that is not based on the underlying data (Reiter, 2018; Nie, Yao, Wang, Pan, & Lin, 2019; Dušek, Howcroft, & Rieser, 2019; Puduppully, Dong, & Lapata, 2019).

While the rule-based generation methods employed by the EMBEDDIA text generation approach do not guarantee that systems built using the approach are unbiased, as we observed in both D6.11 and Leppänen, Tuulonen, and Sirén-Heikel (2020), rule-based generation methods both avoid some sources of bias — notably word embeddings as used in text generation systems that use an encoder-decoder process which produces word embeddings as output — while also allowing for easier identification of biases through their transparency.

(As described in D6.11 and by Leppänen et al. (2020), bias and journalism have a complicated relationship. On one hand, (especially western) journalism is deeply associated with an *objectivity norm*, where news and journalists strive for objectivity, correctness and truth. This objectivity has been traditionally seen as an antonym of bias and partisanship, both of which are viewed as having adverse effects on the journalistic ethos for reporting the reality truthfully (Hackett, 1984). However, the complexity of journalistic bias has gained a new dimension with digitalization. The shift towards mobile and the changes in audience behavior have increased the role of the audience, affecting news values and journalistic work (Harcup & O'neill, 2017; Kunert & Thurman, 2019). Personalization, in effect a form of bias, has become a strategy for media organizations and platforms for creating customer value. Catering for audience tastes based on implicit or explicit user information can also increase the value for automated news, for example based on location, as suggested by Plattner and Orel (2019). However, as Kunert and Thurman (2019) found in their longitudinal study, most news organizations remain committed to exposing their audience to a diversity in news stories, reaffirming the prevailing framing of quality journalism. Distinguishing between *acceptable* bias, such as exhibited in personalized sports news, and *unacceptable* bias, e.g., favoring certain ethnicities, is a value ridden process. Both are examples of *selectivity*, as suggested by Hofstetter and Buss (1978, p. 517), or more generally framing (Entman, 1993; Scheufele, 1999). Only shared values decide that one is acceptable and the other is not. Encoding such values exhaustively into any automated procedure is extremely difficult.)

Consumer prices in Austria

In September 2021, in Austria, the monthly growth rate of the harmonized consumer price index for the category 'clothing and footwear' was 17.1 points. It was 7.6 percentage points more than the EU average. The country had the 5th highest monthly growth rate of the harmonized consumer price index for the category 'clothing and footwear' across the observed countries. In February 2021, the monthly growth rate of the harmonized consumer price index for the category 'clothing and footwear' was -7.5 points. It was 6.3 percentage points less than the EU average.

In October 2021, the monthly growth rate of the harmonized consumer price index for the category 'communication' was 1 points. It was 1.1 percentage points more than the EU average. The country had the 3rd highest monthly growth rate of the harmonized consumer price index for the category 'communication' across the observed countries. In September 2021, the country had the 11th highest value for it across the observed countries. The monthly growth rate of the harmonized consumer price index for the category 'communication' was -0.2 points.

Virossa, toukokuussa 2021, kuukausittainen kasvu kuluttajahintaindeksissä 'koulutus' 18.1 yksikköä

Virossa kuukausittainen kasvu kuluttajahintaindeksissä 'koulutus' oli 18.1 yksikköä toukokuussa 2021. Se oli 18 prosenttiyksikköä yli EU:n keskiarvon. Se oli 10.8 prosenttiyksikköä ali EU:n keskiarvon maaliskuussa 2021. Se oli 30.7 prosenttiyksikköä yli EU:n keskiarvon kesäkuussa 2020. Se oli 30.8 yksikköä.

Kuukausittainen kasvu kuluttajahintaindeksissä 'asuminen, vesi, sähkö ja lämmitys' oli 5.1 yksikköä kesäkuussa 2021. Se oli 4.7 prosenttiyksikköä yli EU:n keskiarvon. Se oli 6.7 prosenttiyksikköä yli EU:n keskiarvon syyskuussa 2021. Se oli 7.5 yksikköä. Se oli -3 yksikköä maaliskuussa 2021.

Figure 1: Examples of texts used in the evaluation in both English (left) and Finnish (right).

2.3 Journalists' evaluation of text quality

The worth of any natural language generation (NLG) system eventually depends on how useful the system is for its intended users. To this end, we conducted a human evaluation of the Eurostat news generation system.

In the evaluation, expert human judges — a total of eight journalists from the three EMBEDDIA media partners — were presented with several texts produced by the news generation system. Examples of the texts are shown in Figure 1. The journalists were shown the texts one at a time, and were asked to indicate their agreement regarding the following statements about the texts:

Newsworthiness: The text contains information that could be published in a news article

Structure: The information is structured or ordered in a logical manner

Grammaticaly: The text is grammatically correct

Fluency: The text is fluent and natural

Usefulness: The text contains information that could be useful in my daily work

Reusability: I could re-use parts of the text in an article I would write based on it.

Agreement was expressed using a 7-step Likert scale consisting of the following options: 'Strongly disagree', 'disagree', 'Somewhat disagree', 'Neutral', 'Somewhat agree', 'Agree', and 'Strongly agree'. For statistical analysis purposes, the values were encoded numerically using the value 1 for 'Strongly disagree' and the value 7 for 'Strongly agree'. Value 4 thus indicates a neutral answer.

English texts In the first part of the evaluation, the eight participating expert evaluators each evaluated 5 English language texts produced by the EMBEDDIA Eurostat news generation system using its default settings and another 5 comparable texts generated using a different document planning method.

Table 1: Human evaluation results for the English language Eurostat news generation system. Care should be taken while interpreting the means and standard deviations, as Likert-scale data is only ordinal, not interval data. We report these numbers only for completeness.

	Median	Mode	Mean	St.dev.
Newsworthiness	5.5	5	5.525	0.716
Structure	5.0	6	5.075	1.248
Grammaticality	6.0	6	5.375	0.952
Fluency	3.5	5	3.850	1.424
Usefulness	4.0	4	4.300	1.363
Reusability	5.0	5	4.800	1.154

Table 2: Median human evaluations Finnish, Estonian and Croatian language texts as created by the Eurostat news generation system. The column *n* indicates the number of judges providing judgements for each language, with all other numbers being reported as medians.

	n	Newsworthiness	Structure	Grammaticality	Fluency	Usefulness	Reusability
Finnish	2	5.0	3.0	4.5	3.0	4.0	3.0
Estonian	3	5.0	3.0	1.0	1.0	4.0	1.0
Croatian	3	3.0	5.0	1.0	1.0	6.0	6.0

The results of these evaluations are presented in Table 1. In this section, we present only the results pertaining to the default system, and will return to the alternative texts in Section 3.2.

The results indicate that the evaluators viewed the system in generally positive terms. The judges agreed (median between 'Somewhat agree' and 'Agree') that the automatically generated texts contained newsworthy information that could be published in news articles. The judges somewhat agreed that they could re-use parts of the text in an article they would write based on it (median 'Somewhat agree'). The judges didn't agree or disagree with the statement that the text contained information that would be useful in their daily work (median 'Neutral'). These results should be interpreted taking into account the fact that the judges were not able to choose the topics of the evaluated texts. The texts discussed consumer price data, and all the stories evaluated in this subsection of the evaluation pertained to countries other than those where the judges' employers were located.

The judges agreed that the texts were grammatical (median 'agree') but slightly disagreed with them being fluent (median between 'somewhat disagree' and 'neutral'). They also viewed the texts as being structured or ordered in a logical manner. We will return to this last point in Section 3.

Texts in native languages In the second part of the evaluation, the same journalists were shown three texts in their native language based on their employer. The three texts also described the consumer price index developments of the nations the media partners were located in. In other words, the STT participants were shown Finnish language texts about Finland, Estonia and Crotia, the Ekspress Meedia participants were shown Estonian language texts about Finland, Estonia and Croatia, etc. The judges were asked to evaluate these texts using the same statements used in part one of the evaluation. All the texts were generated using the standard (heuristic-based) document planner.

The results of the evaluation are shown in Table 2. Due to the limited number of participants per language — the column *n* indiciates the per-language number of judges, ranging from two to three — we do not compute other statistics beyond medians.

Observing the results, we note that the judges were generally positive regarding the usefulness of the results (medians ranging from 'neutral' to 'agree'), but observed grammatical flaws in the text, as indicated by disagreement with the grammaticality statement. Analysis of the judges' free-text answers

indicates that the Estonian and Croatian systems created sentences with incorrect grammatical agreement. This was somewhat expected for the Croatian texts due to a lack of a Croatian morphological analyzer-generator component. For Estonian, the result was unexpected, as we are using a morphological library to conduct analyses and inflections. We suspect that the observed phenomenon results from either a programming or templating error, but more detailed analysis with Estonian native speakers is required to establish the exact root cause.

Interestingly, there is significant variation in the answers to the 'reusability' aspect even when accounting for the grammatical errors. Specifically, we note that while the Crotian evaluators found the texts ungrammatical, they still agreed that they would be useful and reusable (median 'agree'). On the other hand, while the Estonian judges viewed the grammaticality as equally bad, they evaluated both usefulness and reusability more critically ('neutral' and 'completely disagree', respectively). We expect that some of this disagreement can be attributed to differing alignments of the texts with the judges' daily work (i.e. a sports journalist would likely answer differently than an economy journalist) as well as whether the judges considered the system in the abstract (i.e. a system of this type, potentially using different data more related to their daily work) or this specific system. Concurrently, due to the low number of judges and total judgements, the results are likely to be very noisy.

2.4 Qualitative analysis of text generation

The evaluators were also asked to provide free-text feedback in a third part of the evaluation, in addition to the above quantitative questions. No additional texts were presented to the evaluators in this part.

The judges were presented with the following three questions:

Q1: "What kinds of changes would make these kinds of texts more useful for you in your work?"

Q2: "If you could receive these types of automatically generated texts from any data you wanted (for example economic data, sports or election results or COVID-related statistics), what would be the most useful to you or your colleagues? Do not worry about whether the data really exists or not."

Q3: "If you were to use this kind of a system, how would you incorporate it into your daily workflow?"

The answers to Q1 indicate that at least some of the judges perceived the English language texts as of sufficiently high quality to be useful as-is. Additional tweaks would be needed to address the grammatical issues in Estonian and Croatian, which in the case of Croatian appear to be related to grammatical gender. As further improvements to the system, the judges called for inclusion of further context to the information, even more focused texts and a stricter enforcement of temporal ordering in the text.

In terms of Q2, the judges identified as the most promising data sources those discussing topics such as sports, COVID and health, crime and legal matters, economics, elections and weather. They also observed that increased localization of the texts (e.g., tailoring texts to more localized areas or smaller populations) would be beneficial.

For Q3, the judges identified two principal methods to incorporate automated news generation to their daily workflows.

The first of these was to embed machine-generated paragraphs or subsections into otherwise human-written news texts. While not explicitly stated by the judges, we note that this would have synergy with (hyper)localization, for example in a scenario where a human journalist writes a high-level analysis of an economic trend on the national level, which is then enhanced by a computer-generated subsection that describes the same phenomena in terms of its relation to the users' locale.

The second method envisioned by the judges was that the texts would be targeted at the journalist alone, e.g., as news alerts that would be sent to them during their day, or alternatively as a "recap" they

could read when starting their daily work. One judge also identified that in this capacity, the texts could act as starting points for writing, allowing for faster response time to breaking news events.

2.5 Discussion

Above, we have evaluated the work conducted in Task T5.1 using a combination of qualitative analysis of system properties in relation to an analysis of requirements, a statistical analysis of human evaluations of the generated texts, contributed by domain experts, as well as a more qualitative analysis of free-text feedback provided by the judges.

In the research literature, NLG systems are often evaluated with metrics such as BLEU (Papineni, Roukos, Ward, & Zhu, 2002) and ROUGE (Lin, 2004) which measure how well the system's output is aligned with a known-good collection of gold standard outputs. Measures proposed in the literature for this purpose are, in general, modified versions of word set overlap metrics that attempt to account for natural language complications such as synonymy and word order. In this work, we have not employed such methods because they assume the existence of gold standard texts that cover the whole space of acceptable outputs. There are no pre-existing human written texts that would have used the same inputs as the system, and it is not feasible to construct new sets of gold standard texts for our NLG use cases with numerous input data, content selection, and ranges of linguistic expressions potentially used to express those selected contents. Having journalists evaluate generated texts gives much more realistic results but is costly.

We have also contributed to the larger discussion regarding the problematic state of automated measures in NLG evaluation. In a larger collaborative effort, we outlined other potential methods for improving the state of NLG evaluations. This work resulted in a publication entitled "Underreporting of errors in NLG output, and what to do about it" (van Miltenburg et al., 2021), attached to this deliverable as Appendix A.

3 Document planning and content selection

In the scope of Task T5.2, we developed three methods for planning the information content and structure of automatically generated news text.

The first of these methods is a heuristic-based approach described in Deliverables D5.3 and D5.5. This heuristic is dependent on the existence of hierarchical labels which describe semantics of data, and we use the hierarchical labels to estimate the semantic similarities between pieces of information. This method is evaluated below in Subsection 3.1.

The second method is a variant of the first one, wherein the similarity is determined using word embeddings. The benefit of this is that the method is applicable in cases where hierarchical labels are not available (but word embeddings are). This method is described in Deliverable D5.5 and is evaluated in Subsection 3.2.

Finally, we developed a separate machine learning approach, described in Deliverables D5.3 and D5.5. The method is evaluated in Subsection 3.3.

3.1 Heuristic method for document planning

In Deliverables D5.3 and D5.5 we described a heuristic-based document planning method. We implemented the method in the larger Eurostat news generation application described in deliverables D5.2 and D5.4. As a baseline to compare against, we also developed a variant of the same application with a simplified document planner. In this simplified planner, the planner always selects the maximally newsworthy available message as the message without any early stopping threshold. Nuclei, i.e. the

paragraph-first key messages, are selected from a more limited 'core messages set', while satellites (auxiliary messages) can discuss a wider range locales; see the above-mentioned deliverables for details. Contrasting our proposed method with this simplified method enables us to evaluate the quality of narrative coherence in the generated texts.

For this evaluation, three experts were recruited from the Finnish News Agency STT to evaluate documents on the consumer price indices in five different European nations. For all nations, the judges were shown variants produced by the heuristic method and the simpler baseline method. One of the selected countries is the country the news agency is based in, with the assumption that the judges have a good amount of world knowledge that they can use in evaluating these texts. Another variant pair describes a country that is both relatively small and geographically remote (but still within EU), with the assumption that the journalists are unlikely to have much world knowledge about this country's consumer prices. The three other countries were selected from among those bordering the first country, with the assumption that the journalists would have some, but not much, world knowledge relating to these countries. The final output texts were not inspected prior to selecting the countries.

This evaluation was focused on document planning and content selection. Therefore the enclosing system was simplified in some respects, e.g., to not conduct complex sentence aggregation. This was done to minimize the effect of later stages of the generation process on this evaluation. The only manual alteration was the addition of headings to indicate the texts' intended themes, e.g., "Consumer Prices in Estonia".

The evaluations were conducted online. The judges were first provided with some basic information on the type of documents they were to read (i.e. that the texts are intended to be news alerts for journalists, rather than publication ready news texts), the length of the task, etc. All instructions were in the judges' native language, in this case Finnish. The judges were not told which texts were produced by which variants nor how many variants were being tested. Following this, the judges were shown the documents one by one. For each document, the judges were asked to indicate their agreement with the following statements (translated from Finnish):

Q1: The text matches the heading

Q2: The text is coherent

Q3: The text lacks some pertinent information

Q4: The text contains unnecessary information

Q5: The text has a suitable length

For Q1–Q4, the judges indicated their agreement on a 7-point Likert scale ranging from 1 ('completely disagree') to 7 ('completely agree'). For Q5, the answers were provided on 5-point scale ranging from 1 ('clearly too short') to 3 ('length is suitable') to 5 ('clearly too long'). In addition, the judges were able to provide textual feedback for each individual text, as well as for the evaluation task as a whole. The judges' answers to Q1 – Q5, are aggregated in Table 3.

The results indicate that the heuristic-based method statistically significantly increases the document's coherence (Q2, mean 4.33 vs. 1.60, median 5 vs. 2), the matching of the document's content to the document's theme (Q1, mean 4.40 vs. 1.80, median 5 vs. 2), and produces documents of more suitable length (Q5, mean 2.93 vs. 4.07, median 3 vs. 4, with 3 being best). The proposed method also seems to result in less unnecessary information being included in the document (Q4, mean 5.13 vs. 6.33, median 5 vs 6), and in the text missing less necessary information (Q3, mean 4.47 vs. 5.80, median 4 vs. 6), but these effects are not statistically significant after correcting for multiple comparisons with the Bonferroni correction. We hypothesize this difference would become significant in a larger-scale evaluation.

The free-form textual feedback provided by the judges indicates, as expected, that the texts could be further improved. For example, a text discussing consumer prices in Estonia stated that Estonia had the third highest index value for a certain type of goods with the Estonian value being X while the Swedish and North Macedonian values for the same category of goods were X+0.6 and X+0.7, respec-

Table 3: Evaluation of document planning and content selection in texts produced with the heuristic-based method. Parentheses indicate answer ranges and whether the higher (↑), lower (↓) or middle values are to be interpreted as the best. The p_{MWU} column contains the (uncorrected) p-value of a two-sided Mann-Whitney U test. An asterisk indicates the p-value is statistically significant also after applying a Bonferroni correction to account for multiple tests.

	Heuristic method			Baseline			
Statement	Median	Mean	SD.	Median	Mean	SD.	p_{MWU}
Q1, match (1–7, ↑)	5	4.40	1.64	2	1.80	0.41	< 0.001*
Q2, coherence (1–7, ↑)	5	4.33	1.76	2	1.60	0.51	< 0.001*
Q3, lack of information (1–7, ↓)	4	4.47	1.81	6	5.80	1.42	0.049
Q4, unnecessary information (1–7, ↓)	5	5.13	1.55	6	6.33	0.62	0.024
Q5, length (1–5, 3 is best)	3	2.93	0.59	4	4.07	0.70	< 0.001*

tively. Here, the judges called for an explicit statement that North Macedonian value was the highest in EU.

This work is described in full by Leppänen and Toivonen (2021), attached to this deliverable as Appendix B.

3.2 Word embeddings based document planning

In Deliverable D5.5 we also described a variant where the information similarity component of the heuristic was replaced with a word embedding based approach. Briefly, instead of observing the similarity between data point labels (expressed using a hierarchical labeling system), we produce isolated sentences describing each fragment and then calculate the similarities between those sentences using word embeddings.

An evaluation of this approach was conducted concurrently with the evaluation described in Section 2.3 by showing the judges — in addition to the five English-language texts produced by the heuristic-based document planner — five comparable texts generated using this word embedding based document planner. A total of 8 judges evaluated 5 texts belonging to both the heuristic-based method and the word embedding method. The judges were told that the evaluated texts came from multiple variants of the same system, but were not told which text was produced using which document planner, nor how many different system variants being tested. The evaluation statements are given above in Section 2.3.

The results obtained in this evaluation are described in Table 4. We report medians, means and standard deviations, for both the heuristic-based systems ('Heuristic-based') and the word embedding method ('Word embeddings') but as above care should be taken when interpreting the latter two of these values as Likert scale answers are ordinal rather than interval data. We conduct Mann-Whitney U tests to determine whether the two methods are statistically significantly different from each other for each question.

As we are conducting multiple significance tests as a family, it is not statistically sound to apply the standard threshold of statistical significance ($\alpha = 0.05$) as-is. Instead, we need to apply a correction to reduce the likelihood of false positive results. Bonferroni correction is a (conservative) method for adjusting statistical significance levels when multiple tests are carried out simultaneously. Given that we had six tests, the traditional threshold $\alpha = 0.05$ for statistical significance is replaced by the Bonferroni-corrected value $\frac{\alpha}{6} \approx 0.008$. We note, however, that the correction is conservative. It is prone to overcorrecting especially when the individual tests are correlated. This is presumably the case between grammaticality and fluency, as well as between usefulness and reusability. This should be kept in mind when interpreting the values reported below.

Based on the results shown in Table 4, we observe that the proposed word embedding based method is outperformed by the heuristic-based method especially in the 'structure' aspect (median 5.5 vs 3.0,

Table 4: Comparison of contents produced with the heuristic-based and the word embedding-based methods. The p_{MWU} column contains the (uncorrected) p-value of a two-sided Mann-Whitney U test. An asterisk indicates statistical significance prior to applying a Bonferroni correction, with two asterisks indicating statistical significance using the Bonferroni-corrected threshold. See text for discussion on the conservative nature of the Bonferroni correction.

	Heuristic-based			Word embedding based			
	Median	Mean	St.dev.	Median	Mean	St.dev.	p_{MWU}
Newsworthiness	5.5	5.525	0.716	5.0	4.950	1.176	0.019*
Structure	5.0	5.075	1.248	3.0	3.700	1.145	1.538e-05**
Grammaticality	6.0	5.375	0.952	5.0	5.025	1.025	0.063
Fluency	3.5	3.850	1.424	3.0	3.175	1.238	0.017*
Usefulness	4.0	4.300	1.363	4.0	4.025	1.368	0.205
Reusability	5.0	4.800	1.154	5.0	4.300	1.651	0.068

$p = 1.538e - 05$). This result is statistically significant even after applying a Bonferroni correction for multiple comparisons.

For the 'newsworthiness' (median 5.5 vs. 5.0) and 'fluency' (median 3.5 vs. 3.0) aspects, the results are statistically significant using the standard threshold of $\alpha = 0.05$, but would lose statistical significance with a Bonferroni correction for multiple testing (see below). Meanwhile the 'usefulness' (both median 4.0), 'reusability' (both median 5.0) and 'grammaticality' (median 6.0 vs. 5.0) aspects lack a statistically significant difference.

Overall, the heuristic based method outperforms the word embedding based one in terms of text quality. Furthermore, we observed that the word embedding based method demands more computing resources, especially in terms of memory, and results in significantly increased system runtime. Taken together, these limit the usefulness of the word embedding based approach especially in relation to on-demand generation. A conclusion is to use the heuristic-based method whenever suitable semantic hierarchy is available for labels.

When hierarchical labels are not available, however, the embedding based approach is an option. Evaluators agreed (median 'somewhat agree') that the resulting texts contained newsworthy content and were reusable in their daily work (median 'somewhat agree'). We thus believe that the method — following further refinement — could be useful in certain types of generation tasks.

3.3 Machine learning based document planning

In this line of work, we have replaced the heuristic content selection method of Section 3.1 with a machine learning model, namely a neural network. As discussed in Section 2.2, a challenge for this approach is the scarcity of end-to-end training data, that is, data records paired with human-written news outputs. We have circumvented this issue in document planning and content selection by making the reasonable assumption that the position of a sentence in a news article reflects its newsworthiness, and by sampling a training dataset from a statistical news corpora.

We have experimented with several different neural architectures, which we have evaluated on the task of sentence ordering given a set of paragraphs with their sentences shuffled. Due to a good performance on this task as well as some technical advantages, we chose a pairwise convolutional neural network (CNN) classifier for the actual content selection task and human evaluation of the generated news outputs. All details regarding the datasets, the neural architectures and the sentence ordering task have been reported in Deliverables D5.3 and D5.5.

In order to evaluate the news outputs generated using this neural content selection approach, we conducted an evaluative survey where, similarly to Section 2.3, five human judges were presented with

generated texts and a set of statements to be answered using a seven-step Likert scale. The texts were also based on Eurostat consumer price data for European countries, as in Section 3.1, each text focusing on one country. The evaluated texts pertained to four different countries.

The judges were recruited from among journalists from the Finnish News Agency STT and well as journalism researchers, under the assumption that both would be intimately familiar with the news production process (as conducted by human journalists) and various news text artefacts.

The survey was conducted online. Following a general description of the study, the judges were presented with a total of 12 texts on at a time. All texts, instructions and statements were in English. The judges were told that different methods had been used for generation, but they did not know which texts were generated using which method, or how many methods were being compared.

After being presented with each text, the judges were asked to indicate their agreement with the following statements:

Q1: The text corresponds to the heading

Q2: The text is coherent

Q3: The text contains useful information

For all of these statements, 1 indicates complete disagreement while 7 indicates complete agreement.

We tested three variants of document planning and content selection. The first variant (Baseline), is our NLG system using a simplified version of the heuristic method discussed in Section 3.1, such that only the newsworthiness score was used to determine the fitness of a message, discarding the other metrics. The second variant (Heuristic) is the complete heuristic-based method described. Finally, the third variant is based on the pairwise CNN classifier for determining the fitness scores of messages.

In total, the judges were presented with 12 two-paragraph news articles generated by the document planner variants, four texts each. For all variants, each paragraph was restricted to discuss only one theme. Examples of the variants are shown in Figure 2.

The median, mean, and standard deviations of the answers are presented in Table 5. We report the means and standard deviations for the purpose of comparison, but acknowledging that computing these metrics on a Likert scale is not completely unproblematic as it assumes a unit distance between each step of agreement. Care should be taken when drawing conclusions based on these latter values.

As the upper table of Table 5 shows, the medians and means are higher for CNN in comparison with Heuristic and Baseline for all statements Q1-Q3. The scores for Heuristic are also slightly higher than those of Baseline, although the difference is smaller.

In order to determine the statistical significance of these results, we performed first a Kruskal-Wallis statistical test for each statement sample (all methods together), and then a two-sided Mann-Whitney U test for each pair of methods within each statement sample, given that the null hypothesis of the Kruskal-Wallis test had been rejected for the statement in question. The p-values of the Kruskal-Wallis test (p_{KW}) for each statement and the Mann-Whitney U test (p_{MWU}) for each pair of methods are shown in the lower part of Table 5.

We used the Kruskal-Wallis test for each statement sample to test the null hypothesis that the answers regarding the three different methods would originate from the same distribution. For statements Q1 and Q2, this null hypothesis can be rejected with very low p-values ($p_{KW} < 0.01$ for Q1 and $p_{KW} < 0.001$ for Q2), meaning that with very high confidence at least one of the methods is significantly different from the others in performance. Even for Q3, the null hypothesis can be rejected, since $p_{KW} < 0.05$, although not quite with the same level of confidence.

According to the Mann-Whitney U tests, the difference between CNN and Baseline is statistically significant with high confidence, since $p_{MWU} < 0.01$ for all statements. The same applies to the difference

Baseline	Heuristic	CNN
Consumer prices in Finland	**Consumer prices in Finland**	**Consumer prices in Finland**
In March 2020, in Finland, the monthly growth rate of the harmonized consumer price index for the category 'health' was 2.4 points. In Turkey, the harmonized consumer price index for the category 'health' was 70.26 points more than in US. It was 181.7 points. In February 2020, it was 65.53 points more than in US. In March 2020, the monthly growth rate of the harmonized consumer price index for the category 'health' was 2.8 points. In February 2020, the harmonized consumer price index for the category 'health' was 176.79 points.	In March 2020, in Finland, the monthly growth rate of the harmonized consumer price index for the category 'health' was 2.4 points. It was 2.2 percentage points more than in US. In Turkey, the harmonized consumer price index for the category 'health' was 70.26 points more than in US. It was 181.7 points. The monthly growth rate of the harmonized consumer price index for the category 'health' was 2.8 points. Finland had the 2nd highest monthly growth rate of the harmonized consumer price index for the category 'health' across the observed countries.	In March 2020, in Finland, the monthly growth rate of the harmonized consumer price index for the category 'health' was 2.4 points. It was 2.2 percentage points more than in US. The country had the 2nd highest monthly growth rate of the harmonized consumer price index for the category 'health' across the observed countries. In February 2020, the country had the 12th highest monthly growth rate of the harmonized consumer price index for the category 'health' across the observed countries. It was -0.3 points. It was 0.5 percentage points less than in US.
In January 2020, in Finland, the monthly growth rate of the harmonized consumer price index for the category 'education' was 1 percentage points less than in US. In February 2020, in Turkey, it was 0.9 points. It was 0.7 percentage points more than in US. In Sweden, it was 0.8 points. It was 0.6 percentage points more than in US. In January 2020, in Estonia, it was 1.3 percentage points more than in US.	In January 2020, the monthly growth rate of the harmonized consumer price index for the category 'education' was 1 percentage points less than in US. It was -0.8 points. It was 0.8 percentage points less than the EU average. In Estonia, it was 1.3 percentage points more than in US. It was 1.5 points. It was 1.5 percentage points more than the EU average.	In January 2020, the monthly growth rate of the harmonized consumer price index for the category 'education' was 1 percentage points less than in US. It was -0.8 points. It was 0.8 percentage points less than the EU average. In February 2020, it was 0 points. It was 0.2 percentage points less than in US. The country had the 8th highest monthly growth rate of the harmonized consumer price index for the category 'education' across the observed countries.

Figure 2: Example texts used when evaluating the machine learning based document planning method. Texts generated by the baseline document planner (left), the heuristic-based method (middle), and the CNN-based method (right).

Table 5: Upper table: medians, means, and standard deviations of the answers to the questionnaire comparing our neural and heuristic content selection methods with a baseline. For each statement, the answer ranges are 1–7, where higher values are better. Lower table: *p*-values for the Kruskal-Wallis test for each statement sample as well as for the Mann-Whitney U test between each pair of methods within each statement sample.

	Baseline			Heuristic			CNN		
Statement	Median	Mean	St.dev.	Median	Mean	St.dev.	Median	Mean	St.dev.
Q1, corr. to heading	4.5	4	1.97	4.5	4.65	1.27	6	5.8	0.94
Q2, coherence	3.5	3.45	1.58	4.5	4.4	1.11	5	5.15	0.94
Q3, usefulness	4	3.9	1.85	4	4.55	1.28	6	5.3	1.18

		Statement		
Test	Variants	Q1	Q2	Q3
Kruskall-Wallis	All	0.0026	0.0009	0.0168
Mann-Whitney U	Baseline – Heuristic	0.1552	0.0185	0.1082
Mann-Whitney U	CNN – Baseline	0.0010	0.0003	0.0036
Mann-Whitney U	CNN – Heuristic	0.0038	0.0161	0.0307

between CNN and Heuristic, although not with as high confidence, and for Q3 the difference is not significant after Bonferroni correction (corrected significance level is $\frac{0.05}{3} \approx 0.017$).

In contrast, the difference between Baseline and Heuristic is not statistically significant for any of the three statements. This result reflects our observation about the survey results, where the difference between these two methods was relatively small.

Altogether, the above results of the survey and the statistical tests indicate that our neural content selection method has superior performance in relation to the simpler baseline method. It also modestly overperforms the heuristic method evaluated in Section 3.1. This suggests that the neural network is able to learn patterns about the relationship between sentence position and newsworthiness, and supports our initial assumption that more newsworthy sentences tend to appear earlier than others. In addition, being data-driven, the method requires fewer assumptions to be made about the data.

Although our neural method brings about better performance in terms of the survey results, we acknowledge that it has a couple of disadvantages in comparison with the heuristic method. First, it is computationally heavier than the heuristic method, which might be an issue for online use cases. In other use cases, where the user does not need to wait for the output, the method is probably more suitable. Second, the method requires a suitable training corpus, the availability of which varies according to the application domain. In our experiments, the corpus was of the same genre of text as in the generation, but including a much wider domain of news topics. This suggests that our approach could potentially be well transferable at least within the same genre. Otherwise, the method's performance will most likely be affected by the similarity of chosen training data to the generation domain.

4 Headline generation

A mixed UH and JSI team has continued its work on headline generation in the low-resource scenario using encoder-decoder architectures. While the initial stages of this work have already been described in Deliverable D5.6, in this deliverable we report on some additional experiments using two distinct neural architectures and the final evaluation results on two Embeddia datasets.

4.1 Methodology

The main objective was to adapt the current state-of-the-art Transformer based approaches to settings with limited amount of data and find the most suitable headline generation strategy for the Embeddia languages. We tested two distinct encoder-decoder neural approaches.

The first approach involved a multilingual encoder-decoder model BART (Tang et al., 2020). More specifically, we employed monolingual BART for English, while for experiments on Estonian and Croatian we used its multilingual version mBART-50 (Tang et al., 2020).

The second approach involved an encoder-decoder architecture consisting of two pretrained BERT (Devlin, Chang, Lee, & Toutanova, 2019) models connected by a randomly initialised cross-attention layer, for which weights need to be learned from scratch. We refer to this model as BERT encoder-decoder (BERT-ED). For English we used two "bert-base-uncased" models, for Estonian and Croatian we used the FinEst BERT and CroSloEngual BERT described in Ulčar and Robnik-Šikonja (2020).

We test the models in a low-resource setting, on two Embeddia languages (Croatian and Estonian), on datasets ranging from 10 000 to roughly 30 000 documents. In order to compare the models' performance in this low-resource scenario to their performance in an ideal high-resource scenario, the models are also tested on two English datasets, one small and one large (see Section 4.2 for details).

We investigate whether using and combining different pretraining schemes can improve the performance of the model. More specifically, we test three distinct pretraining techniques:

- **Text infilling**: As proposed by Lewis et al. (2019), about 20% of the training corpus is corrupted by an in-filling scheme, where spans of text are replaced with a single mask token. The encoder-decoder is then trained to generate the original text from the corrupted input.
- **Sentence shuffling**: Same as in Lewis et al. (2019), the input sentences are randomly shuffled and the model is trained to generate the original text with the correct sentence order.
- **Two tasks**: The model is first trained to restore the correct order of shuffled sentences and than to restore the corpus corrupted by the text in-filling scheme.

Note that pretraining is performed using only the headline generation training dataset and no additional data is used. This way, we inspect if the model's performance can be improved by extensive pretraining instead of obtaining more data.

To increase the size of the training corpus we employ several data augmentation techniques.

- **BERT-based augmentation**: 20% of the words in the news article are masked. Then, the masked article is fed to the BERT model, who proposes probable candidates for the masked tokens. These tokens are replaced by the most probable candidates, creating new articles to be added to the training set.
- **Word2vec augmentation**: For each news article in the train set, we replace random words in the articles by synonyms proposed by the Word2vec model.[2]
- **Wordnet augmentation**: This method is similar to the previous one, but replacement candidates are obtained from Wordnet.
- **EDA augmentation**: EDA, proposed by Wei and Zou (2019), consists of four operations: Wordnet synonym replacement, random insertion, random swap, and random deletion.

[2]We set the number of runs parameter to 5 and probability of replacement to 0.3 (i.e. the algorithm goes through the text five times and tries to augment each sentence with a 0.3 probability). English word2vec embeddings are trained on the Google News dataset, Croatian word2vec embeddings are trained on the Croatian Web Corpus (HrWAC) (Ljubešić & Erjavec, 2011; Šnajder, 2014) while the Estonian embeddings are trained on the Estonian Reference Corpus (Kaalep, Muischnek, Uiboaed, & Veskis, 2010).

- **Mixed augmentation**: Each article in the train set is first augmented with Word2vec. The augmented article is fed to the EDA-based augmentation and the output of this augmentation is additionally fed to the Wordnet-based augmentation.

All augmentation techniques except for BERT have been previously established and are available in the TextAugment library[3]: For English, we used all augmentation strategies. For Croatian and Estonian only BERT and word2vec augmentations are available since Wordnet is not available for these languages.

For each original article in the train set, we generate 5 augmented articles using the algorithms described above. These new articles are inserted into the original training set and used for training of the headline generation model. We opted to generate five augmented texts per article, as initial experiments suggested that using a smaller number results in an insufficient increase of the training dataset, and using a larger number results in repetitions of the training examples.

4.2 Experimental Setting

Experiments were conducted on three datasets, namely the Estonian ExM news article dataset (Purver, Pollak, et al., 2021), the Croatian 24sata news article dataset (Purver, Shekhar, Pranjić, Pollak, & Martinc, 2021) and the English KPTimes dataset (Gallina, Boudin, & Daille, 2019). The dataset statistics are presented in Table 6. For Croatian and Estonian, we use the same train and test dataset splits as in the recent study on keyword extraction (Koloski, Pollak, Škrlj, & Martinc, 2021).

The English dataset is included in our experiments to serve as a benchmark for several comparisons. First, we wish to research whether there is a discrepancy in the quality of produced headlines between English (for which most NLG models are originally produced) and two low-resource languages, Estonian and Croatian. Second, besides conducting low-resource experiments, the abundance of resources in English allows us to obtain results for the high-resource scenario, to which we can compare our low-resource results. For this reason, we use both the large KPTimes train set, containing about 260,000 news articles, and the original KPTimes validation set, containing 10,000 articles, which we employ as a 'low-resource' English train set and train models on it. Since we do not use these datasets as training and validation sets, we refer to them as 260K and 10K respectively to avoid terminology confusion.

Both BART and BERT-ED approaches are first tested in a high resource scenario, i.e. by training them on the 260K KPTimes train set. The results of these experiments are used as a reference point of how well these models work in an ideal scenario with plenty of data available, to which we can compare results of our low-resource experiments. Next, both of these models are trained on the 10K set, the Estonian train set, and the Croatian train set without any additional pretraining or data augmentation. These low-resource reference points are used as baselines that we want to improve through various pretraining and data augmentation methods.

In our experiments, we employ the same training and generation regime for both models. The input news articles are truncated at 128 tokens, since we assume that the most important content of the news, to which the title most likely refers to, is covered at the beginning of the article. The length of the output is limited to 30 tokens; finally, for generation we employ a beam search of size 5 and early stopping.

The quality of the generated headlines is evaluated by the standard ROGUE[4] score and two newly proposed measures:

- Semantic similarity (SS) between true and generated headline, using sentence embeddings

[3] https://github.com/dsfsi/textaugment

[4] More specifically, we evaluate the headlines by employing ROUGE-1, which measures the overlap of unigrams between the original headlines and headlines generated by the system, ROUGE-2, which measures the overlap of bigrams, and ROUGE-L, which is the longest common subsequence based statistics that also considers sentence level structure similarity and identifies longest co-occurring n-grams.

Table 6: News datasets used for empirical evaluation of headline generation (number of documents).

Language	train set	test set
English 260K (KPTimes train)	259,923	10,000
English 10K (KPTimes valid)	10,000	10,000
Croatian	32,223	3,582
Estonian	10,750	7,747

- Textual entailment between both headlines, i.e. Natural language inference (NLI)

4.3 Results

The results of the experiments on the English dataset are presented in Table 7 and the results of the experiments on Estonian and Croatian datasets are presented in Table 8.

Both BERT-ED and BART models perform well in the ideal high-resource scenario when trained on the large 260K train set (see approaches labeled as "BASELINE 260K" for both English models), with BART outperforming BERT-ED by roughly 4 points according to all three ROUGE scores, by about 2 points according to SS and by almost 5 points according to NLI.

On the other hand, when the models are compared in a low-resource scenario, the gap between the model's performance drastically increases (see approaches labeled as BASELINE for Estonian and Croatian models and the approach labeled as "BASELINE 10K" for English models). This is due to the drastic decrease in BERT-ED's performance when trained on the small 10K dataset.

While the results for BERT-ED clearly indicate that only training the model from scratch on a corpus of limited size is not a viable option, BART-based models on the other hand show more robust performance, even when trained in the low-resource scenario. For English, training the BART model on the 10K dataset results in a modest drop of about 3 points according to all criteria, when compared to the BART model trained on the 260K dataset. The results for Estonian and Croatian are worse, yet still much better than for the BERT-ED-based models. On Estonian, the multilingual mBART model achieves ROUGE-1 of 26.2, ROUGE-2 of 12.3, ROUGE-L of 24.3 and SS score of 56.7.

While comparison of ROUGE and SS scores across languages is problematic,[5] these scores—and the manual inspection confirming the quality of the produced headlines—indicate that an extensively pretrained multilingual model can be successfully applied in a low-resource scenario. The mBART results for Croatian are worse, which is interesting, since the Croatian train set is three times the size of the Estonian one. They can nevertheless be explained by the fact that mBART-50 was pretrained on a much smaller Croatian corpus than the Estonian one (Tang et al., 2020).

Next, we discuss the results of the **data augmentation** and pretraining experiments. Generally speaking, the results indicate that these experiments have on the one hand a significant influence on the performance of BERT-ED-based models and a negligible influence on the performance of the BART-based models. When it comes to English data augmentation, all but one (Word2Vec augmentation) method manage to beat the BERT-ED 10K baseline score. The biggest improvement can be observed for the BERT augmentation. Decent improvements according to all criteria can also be observed when EDA and Wordnet augmentation are used. Mix augmentation does not work that well, probably because texts become very different after the multi-step process and not always preserve the original meaning. It is hard to fine-tune augmentation parameters, since this would require retraining of the corresponding headline generation model.

For Croatian, the data augmentation improvements are smaller than for English; BERT data augmentation does not work at all. As the Croatian training dataset is three times bigger than the English

[5] This is especially true when comparison needs to be made between a morphologically rich language, such as Estonian, and a morphologically less diverse language, such as English.

Table 7: Results of experiments on the English datasets. Best results in a low resource setting (i.e. excluding the BART and BERT-ED models trained on English 260K dataset) per evaluation measure are **bolded**. For each measure, we report its absolute value (the first number) and the difference with the baseline model (the second colored, number). Since all experiments with data augmentation and pretraining are run on the 10K dataset, differences are computed respectively to the 10K baseline, i.e. the first row of results for each model.

Approach		ROUGE-1		ROUGE-2		ROUGE-L		SS		NLI	
English BERT-ED models											
BASELINE	10K	10.2		1.4		9.6		24.6		15.4	
	260K	27.6		10.1		25.1		49.6		32.1	
AUGMENTATION	bert	13.2	3.0	2.3	0.9	12.2	2.6	30.8	6.2	15.7	0.3
	w2v	9.7	-0.5	1.6	0.2	8.9	-0.7	26.5	1.9	14.9	-0.5
	mix	10.4	0.2	1.7	0.3	9.6	0.0	23.9	-0.7	13.1	-2.3
	eda	12.8	2.6	2.2	0.8	11.9	2.3	29.5	4.9	15.2	-0.2
	wordnet	12.4	2.2	2.1	0.7	11.5	1.9	29.3	4.7	15.2	-0.2
PRETRAINING	infilling	11.7	1.5	1.9	0.5	10.7	1.1	31.0	6.4	18.9	3.5
	shuffling	12.9	2.7	2.6	1.2	11.8	2.2	36.0	11.4	18.8	3.4
	two tasks	16.5	6.3	4.6	3.2	15.1	5.5	42.0	17.4	25.9	10.5
English BART models											
BASELINE	10K	**29.0**		**10.9**		**26.0**		49.3		34.1	
	260K	31.9		13.1		28.7		51.7		36.8	
AUGMENTATION	bert	28.5	-0.5	10.5	-0.4	25.6	-0.4	49.1	-0.2	34.0	-0.1
	w2v	27.8	-1.2	10.1	-0.8	25.1	-0.9	48.2	-1.1	32.0	-2.1
	mix	27.7	-1.3	10.2	-0.7	25.0	-1.0	47.9	-1.4	32.2	-1.9
	eda	28.3	-0.7	10.4	-0.5	25.5	-0.5	49.0	-0.3	33.2	-0.9
	wordnet	28.2	-0.8	10.3	-0.6	25.3	-0.7	48.7	-0.6	33.4	-0.7
PRETRAINING	infilling	29.0	0.0	10.9	0.0	26.0	0.0	**49.5**	**0.2**	34.2	0.1
	shuffling	28.8	-0.2	10.8	-0.1	25.9	-0.1	49.4	0.1	**34.3**	**0.2**
	two tasks	28.7	-0.3	10.7	-0.2	25.9	-0.1	49.2	-0.1	34.1	0.0

and Estonian ones, we deduce that increasing the dataset size with data augmentation techniques might be less beneficial for larger datasets. The highest improvements over the BERT-ED baseline for data augmentation are observed for the Estonian dataset. Indeed, the BERT-ED baseline—which most likely did not converge due to the lack of training data—returns mostly repetitive or empty strings, while data augmentations apparently creates enough additional training data to generate more coherent content.

For the BART-based models, all data augmentation strategies lead to scores lower than the baseline for all languages. While the reduction is in most cases minimal, these scores nevertheless do indicate that the augmented data is not of sufficient quality for the pretrained model to obtain useful information that can be successfully leveraged during NLG training.

By **pretraining** the BERT-ED-based models, using text infilling and sentence shuffling tasks, on the same datasets on which they are later fine-tuned for headline generation, we obtain substantial performance boosts. The increase in performance is even larger than with data augmentation. For English and Estonian, it is especially useful to apply both pretraining regimes, sentence shuffling and text infilling, sequentially (see the row in Tables 7 and 8 labeled as “PRETRAINING two tasks”). For Croatian, text infilling works slightly better than sentence shuffling according to most criteria, but combining these two approaches does not improve the performance.

Pretraining the BART-based models leads to small improvements for Estonian and Croatian, and to small reduction for English. The monolingual English BART, which was extensively pretrained on a massive English corpus using the same denoising tasks we employ here, apparently does not profit from the additional pretraining on a small corpus. The pretraining experiments for the multilingual mBART-50 on the other hand consistently show small improvements across all three pretraining regimes and for both languages.

The average increase in performance for data augmentation and pretraining across all languages and

Table 8: Results of experiments on the Croatian and Estonian datasets. Best results per language and per evaluation measure are **bolded**. For each measure, we report its absolute value (the first number) and the difference with the baseline model (the second, colored, number). The differences are computed in respect to the baseline.

Approach		ROUGE-1		ROUGE-2		ROUGE-L		SS	
Croatian BERT-ED models									
BASELINE		9.6		1.0		8.9		29.7	
AUGMENTATION	bert	2.5	-7.1	0.0	-1.0	2.5	-6.4	10.2	-19.5
	w2v	11.0	1.4	1.4	0.4	0.1	1.1	33.7	4.0
PRETRAINING	infilling	16.6	7.0	4.2	3.2	14.8	5.9	44.9	15.2
	shuffling	15.2	5.6	3.6	2.6	13.4	4.5	43.9	14.2
	two tasks	15.4	5.8	4.2	3.2	13.6	4.7	45.9	16.2
Croatian mBART models									
BASELINE		20.5		7.3		18.1		49.6	
AUGMENTATION	bert	19.8	-0.7	6.8	-0.5	17.6	-0.5	49.8	0.2
	w2v	18.3	-2.2	5.8	-1.5	16.3	-1.8	47.9	-1.7
PRETRAINING	infilling	21.0	0.5	**7.5**	**0.2**	18.6	0.5	**51.1**	**1.5**
	shuffling	**21.2**	**0.7**	7.4	0.1	**18.7**	**0.6**	50.8	1.2
	two tasks	20.8	0.3	7.2	-0.1	18.4	0.3	50.9	1.3
Estonian BERT-ED models									
BASELINE		3.9		0.3		3.8		17.9	
AUGMENTATION	bert	9.8	5.9	2.5	2.2	9.4	5.6	36.9	19.0
	w2v	8.5	4.6	2.1	1.8	8.1	4.3	34.4	16.5
PRETRAINING	infilling	13.9	0.1	4.3	4.0	13.2	9.4	44.0	26.1
	shuffling	11.3	7.4	2.8	2.5	10.7	6.9	40.7	22.8
	two tasks	17.6	13.7	6.5	6.2	16.3	12.5	49.8	31.9
Estonian mBART models									
BASELINE		26.2		12.3		24.4		56.7	
AUGMENTATION	bert	25.4	-0.8	11.6	-0.7	23.8	-0.6	55.9	-0.8
	w2v	23.0	-3.2	9.8	-2.5	21.5	-2.9	53.5	-3.2
PRETRAINING	infilling	**27.1**	**0.9**	**12.9**	**0.6**	**25.2**	**0.8**	**57.2**	**0.5**
	shuffling	26.6	0.4	12.6	0.3	24.8	0.4	56.9	0.2
	two tasks	26.6	0.4	12.3	0.0	24.6	0.2	56.6	-0.1

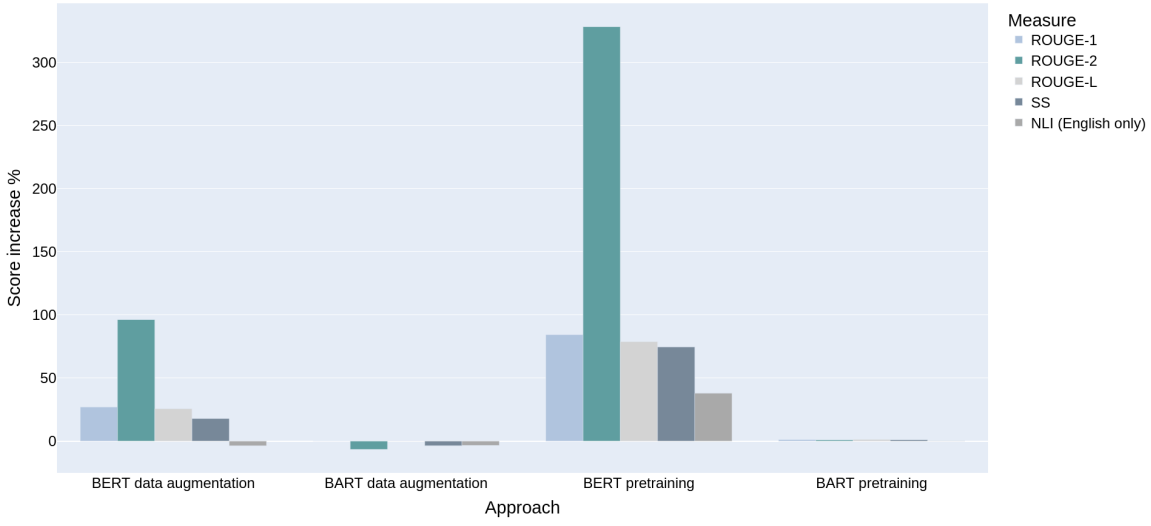

Figure 3: Average *increase* in performance for pretraining and data augmentation approaches for both models across the three languages according to five evaluation measures: three ROUGE scores, semantic similarity (SS) and NLI (only for English).

Table 9: Examples of English headlines generated by various models.

	EXAMPLE 1	EXAMPLE 2
True headline	martial law is rescinded in a philippine province	fighting n. y. c. soda ban, industry focuses on personal choice
BART 260K baseline	philippine president lifts martial law	soda industry fights new york city's soda ban
BART 10K baseline	philippine president lifts martial law	soft-drink industry takes aim at sugary drinks
BERT-ED baseline 260K	philippine president lifts martial law in southern philippines	soft - drink industry seeks to fight sugary drinks ban on sugary drinks
BERT-ED baseline 10K	obama's court's ban in court	in new yorks's york taxs's taxs s
BERT-ED 10K + BERT augmentation	philippines's ban in philippines' ban in philippines' ban in philippines	in new york city, new york city's new york city's bans law
BERT-ED 10K + shuffling	philippines : lawmakers seek lawmaker's ban on lawmakers obama lawmakers arroyo's lawmakers arroyo's lawmakers	u. s. and new york's new new york city mayor' campaign campaign moves new york's mayor's campaign campaign
BERT-ED 10k + infilling	new new new new york city party party leader s. o. p. s. a. leader s. o. p.	philippines : s. o. p. to be suspended s. a. lawmakers s. ban s. a.'s
BERT-ED 10K + two tasks	president's decision to rebuke military law ends in conflict philippines arroyo's rebuke philippines's supreme court in	new yorkers face a challenge to soda industry in new yorkers in new yorkers' campaign campaign in new york city's

for both models is visualized in Figure 3. It is visible that the employment of data augmentation or pretraining leads to on average much larger increase in performance when BERT-ED-based models are used. The measure that benefits the most from these additional steps is ROUGE-2, most likely since this is the hardest criterion of the model's quality, which is only slightly above zero for most baseline BERT-ED-based approaches. On the other hand, the figure clearly shows that both pretraining and data augmentation have only a marginal effect on the BART-based models.

4.4 Qualitative assessment

We manually checked the outputs of several English models[6]. The BART model, fine-tuned on the 10K dataset produces one of the the best results. However, it can hallucinate (see Example 2 in Table 9) or shift the focus of the headline. The manual inspection did not reveal any large differences between the BART-based model trained on the 10K dataset and on the 260K dataset. Interestingly, Example 1 (Table 9) results in identical outputs for BART models trained on both datasets, as well as in *all* other modifications we try with BART. Variation between outputs are rare and, in most cases, not significant; thus, it is hard to judge which outputted headline is better. On the contrary, the performance of the BERT-ED-based model trained on the 10K dataset drops drammatically compared to the one trained on the 260K dataset, as could be seen in the same table. In most cases, it produces ungrammatical sequences with many repetitions.

Data augmentation only slightly improves the performance on English. According to numerical results in Table 8, the best augmentation method is BERT-based augmentation. However, as could be seen in Table 9, the outputs are still ungrammatical, though the meaning is closer to the true headlines. Similar results were obtained with other augmentation strategies.

In our experiments, pretraining has a more positive effect, though repetitions and hallucinations are still possible, as can be seen in the last row in Table 9. Pretraining results in much longer output sequences, where in most of the cases only the first 5-6 words make sense, and then the model starts making repetitions as if it did not know where to stop.

All BERT-ED-based models overuse possessive suffixes in an ungrammatical way. Text infilling strategy

[6]Note that here we do not conduct a Likert scale-based expert evaluation, as we do for the generated documents in Sections 2 and 3. The reason for this lies in the availability of the gold standard headlines, to which we can compare the generated headlines using a well established ROUGE measure and two novel measures, which also consider semantic similarity. We do however acknowledge that there is as of yet no sufficient substitute for a proper manual evaluation, therefore we plan to do this in the future.

also results in overusing of abbreviations, though this problem disappears in a “two task” pretraining (the last two rows in Table 9).

This work is described in full in Martinc, Montariol, Pivovarova, and Zosa (2022), attached here as Appendix C.

5 Conclusions

In this deliverable we have described the final evaluation efforts relating to work conducted within Work Package WP5.

In terms of the work conducted in Task T5.1, our results indicate that the fundamental technology developed therein is sound. For one, our qualitative analysis of the technical features, taken together with our ability to adapt the basic technologies to multiple languages and domains as well as being able to augment the basic rule-based approach with various neural and non-neural processing methods, indicate that the basic technological approach fits the our design goals. These results are complemented by the human evaluations conducted by domain experts, i.e. journalists. While there clearly are further improvements that could be made both to the technology in general and as well as the specific case study systems, the results in general indicate that the journalists see promise in the evaluated systems and view them as useful.

In Task T5.2, we developed several content selection and document planning methods which were evaluated in the context of the Eurostat news generation case study. Our results indicate that the three approaches evaluated here have different upsides and downsides. For example, it appears that the neural machine learning method described in Section 3.3 performs better than the heuristic approach evaluated in Section 3.1), but on the other hand depends on the existence of a text corpus containing texts from the same genre and increases the compute times and system resource requirements. Concurrently, while the heuristic-based approach outperforms a word embedding based approach (Section 3.2, it assumes that data points can be associated with hierarchical labels that can be in turn analyzed to determine the degree of semantic similarity between the data points, while the word embedding based approach makes no assumption. These differences between the assumptions made by the different approaches make them suitable for different concrete use cases.

In the context of headline generation (T5.3), the results suggest that if there exists a pretrained multilingual NLG model for a specific low-resource language, this option for headline generation should be picked over the employment of the encoder-decoder architecture consisting of two pretrained BERT (Devlin et al., 2019) models connected by the randomly initialised cross-attention layer. The successful training of a randomly initialized cross-attention layer, connecting the two language models, is crucial for the model’s performance and is dependent on a large corpus not available in low-resource languages. We have shown that while pretraining and data augmentation can drastically improve the performance of the BERT-based models in headline generation, it has little effect on the BART-based models which have already been extensively pretrained on the same denoising tasks, text infilling and sentence shuffling, that we employ in our experiments. The experiments also suggest that pretraining on the train set is a better option than data augmentation since the improvements are larger and since data augmentation had a negative effect on the performance of the BART-based models, most likely due to the insufficient quality of the data augmentation algorithms.

6 Associated outputs

This work is associated with the following publications:

Citation	Status	Appendix
van Miltenburg, E. et al. (2021) Underreporting of errors in NLG output, and what to do about it. In *Proceedings of the 14th International Conference on Natural Language Generation*. Association for Computational Linguistics.	Published	Appendix A
Leppänen, L., & Toivonen, H. (2021) A Baseline Document Planning Method for Automated Journalism. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*. Linköping University Electronic Press.	Published	Appendix B
Martinc, M. et al. (2022) Data Augmentation and Pretraining to Improve Neural Headline Generation in Low-Resource Setting. In *Proceedings of the 13th Language Resources and Evaluation Conference (LREC 2022)*.	Submitted	Appendix C

As noted above, some evaluations were already reported in earlier deliverables (especially in D5.6 for various creative tasks) and in the respective original articles (Alnajjar et al., 2019; Alnajjar & Toivonen, 2021; Alnajjar & Hämäläinen, 2021; Wright & Purver, 2021) and are thus not repeated here.

References

Alnajjar, K., & Hämäläinen, M. (2021). When a computer cracks a joke: Automated generation of humorous headlines. In *Proceedings of the 12th international conference on computational creativity (iccc 2021)* (p. 292-299). Coimbra, Portugal.

Alnajjar, K., Leppänen, L., & Toivonen, H. (2019). No time like the present: Methods for generating colourful and factual multilingual news headlines. In *The 10th international conference on computational creativity* (pp. 258–265).

Alnajjar, K., & Toivonen, H. (2021). Computational generation of slogans. *Natural Language Engineering*, *27*(5), 575–607. doi: 10.1017/S1351324920000236

Angwin, J., Larson, J., Mattu, S., & Kirchner, L. (2016). Machine bias. *ProPublica, May*, *23*, 2016.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019, June). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)* (pp. 4171–4186). Minneapolis, Minnesota: Association for Computational Linguistics. Retrieved from `https://www.aclweb.org/anthology/N19-1423` doi: 10.18653/v1/N19-1423

Dušek, O., Howcroft, D. M., & Rieser, V. (2019). Semantic noise matters for neural natural language generation. In *Proceedings of the 12th international conference on natural language generation* (pp. 421–426).

Entman, R. M. (1993). Framing: Towards clarification of a fractured paradigm. *McQuail's reader in mass communication theory*, 390–397.

Gallina, Y., Boudin, F., & Daille, B. (2019, October–November). KPTimes: A large-scale dataset for keyphrase generation on news documents. In *Proceedings of the 12th international conference on natural language generation* (pp. 130–135). Tokyo, Japan: Association for Computational Linguistics. Retrieved from `https://www.aclweb.org/anthology/W19-8617` doi: 10.18653/v1/W19-8617

Goldberg, E., Driedger, N., & Kittredge, R. I. (1994). Using natural-language processing to produce weather forecasts. *IEEE Expert*, *9*(2), 45–53.

Hackett, R. A. (1984). Decline of a paradigm? bias and objectivity in news media studies. *Critical Studies in Media Communication*, *1*(3), 229–259.

Harcup, T., & O'neill, D. (2017). What is news? news values revisited (again). *Journalism studies*, *18*(12), 1470–1488.

Hofstetter, C. R., & Buss, T. F. (1978). Bias in television news coverage of political events: A methodological analysis. *Journal of Broadcasting & Electronic Media*, *22*(4), 517–530.

Kaalep, H.-J., Muischnek, K., Uiboaed, K., & Veskis, K. (2010). The estonian reference corpus: Its composition and morphology-aware user interface. In *Baltic hlt* (pp. 143–146).

Koloski, B., Pollak, S., Škrlj, B., & Martinc, M. (2021). Keyword extraction datasets for Croatian, Estonian, Latvian and Russian 1.0.

Kunert, J., & Thurman, N. (2019). The form of content personalisation at mainstream, transatlantic news outlets: 2010–2016. *Journalism Practice*, *13*(7), 759–780.

Leppänen, L., Munezero, M., Granroth-Wilding, M., & Toivonen, H. (2017). Data-driven news generation for automated journalism. In *Proceedings of the 10th international conference on natural language generation* (pp. 188–197).

Leppänen, L., & Toivonen, H. (2021). A baseline document planning method for automated journalism. In *Proceedings of the 23rd nordic conference on computational linguistics (nodalida).*

Leppänen, L., Tuulonen, H., & Sirén-Heikel, S. (2020). Automated journalism as a source of and a diagnostic device for bias in reporting. *Media and Communication*.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text summarization branches out* (pp. 74–81).

Linden, C.-G. (2017). Decades of automation in the newsroom: Why are there still so many jobs in journalism? *Digital Journalism*, *5*(2), 123–140.

Ljubešić, N., & Erjavec, T. (2011). hrWaC and slWaC: Compiling web corpora for Croatian and Slovene. In *International conference on text, speech and dialogue* (pp. 395–402).

Martinc, M., Montariol, S., Pivovarova, L., & Zosa, E. (2022). Data augmentation and pretraining to improve neural headline generation in low-resource setting. In *Proceedings of the 13th language resources and evaluation conference (lrec 2022).*

McBride, K., & Rosenstiel, T. (2013). *The new ethics of journalism: Principles for the 21st century*. CQ Press.

Mindich, D. T. (2000). *Just the facts: How" objectivity" came to define american journalism*. NYU Press.

Nie, F., Yao, J.-G., Wang, J., Pan, R., & Lin, C.-Y. (2019). A simple recipe towards reducing hallucination in neural surface realisation. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 2673–2679).

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics* (pp. 311–318).

Plattner, T., & Orel, D. (2019). Addressing microaudiences at scale. In *communication présentée à computation+ journalism conference, miami university, floride* (pp. 1–2).

Puduppully, R., Dong, L., & Lapata, M. (2019). Data-to-text generation with content selection and planning. In *Proc. 33rd aaai conference on artificial intelligence.*

Purver, M., Pollak, S., Freienthal, L., Kuulmets, H.-A., Krustok, I., & Shekhar, R. (2021). Ekspress news article archive (in estonian and russian) 1.0.

Purver, M., Shekhar, R., Pranjić, M., Pollak, S., & Martinc, M. (2021). 24sata news article archive 1.0.

Reiter, E. (2018). *Hallucination in neural NLG.* https://ehudreiter.com/2018/11/12/hallucination-in-neural-nlg/. (Accessed: 2020-03-02)

Scheufele, D. A. (1999). Framing as a theory of media effects. *Journal of communication*, *49*(1), 103–122.

Schmitt, M., Sharifzadeh, S., Tresp, V., & Schütze, H. (2019). Unsupervised text generation from structured data. *arXiv preprint arXiv:1904.09447*.

Šnajder, J. (2014). DerivBase. hr: A high-coverage derivational morphology resource for Croatian. In *Proceedings of the 9th international conference on language resources and evaluation* (pp. 3371–3377).

Stark, J. A., & Diakopoulos, N. (2016). Towards editorial transparency in computational journalism. *Computation + Journalism Symposium*.

Stray, J. (2019). Making artificial intelligence work for investigative journalism. *Digital Journalism*, *7*(8), 1076–1097.

Tang, Y., Tran, C., Li, X., Chen, P.-J., Goyal, N., Chaudhary, V., ... Fan, A. (2020). Multilingual translation with extensible multilingual pretraining and finetuning. *arXiv preprint arXiv:2008.00401*.

Ulčar, M., & Robnik-Šikonja, M. (2020). Finest bert and crosloengual bert: less is more in multilingual models. *arXiv preprint arXiv:2006.07890*.

van Miltenburg, E., Clinciu, M., Dušek, O., Gkatzia, D., Inglis, S., Leppänen, L., ... Wen, L. (2021, August). Underreporting of errors in NLG output, and what to do about it. In *Proceedings of the 14th international conference on natural language generation* (pp. 140–153). Aberdeen, Scotland, UK: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.inlg-1.14

Wei, J., & Zou, K. (2019). Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*.

Wright, G., & Purver, M. (2021, September). Evaluating natural language descriptions generated in a workspace-based architecture. In A. Gómez de Silva Garza, T. Veale, W. Aguilar, & R. Pérez y Pérez (Eds.), *Proceedings of the 12th International Conference on Computational Creativity (ICCC)* (p. 87-91). Online: Association for Computational Creativity. Retrieved from https://computationalcreativity.net/iccc21/wp-content/uploads/2021/09/ICCC_2021_paper_97.pdf

Yu, R. (2014). How robots will write earnings stories for the ap. *USA Today*, *30*.

Appendix A: Underreporting of errors in NLG output, and what to do about it

Underreporting of errors in NLG output, and what to do about it

Emiel van Miltenburg,[1*] **Miruna Clinciu,**[2,3,4] **Ondřej Dušek,**[5] **Dimitra Gkatzia,**[6]
Stephanie Inglis,[7] **Leo Leppänen,**[8] **Saad Mahamood,**[9] **Emma Manning,**[10]
Stephanie Schoch,[11] **Craig Thomson,**[7] and **Luou Wen**[12]
[1]Tilburg University, [2]Edinburgh Centre for Robotics, [3]Heriot-Watt University,
[4]University of Edinburgh, [5]Charles University, Prague, [6]Edinburgh Napier University,
[7]University of Aberdeen, [8]University of Helsinki, [9]trivago N.V., [10]Georgetown University,
[11]University of Virginia, [12]University of Cambridge
Contact: c.w.j.vanmiltenburg@tilburguniversity.edu

Abstract

We observe a severe under-reporting of the different kinds of errors that Natural Language Generation systems make. This is a problem, because mistakes are an important indicator of where systems should still be improved. If authors only report overall performance metrics, the research community is left in the dark about the specific weaknesses that are exhibited by 'state-of-the-art' research. Next to quantifying the extent of error under-reporting, this position paper provides recommendations for error identification, analysis and reporting.

1 Introduction

This paper turned out very differently from the one we had initially intended to write. Our original intention was to write an overview of the different kinds of errors that appear in the output of different kinds of Natural Language Generation (NLG) systems, and to develop a general taxonomy of NLG output errors, based on the publications that have appeared at previous INLG conferences (similar to Howcroft et al. 2020; Belz et al. 2020). This, however, turned out to be impossible. The reason? There is a severe under-reporting of the different kinds of errors that NLG systems make. By this assertion, we mean that authors neither include any error analysis nor provide any examples of errors made by the system, and they do not make reference to different kinds of errors that may appear in the output. The latter is a lower bar than carrying out an error analysis, which requires a more systematic approach where several outputs are sampled and analysed for the presence of errors, which are then categorised (ideally through a formal procedure with multiple annotators). Section 3 provides more detailed statistics about error reporting in different years of INLG (and ENLG), and the amount of papers that discuss the kinds of errors that may appear in NLG output.

The fact that errors are under-reported in the NLG literature is probably unsurprising to experienced researchers in this area. The lack of reporting of negative results in AI has been a well-known issue for many years (Reiter et al., 2003). With the classic NLG example being the reporting of negative results for the STOP project on smoking cessation (Reiter et al., 2001, 2003). But even going in with (relatively) low expectations, it was confronting just to see how little we as a community look at the mistakes that our systems make.

We believe that it is both necessary *and* possible to improve our ways. One of the reasons why it is necessary to provide more error analyses (see §2.2 for more), is that otherwise, it is unclear what are the strengths and weaknesses of current NLG systems. In what follows, we provide guidance on how to gain more insight into system behavior.

This paper provides a general framework to carry out error analyses. First we cover the terminology and related literature (§2), after which we quantify the problem of under-reporting (§3). Following up on this, we provide recommendations on how to carry out an error analysis (§4). We acknowledge that there are barriers to a more widespread adoption of error analyses, and discuss some ways to overcome them (§5). Our code and data are provided as supplementary materials.

2 Background: NLG systems and errors

2.1 Defining errors

There are many ways in which a given NLG system can fail. Therefore it can be difficult to exactly define all the different types of errors that can possibly occur. Whilst error analyses in past NLG literature were not sufficient for us to create a taxonomy, we will instead propose high-level distinctions to help

[*]This project was led by the first author. Remaining authors are presented in alphabetical order.

Proceedings of the 14th International Conference on Natural Language Generation (INLG), pages 140–153,
Aberdeen, Scotland, UK, 20-24 September 2021.

bring clarity within the NLG research community.

This paper focuses on text errors, which we define as countable instances of things that went wrong, as identified from the generated text.[1] Text errors apply when something is incorrect in the generated text with respect to the data, an external knowledge source, or the communicative goal.

Through our focus on text errors, we only look at the *product* (what comes out) of an NLG system, so that we can compare the result of different kinds of systems (e.g., rule-based pipelines versus neural end-to-end systems), with error categories that are independent of the *process* (how the text is produced).[2] For completeness, we discuss errors related to the production process in §2.3.

By *error analysis* we mean the identification and categorisation of errors, after which statistics about the distribution of error categories are reported. It is an annotation process (Pustejovsky and Stubbs, 2012; Ide and Pustejovsky, 2017), similar to Quantitative Content Analysis in the social sciences (Krippendorff, 2018; Neuendorf, 2017).[3] Error analysis can be carried out during development (to see what kinds of mistakes the system is currently making), as the last part of a study (evaluating a new system that you are presenting), or as a standalone study (comparing different systems). The latter option requires output data to be available, ideally for both the validation and test sets. A rich source of output data is the GEM shared task (Gehrmann et al., 2021).

Text errors can be categorised in several different types, including factual errors (e.g. incorrect number; Thomson and Reiter 2020), and errors related to form (spelling, grammaticality), style (formal versus informal, empathetic versus neutral), or behavior (over- and under-specification). Some of these are universally wrong, while others may be 'contextually wrong' with respect to the task success or for a particular design goal. For example, formal texts aren't wrong *per se*, but if the goal is to produce informal texts, then any semblance of formality may be considered incorrect.

It may be possible to relate different kinds of errors to the different dimensions of text quality identified by Belz et al. (2020). What is crucial here, is that we are able to identify the specific thing which went wrong, rather than just generate a number that is representative of overall quality.

2.2 Why do authors need to report errors?

There is a need for realism in the NLG community. By providing examples of different kinds of errors, we can show the complexity of the task(s) at hand, and the challenges that still lie ahead. This also helps set realistic expectations for users of NLG technology, and people who might otherwise build on top of our work. A similar argument has been put forward by Mitchell et al. (2019), arguing for 'model cards' that provide, inter alia, performance metrics based on quantitative evaluation methods. We encourage authors to also look at the data and provide examples of where systems produce errors. Under-reporting the types of errors that a system makes is harmful because it leaves us unable to fully appreciate the system's performance.

While some errors may be detected automatically, e.g., using information extraction techniques (Wiseman et al., 2017) or manually defined rules (Dušek et al., 2018), others are harder or impossible to identify if not reported. We rely on researchers to communicate the less obvious errors to the reader, to avoid them going unnoticed and causing harm for subsequent users of the technology.

Reporting errors is also useful when comparing different implementation paradigms, such as pipeline-based data-to-text systems versus neural end-to-end systems. It is important to ask where systems fall short, because different systems may have different shortcomings. One example of this is the E2E challenge, where systems with similar human rating scores show very different behavior (Dušek et al., 2020).

Finally, human and automatic evaluation metrics, or at least the ones that generate some kind of intrinsic rating, are too coarse-grained to capture relevant information. They are general evaluations of system performance that estimate an average-case performance across a limited set of abstract dimensions (if they measure anything meaningful

[1] We use the term 'text' to refer to any expression of natural language. For example, sign language (as in Mazzei 2015) would be considered 'text' under this definition.

[2] By focusing on countable instances of things that went wrong in the output text, we also exclude issues such as bias and low output diversity, that are global properties of the collection of outputs that a system produces for a given amount of inputs, rather than being identifiable in individual outputs.

[3] There has been some effort to automate this process. For example, Shimorina et al. (2021) describe an automatic error analysis procedure for shallow surface realisation, and Stevens-Guille et al. (2020) automate the detection of repetitions, omissions, and hallucinations. However, for many NLG tasks, this kind of automation is still out of reach, given the wide range of possible correct outputs that are available in language generation tasks.

at all; see Reiter 2018). We don't usually know the worst-case performance, and we don't know what kinds of errors cause the metrics or ratings to be sub-optimal. Additionally, the general lack of extrinsic evaluations among NLG researchers (Gkatzia and Mahamood, 2015) means that in some cases we only have a partial understanding of the possible errors for a given system.

2.3 Levels of analysis

As noted above, our focus on errors in the output text is essential to facilitate framework-neutral comparisons between the performance of different systems. When categorizing the errors made by different systems, it is important to be careful with terms such as *hallucination* and *omission*, since these are process-level (pertaining to the system) rather than product-level (pertaining to the output) descriptions of the errors.[4] Process-level descriptions are problematic because we cannot reliably determine how an error came about, based on the output alone.[5] We can distinguish between at least two causes of errors, which we define below: system problems and data problems. While these problems should be dealt with, we do not consider them to be the subject of error analysis.

System problems can be defined as the malfunctioning of one or several components in a given system, or the malfunctioning of the system as a whole. System problems in rule/template-based systems could be considered as synonymous to 'bugs,' which are either semantic and/or syntactic in nature. If the system has operated in a mode other than intended (e.g., as spotted through an error analysis), the problem has to be identified, and then corrected. Identifying and solving such problems may require close involvement of domain experts for systems that incorporate significant domain knowledge or expertise (Mahamood and Reiter, 2012). Van Deemter and Reiter (2018) provide further discussion of how errors could occur at different stages of the NLG pipeline system. System problems in end-to-end systems are harder to identify, but recent work on interpretability/explainability aims to improve this (Gilpin et al., 2019).

Data problems are inaccuracies in the input that are reflected in the output. For example: when a player scored three goals in a real-world sports game, but only one goal is recorded (for whatever reason) in the data, even a perfect NLG system will generate an error in its summary of the match. Such errors may be identified as factual errors by cross-referencing the input data with external sources. They can then be further diagnosed as data errors by tracing back the errors to the data source.

3 Under-reporting of errors

We examined different *NLG conferences to determine the amount of papers that describe (types of) output errors, and the amount of papers that actually provide a manual error analysis.

3.1 Approach

We selected all the papers from three SIGGEN conferences, five years apart from each other: INLG2010, ENLG2015, and INLG2020. We split up the papers such that all authors looked at a selection of papers from one of these conferences, and informally marked all papers that discuss NLG errors in some way. These papers helped us define the terms 'error' and 'error analysis' more precisely.

In a second round of annotation, multiple annotators categorised all papers as 'amenable' or 'not amenable' to an error analysis. A paper is amenable to an error analysis if one of its primary contributions is presenting an NLG system that produces some form of output text. So, NLG experiments are amenable to an error analysis, while survey papers are not.[6] For all amenable papers, the annotator indicated whether the paper (a) mentions any errors in the output and (b) whether it contains an error analysis.[7] We encouraged discussion between annotators whenever they felt uncertain (details in Appendix A). The annotations for each paper were subsequently checked by one other annotator, after which any disagreements were adjudicated through

[4] Furthermore, terms like *hallucination* may be seen as unnecessary anthropomorphisms that trivialise mental illness.

[5] A further reason to avoid process-level descriptors is that they are often strongly associated with one type of approach. For example, the term 'hallucination' is almost exclusively used with end-to-end systems, as it is common for these systems to add phrases in the output text that are not grounded in the input. In our experience, pipeline systems are hardly ever referred to as 'hallucinating.' As such, it is better to avoid the term and instead talk about concrete phenomena in the output.

[6] Examples of other kinds of papers that are not amenable include evaluation papers, shared task proposals, papers which analyze patterns in human-produced language, and papers which describe a component in ongoing NLG work which does not yet produce textual output (e.g. a ranking module).

[7] As defined in § 2, errors are (countable) instances of something that is wrong about the output. An 'error mention' is a reference to such an instance or a class of such instances. Error analyses are formalised procedures through which annotators identify and categorise errors in the output.

Venue	Total	Amenable	Error mention	Error analysis	Percentage with error analysis
INLG2010	37	16	6	0	0%
ENLG2015	28	20	4	1	5%
INLG2020	46	35	19	4	11%

Table 1: Annotation results for different SIGGEN conferences, showing the percentage of amenable papers that included error analyses. Upon further inspection, most error mentions are relatively general/superficial.

a group discussion.

3.2 Results

Table 1 provides an overview of our results. We found that only five papers at the selected *NLG conferences provide an error analysis,[8] and more than half of the papers fail to mention any errors in the output. This means that the INLG community is systematically under-informed about the weaknesses of existing approaches. In light of our original goal, it does not seem to be a fruitful exercise to survey all SIGGEN papers if so few authors discuss any output errors. Instead, we need a culture change where authors discuss the output of their systems in more detail. Once this practice is more common, we can start to make generalisations about the different kinds of errors that NLG systems make. To facilitate this culture change, we give a set of recommendations for error analysis.

4 Recommendations for error analysis

We provide general recommendations for carrying out an error analysis, summarized in Figure 1.

4.1 Setting expectations

Before starting, it is important to be clear about your goals and expectations for the study.

Goal Generally speaking, the goal of an error analysis is to find and quantify system errors statistically, to allow a thorough comparison of different systems, and to help the reader understand the shortcomings of your system. But your personal goals and interests may differ. For example, you may only be interested in *grammatical* errors, and less so in factual errors.

Expected errors When starting an error analysis, you may already have some ideas about what kinds of errors might appear in the outputs of different systems. These ideas may stem from the literature (theoretical limitations, or discussions of errors), from your personal experience as an NLG researcher, or it might just be an impression you have from talking to others. You might also have particular expectations about what the distribution of errors will look like.

Both goals and expectations may bias your study, and cause you to overlook particular kinds of errors. But if you are aware of these biases, you may be able to take them into account, and later check if the results confirm your original expectations. Hence, it may be useful to preregister your study, so as to make your thoughts and plans explicit (Haven and Grootel, 2019; van Miltenburg et al., 2021). This also makes it easier for others to check whether they agree with the assumptions behind your study.

4.2 Set-up

Given your goals and expectations, there are several design choices that you have to make, in order to carry out your study.

Systems and outputs Since error analysis is relatively labor-intensive, it may not be feasible to look at a wide array of different systems. In that case, you could pre-select a smaller number of models, either based on automatic metric scores, or based on specific model features you are interested in. Alternatively, you could see to what extent the model outputs overlap, given the same input. If two models produce exactly the same output, you only need to annotate that output once.

Number of outputs Ideally, the number of outputs should be based on a power analysis to provide meaningful comparisons (Card et al., 2020; van der Lee et al., 2021), but other considerations, such as time and budget, may be taken into account.

Sample selection Regarding the selection of examples to analyze, there are three basic alternatives: The most basic is *random sampling* from the validation/test outputs. Another option is selecting *specific kinds of inputs* and analysing all corresponding outputs. Here, inputs known to be difficult/adversarial or inputs specifically targeting system properties or features may be selected

[8] Summaries of these error analyses are in Appendix B.

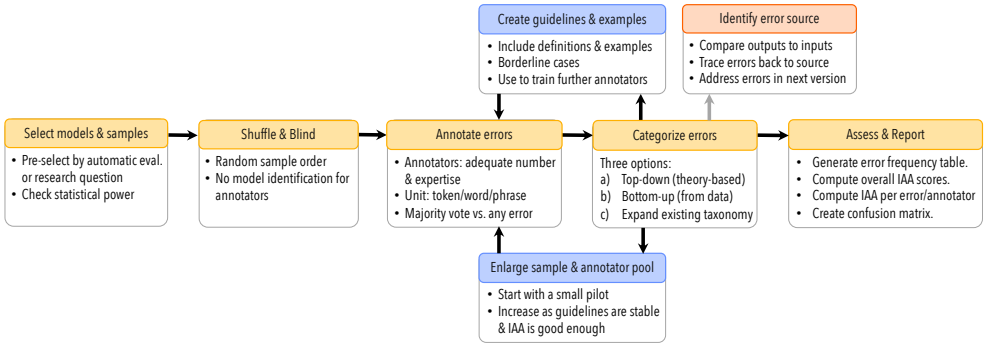

Figure 1: Flowchart depicting recommended analysis steps, as described in §4. IAA stands for Inter-Annotator Agreement, as measured through Cohen's kappa or Krippendorff's alpha, for example.

(Ribeiro et al., 2020). Finally, examples to analyze may also be selected based on *quantitative values*: automatic metric scores or ratings in a preceding general human evaluation. This way, error analysis can provide explanation for the automatic or human scores. The most suitable option depends on your specific use case: While random selection gives the least biased picture of the model performance, selecting specifically hard and/or low-rated samples may be more efficient. Also note that the sample selection should always be independent of any samples you may have previously examined during system development, since any errors for those cases are likely to have been resolved already (although you cannot be sure until you have verified these cases as well).

Presentation The order of the items should be randomized (to reduce possible order effects), and if multiple system variants are considered, the annotators must not know which system produced which output (to minimise annotator bias).

Interface The efficiency of any annotation task depends on the quality of the interface. With the right interface, annotators may be able to annotate more data in a shorter time frame. Monarch (2021, Chapter 11) provides recommendations on interface design based on principles from the field of Human-Computer Interaction (HCI). Once you have a working interface, it is important to test the interface and obtain feedback from annotators to see whether it can be made more intuitive or efficient (e.g. by adding keyboard shortcuts to perform common operations).[9]

[9]Note that keyboard operations are generally quicker than using the mouse (Monarch, 2021).

Annotators and the annotation process The annotation process can be split into two parts: identifying the errors (§4.3), and categorising the errors (§4.4). These can either be carried out sequentially (first identify, then categorize) or simultaneously (asking annotators to both identify and categorize errors at the same time). The choices you make here also impact annotator requirements, and the evaluation of the annotation procedure.

Number of annotators Generally speaking, having more annotators reduces the prevalence of the individual bias (Artstein and Poesio, 2005). This is particularly relevant if we want to detect all the errors in the output data. Having more annotators means that we are less likely to overlook individual instances of errors. Once those errors are identified, it may make more sense to rely on a smaller set of well-trained annotators to categorise the different errors. In the ideal situation, all errors are annotated by (at least) two judges so as to be able to detect and resolve any disagreements afterwards. If this is not possible, then you should at least double-annotate a large enough sample to reliably estimate inter-annotator agreement.[10]

Role of the annotators Ideally, the annotators should be independent of the authors reporting the error analysis (Neuendorf, 2017), to ensure that the results are not influenced by any personal biases about the systems involved, and that the annotations are indeed based on the guidelines themselves rather than on discussions between the authors. If this is not feasible, then the authors should at least ensure that they remain ignorant of the identity of

[10]See Krippendorff 2011 for a reference table to determine the sample size for Krippendorff's α. Similar studies exist for Cohen's κ, e.g. Flack et al. 1988; Sim and Wright 2005.

the system that produced the relevant outputs.

Degree of expertise Depending on the complexity of the annotation guidelines, the error analysis may require expertise in linguistics (in the case of a theory-driven error categorisation scheme), or the relevant application area (with a context-driven error categorisation scheme). For example, Mahamood and Reiter (2012) worked with nurses to identify errors in reports generated for parents of neonatal infants. Taking into consideration the costly process of selecting domain expert annotators and the importance of quality control, non-domain experts might be also considered, ensuring their qualification through (intensive) training (Artstein and Poesio, 2005; Carlson et al., 2001).[11]

Compensation and treatment of workers If annotators are hired, either directly or via a crowdsourcing platform such as MTurk, they should be compensated and treated fairly (Fort et al., 2011). Silberman et al. (2018) provide useful guidelines for the treatment of crowd-workers. The authors note that they should at least be paid the minimum wage, they should be paid promptly, and they should be treated with respect. This means you should be ready to answer questions about the annotation task, and to streamline the task based on worker feedback. If you use human participants to annotate the data, you likely also need to apply for approval by an Institutional Review Board (IRB).

Training Annotators should receive training to be able to carry out the error analysis, but the amount of training depends on the difficulty of the task (which depends, among other factors, on the *coding units* (see §4.3), and the number of error types to distinguish). They should be provided with the annotation guidelines (§4.5), and then be asked to annotate texts where the errors are known (but not visible). The solutions would ideally be created by experts, although in some cases, solutions created by researchers may be sufficient (Thomson and Reiter, 2020). It should be decided in advance what the threshold is to accept annotators for the reaming work, and, if they fail, whether to provide additional training or find other candidates. Note that annotators should also be compensated for taking part in the training (see previous paragraph).

[11] At least on the MTurk platform, Requesters can set the entrance requirements for their tasks such that only Workers who passed a qualifying test may carry out annotation tasks.

4.3 Identifying the errors

Error identification focuses on discovering all errors in the chosen output samples (as defined in the introduction). Previously, Popović (2020) asked error annotators to identify issues with comprehensibility and adequacy in machine-translated text. Similarly, Freitag et al. (2021) proposed a manual error annotation task where the annotators identified and highlighted errors within each segment in a document, taking into account the document's context as well as the severity of the errors.

The major challenge in this annotation step is how to determine the units of analysis; should annotators mark individual tokens, phrases, or constituents as being incorrect, or can they just freely highlight any sequence of words? In content analysis, this is called *unitizing*, and having an agreed-upon unit of analysis makes it easier to process the annotations and compute inter-annotator agreement (Krippendorff et al., 2016).[12] What is the right unit may depend on the task at hand, and as such is beyond the scope of this paper.[13]

A final question is what to do when there is disagreement between annotators about what counts as an error or not. When working with multiple annotators, it may be possible to use majority voting, but one might also be inclusive and keep all the identified errors for further annotation. The error categorization phase may then include a category for those instances that are not errors after all.

4.4 Categorizing errors

There are three ways to develop an error categorisation system:

1. **Top-down** approaches use existing theory to derive different types of errors. For example, Higashinaka et al. (2015a) develop an error taxonomy based on Grice's (1975) Maxims of conversation. And the top levels of Costa et al.'s (2015) error taxonomy[14] are based on general linguistic theory, inspired by Dulay et al. (1982).

2. **Bottom-up** approaches first identify different

[12] Though note that Krippendorff et al. do provide a metric to compute inter-annotator agreement for annotators who use units of different lengths.

[13] One interesting solution to the problem of unitization is provided by Pagnoni et al. (2021), who do not identify individual errors, but do allow annotators to "check all types that apply" at the sentence level. The downside of this approach is that it is not fine-grained enough to be able to count individual instances of errors, but you do get an overall impression of the error distribution based on the sentence count for each type.

[14] Orthography, Lexis, Grammar, Semantic, and Discourse.

errors in the output, and then try to develop coherent categories of errors based on the different kinds of attested errors. An example of this is provided by Higashinaka et al. (2015b), who use a clustering algorithm to automatically group errors based on comments from the annotators (verbal descriptions of the nature of the mistakes that were made). Of course, you do not have to use a clustering algorithm. You can also manually sort the errors into different groups (either digitally[15] or physically[16]).

3. **Expanding on existing taxonomies**: here we make use of other researchers' efforts to categorize different kinds of errors, by adding, removing, or merging different categories. For example, Costa et al. (2015) describe how different taxonomies of errors in Machine Translation build on each other. In NLG, if you are working on data-to-text, then you could take Thomson and Reiter's (2020) taxonomy as a starting point. Alternatively, Dou et al. (2021) present a crowd-sourced error annotation schema called SCARECROW. For image captioning, there is a more specific taxonomy provided by van Miltenburg and Elliott (2017). Future work may also investigate the possibility of merging all of these taxonomies and relating the categories to the quality criteria identified by Belz et al. (2020).

The problem of error ambiguity To be able to categorize different kinds of errors, we often rely on the *edit-distance heuristic*. That is: we say that the text contains particular kinds of errors, because fixing those errors will give us the desired output. With this reasoning, we take the mental 'shortest path' towards the closest correct text.[17] This at least gives us a set of 'perceived errors' in the text, that provides a useful starting point for future research. However, during the process of identifying errors, we may find that there are multiple 'shortest paths' that lead to a correct utterance, resulting in error ambiguity (see, e.g., Van Miltenburg and Elliott 2017; Thomson and Reiter 2020, §3.3).

For example, if the output text from a sports summary system notes that Player A scored 2 points, while in fact Player A scored 1 point and Player B scored 2 points, should we say that this is a number error (2 instead of 1) or a person error (Player A instead of B)? This example also shows the fragility of the distinction between product and process. It is very tempting to look at what the system did to determine the right category, but it is unclear whether the 'true error category' is always knowable.

There are multiple ways to address the problem of error ambiguity. For instance, we may award partial credit ($1/n$ error categories), mark both types of errors as applying in this situation (overgeneralising, to be on the safe side), or count all ambiguous cases to separately report on them in the overall frequency table. Another solution, used by Thomson and Reiter (2020) is to provide the annotators with a fixed preference order (NAME, NUMBER, WORD, CONTEXT), so that similar cases are resolved in a similar fashion.

4.5 Writing annotation guidelines

Once you have determined an error identification strategy and developed an error categorisation system, you should describe these in a clear set of annotation guidelines. At the very least, these guidelines should contain relevant definitions (of each error category, and of errors in general), along with a set of examples, so that annotators can easily recognize different types of errors. For clarity, you may wish to add examples of borderline cases with an explanation of why they should be categorized in a particular way.

Pilot The development of a categorisation system and matching guidelines is an iterative process. This means that you will need to carry out multiple pilot studies in order to end up with a reliable set of guidelines,[18] that is easily understood by the annotators, and provides full coverage of the data. Pilot studies are also important to determine how long the annotation will take. This is not just practical to plan your study, but also essential to determine how much crowd-workers should be paid per task, so that you are able to guarantee a minimum wage.

4.6 Assessment

Annotators and annotations can be assessed during or after the error analysis.[19]

[15] E.g. via a program like Excel, MaxQDA or Atlas.ti, or a website like https://www.well-sorted.org.

[16] A good example of this *pile sorting* method is provided by Yeh et al. (2014). Blanchard and Banerji (2016) give further recommendations.

[17] Note that we don't know whether the errors we identified are actually the ones that the system internally made. This would require further investigation, tracing back the origins of each different instance of an error.

[18] As determined by an inter-annotator agreement that exceeds a particular threshold, e.g. Krippendorff's $\alpha \geq 0.8$.

[19] And in many cases, the annotators will already have been assessed during the training phase, using the same measures.

During the error analysis Particularly with crowd-sourced annotations it is common to include gold-standard items in the annotation task, so that it is possible to flag annotators who provide too many incorrect responses. It is also possible to carry out an intermediate assessment of inter-annotator agreement (IAA), described in more detail below. This is particularly relevant for larger projects, where annotators may diverge over time.

After the error analysis You can compute IAA scores (e.g., Cohen's κ or Krippendorff's α, see: Cohen 1960; Krippendorff 1970, 2018), to show the overall reliability of the annotations, the pairwise agreement between different annotators, and the reliability of the annotations for each error type. You can also produce a confusion matrix; a table that takes one of the annotators (or the adjudicated annotations after discussion) as a reference, and provides counts for how often errors from a particular category were annotated as belonging to any of the error categories (Pustejovsky and Stubbs, 2012). This shows all disagreements at a glance.

Any analysis of (dis)agreement or IAA scores requires there to be overlap between the annotators. This overlap should be large enough to reliably identify any issues with either the guidelines or the annotators. Low agreement between annotators may be addressed by having an adjudication round, where the annotators (or an expert judge) resolve any disagreements; rejecting the work of unreliable annotators; or revising the task or the annotation guidelines, followed by another annotation round (Pustejovsky and Stubbs, 2012).

4.7 Reporting

We recommend that authors should provide a table reporting the frequency of each error type, along with the relevant IAA scores. The main text should at least provide the overall IAA score, while IAA scores for the separate error categories could also be provided in the appendix. For completeness, it is also useful to include a confusion matrix, but this can also be put in the appendix. The main text should provide a discussion of both the frequency table, as well as the IAA scores. What might explain the distribution of errors? What do the examples from the *Other*-category look like? And how should we interpret the IAA score? Particularly with low IAA scores, it is reasonable to ask why the scores are so low, and how this could be improved. Reasons for low IAA scores include: unclear annotation guidelines, ambiguity in the data, and having one or more unreliable annotator(s). The final annotation guidelines should be provided as supplementary materials with your final report. All annotations and output data (e.g. train, validation, and test outputs, possibly with confidence scores) should of course also be shared.

5 (Overcoming) barriers to adoption

One reason why authors may feel hesitant about providing an error analysis is that it takes up significantly more space than the inclusion of some overall performance statistics. The current page limits in our field may be too tight to include an error analysis. Relegating error analyses to the appendix does not feel right, considering the amount of work that goes into providing such an analysis. Given the effort that goes into an error analysis, authors have to make trade-offs in their time spent doing research. If papers can easily get accepted without any error analysis, it is understandable that this additional step is often avoided. How can we encourage other NLG researchers to provide more error analyses, or even just examples of errors?

Improving our standards We should adopt reporting guidelines that stress the importance of error analysis in papers reporting NLG experiments. The NLP community is already adopting such guidelines to improve the reproducibility of published work (see Dodge et al.'s (2019) reproducibility checklist that authors for EMNLP2020 need to fill in). We should also stress the importance of error reporting in our reviewing forms; authors should be rewarded for providing insightful analyses of the outputs of their systems. One notable example here is COLING 2018, which explicitly asked about error analyses in their reviewing form for NLP engineering experiments, and had a 'Best Error Analysis' award.[20,21]

Making space for error analyses We should make space for error analyses. The page limit in *ACL conferences is already expanding to incorporate ethics statements, to describe the broader impact of our research. This suggests that we have reached the limits of what fits inside standard papers, and an expansion is warranted. An alternative is to publish more journal papers, where there is more space to fit an error analysis, but then we as

[20] https://coling2018.org/paper-types/
[21] http://coling2018.org/index.html%3Fp=1558.html

a community also need to encourage and increase our appreciation of journal submissions.

Spreading the word Finally, we should inform others about how to carry out a proper error analysis. If this is a problem of exposure, then we should have a conversation about the importance of error reporting. This paper is an attempt to get the conversation started.

6 Follow-up work

What should you do after you have carried out an error analysis? We identify three directions for follow-up studies.

Errors in inputs An additional step can be added during the identification of errors which focuses on observing the system inputs and their relation to the errors. Errors in the generated text may occur due to semantically noisy (Dušek et al., 2019) or incorrect system input (Clinciu et al., 2021); for instance, input data values might be inaccurate or the input might not be updated due to a recent change (e.g., new president). To pinpoint the source of the errors, we encourage authors to look at their input data jointly with the output, so that errors in inputs can be identified as such.

Building new evaluation sets Once you have identified different kinds of errors, you can try to trace the origin of those errors in your NLG model, or posit a hypothesis about what kinds of inputs cause the system to produce faulty output. But how can you tell whether the problem is really solved? Or how can you stimulate research in this direction? One solution, following McCoy et al. (2019), is to construct a new evaluation set based on the (suspected) properties of the errors you have identified. Future research, knowing the scope of the problem from your error analysis, can then use this benchmark to measure progress towards a solution.

Scales and types of errors Error types and human evaluation scales are closely related. For example, if there are different kinds of grammatical errors in a text, we expect human grammaticality ratings to go down as well. But the relation between errors and human ratings is not always as transparent as with grammaticality. Van Miltenburg et al. (2020) show that different kinds of semantic errors have a different impact on the perceived overall quality of image descriptions.[22] Future research should aim to explore the connection between the two in more detail, so that there is a clearer link between different kinds of errors and different quality criteria (Belz et al., 2020).

7 Conclusion

Having found that NLG papers tend to underreport errors, we have motivated why authors should carry out error analyses, and provided a guide on how to carry out such analyses. We hope that this paper paves the way for more in-depth discussions of errors in NLG output.

Acknowledgements

We would like to thank Emily Bender and the anonymous reviewers for their feedback. Dimitra Gkatzia's contribution was supported under the EPSRC projects CiViL (EP/T014598/1) and NLG for Low-resource Domains (EP/T024917/1). Miruna Clinciu's contribution is supported by the EPSRC Centre for Doctoral Training in Robotics and Autonomous Systems at Heriot-Watt University and the University of Edinburgh. Miruna Clinciu's PhD is funded by Schlumberger Cambridge Research Limited (EP/L016834/1, 2018-2021). Ondřej Dušek's contribution was supported by Charles University grant PRIMUS/19/SCI/10. Craig Thomson's work is supported under an EPSRC NPIF studentship grant (EP/R512412/1). Leo Leppänen's work has been supported by the European Union's Horizon 2020 research and innovation program under grant 825153 (EMBEDDIA).

References

Imen Akermi, Johannes Heinecke, and Frédéric Herledan. 2020. Tansformer based natural language generation for question-answering. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 349–359, Dublin, Ireland. Association for Computational Linguistics.

Ron Artstein and Massimo Poesio. 2005. Bias decreases in proportion to the number of annotators. *Proceedings of FG-MoL.*

Cristina Barros and Elena Lloret. 2015. Input seed features for guiding the generation process: A statistical approach for Spanish. In *Proceedings of the 15th European Workshop on Natural Language Generation*

[22] Relatedly, Freitag et al. (2021) ask annotators to rate the severity of errors in machine translation output, rather than simply marking errors.

(ENLG), pages 9–17, Brighton, UK. Association for Computational Linguistics.

David Beauchemin, Nicolas Garneau, Eve Gaumond, Pierre-Luc Déziel, Richard Khoury, and Luc Lamontagne. 2020. Generating intelligible plumitifs descriptions: Use case application with ethical considerations. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 15–21, Dublin, Ireland. Association for Computational Linguistics.

Anya Belz, Simon Mille, and David M. Howcroft. 2020. Disentangling the properties of human evaluation methods: A classification system to support comparability, meta-evaluation and reproducibility testing. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 183–194, Dublin, Ireland. Association for Computational Linguistics.

Emily M. Bender, Leon Derczynski, and Pierre Isabelle, editors. 2018. *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, Santa Fe, New Mexico, USA.

Simon J. Blanchard and Ishani Banerji. 2016. Evidence-based recommendations for designing free-sorting experiments. *Behavior Research Methods*, 48(4):1318–1336.

Dallas Card, Peter Henderson, Urvashi Khandelwal, Robin Jia, Kyle Mahowald, and Dan Jurafsky. 2020. With little power comes great responsibility. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9263–9274, Online. Association for Computational Linguistics.

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. 2001. Building a discourse-tagged corpus in the framework of Rhetorical Structure Theory. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue - Volume 16*, SIGDIAL '01, page 1–10.

Miruna-Adriana Clinciu, Dimitra Gkatzia, and Saad Mahamood. 2021. It's commonsense, isn't it? demystifying human evaluations in commonsense-enhanced NLG systems. In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 1–12, Online. Association for Computational Linguistics.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Ângela Costa, Wang Ling, Tiago Luís, Rui Correia, and Luísa Coheur. 2015. A linguistically motivated taxonomy for machine translation error analysis. *Machine Translation*, 29(2):127–161.

Kees van Deemter and Ehud Reiter. 2018. Lying and computational linguistics. In Jörg Meibauer, editor, *The Oxford Handbook of Lying*. Oxford University Press.

Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. 2019. Show your work: Improved reporting of experimental results. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2185–2194, Hong Kong, China. Association for Computational Linguistics.

Yao Dou, Maxwell Forbes, Rik Koncel-Kedziorski, Noah A. Smith, and Yejin Choi. 2021. Scarecrow: A framework for scrutinizing machine text.

Heidi C. Dulay, Marina K. Burt, and Stephen D. Krashen. 1982. *Language two*. New York : Oxford University Press.

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

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2020. Evaluating the State-of-the-Art of End-to-End Natural Language Generation: The E2E NLG Challenge. *Computer Speech & Language*, 59:123–156.

Virginia F Flack, AA Afifi, PA Lachenbruch, and HJA Schouten. 1988. Sample size determinations for the two rater kappa statistic. *Psychometrika*, 53(3):321–325.

Karën Fort, Gilles Adda, and K. Bretonnel Cohen. 2011. Last words: Amazon Mechanical Turk: Gold mine or coal mine? *Computational Linguistics*, 37(2):413–420.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation.

Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Aremu Anuoluwapo, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna Clinciu, Dipanjan Das, Kaustubh D. Dhole, Wanyu Du, Esin Durmus, Ondřej Dušek, Chris Emezue, Varun Gangal, Cristina Garbacea, Tatsunori Hashimoto, Yufang Hou, Yacine Jernite, Harsh Jhamtani, Yangfeng Ji, Shailza Jolly, Mihir Kale, Dhruv Kumar, Faisal Ladhak, Aman Madaan,

Mounica Maddela, Khyati Mahajan, Saad Mahamood, Bodhisattwa Prasad Majumder, Pedro Henrique Martins, Angelina McMillan-Major, Simon Mille, Emiel van Miltenburg, Moin Nadeem, Shashi Narayan, Vitaly Nikolaev, Rubungo Andre Niyongabo, Salomey Osei, Ankur Parikh, Laura Perez-Beltrachini, Niranjan Ramesh Rao, Vikas Raunak, Juan Diego Rodriguez, Sashank Santhanam, João Sedoc, Thibault Sellam, Samira Shaikh, Anastasia Shimorina, Marco Antonio Sobrevilla Cabezudo, Hendrik Strobelt, Nishant Subramani, Wei Xu, Diyi Yang, Akhila Yerukola, and Jiawei Zhou. 2021. The gem benchmark: Natural language generation, its evaluation and metrics.

Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. 2019. Explaining explanations: An overview of interpretability of machine learning. In *Proceedings - 2018 IEEE 5th International Conference on Data Science and Advanced Analytics, DSAA 2018*.

Dimitra Gkatzia and Saad Mahamood. 2015. A Snapshot of NLG Evaluation Practices 2005-2014. In *Proceedings of the 15th European Workshop on Natural Language Generation (ENLG)*, pages 57–60.

H. P. Grice. 1975. *Logic and Conversation*, pages 41 – 58. Brill, Leiden, The Netherlands.

Tamarinde L. Haven and Dr. Leonie Van Grootel. 2019. Preregistering qualitative research. *Accountability in Research*, 26(3):229–244. PMID: 30741570.

Ryuichiro Higashinaka, Kotaro Funakoshi, Masahiro Araki, Hiroshi Tsukahara, Yuka Kobayashi, and Masahiro Mizukami. 2015a. Towards taxonomy of errors in chat-oriented dialogue systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 87–95, Prague, Czech Republic. Association for Computational Linguistics.

Ryuichiro Higashinaka, Masahiro Mizukami, Kotaro Funakoshi, Masahiro Araki, Hiroshi Tsukahara, and Yuka Kobayashi. 2015b. Fatal or not? finding errors that lead to dialogue breakdowns in chat-oriented dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2243–2248, Lisbon, Portugal. Association for Computational Linguistics.

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

Nancy Ide and James Pustejovsky, editors. 2017. *Handbook of linguistic annotation*. Springer. ISBN 978-94-024-1426-4.

Taichi Kato, Rei Miyata, and Satoshi Sato. 2020. BERT-based simplification of Japanese sentence-ending predicates in descriptive text. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 242–251, Dublin, Ireland. Association for Computational Linguistics.

Klaus Krippendorff. 1970. Bivariate agreement coefficients for reliability of data. *Sociological methodology*, 2:139–150.

Klaus Krippendorff. 2011. Agreement and information in the reliability of coding. *Communication Methods and Measures*, 5(2):93–112.

Klaus Krippendorff. 2018. *Content Analysis: An Introduction to Its Methodology*, fourth edition edition. SAGE Publications, Inc.

Klaus Krippendorff, Yann Mathet, Stéphane Bouvry, and Antoine Widlöcher. 2016. On the reliability of unitizing textual continua: Further developments. *Quality & Quantity*, 50(6):2347–2364.

Zewang Kuanzhuo, Li Lin, and Zhao Weina. 2020. SimpleNLG-TI: Adapting SimpleNLG to Tibetan. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 86–90, Dublin, Ireland. Association for Computational Linguistics.

Chris van der Lee, Albert Gatt, Emiel van Miltenburg, and Emiel Krahmer. 2021. Human evaluation of automatically generated text: Current trends and best practice guidelines. *Computer Speech & Language*, 67:101–151.

Saad Mahamood and Ehud Reiter. 2012. Working with clinicians to improve a patient-information NLG system. In *INLG 2012 Proceedings of the Seventh International Natural Language Generation Conference*, pages 100–104.

Alessandro Mazzei. 2015. Translating Italian to LIS in the rail stations. In *Proceedings of the 15th European Workshop on Natural Language Generation (ENLG)*, pages 76–80, Brighton, UK. Association for Computational Linguistics.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Emiel van Miltenburg and Desmond Elliott. 2017. Room for improvement in automatic image description: an error analysis. *CoRR*, abs/1704.04198.

Emiel van Miltenburg, Chris van der Lee, and Emiel Krahmer. 2021. Preregistering NLP research. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*,

pages 613–623, Online. Association for Computational Linguistics.

Emiel van Miltenburg, Wei-Ting Lu, Emiel Krahmer, Albert Gatt, Guanyi Chen, Lin Li, and Kees van Deemter. 2020. Gradations of error severity in automatic image descriptions. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 398–411, Dublin, Ireland. Association for Computational Linguistics.

Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. Model cards for model reporting. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 220–229.

Robert (Munro) Monarch. 2021. *Human-in-the-Loop Machine Learning*. Manning Publications Co., Shelter Island, New York. ISBN 9781617296741.

Adrian Muscat and Anja Belz. 2015. Generating descriptions of spatial relations between objects in images. In *Proceedings of the 15th European Workshop on Natural Language Generation (ENLG)*, pages 100–104, Brighton, UK. Association for Computational Linguistics.

Kimberly A. Neuendorf. 2017. *The Content Analysis Guidebook*. SAGE Publications. Second edition, ISBN 9781412979474.

Jason Obeid and Enamul Hoque. 2020. Chart-to-text: Generating natural language descriptions for charts by adapting the transformer model. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 138–147, Dublin, Ireland. Association for Computational Linguistics.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829, Online. Association for Computational Linguistics.

Maja Popović. 2020. Informative manual evaluation of machine translation output. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5059–5069, Barcelona, Spain (Online). International Committee on Computational Linguistics.

James Pustejovsky and Amber Stubbs. 2012. *Natural Language Annotation for Machine Learning: A guide to corpus-building for applications.* " O'Reilly Media, Inc.".

Ehud Reiter. 2018. A structured review of the validity of BLEU. *Computational Linguistics*, 44(3):393–401.

Ehud Reiter, Roma Robertson, A. Scott Lennox, and Liesl Osman. 2001. Using a randomised controlled clinical trial to evaluate an NLG system. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 442–449, Toulouse, France. Association for Computational Linguistics.

Ehud Reiter, Roma Robertson, and Liesl M. Osman. 2003. Lessons from a failure: Generating tailored smoking cessation letters. *Artificial Intelligence*, 144(1):41–58.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Anastasia Shimorina, Yannick Parmentier, and Claire Gardent. 2021. An error analysis framework for shallow surface realisation. *Transactions of the Association for Computational Linguistics*, 9.

M. S. Silberman, B. Tomlinson, R. LaPlante, J. Ross, L. Irani, and A. Zaldivar. 2018. Responsible research with crowds: Pay crowdworkers at least minimum wage. *Commun. ACM*, 61(3):39–41.

Julius Sim and Chris C Wright. 2005. The Kappa Statistic in Reliability Studies: Use, Interpretation, and Sample Size Requirements. *Physical Therapy*, 85(3):257–268.

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.

Mariët Theune, Ruud Koolen, and Emiel Krahmer. 2010. Cross-linguistic attribute selection for REG: Comparing Dutch and English. In *Proceedings of the 6th International Natural Language Generation Conference*.

Craig Thomson and Ehud Reiter. 2020. A gold standard methodology for evaluating accuracy in data-to-text systems. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 158–168, Dublin, Ireland. Association for Computational Linguistics.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

Hung-Wen Yeh, Byron J Gajewski, David G Perdue, Angel Cully, Lance Cully, K Allen Greiner, Won S Choi, and Christine Makosy Daley. 2014. Sorting it out: Pile sorting as a mixed methodology for exploring barriers to cancer screening. *Quality & quantity*, 48(5):2569–2587.

A Annotation

A.1 Procedure and definitions

We annotated all papers from INLG2010, ENLG2015, and INLG2020 in two rounds. Round 1 was an informal procedure where we generally checked whether the papers mentioned any errors at all (broadly construed, without defining the term 'error'). Following this, we determined our formal annotation procedure, based on the example papers: first check if the paper is amenable. If so, check if it (a) mentions any errors in the output or (b) contains an error analysis. We used the following definitions:

Amenable A paper is amenable to an error analysis if one of its primary contributions is presenting an NLG system that produces some form of output text. So, NLG experiments are amenable to an error analysis, while survey papers are not.

Error Errors are (countable) instances of something that is wrong about the output.

Error mention An 'error mention' is a reference to such an instance or a class of such instances.

Error analysis Error analyses are defined as formalised procedures through which annotators identify and categorise errors in the output.

A.2 Discussion points

The most discussion took place on the topic of amenability. Are papers that just generate prepositions (Muscat and Belz, 2015) or attributes for referring expressions (Theune et al., 2010) amenable to error analysis? And what about different versions of SimpleNLG? (E.g., Kuanzhuo et al. 2020.) Although these topics feel different from, say, data-to-text systems, we believe it should be possible to carry out an error analysis in these contexts as well. In the end, amenability for us is just an artificial construct to address the (potential) criticism that we cannot just report the amount of error analyses as a proportion of all *NLG papers. As such, our definition for amenability is just a quick heuristic. Determining whether a paper really benefits from an error analysis is a more complex issue, that depends on many contextual factors.

B Papers containing error analyses

Below is a brief summary of the error analyses that we found in our annotation study.

1. Barros and Lloret (2015) investigate the use of different seed features for controlled neural NLG. They analysed all the outputs of their model, and categorised them based on existing lists of common grammatical errors and drafting errors.

2. Akermi et al. (2020) explore the use of pre-trained transformers for question-answering. They conducted a human evaluation study, asking 20 native speakers to indicate the presence of errors in the outputs of a French and English system. These errors were categorised as: *extra words, grammar, missing words, wrong preposition, word order*.

3. Beauchemin et al. (2020) aim to generate explanations of *plumitifs* (dockets), based on the text of the dockets themselves. Following the identification of different errors (defined by the authors as "the lack of realizing a specific part (accused, plaintiff or list of charges paragraphs), instead of evaluating the textual generation," they trace the source of the error back to either an earlier information extraction step, or to the generation procedure.

4. Kato et al. (2020) present a BERT-based approach to simplify Japanese sentence-ending predicates. They took a bottom-up approach to classify the 140 cases where their model could not generate any acceptable cases. The authors then relate the error types to different stages of the generation process, and to the general architecture of their system.

5. Obeid and Hoque (2020) present a neural NLG model for automatically providing natural language descriptions of information visualisations (i.e., charts). They manually assessed 50 output examples, and highlighted the different errors in the text. The authors find that, despite their efforts to prevent it, their model still suffers from hallucination. They identify two kinds of hallucination: either the model associates an existing value with the wrong data point, or it simply predicts an irrelevant token.

A **notable exception** is the paper by Thomson and Reiter (2020), who carry out an error analysis of existing output data from three different systems. This paper was not considered amenable, because

it does not present an NLG system of its own, and thus it was not included in our counts. But even if we were to count this paper among the error analyses, the trend remains the same: very few papers discuss errors in NLG output.

Appendix B: A Baseline Document Planning Method for Automated Journalism

A Baseline Document Planning Method for Automated Journalism

Leo Leppänen
University of Helsinki
Department of Computer Science
leo.leppanen@helsinki.fi

Hannu Toivonen
University of Helsinki
Department of Computer Science
hannu.toivonen@helsinki.fi

Abstract

In this work, we present a method for content selection and document planning for automated news and report generation from structured statistical data such as that offered by the European Union's statistical agency, Eurostat. The method is driven by the data and is highly topic-independent within the statistical dataset domain. As our approach is not based on machine learning, it is suitable for introducing news automation to the wide variety of domains where no training data is available. As such, it is suitable as a low-cost (in terms of implementation effort) baseline for document structuring prior to introduction of domain-specific knowledge.

1 Introduction

Automated generation of news texts from structured data – often referred to as 'automated journalism' (Graefe, 2016; Dörr, 2015; Caswell and Dörr, 2018) or 'news automation' (Linden, 2017; Sirén-Heikel et al., 2019; Dierickx, 2019) – is of great interest to various news producers. It is seen as a way of 'providing efficiency, increasing output and aiding in reallocating resources to pursue quality journalism' (Sirén-Heikel et al., 2019, p. 47). While data-to-text NLG systems are still far from common especially among the smaller, regional news industry players, at least among the larger newsrooms the use of NLG approaches has clearly been established (Fanta, 2017).

While secrecy in the industry makes it difficult to establish the commercial reality as an outsider, the limited available evidence indicates that commercial automated journalism is mostly done using rule-based methods despite a surge of academic interest in increasingly complex neural methods for NLG (e.g. Puduppully et al., 2019; Ferreira et al., 2019): Interviews of news automation users indicate that the employed methods are mostly based on templates (Sirén-Heikel et al., 2019), as are the few open source code repositories of real-world news automation systems (Yleisradio, 2018). Indeed, some NLG industry experts believe that especially end-to-end neural models do not match customer needs at this time (Reiter, 2019).

Contributing factors include a lack of control (Reiter, 2019); issues with hallucination of non-grounded output (Nie et al., 2019; Dušek et al., 2019; Reiter, 2018); the difficulty in surgically correcting any issues identified in trained neural models beyond additional training; as well as the difficulty of establishing what the 'worst case' performance of a neural model is.

In addition, we believe that that while neural NLG methods are theoretically highly transferable, the *practical* transferability of neural NLG solutions to many news domains is limited by a lack of training data. While newsrooms have extensive archives of news text, these are rarely associated with the matching data that is the 'input' for each piece of news text (E.g., MacKová and Sido, 2020, pp. 43–44, Kanerva et al., 2019, p. 247). At the same time, the non-trainable methods for NLG, too, suffer from difficulties in transferability and reusability (Linden, 2017).

In this work, we investigate document planning (selecting what content and in what order should appear in the document) for structured, statistical data-to-text NLG in the context of automated journalism targeting human journalists. We are not in search of a perfect method, but rather something that is relatively easy to implement as a subdomain-independent baseline and which can then be enhanced with domain-specific processing later-on. Such a method would make it easier to introduce automated journalism solutions to completely new subdomains within the larger statistical data domain.

2 Structuring Hard News

When queried for insight into news structure, journalists and academics often recite the concept of the "(inverted) news pyramid", where the news article is structured so that the order in which information appears in the text reflects the journalist's belief about the importance of the piece of information (Thomson et al., 2008). While the precise origin of the structure is not clear (Pöttker, 2003), it has become so prototypical that it is held self-evident in the journalistic trade literature: "*Every journalist knows how to write a traditional news text: start with the most important thing and continue until you have either said everything relevant or the space reserved for the story runs out*" (Sulopuisto, 2018, translated from Finnish).

A more rigorous analysis of the structures employed in 'hard' news is presented by White (1997), who argues that hard news articles have an 'orbital' structure consisting of a *nucleus* which represents the main point of the article and *satellites* that give context and additional information about the nucleus. White (1997) assigns the role of the nucleus to the combination of the headline and the lead paragraph of the article, and describes the subsequent paragraphs as the satellites. White (1997) identifies five possible relations between a satellite and the nucleus: elaboration, cause-and-effect, justification, contextualization and apprisal. Thomson et al. (2008), in turn, identify that the satellites can elaborate, reiterate, describe causes or consequences, contextualize or provide additional assessment. An important observation is that – as indicated by 'orbital' – these satellites are relatively freely reorderable without affecting readability or meaning. Together, these two observations indicate that a good document plan for hard news (1) prioritizes more newsworthy items and (2) contains some overarching theme (exemplified by the nucleus) so that the text as a whole is coherent, i.e. the satellites are in some way related to the nucleus.

The relations identified by White (1997) and Thomson et al. (2008) are highly similar to those identified in the more general Rhetorical Structure Theory (RST) (Mann and Thompson, 1988), which uses similar nucleus-satellite terminology. However, whereas White (1997) and Thomson et al. (2008) analyze news text on the level of paragraphs, RST can be applied on a more fine-grained level to much shorter text spans. As RST shows that similar relations can be applied on a sub-paragraph level, we hypothesize that a reasonably approximation of a news article might be constructed by applying White's (1997) orbital theory also *within* paragraphs, by considering the first *sentence* of the paragraph a nucleus, and the others as satellites.

Importantly, we interpret the orbital theory of news structuring to suggest that – as the satellites are freely orderable – the actual *type* of relation is not as important for document planning as knowing that *some* relation exists between the satellite and the nucleus. We hypothesize that while identifying whether a specific (RST) relation exists between two arbitrary pieces of information requires domain knowledge, an approximation of whether two arbitrary pieces of information are related in *some* way could be obtained by inspecting their similarity in a domain-independent fashion.

That is, we expect that a piece of information regarding the US health care funding in 2020 is more likely to be related in *some way* to a piece of information discussing the US health care funding in 2020 than to another piece of information discussing the health care funding in Sweden in 1978. If a heuristic or similarity measure identifying such relations could be identified, it could be used together with some estimate of newsworthiness to construct paragraph and document plans that seek to maximize both the key aspects identified above: newsworthiness and the relatedness of the content.

As noted in the introduction, there is a distinction between the theoretical and the practical transferability of neural processing methods. We believe that a good baseline document planning and content selection approach should avoid the need for training data present in the many of recently proposed document planning and content selection approaches. This rules out as unsuitable most recent work that are based on learning from an aligned corpus of data and human-written texts, such as Angeli et al. (2010), Konstas and Lapata (2013), Wiseman et al. (2017), Zhang et al. (2017), Li and Wan (2018), Dou et al. (2018) and Puduppully et al. (2019).

Outside of these trainable approaches, to our knowledge, most other document planning approaches are based on '*hand-engineered*' (Konstas and Lapata, 2013), domain-specific methods. A highly relevant survey of various document planning methods is presented by Gkatzia (2016). While these previous works are – to at least some degree – domain-specific, they establish concepts

and ideas that are highly relevant for our goal. Both Hallett et al. (2006) and Gatt et al. (2009) describe a core set of information, called 'summary spine' or 'key events', that they hold as more important than the rest of the available information. They, as well as Banaee et al. (2013), also employ a numeric estimate of importance. Demir et al. (2010) identify that content already selected for inclusion in the document plan affects how well suited so-far unselected content is for inclusion. Sripada et al. (2003) identify Gricean maxims (Grice, 1975) as providing requirements for document planning and content selection.

3 Context

Our work on document planning is done in the context of a series of data-to-text NLG applications producing short highlights of structured statistical data. Importantly, the applications are intended to be deployed in contexts where they must be able to produce texts highlighting between 10 and 30 data points from datasets measured in 100.000s of data points. The resulting texts are intended to both alert journalists to potential news and to provide them with a starting place from which to write the final news text.

Our system, adapted from Leppänen et al. (2017a), is based on a pipeline of components with dedicated responsibilities similar to those described by Reiter and Dale (2000) and Reiter (2007). For this work, the relevant part of the architecture is the Document Planner component. This component receives as input two sets of *message* data structures, an example of which is shown in Table 1.[1] The messages are extracted automatically from tables of statistical data obtained from Eurostat.

The *core set* contains messages that are known to be highly relevant to the generation task. Unlike the "summary spine' of Hallett et al. (2006), the set is unlinked and unordered, and not all members of the set are guaranteed to be included in the document plan. The *expanded set*, contains messages that *can* be, but are not guaranteed to be, relevant for the document. Expressed using the terminology from Section 2, we assume that only messages in the core set can be nuclei, while messages from either set can be satellites.

These core and expanded sets are determined automatically from user input. When requesting a new text, the user of the system must define a dataset the text is to be generated from, for example the consumer price data available from Eurostat. This dataset is then divided into the core set and the expanded set by the user when they select what country the generated text should focus on. For example, if the user were to select that the text should discuss French consumer prices, the core set would contain all data from the consumer price dataset that pertains directly to France, while the rest of the consumer price dataset (including data pertaining to the UK, Finland, Croatia, etc.) would be set as the expanded set.

We estimate each message's 'newsworthiness' using the Interquartile Range based method described by Leppänen et al. (2017b) with the values scaled to have mean 0 and standard deviation 1 for the purposes of this computation. The resulting value is conceptually similar to 'importance' of Gatt et al. (2009) and 'risk' of Banaee et al. (2013). The IQR based method compares each data point in turn to a larger distribution, giving it higher scores the further it is from the area between the first and the third quartile of the larger distribution. Values between the quartiles are given a minimal, uniform, score that is dependent on the shape of the distribution. In other words, higher IQR values indicate that the value is more of an outlier compared to the rest of related data in the dataset. As such, it captures a degree of 'unexpectedness', which is an important aspect of newsworthiness (Galtung and Ruge, 1965).

We do not use the domain-specific parts of the method described by Leppänen et al. (2017b). That is, we make no value judgement of whether messages pertaining to French consumer prices are more newsworthy than messages pertaining to Croatian consumer prices, nor do we make judgements of whether changes in the price of education are more or less newsworthy than changes in the price of alcohol and tobacco. However, we do weight the scores so that messages with the `timestamp` field being closer to present receive higher weights, as recency is an important aspect of newsworthiness. While we have described our method for computing the `newsworthiness` value in some detail, we emphasize that for the rest of this article we only assume that the `newsworthiness` values are non-negative and that higher values indicate higher newsworthiness.

More crucially for the method described be-

[1] The concrete implementation details are somewhat more complex. We omit details irrelevant for this work.

low, we specify that the `value_type` fields (which describe how the messages' values are to be interpreted) contain members of a hierarchical taxonomy of data types represented as colon-separated hierarchies of labels. For example, the `value_type` field value `health:cost:hc2:mio_eur` would indicate that the number in the `value` field is the amount of money (`cost`), measured in millions of euros (`mio_eur`), spent by some nation (as defined by the `location` and `location_type` fields) on rehabilitative care (`hc2`) in some time period (as defined by the `timestamp` and `timestamp_type` fields) and that this is part of the larger health care topic (`health`). In our case, these labels are automatically established from the headers of the input data tables.

The goal of document structuring is to produce a three-level tree-structure with ordered children. The root node corresponds to the document as a whole and the mid-level structures correspond to paragraphs. The leaves are the messages selected for inclusion in the document. While the messages have not yet, at this stage, been associated with any linguistic structures, they can be conceptualized as being phrases or very short sentences. We are thus concurrently determining both the content and the structure the document.

We emphasize that our applications are employed in domains where they must be able to select some 10-30 messages from a pool of potential messages numbering in 100,000s. Given infinite computational resources, it would be preferential to construct all possible document plans and then score them in some fashion. This, however, is infeasible given the size of the search space. Previously, other authors have employed, for example, stochastic searches with significantly smaller search spaces (Mellish et al., 1998). Indeed, some kind of a beam search approach could be very useful in smartly searching a subset of the search space. However, we have thus far been unable to identify a document-level metric that adequately balances the 'total amount of newsworthiness' in a text with the length of the text, a requirement for beam search.

4 Research Objective

Based on the above considerations, our main goal is to identify a widely applicable method for content selection and document planning that matches the following requirements:

REQ1: The method needs to be highly performant

REQ2: The method should not be dependent on domain knowledge

REQ3: The document should have a theme

REQ4: The document should have multiple paragraphs but not be excessively long

REQ5: The paragraphs should have distinct themes related to the document theme

REQ6: The paragraph themes should be newsworthy in their own right

REQ7: The paragraphs should not be excessively long or short

REQ8: All messages should relate to the paragraph theme

REQ9: All messages should be newsworthy

REQ10: Within each paragraph, the messages should be presented in an order that produces a coherent narrative

Again, we emphasize that our goal is not to identify a method that is optimal for any specific scenario, but rather to determine a baseline method that is *adequate* for a broad spectrum of applications and sub-domains.

5 A Baseline Approach to Document Planning

Optimally, we would wish to produce some sort of a *globally optimal* document plan. However, as discussed above, this would entail significant computational costs and require a scoring function applicable to the document as a whole. As such, we propose a method for producing document plans in a greedy, linear, and iterative fashion. At every stage, decisions are made considering only a limited local context, thus avoiding the need for a method of determining the global quality of the document plan, thus fulfilling REQ1 ('The method needs to be highly performant').

The document's overall theme, in our use case, is selected by the user who initiates the generation task. In initiating the task, the users selects both a dataset and a focus location. The generation process then derives the *core messages* and *expanded messages* sets (the inputs to the Document Planner, see Section 3) so that both sets discuss the dataset

Field	Description	Example value
`where`	What location the fact relates to	Finland
`where_type`	What the type of the location is	country
`timestamp`	The time (or time range) the fact relates to	2020M05
`timestamp_type`	The type of the timestamp	month
`value`	A (usually) numeric value	0.01
`value_type`	Interpretation of `value`	cphi:hicp2015:cp-hi02:rt01
`newsworthiness`	An estimate of how newsworthy the message is	1

Table 1: An example of a message. The hypothetical message states that in the fifth month of 2020, in Finland, the consumer price index, using the year 2015 as the start of the index, of alcoholic beverages and tobacco changed by 0.01 points with respect to the value of the index during the previous month.

indicated by the user (i.e. messages from other datasets are not generated) and that the core set contains messages pertaining to the user's indicated focus location, while messages pertaining to all other locations are in the expanded set. This fulfills REQ3 ('The document should have a theme'). This step is also independent of the specific subdomain, thus fulfilling REQ2 ('The method should not be dependent on domain knowledge'). This step thus fulfills all the relevant requirements. Next, we'll describe how both the first and subsequent paragraphs can be planned in a way consistent with the requirements defined above.

5.1 Planning the First Paragraph

At the start of the document planning process, we select the most newsworthy message from the *core messages* set to act as the nucleus (n_1) of the first paragraph (p_1). This nucleus establishes the theme of the first paragraph as follows: We inspect the `value_type` field of this first nucleus n_1, and retrieve a prefix `Prefix`(n_1). The prefix is the least amount of colon-separated labels wherein the total amount of prefixes in the core set is greater than the minimal amount of paragraphs a document can have, in our case two. In our case, as a consequence of our label hierarchy, this is always the first three colon-separated units. For the message shown in Table 1, the prefix would thus be `cphi:hicp2015:cp-hi02`, meaning that the first paragraph's theme would be the prices of alcoholic beverages and tobacco. This fulfills REQ5, 'the paragraphs should have distinct themes related to the document theme' for the first paragraph.

Next, the first paragraph is completed with satellites from the union of the *core messages* and the *expanded messages* sets. These satellites are initially filtered so that only messages that have the same prefix as the nucleus n_i are considered in paragraph p_i to fulfill REQ8 ('All messages should relate to the paragraph theme'). The satellites are then selected in a linear, greedy, and iterative manner to fulfill REQ1.

For selecting the k'th satellite to a partially constructed paragraph already containing $k-1$ satellites and one nucleus, we consider both the newsworthiness of the available messages (REQ9), as well as how well they would fit the already constructed segment (REQ8). Observing only the newsworthiness would produce a highly incoherent narrative, whereas focusing only on the narrative risks leaving out highly important information.

Following the reasoning in Section 2, we assume that two subsequent messages are more likely to form a good narrative if they are similar. As such, we need a method for weighing the message's newsworthiness by the similarity of the message to the last message of the under-construction paragraph, thus balancing the requirements of REQ8 and REQ9. In terms of the message objects described in Table 1, it seems to us that the intuitive aspects of similarity are related to the degree of similarity within the 'meta' fields such as `timestamp`, `location` and `value_type`.

For the `timestamp` and `location` fields, we can state that two messages that have identical values in the fields are more similar that two messages that are otherwise the same but have distinct values for said fields. We call this the *contextual* similarity of the messages, and the fields the *contextual fields* (F_c), as these fields provide us access to the larger context in which the `value` and `value_type` fields can be interpreted. Contextual similarity captures the notion that it is likely better to follow a fact about French healthcare spending in 2020 with another piece of information about France in 2020,

rather than about Austria in 1990.

In more precise terms, we propose the following weighing scheme for contextual similarity: The similarity $sim_c(A, B)$ of two messaged A and B is the product of weights $w_f > 1$ for each field f among the contextual fields F_c, where both A and B have the same value for the field:

$$sim_c(A, B) = \prod_{\{f \in F_c | A.f = B.f\}} w_f \quad (1)$$

This value strictly increases as more fields are shared between A and B. We explicitly define the similarity to be zero if there are no fields f where A and B share a value. If w_f is a uniform value for all fields f, this scheme is completely domain-agnostic. Setting different weights w_f for each field $f \in F_c$ allows for encoding some domain knowledge about which fields are the most important for the text, thus providing a method for producing more tailored texts at the cost of slightly violating REQ2. In our case study, we set $w_{timestamp} = 1.1$ and $w_{location} = 1.5$.

The above consideration of similarity still ignores valuable information available from the `value_type` field, which describes how the value in the `value` field is to be interpreted. Denoting `health:cost:hc2:mio_eur` (the cost of rehabilitative care in millions of euros) by T_1, consider its similarity to T_2 = `health:cost:hc2:eur_hab`, the cost of rehabilitative care as euros per inhabitant, and T_3 = `health:cost:hc41:mio_eur`, the cost of health care related imaging services in millions of euros. Intuitively, T_1 and T_2 are thematically closer than T_1 and T_3. We model this similarity between two facts A and B simply as

$$sim_t(A, B) = \frac{1}{s(A, B)} \quad (2)$$

where $s(A, B)$ is the length – in colon-separated units – of the unshared suffix between A and B's `value_type` fields. That is, $s(T_1, T_2) = 1$ whereas $s(T_1, T_3) = 2$. We specify that $sim_t(\cdot, \cdot)$ is zero for all pairs without any shared prefix.

Our formulation of $sim_t(\cdot, \cdot)$ was influenced by the observation that in our context the messages' `value_type` values have a constant number of colon-separated segments. In cases where the lengths of the `value_type` values differ, an alternative formulation of

$$sim'_t(A, B) = \frac{2p(A, B)}{\ell(A) + \ell(B)} \quad (3)$$

where $\ell(\cdot)$ provides the length of the `value_type` value, and $p(\cdot, \cdot)$ is the length of shared *prefix* between A and B, both measured as colon-separated units, might be preferable if also more complex.

When considering whether the k'th satellite s_i^k of paragraph p_i should be a specific candidate $c \in C$, where C is all so far unused messages, we can combine the similarity metrics with the newsworthiness of c into a general fitness value as follows:

$$\begin{aligned} fit(c, x) = c.newsworthiness \\ \times sim_c(c, x) \\ \times sim_t(c, x) \\ \times set\_penalty(c) \end{aligned}$$

The $set\_penalty(c)$ factor depends on whether the message originates from the *core messages* set, or the *extended messages* set. For messages originating from the core message set, the penalty is 1. For messages originating from the extended messages set, the penalty is $\frac{1}{dist+1}$, where $dist$ is the distance from the previous core message.

The final score describing how good of an addition c would be as the kth satellite of the ith paragraph s_i^k is then obtained by taking the average of fitnesses of c in relation to both the nucleus n_i and the previous satellite s_i^{k-1} by computing:

$$score(c, n_i, s_i^{k-1}) = \frac{fit(c, n_i) + fit(c, s_i^{k-1})}{2}$$

This maximizes the newsworthiness of the paragraph's contents (fulfilling REQ9, 'all messages should be newsworthy'), while also enforcing relatedness to the theme of the paragraph (fulfilling REQ8, 'all messages should relate to the paragraph theme') by measuring against the nucleus and with the inclusion of the *set_penalty*. By continuously measuring against the previously selected satellite, the procedure also allows for interludes to e.g. discuss highly newsworthy information related to but not strictly about the paragraph's main topic, or 'thematic drift'. It thus fulfills REQ10 ('Within each paragraph, the messages should be presented in an order that produces a coherent narrative') while also paying attention to the pyramid model of news (See Section 2).

Using $score$, the highest scoring candidate $c_{top} = \arg\max_{c \in C} score(c, n_i, s_i^{k-1})$ is then compared to both an absolute threshold t_{abs} and the newsworthiness of the nucleus n_i multiplied by relative threshold value t_{rel}. Provided that the

maximal paragraph length has not been reached, the top candidate message c_{top} is appended to the paragraph p_i as the k'th satellite s_i^k in the document plan provided that either $score(c_{top}, n_i, s_i^{k-1}) \geq t_{abs}$ or $score(c_{top}, n_i, s_i^{k-1}) \geq t_{rel} \times n_i.newsworthiness$.

These thresholds ensure that the paragraph does not stray into minutiae, whether considered in absolute terms or in relation to the nucleus of the paragraph. In cases where the minimum paragraph length has not been reached, the thresholds are ignored and the top candidate is always appended. This accounts for REQ7 ('The paragraphs should not be excessively long or short').

The above considerations take into account several free parameters, namely the maximal and minimal paragraph lengths as well as the threshold values t_{rel} and t_{abs}. In our case study, we selected the minimal and maximal paragraph lengths as 2 and 5 messages empirically by trialing out various values and observing the resulting texts. These should, naturally, be based on the genre of text and the target audience. For the threshold values we selected 0.2 and 0.5, respectively, using the same method as with the paragraph lengths above. Both the thresholds and the minimal and maximal paragraph lengths should be viewed as (manually) tuneable hyperparameters.

5.2 Planning Subsequent Paragraphs

We then proceed to generate further paragraphs in a manner highly similar to that used when planning the first paragraph. The only distinction is that, when selecting the nucleus n_i for a subsequent paragraph p_i, we obtain the message from the *core messages* set with a highest newsworthiness value that has a prefix (theme) not yet discussed among the previously planned paragraphs p_1 - p_{i-1}:

$$n_i = \arg\max_{c \in C} c.newsworthiness \tag{4}$$

where

$$C = \Big\{ c \in CoreMessages | \texttt{Prefix}(c) \notin \{\texttt{Prefix}(n_k) | k \in [1..i-1]\} \Big\} \tag{5}$$

This ensures that the different paragraphs are highly newsworthy, thus fulfilling REQ6, while also fulfilling REQ5 for having distinct themes for the different paragraphs.

As when constructing the subsequent paragraphs, the total length of the document also needs to be considered. To fulfill REQ4 ('The document should have multiple paragraphs but not be excessively long'), we employ a variation of the method described in the previous section for ending individual paragraphs. A maximal length (in our case, 3 paragraphs) ensures that the document is not allowed to grow beyond reason, whereas a minimal length (for us, 2 paragraphs) ensures that the document is not unreasonably short. After the minimal length has been reached (but not yet the maximal length), a new paragraph is only started if the nucleus of the potential paragraph has a newsworthiness value that is at least 30 % of the newsworthiness value of the first nucleus of the document. This, as with the satellites, ensures that the the document does not stray into minutiae, balancing REQs 4 and 6. the maximal and minimal lengths, as well as the 30 % threshold, were determined by manual fine-tuning and should be viewed as tuneable hyperparameters.

6 Evaluation

The method described above was implemented in a larger NLG application producing news alerts for journalists from datasets provided by Eurostat. A variation of the same application was also developed with a simplified document planner. In this simplified planner, the planner always selects the maximally newsworthy available message as the message without any early stopping threshold. Nuclei are selected from the core messages set, while satellites can be from either set. Contrasting our proposed method with this simplified method enables us to evaluate the importance of narrative coherence in the generated texts. The larger application is multilingual, but the evaluation was conducted using English language texts.

Three experts were recruited from the Finnish News Agency STT, a national European news agency, to evaluate documents on the consumer price indices in five different European nations. For all nations, the judges were shown variants produced by both our proposed method and the simplified method. One of the selected countries is the country the news agency is based in, with the assumption that the judges would have high amounts of world knowledge they would be able to use in evaluating these texts. Another variant pair describes a country that is both relatively small and geographically remote (but still within EU), with the assumption that the journalists are unlikely to

Consumer Prices in Estonia

In June 2020, in Estonia, the monthly growth rate of the harmonized consumer price index for the category 'education' was 30.8 points. It was 30.7 percentage points more than the EU average. In July 2020, it was 0.4 percentage points less than the EU average. It was -0.4 points. In May 2020, the yearly growth rate of the harmonized consumer price index for the category 'education' was -20.5 points. It was 21.9 percentage points less than the EU average.

In August 2020, the monthly growth rate of the harmonized consumer price index for the category 'housing, water, electricity, gas and other fuels' was 2.5 points. It was 2.3 percentage points more than the EU average. In North Macedonia, it was 3 percentage points more than the EU average. It was 3.2 points. Estonia had the 3rd highest monthly growth rate of the harmonized consumer price index for the category 'housing, water, electricity, gas and other fuels' across the observed countries. In Sweden, the monthly growth rate of the harmonized consumer price index for the category 'housing, water, electricity, gas and other fuels' was 3.1 points.

Figure 1: Example output regarding Eurostat statistics on consumer prices. The text contains 12 messages, selected from among 207,210 messages available during generation.

have much world knowledge about this country's consumer prices. The three other countries were selected from among those bordering the first country, with the assumption that the journalists would have some, but not much, world knowledge relating to these countries. The final output texts were not inspected prior to selecting the countries.

All of the texts used in the evaluation were generated from a copy of the same underlying Eurostat dataset, entitled 'Harmonised index of consumer prices - monthly data [ei_cphi_m]'[2] downloaded in September 2020. It contains country-level data regarding the harmonized consumer prices indices, and their change over time, for various EU nations starting from January 1996. We preprocess the data by adding monthly rankings (i.e. determine what country had the greatest, the second greatest, etc. value for a specific index category during any specific month) and comparisons to the EU average values.

As the evaluation was focused on document planning and content selection, the larger system was simplified in some respects, e.g., to not conduct complex aggregation. This was done to minimize the effect of later stages of the generation process on the evaluation. As a result, the language in the evaluated documents was relatively stilted, as exemplified by Figure 1. The only manual alteration was the addition of headings to indicate the texts' intended themes.

The judges did not receive any direct compensation but their employer, the news agency, is a member of the EU-wide EMBEDDIA research project within which parts of this work was conducted. The evaluations were conducted online. The judges were first provided with some basic information on the type of documents they were to read (i.e. that the texts are intended to be news alerts for journalists, rather than publication ready news texts), the length of the task, etc. All instructions were in the judges' native language, in this case Finnish. The judges were not told which texts were produced by which variants nor how many variants were being tested. Following this, the judges were shown the documents one by one. For each document, the judges were asked to indicate their agreement with the following statements (translated from Finnish):

Q1: The text matches the heading

Q2: The text is coherent

Q3: The text lacks some pertinent information

Q4: The text contains unnecessary information

Q5: The text has a suitable length

For Q1–Q4, the judges indicated their agreement on a 7-point Likert scale ranging from 1 ('completely disagree') to 7 ('completely agree'). For Q5, the answers were provided on 5-point scale ranging from 1 ('clearly too short') to 3 ('length is suitable') to 5 ('clearly too long'). In addition, the judges were able to provide textual feedback for each individual text, as well as for the evaluation task as a whole. The judges' answers to Q1 – Q5, are aggregated in Table 2.

The results indicate that the proposed method statistically significantly increases the document's coherence (Q2, mean 4.33 vs. 1.60, median 5 vs 2), the matching of the document's content to the document's theme (Q1, mean 4.40 vs. 1.80, median 5 vs 2), and produces documents of more suitable length (Q5, mean 2.93 vs. 4.07, median 3 vs 4, with 3 being best). The proposed method also seems

[2] Available for download and browsing from `http://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=ei_cphi_m`

	Our method			Baseline			
Statement	Median	Mean	SD.	Median	Mean	SD.	p_{MWU}
Q1 (1–7, ↑)	5	4.40	1.64	2	1.80	0.41	< 0.001*
Q2 (1–7, ↑)	5	4.33	1.76	2	1.60	0.51	< 0.001*
Q3 (1–7, ↓)	4	4.47	1.81	6	5.80	1.42	0.049
Q4 (1–7, ↓)	5	5.13	1.55	6	6.33	0.62	0.024
Q5 (1–5, 3 best)	3	2.93	0.59	4	4.07	0.70	< 0.001*

Table 2: Results obtained during the evaluation. Parentheses indicate answer ranges and whether the higher (↑), lower (↓) or middle values are to be interpreted as the best. The p_{MWU} column contains the (uncorrected) p-value of a two-sided Mann-Whitney U test. An asterisk indicates the p-value is statistically significant also after applying a Bonferroni correction to account for multiple tests.

to result in less unnecessary information being included in the document (Q4, mean 5.13 vs 6.33, median 5 vs 6), and in the text missing less necessary information (Q3, mean 4.47 vs 5.80, median 4 vs 6), but these effects are not statistically significant after correcting for multiple comparisons with the Bonferroni correction. We hypothesize this difference would become significant in a larger-scale evaluation.

The free-form textual feedback provided by the judges, as expected, indicates that the texts could be further improved. For example, in the case of the text shown in Figure 1, the judges called for a sentence explicitly noting that North Macedonia had the highest monthly growth rate. In addition, they noted it might be better to produce distinct, even shorter, texts as 'news alerts' while reserving the evaluated texts for use as a starting point when the journalist starts writing.

7 Conclusions

In this work, we have identified a need for, and proposed, a widely applicable baseline document planning method for generating journalistic texts from statistical datasets. Our method is based on observations on the similarities between the orbital theory of news structure (White, 1997) and Rhetorical Structure Theory (Mann and Thompson, 1988). While our proposed method is likely to fall short of the performance of subdomain-specific planning methods, results indicate that it achieves adequate performance while fulfilling a set of requirements identified based on the larger application domain of news generation.

Acknowledgements

This work is supported by the European Union's Horizon 2020 research and innovation program under grant agreement No 825153, project EMBEDDIA (Cross-Lingual Embeddings for Less-Represented Languages in European News Media), and grant agreement No 770299, project NewsEye (A Digital Investigator for Historical Newspapers).

References

Gabor Angeli, Percy Liang, and Dan Klein. 2010. A simple domain-independent probabilistic approach to generation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 502–512.

Hadi Banaee, Mobyen Uddin Ahmed, and Amy Loutfi. 2013. Towards NLG for physiological data monitoring with body area networks. In *14th European Workshop on Natural Language Generation, Sofia, Bulgaria, August 8-9, 2013*, pages 193–197.

David Caswell and Konstantin Dörr. 2018. Automated journalism 2.0: Event-driven narratives: From simple descriptions to real stories. *Journalism practice*, 12(4):477–496.

Seniz Demir, Sandra Carberry, and Kathleen F. McCoy. 2010. A discourse-aware graph-based content-selection framework. In *Proceedings of the 6th International Natural Language Generation Conference*.

Laurence Dierickx. 2019. Why news automation fails. In *Computation+ Journalism Symposium, Miami, FL*.

Konstantin Nicholas Dörr. 2015. Mapping the field of algorithmic journalism. *Digital journalism*.

Longxu Dou, Guanghui Qin, Jinpeng Wang, Jin-Ge Yao, and Chin-Yew Lin. 2018. Data2text studio: Automated text generation from structured data. In

Proc. 2018 Conference on Empirical Methods in Natural Language Processing.

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.

Alexander Fanta. 2017. Putting Europe's robots on the map: automated journalism in news agencies. *Reuters Institute Fellowship Paper*, pages 2017–09.

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*.

Johan Galtung and Mari Holmboe Ruge. 1965. The structure of foreign news: The presentation of the Congo, Cuba and Cyprus crises in four Norwegian newspapers. *Journal of peace research*, 2(1):64–90.

Albert Gatt, Francois Portet, Ehud Reiter, Jim Hunter, Saad Mahamood, Wendy Moncur, and Somayajulu Sripada. 2009. From data to text in the neonatal intensive care unit: Using NLG technology for decision support and information management. *Ai Communications*, 22(3):153–186.

Dimitra Gkatzia. 2016. Content selection in data-to-text systems: A survey. *arXiv preprint*. Available at `https://arxiv.org/abs/1610.08375`.

Andreas Graefe. 2016. Guide to automated journalism.

Herbert P Grice. 1975. Logic and conversation. In *Speech acts*, pages 41–58. Brill.

Catalina Hallett, Richard Power, and Donia Scott. 2006. Summarisation and visualisation of e-health data repositories. In *UK E-Science All-Hands Meeting*.

Jenna Kanerva, Samuel Rönnqvist, Riina Kekki, Tapio Salakoski, and Filip Ginter. 2019. Template-free data-to-text generation of Finnish sports news. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 242–252, Turku, Finland. Linköping University Electronic Press.

Ioannis Konstas and Mirella Lapata. 2013. Inducing document plans for concept-to-text generation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1503–1514.

Leo Leppänen, Myriam Munezero, Mark Granroth-Wilding, and Hannu Toivonen. 2017a. Data-driven news generation for automated journalism. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 188–197.

Leo Leppänen, Myriam Munezero, Stefanie Sirén-Heikel, Mark Granroth-Wilding, and Hannu Toivonen. 2017b. Finding and expressing news from structured data. In *Proceedings of the 21st International Academic Mindtrek Conference*, pages 174–183. ACM.

Liunian Li and Xiaojun Wan. 2018. Point precisely: Towards ensuring the precision of data in generated texts using delayed copy mechanism. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1044–1055, Santa Fe, New Mexico, USA. ACL.

Carl-Gustav Linden. 2017. Decades of Automation in the Newsroom: Why are there still so many jobs in journalism? *Digital Journalism*, 5(2):123–140.

Veronika MacKová and Jakub Sido. 2020. The robotic reporter in the Czech News Agency: Automated journalism and augmentation in the newsroom. *Communication Today*, 11(1):36–53.

William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-Interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

Chris Mellish, Alistair Knott, Jon Oberlander, and Mick O'Donnell. 1998. Experiments using stochastic search for text planning. In *Natural Language Generation*.

Feng Nie, Jin-Ge Yao, Jinpeng Wang, Rong Pan, and Chin-Yew Lin. 2019. A simple recipe towards reducing hallucination in neural surface realisation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2673–2679.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with content selection and planning. In *Proc. 33rd AAAI Conference on Artificial Intelligence*.

Horst Pöttker. 2003. News and its communicative quality: The inverted pyramid—when and why did it appear? *Journalism Studies*, 4(4):501–511.

Ehud Reiter. 2007. An architecture for data-to-text systems. In *Proceedings of the Eleventh European Workshop on Natural Language Generation*, pages 97–104. Association for Computational Linguistics.

Ehud Reiter. 2018. Hallucination in neural NLG. `https://ehudreiter.com/2018/11/12/hallucination-in-neural-nlg/`. Accessed: 2020-03-02.

Ehud Reiter. 2019. ML is used more if it does not limit control. `https://ehudreiter.com/2019/08/15/ml-limits-control/`. Accessed: 2020-07-25.

Ehud Reiter and Robert Dale. 2000. *Building natural language generation systems*. Studies in Natural Language Processing. Cambridge University Press.

Stefanie Sirén-Heikel, Leo Leppänen, Carl-Gustav Lindén, and Asta Bäck. 2019. Unboxing news automation: Exploring imagined affordances of automation in news journalism. *Nordic Journal of Media Studies*, 1(1):47–66.

Somayajulu G Sripada, Ehud Reiter, Jim Hunter, and Jin Yu. 2003. Generating English summaries of time series data using the Gricean maxims. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 187–196.

Olli Sulopuisto. 2018. Uutisia kortti kerrallaan. *Suomen Lehdistö*. `https://suomenlehdisto.fi/uutisia-kortti-kerrallaan/`.

Elizabeth A Thomson, Peter RR White, and Philip Kitley. 2008. "Objectivity" and "hard news" reporting across cultures: Comparing the news report in English, French, Japanese and Indonesian journalism. *Journalism studies*, 9(2):212–228.

Peter White. 1997. Death, disruption and the moral order: the narrative impulse in mass-media 'hard news' reporting. *Genres and institutions: Social processes in the workplace and school*, 101:133.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proc. 2017 Conference on Empirical Methods in Natural Language Processing*.

Yleisradio. 2018. Avoin voitto. `https://github.com/Yleisradio/avoin-voitto`.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark. ACL.

Appendix C: Data Augmentation and Pretraining to Improve Neural Headline Generation in Low-Resource Setting

Data Augmentation and Pretraining to Improve Neural Headline Generation in Low-Resource Setting

Matej Martinc[1], **Syrielle Montariol**[2], **Lidia Pivovarova**[3], **Elaine Zosa**[3]
[1]Jozef Stefan Institute, [2]INRIA Paris, [3]University of Helsinki
matej.martinc@ijs.si, syrielle.montariol@inria.fr, first.last@helsinki.fi

Abstract

We tackle the problem of neural headline generation in a low-resource setting, where only limited amount of data is available to train a model. We compare the ideal high-resource scenario on English with results obtained on a smaller subset of the same data and also run experiments on two small news corpora covering low-resource languages, Croatian and Estonian. Two options for headline generation in a multilingual low-resource scenario are investigated: a pretrained multilingual encoder-decoder model and a combination of two pretrained language models, one used as an encoder and the other as a decoder, connected with a cross-attention layer that needs to be trained from scratch. The results show that the first approach outperforms the second one by a large margin. We explore several data augmentation and pretraining strategies in order to improve the performance of both models and show that while we can drastically improve the second approach using these strategies, they have little to no effect on the performance of the pretrained encoder-decoder model. Finally, we propose two new measures for evaluating the performance of the models besides the classic ROUGE scores.

Keywords: Natural language generation, Headline generation, Low resource languages

1. Introduction

Neural approaches for natural language generation (NLG) have mushroomed during past few years. The most common idea is to employ approaches that have shown good performance in machine translation (or another sequence-to-sequence task) and treat the generation task as a translation task between an input text and the generated output text (Wen et al., 2015; Cho et al., 2014). The most popular text generation is automatic summarization, and recent years have seen huge advances in automatic generation of high-quality summaries. The newest approaches, such as BART (Lewis et al., 2020), employ an encoder-decoder transformer architecture (Vaswani et al., 2017), which "translates" the input text into an output summary.

Due to large textual resources required by these NLG systems, research on this topic mostly focused on high-resource languages such as English, since the lack of data makes the training of these approaches from scratch infeasible in some low-resource domains and languages (Gkatzia, 2016). While recently some multilingual models which also cover low-resourced languages (Liu et al., 2020) have been proposed, most low-resource languages still lack efficient monolingual language generation systems. Therefore, to generate texts for these languages with a neural architecture but without large datasets and substantial computational resources—required for extensive pretraining of encoder-decoder models—we are left with two options:

Using a multilingual NLG system that supports the low-resource language in which we wish to generate text. The options are limited here, with the multilingual generation models ProphetNet-Multi (Qi et al., 2021b) and mBART-50 (Tang et al., 2020) currently being the models supporting the most languages (52 and 50 respectively, including some low-resource ones). The possible downside of using this approach is the so-called curse of multilinguality (Conneau et al., 2020), i.e., a trade-off between the number of languages the model supports and the overall decrease in performance on monolingual and cross-lingual benchmarks.

Training a multilingual encoder-decoder NLG system from scratch, with the downside being that the performance of the model will be most likely directly correlated to the amount of available training data. One possible solution to partially circumvent this problem is to employ an approach proposed by Rothe et al. (2020), which relies on the usage of two pretrained transformers, combined into an encoder-decoder NLG architecture. In this case, only the cross-attention layer needs to be trained from scratch, and since the combined model can leverage the knowledge gained during the language model pretraining, it requires less training data for optimal performance, at least in theory. An upside of this approach is that these multilingual pretrained transformer-based language models (Vaswani et al., 2017) have been recently trained for a plethora of low-resource languages[1], meaning that this approach can be used for much more languages than by using a pretrained multilingual NLG system.

While automatic summary generation is very popular, generation tasks, which focus on production of more creative content such as headlines or slogans, receive less focus. However, a headline can also be considered as a sort of summary, since it is a vehicle that carries the most important information about the news article content. The newest approaches for headline generation

[1]The Huggingface library currently offers pretrained transformers for 168 languages: `https://huggingface.co/models`

based on this idea have obtained promising results, but this research is once again mostly focused on English (Shen et al., 2016). On the down side, these approaches are difficult to employ in real-life scenarios due to a special type of overfitting called "hallucination", where the system produces non-factual outputs that are not based on the data presented in the input (Reiter, 2018; Dušek et al., 2019). This severely limits the application of these systems in the domain of newspaper articles, where the production of factual text is essential. These systems also lack interpretability and their evaluation could be unreliable unless conducted manually by humans. It has been shown that commonly used automated evaluation metrics do not necessarily correlate well with human judgement (Reiter and Belz, 2009; Dušek et al., 2018).
We tackle some of the problems and research gaps introduced above. These are our main contributions:

- We address the generation of creative texts, news headlines, in a low-resource multilingual setting with neural encoder-decoder architectures. More specifically, we compare the two distinct approaches for NLG described above. In the first approach, we use a pretrained monolingual NLG system BART (Lewis et al., 2020) or multilingual mBART (Liu et al., 2020) (depending on the language). In the second approach, we train the NLG model from scratch, relying on pretrained BERT models combined into an NLG encoder-decoder, same as in Rothe et al. (2020).
- We explore two techniques for reducing the needed amount of training data, namely data augmentation and domain-specific pretraining. We focus on evaluating how these strategies affect both types of models and conclude that they have a significant influence only in the second approach, where pretrained BERT models are combined into an NLG encoder-decoder.
- We propose two evaluation measures that have not been applied for headlines generation in the literature. Both measures focus on the semantic similarity between correct and generated headlines and therefore complement the established ROUGE score, which measures a word overlap and was criticized in the past for not considering semantic similarity.
- We offer a manual error analysis in order to determine how the proposed data augmentation and pretraining tactics affect both models and to pinpoint mistakes specific for each model.

2. Related Work

As stated above, most recent approaches to headline generation consider it as a summarization task and employ state-of-the-art neural summarization models. These models have been used to tackle several distinct variants of the headline generation task, such as bilingual headline generation (Shen et al., 2018), headlines for community question answering (Higurashi et al., 2018), multiple headline generation (Iwama and Kano, 2019) and also user-specific headline generation used in the recommendation systems (Liu et al., 2018).
Liang et al. (2020) compare multiple text noising strategies for training, showing large improvements on the headline generation task. The best noising strategy consists of sampling a number of token spans from the original text with span lengths drawn from a Poisson distribution, and then replacing each token span with a single [MASK] token.
While most research is still focused on English, recently some multilingual benchmarks for news headline generation were proposed. Among the well-known benchmarks, X-GLUE (Liang et al., 2020) includes a headline generation task, covering 5 high-resource languages (German, English, French, Spanish and Russian) and using BLEU-4 score as the metric. The training dataset contains 300K examples, and development and test datasets contain 10k examples. In this benchmark, XLM-R (Conneau et al., 2020) and M-BERT (Devlin et al., 2019), initialized as encoder-decoder models and fine-tuned on the downstream task, are outperformed by the Unicoder (Huang et al., 2019), a universal language encoder trained to be language-agnostic by being pretrained on cross-lingual tasks.
A more general benchmark for text generation is GLGE[2], including 4 abstractive text summarization tasks, CNN/DailyMail (Hermann et al., 2015) (See et al., 2017), Gigaword (Rush et al., 2015) (Graff et al., 2003), XSum (Narayan et al., 2018), and MSNews. Gigaword and MSNews both use news headlines as targets, while in the other two tasks informative summaries need to be generated. All tasks are in English, and the benchmarks are divided into three versions, from easy to hard. Prophetnet (Qi et al., 2020) and its other version ProphetNet-X (Qi et al., 2021a) beat Unicoder on this second benchmark, but are outperformed by BART (Lewis et al., 2020) on the hard version of the benchmark. ProphetNet and BART were also trained on a multilingual corpus. ProphetNet-Multi is trained on the 101GB Wiki-100 corpus and 1.5TB Common Crawl2 data. Similarly, mBART, which we employ in this study and is described in more detail in Section 3.1, is trained on 25 languages and its bigger version mBART-50 on 50 languages.

3. Methodology

3.1. The Models

For our experiments, we use two state-of-the-art summarization systems. The first system is BART (Lewis et al., 2020), a denoising autoencoder for pretraining sequence-to-sequence models[3]. BART employs a stan-

[2] https://github.com/microsoft/glge

[3] We opted to test this model instead of the alternative ProphetNet-X (Qi et al., 2021a) since it is more compara-

dard transformer-based neural machine translation architecture and is pretrained on several denoising tasks, in which the original text is corrupted and the model is trained to generate an uncorrupted output. To be more specific, the training corpus is corrupted by either randomly shuffling the original sentences or by using an in-filling scheme, where spans of text are replaced with a single mask token. BART achieved new state-of-the-art results on a set of tasks, among them classification, abstractive dialogue, question answering, and summarization. We employ BART for English, while for experiments on Estonian and Croatian we use its multilingual version mBART-50 (Tang et al., 2020).

The other approach, proposed in Rothe et al. (2020), relies on a combination of pretrained transformer-based language models. Using one language model as an encoder and the other as a decoder, the authors demonstrate the efficacy of pretrained language models for sequence generation, leading to state-of-the-art results on several tasks, among which machine translation and text summarization. We use as encoders and decoders two pretrained BERT models (Devlin et al., 2019), which are available for all languages covered in our experiments, and name this approach BERT-ED.

The main difference between the two approaches is that BART has already been pretrained as an encoder-decoder model on a large corpus consisting of books and Wikipedia (i.e. the same corpus as BERT), and mBART-50 on a large dataset containing texts from 50 languages extracted from the Common Crawl (CC) (Wenzek et al., 2020). On the other hand, BERT-ED consists of two pretrained BERT models[4] connected by a cross-attention layers, which are *randomly initialized*. We suspect this difference would result in a gap in performance between the two systems when trained on a relatively small corpora in a low-resource setting. We hypothesise that while BART will be harder to adapt for a specific headline generation task due to its extensive pretraining as an encoder-decoder, it would nevertheless return semantically and grammatically better headlines. Since the cross-attention layer in the system composed of two BERT models has not been pretrained, this approach might require more training data to generate semantically and grammatically correct headlines. It would nevertheless be easier to adapt to a specific task and domain at hand.

3.2. Training Schemes

As mentioned above, our main focus is to evaluate these systems in a low-resource setting. Most related work train neural models on large datasets consisting of more than 100,000 documents. In contrast, we test the models in a low-resource setting, on datasets ranging from 10 000 to roughly 30 000 documents, and investigate whether using and combining different pretraining schemes can improve the performance of the model. More specifically, we test three distinct pretraining techniques:

- **Text infilling**: As proposed by Lewis et al. (2020), about 20% of the training corpus is corrupted by an in-filling scheme, where spans of text are replaced with a single mask token. The encoder-decoder is then trained to generate the original text from the corrupted input.
- **Sentence shuffling**: Same as in Lewis et al. (2020), the input sentences are randomly shuffled and the model is trained to generate the original text with the correct sentence order.
- **2 tasks**: The model is first trained to restore the correct order of shuffled sentences and than to restore the corpus corrupted by the text in-filling scheme.

Note that pretraining is performed using only the headline generation training dataset and no additional data is used. This way, we inspect if the model's performance can be improved by extensive pretraining instead of obtaining more data.

3.3. Data Augmentation

To increase the size of the training corpus we employ several data augmentation techniques.

- **BERT-based augmentation**: 20% of the words in the news article are masked. Then, the masked article is fed to the BERT model, who proposes probable candidates for the masked tokens. These tokens are replaced by the most probable candidates, creating new articles to be added to the training set.
- **Word2vec augmentation**: For each news article in the train set, we replace random words in the articles by synonyms proposed by the Word2vec model.[5]
- **Wordnet augmentation**: This method is similar to the previous one, but replacement candidates are obtained from Wordnet.

ble in size to the other tested model BERT-ED (see below), making the comparison fairer.

[4]For English we used two "bert-base-uncased" models, for Estonian and Croatian we used the FinEst BERT and CroSloEngual BERT described in (Ulčar and Robnik-Šikonja, 2020), respectively.

[5]We set the number of runs parameter to 5 and probability of replacement to 0.3 (i.e., the algorithm goes through the text five times and tries to augment each sentence with a 0.3 probability). English word2vec embeddings are trained on the Google News dataset, Croatian word2vec embeddings are trained on the Croatian Web Corpus (HrWAC) (Ljubešić and Erjavec, 2011; Šnajder, 2014) while the Estonian embeddings are trained on the Estonian Reference Corpus (Kaalep et al., 2010).

- **EDA augmentation**: EDA, proposed by Wei and Zou (2019), consists of four operations: Wordnet synonym replacement, random insertion, random swap, and random deletion.
- **Mixed augmentation**: Each article in the train set is first augmented with Word2vec. The augmented article is fed to the EDA-based augmentation and the output of this augmentation is additionally fed to the Wordnet-based augmentation.

All augmentation techniques except for BERT have been previously established and are available in the TextAugment library[6]: For English, we used all augmentation strategies. For Croatian and Estonian only BERT and word2vec augmentations are available since Wordnet is not available for these languages.
For each original article in the train set, we generate 5 augmented articles using the algorithms described above. These new articles are inserted into the original training set and used for training of the headline generation model. We opted to generate five augmented texts per article, as initial experiments suggested that using a smaller number results in an insufficient increase of the training dataset, and using a larger number results in repetitions of the training examples.

3.4. Evaluation

For evaluation, we employ the ROUGE score, which is the current standard for evaluating generated summaries and headlines. However, ROUGE score does not necessarily have sufficient correlation with human judges (Reiter and Belz, 2009; Dušek et al., 2018) because it only compares n-gram overlap and therefore does not represent well the semantic similarity between true and generated headlines. To alleviate this problem, we propose two new evaluation measures that consider semantic similarity. The first measure, *semantic similarity* (SS), measures cosine distance (CD) between the embedding of the true and generated headline. We employ sentence transformers (Reimers and Gurevych, 2019) for generating embeddings for true and generated headlines.[7] The second evaluation approach is motivated by Yin et al. (2019), who used a pretrained *natural language inference* (NLI) sequence-pair classifier as a zero-shot text classifier. Considering the true headline as the "premise" and each generated headline as the "hypothesis", we use the NLI model to predict whether the premise entails the hypothesis. We take the probability of the entailment between a true and a generated headline as a measure of headline quality. Note that this measure is only used for English experiments, since there is no available model pretrained for NLI that covers Croatian and Estonian.[8]

4. Experiments

4.1. Experimental Setting

Experiments were conducted on three datasets, namely the Estonian ExM news article dataset (Purver et al., 2021a), the Croatian 24sata news article dataset (Purver et al., 2021b) and the English KPTimes dataset (Gallina et al., 2019). The dataset statistics are presented in Table 1. For Croatian and Estonian, we use the same train and test dataset splits as in the recent study on keyword extraction (Koloski et al., 2021).
The English dataset is included in our experiments to serve as a benchmark for several comparisons. First, we wish to research whether there is a discrepancy in the quality of produced headlines between English (for which most NLG models are originally produced) and two low-resource languages, Estonian and Croatian. Second, besides conducting low-resource experiments, the abundance of resources in English allows us to obtain results for the high-resource scenario, to which we can compare our low-resource results. For this reason, we use both the large KPTimes train set, containing about 260,000 news articles, and the original KPTimes validation set, containing 10,000 articles, which we employ as a 'low-resource' English train set and train models on it. Since we do not use these datasets as training and validation sets, we refer to them as 260K and 10K respectively to avoid terminology confusion.
Both BART and BERT-ED approaches are first tested in a high resource scenario, i.e., by training them on the 260K KPTimes train set. The results of these experiments are used as a reference point of how well these models work in an ideal scenario with plenty of data available, to which we can compare results of our low-resource experiments. Next, both of these models are trained on the 10K set, the Estonian train set, and the Croatian train set without any additional pretraining or data augmentation. These low-resource reference points are used as baselines that we want to improve through various pretraining and data augmentation methods.
In our experiments, we employ the same training and generation regime for both models. The input news articles are truncated at 128 tokens, since we assume that the most important content of the news, to which the title most likely refers to, is covered at the beginning of the article. The length of the output is limited to 30 tokens; finally, for generation we employ a beam search of size 5 and early stopping.

4.2. Results

The results of the experiments on the English dataset are presented in Table 2 and the results of the experi-

[6] https://github.com/dsfsi/textaugment

[7] More specifically, we employ the "sentence-transformers/paraphrase-MiniLM-L6-v2" for experiments on English and "sentence-transformers/paraphrase-xlm-r-multilingual-v1" for experiments on Croatian and Estonian. Both models are available in the Huggingface library.

[8] For English, we employ the "typeform/distilbert-base-uncased-mnli" for entailment predictions.

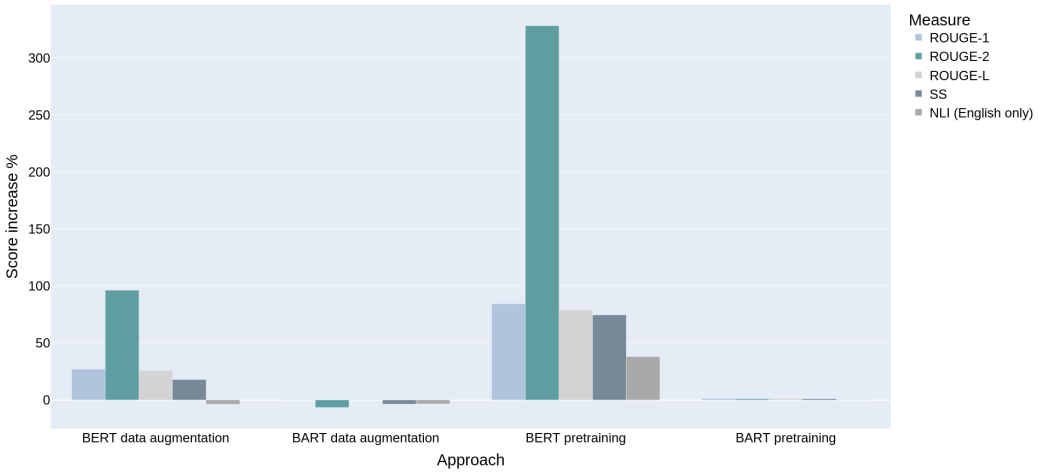

Figure 1: Average *increase* in performance for pretraining and data augmentation approaches for both models across the three languages according to five evaluation measures: three ROUGE scores, semantic similarity (SS) and NLI (only for English).

Language	train set	test set
English 260K (KPTimes train)	259,923	10,000
English 10K (KPTimes valid)	10,000	10,000
Croatian	32,223	3,582
Estonian	10,750	7,747

Table 1: News datasets used for empirical evaluation of headline generation (number of documents).

ments on Estonian and Croatian datasets are presented in Table 3.

Both BERT-ED and BART models perform well in the ideal high-resource scenario when trained on the large 260K train set (see approaches labeled as "BASELINE 260K" for both English models), with BART outperforming BERT-ED by roughly 4 points according to all three ROUGE scores, by about 2 points according to SS and by almost 5 points according to NLI.

On the other hand, when the models are compared in a low-resource scenario, the gap between the model's performance drastically increases (see approaches labeled as BASELINE for Estonian and Croatian models and the approach labeled as "BASELINE 10K" for English models). For example, the English BART model trained on the English 10K dataset outperforms BERT-ED trained on the same dataset by about 20 points according to ROUGE-1, by about 10 points according to ROUGE-2 and NLI, by about 16 points according to ROUGE-3, and by about 25 points according to SS. This is due to the drastic decrease in BERT-ED's performance when trained on the small 10K dataset. Similar phenomena can be observed for the other two languages, Croatian and Estonian, with the performance being especially bad on the Estonian corpus, where the model has trouble converging and achieves very low ROUGE scores.

While the results for BERT-ED clearly indicate that only training the model from scratch on a corpus of limited size is not a viable option, BART-based models on the other hand show more robust performance, even when trained in the low-resource scenario. For English, training the BART model on the 10K dataset results in a modest drop of about 3 points according to all criteria, when compared to the BART model trained on the 260K dataset. The results for Estonian and Croatian are worse, yet still much better than for the BERT-ED-based models. On Estonian, the multilingual mBART model achieves ROUGE-1 of 26.2, ROUGE-2 of 12.3, ROUGE-L of 24.3 and SS score of 56.7.

While comparison of ROUGE and SS scores across languages is problematic,[9] these scores—and the manual inspection confirming the quality of the produced headlines—indicate that an extensively pretrained multilingual model can be successfully applied in a low-resource scenario. The mBART results for Croatian are worse, which is interesting, since the Croatian train set is three times the size of the Estonian one. They can nevertheless be explained by the fact that mBART-50 was pretrained on a much smaller Croatian corpus than the Estonian one (Tang et al., 2020).

Next, we discuss the results of the **data augmentation** and pretraining experiments. Generally speaking, the results indicate that these experiments have on the one hand a significant influence on the performance of BERT-ED-based models and a negligible influence on the performance of the BART-based models. When it comes to English data augmentation, all but one (Word2Vec augmentation) method manage to

[9]This is especially true when comparison needs to be made between a morphologically rich language, such as Estonian, and a morphologically less diverse language, such as English.

Approach		ROUGE-1		ROUGE-2		ROUGE-L		SS		NLI	
English BERT-ED-based models											
BASELINE	10K	10.2		1.4		9.6		24.6		15.4	
	260K	27.6		10.1		25.1		49.6		32.1	
AUGMENTATION	bert	13.2	3.0	2.3	0.9	12.2	2.6	30.8	6.2	15.7	0.3
	w2v	9.7	-0.5	1.6	0.2	8.9	-0.7	26.5	1.9	14.9	-0.5
	mix	10.4	0.2	1.7	0.3	9.6	0.0	23.9	-0.7	13.1	-2.3
	eda	12.8	2.6	2.2	0.8	11.9	2.3	29.5	4.9	15.2	-0.2
	wordnet	12.4	2.2	2.1	0.7	11.5	1.9	29.3	4.7	15.2	-0.2
PRETRAINING	infilling	11.7	1.5	1.9	0.5	10.7	1.1	31.0	6.4	18.9	3.5
	shuffling	12.9	2.7	2.6	1.2	11.8	2.2	36.0	11.4	18.8	3.4
	2 tasks	16.5	6.3	4.6	3.2	15.1	5.5	42.0	17.4	25.9	10.5
English BART-based models											
BASELINE	10K	**29.0**		**10.9**		**26.0**		49.3		34.1	
	260K	31.9		13.1		28.7		51.7		36.8	
AUGMENTATION	bert	28.5	-0.5	10.5	-0.4	25.6	-0.4	49.1	-0.2	34.0	-0.1
	w2v	27.8	-1.2	10.1	-0.8	25.1	-0.9	48.2	-1.1	32.0	-2.1
	mix	27.7	-1.3	10.2	-0.7	25.0	-1.0	47.9	-1.4	32.2	-1.9
	eda	28.3	-0.7	10.4	-0.5	25.5	-0.5	49.0	-0.3	33.2	-0.9
	wordnet	28.2	-0.8	10.3	-0.6	25.3	-0.7	48.7	-0.6	33.4	-0.7
PRETRAINING	infilling	29.0	0.0	10.9	0.0	26.0	0.0	**49.5**	**0.2**	34.2	0.1
	shuffling	28.8	-0.2	10.8	-0.1	25.9	-0.1	49.4	0.1	**34.3**	**0.2**
	2 tasks	28.7	-0.3	10.7	-0.2	25.9	-0.1	49.2	-0.1	34.1	0.0

Table 2: Results of experiments on the English datasets. Best results in a low resource setting (i.e., excluding the BART and BERT-ED models trained on English 260K dataset) per evaluation measure are **bolded**. For each measure, we report its absolute value (the first number) and the difference with the baseline model (the second, colored, number). Since all experiments with data augmentation and pretraining are run on the 10K dataset, differences are computed respectively to the 10K baseline, i.e. the first row of results for each model.

beat the BERT-ED 10K baseline score. The biggest improvement can be observed for the BERT augmentation. Decent improvements according to all criteria can also be observed when EDA and Wordnet augmentation are used. Mix augmentation does not work that well, probably because texts become very different after the multi-step process and not always preserve the original meaning. It is hard to fine-tune augmentation parameters, since this would require retraining of the corresponding headline generation model.

For Croatian, the data augmentation improvements are smaller than for English; BERT data augmentation does not work at all. As the Croatian training dataset is three times bigger than the English and Estonian ones, we deduce that increasing the dataset size with data augmentation techniques might be less beneficial for larger datasets. The highest improvements over the BERT-ED baseline for data augmentation are observed for the Estonian dataset. Indeed, the BERT-ED baseline—which most likely did not converge due to the lack of training data—returns mostly repetitive or empty strings, while data augmentations apparently creates enough additional training data to generate more coherent content.

For the BART-based models, all data augmentation strategies lead to scores lower than the baseline for all languages. While the reduction is in most cases minimal, these scores nevertheless do indicate that the augmented data is not of sufficient quality for the pretrained model to obtain useful information that can be successfully leveraged during NLG training.

By **pretraining** the BERT-ED-based models, using text infilling and sentence shuffling tasks, on the same datasets on which they are later fine-tuned for headline generation, we obtain substantial performance boosts. The increase in performance is even larger than with data augmentation. For English and Estonian, it is especially useful to apply both pretraining regimes, sentence shuffling and text infilling, sequentially (see the row in Tables 2 and 3 labeled as "PRETRAINING 2 tasks"). For Croatian, text infilling works slightly better than sentence shuffling according to most criteria, but combining these two approaches does not improve the performance.

Pretraining the BART-based models leads to small improvements for Estonian and Croatian, and to small reduction for English. The monolingual English BART, which was extensively pretrained on a massive English corpus using the same denoising tasks we employ here, apparently does not profit from the additional pretraining on a small corpus. The pretraining experiments for the multilingual mBART-50 on the other hand consistently show small improvements across all three pretraining regimes and for both languages.

The average increase in performance for data augmentation and pretraining across all languages and for both

Approach		ROUGE-1		ROUGE-2		ROUGE-L		SS	
Croatian BERT-ED-based models									
BASELINE		9.6		1.0		8.9		29.7	
AUGMENTATION	bert	2.5	-7.1	0.0	-1.0	2.5	-6.4	10.2	-19.5
	w2v	11.0	1.4	1.4	0.4	0.1	1.1	33.7	4.0
PRETRAINING	infilling	16.6	7.0	4.2	3.2	14.8	5.9	44.9	15.2
	shuffling	15.2	5.6	3.6	2.6	13.4	4.5	43.9	14.2
	2 tasks	15.4	5.8	4.2	3.2	13.6	4.7	45.9	16.2
Croatian BART-based models									
BASELINE		20.5		7.3		18.1		49.6	
AUGMENTATION	bert	19.8	-0.7	6.8	-0.5	17.6	-0.5	49.8	0.2
	w2v	18.3	-2.2	5.8	-1.5	16.3	-1.8	47.9	-1.7
PRETRAINING	infilling	21.0	0.5	**7.5**	**0.2**	18.6	0.5	**51.1**	**1.5**
	shuffling	**21.2**	**0.7**	7.4	0.1	**18.7**	**0.6**	50.8	1.2
	2 tasks	20.8	0.3	7.2	-0.1	18.4	0.3	50.9	1.3
Estonian BERT-ED-based models									
BASELINE		3.9		0.3		3.8		17.9	
AUGMENTATION	bert	9.8	5.9	2.5	2.2	9.4	5.6	36.9	19.0
	w2v	8.5	4.6	2.1	1.8	8.1	4.3	34.4	16.5
PRETRAINING	infilling	13.9	0.1	4.3	4.0	13.2	9.4	44.0	26.1
	shuffling	11.3	7.4	2.8	2.5	10.7	6.9	40.7	22.8
	2 tasks	17.6	13.7	6.5	6.2	16.3	12.5	49.8	31.9
Estonian BART-based models									
BASELINE		26.2		12.3		24.4		56.7	
AUGMENTATION	bert	25.4	-0.8	11.6	-0.7	23.8	-0.6	55.9	-0.8
	w2v	23.0	-3.2	9.8	-2.5	21.5	-2.9	53.5	-3.2
PRETRAINING	infilling	**27.1**	**0.9**	**12.9**	**0.6**	**25.2**	**0.8**	**57.2**	**0.5**
	shuffling	26.6	0.4	12.6	0.3	24.8	0.4	56.9	0.2
	2 tasks	26.6	0.4	12.3	0.0	24.6	0.2	56.6	-0.1

Table 3: Results of experiments on the Croatian and Estonian datasets. Best results per language and per evaluation measure are **bolded**. For each measure, we report its absolute value (the first number) and the difference with the baseline model (the second, colored, number). The differences are computed in respect to the baseline.

models is visualized in Figure 1. It is visible that the employment of data augmentation or pretraining leads to on average much larger increase in performance when BERT-ED-based models are used. The measure that benefits the most from these additional steps is ROUGE-2, most likely since this is the hardest criterion of the model's quality, which is only slightly above zero for most baseline BERT-ED-based approaches. On the other hand, the figure clearly shows that both pretraining and data augmentation have only a marginal effect on the BART-based models.

5. Qualitative results

We manually checked the outputs of several English models. The BART model, fine-tuned on the 10K dataset produces one of the the best results. However, it can hallucinate (see Example 2 in Table 4) or shift the focus of the headline. The manual inspection did not reveal any large differences between the BART-based model trained on the 10K dataset and on the 260K dataset. Interestingly, Example 1 results in identical outputs for BART models trained on both datasets, as well as in *all* other modifications we try with BART. Variation between outputs are rare and, in most cases, not significant; thus, it is hard to judge which outputted headline is better. On the contrary, the performance of the BERT-ED-based model trained on the 10K dataset drops drammatically compared to the one trained on the 260K dataset, as could be seen in the same table. In most cases, it produces ungrammatical sequences with many repetitions.

Data augmentation only slightly improves the performance on English. According to numerical results in Table 3, the best augmentation method is BERT-based augmentation. However, as could be seen in Table 4, the outputs are still ungrammatical, though the meaning is closer to the true headlines. Similar results were obtained with other augmentation strategies.

In our experiments, pretraining has a more positive effect, though repetitions and hallucinations are still possible, as can be seen in the last row in Table 4. Pretraining results in much longer output sequences, where in most of the cases only the first 5-6 words make sense, and then the model starts making repetitions as if it did

Table 4: Examples of English headlines generated by various models.

	EXAMPLE 1	EXAMPLE 2
True headline	martial law is rescinded in a philippine province	fighting n. y. c. soda ban, industry focuses on personal choice
BART 260K	philippine president lifts martial law	soda industry fights new york city's soda ban
BART 10K	philippine president lifts martial law	soft-drink industry takes aim at sugary drinks
BERT-ED 260K	philippine president lifts martial law in southern philippines	soft - drink industry seeks to fight sugary drinks ban on sugary drinks
BERT-ED 10K	obama's court's ban in court	in new yorks's york taxs's taxs s
BERT-ED 10K + BERT aug	philippines's ban in philippines' ban in philippines' ban in philippines	in new york city, new york city's new york city's bans law
BERT-ED 10K + shuffling	philippines : lawmakers seek lawmaker's ban on lawmakers obama lawmakers arroyo's lawmakers arroyo's lawmakers	u. s. and new york's new new york city mayor' campaign campaign moves new york's mayor's campaign campaign
BERT-ED 10k + infilling	new new new new york city party party leader s. o. p. s. a. leader s. o. p.	philippines : s. o. p. to be suspended s. a. lawmakers s. ban s. a.'s
BERT-ED 10K + 2 tasks	president's decision to rebuke military law ends in conflict philippines arroyo's rebuke philippines's supreme court in	new yorkers face a challenge to soda industry in new yorkers in new yorkers' campaign campaign in new york city's

not know where to stop.

All BERT-ED-based models overuse possessive suffixes in an ungrammatical way. Text infilling strategy also results in overusing of abbreviations, though this problem disappears in a "2-task" pretraining (the last two rows in Table 4).

6. Conclusion

We investigated two systems for headline generation in a multilingual low-resource scenario. The first option is the employment of a pretrained multilingual encoder-decoder summarization model and the second one is combining two pretrained language models into an encoder-decoder architecture that is trained from scratch. We suggest that if the first option is available, i.e., there exists a pretrained multilingual NLG model for a specific low-resource language, it should be picked over the second one. The successful training of a randomly initialized cross-attention layer, connecting the two language models, is crucial for the model's performance and is dependent on a large corpus, such as the KPTimes train dataset containing round 260 K document. However, even in that scenario, the BERT-ED model is outperformed by an English BART model. We have shown that while pretraining and data augmentation can drastically improve the performance of the BERT-ED models, it has little effect on the BART-based models, which have already been extensively pretrained on the same denoising tasks, text infilling and sentence shuffling, that we employ in our experiments. The experiments also suggest that pretraining on the train set is a better option than data augmentation since the improvements are larger and since data augmentation had a negative effect on the performance of the BART-based models, most likely due to the insufficient quality of the data augmentation algorithms.

The best performance is achieved by the BART model trained on a large KPTimes train set. While this indicates that currently there is still no substitution for a large dataset, the BART model trained on the magnitudes smaller 10K dataset nevertheless still offers competitive performance. Scores that mBART achieves on the Estonian and Croatian datasets are lower, which could be caused by the fact that these languages are morphologically much richer languages than English. It might however also indicate that multilingual models cannot compete with the monolingual one, confirming the curse of multilinguality (Conneau et al., 2020).

On top of ROUGE-1, -2 and -L, we use two metrics to evaluate the quality of the generated headlines that are less broadly used in the literature, measuring semantic similarity (SS) and sentence entailment (NLI). They are globally highly correlated with ROUGE scores, but allow for more fine-grained comparison when evaluating the impact of different augmentation and pretraining regimes.

The main focus of the future work will be on improving the quality of generated headlines in low-resource scenarios, by (1) introducing novel pretraining tasks and data augmentation techniques and by (2) pretraining monolingual encoder-decoder models on denoising tasks on as large corpora as can be obtained for low-resource languages. We will expand our evaluation setting, by introducing both novel measures and manual evaluation. Finally, we will consider several techniques for the generation of headlines with specific style.

7. Acknowledgements

This work has been supported by the European Union's Horizon 2020 research and innovation programme under grant 825153 (EMBEDDIA).

References

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online, July. Association for Computational Linguistics.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Dušek, O., Novikova, J., and Rieser, V. (2018). Findings of the E2E NLG challenge. *arXiv preprint arXiv:1810.01170*.

Dušek, O., Howcroft, D. M., and Rieser, V. (2019). Semantic noise matters for neural natural language generation. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 421–426.

Gallina, Y., Boudin, F., and Daille, B. (2019). KPTimes: A large-scale dataset for keyphrase generation on news documents. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 130–135, Tokyo, Japan, October–November. Association for Computational Linguistics.

Gkatzia, D. (2016). Content selection in data-to-text systems: A survey. *arXiv preprint 1610.08375*.

Graff, D., Kong, J., Chen, K., and Maeda, K. (2003). English gigaword. *Linguistic Data Consortium, Philadelphia*, 4(1):34.

Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. (2015). Teaching machines to read and comprehend. In *NIPS*, pages 1693–1701.

Higurashi, T., Kobayashi, H., Masuyama, T., and Murao, K. (2018). Extractive headline generation based on learning to rank for community question answering. In *COLING*, pages 1742–1753.

Huang, H., Liang, Y., Duan, N., Gong, M., Shou, L., Jiang, D., and Zhou, M. (2019). Unicoder: A universal language encoder by pre-training with multiple cross-lingual tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2485–2494, Hong Kong, China, November. Association for Computational Linguistics.

Iwama, K. and Kano, Y. (2019). Multiple news headlines generation using page metadata. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 101–105, Tokyo, Japan, October–November. Association for Computational Linguistics.

Kaalep, H.-J., Muischnek, K., Uiboaed, K., and Veskis, K. (2010). The estonian reference corpus: Its composition and morphology-aware user interface. In *Baltic HLT*, pages 143–146.

Koloski, B., Pollak, S., Škrlj, B., and Martinc, M. (2021). Keyword extraction datasets for Croatian, Estonian, Latvian and Russian 1.0.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (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, July. Association for Computational Linguistics.

Liang, Y., Duan, N., Gong, Y., Wu, N., Guo, F., Qi, W., Gong, M., Shou, L., Jiang, D., Cao, G., et al. (2020). Xglue: A new benchmark datasetfor cross-lingual pre-training, understanding and generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6008–6018.

Liu, T., Li, H., Zhu, J., Zhang, J., and Zong, C. (2018). Review headline generation with user embedding. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 324–334. Springer.

Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., Lewis, M., and Zettlemoyer, L. (2020). Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Ljubešić, N. and Erjavec, T. (2011). hrWaC and slWaC: Compiling web corpora for Croatian and Slovene. In *International Conference on Text, Speech and Dialogue*, pages 395–402. Springer.

Narayan, S., Cohen, S. B., and Lapata, M. (2018). Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *EMNLP*, pages 1797–1807.

Purver, M., Pollak, S., Freienthal, L., Kuulmets, H.-A., Krustok, I., and Shekhar, R. (2021a). Ekspress news article archive (in estonian and russian) 1.0.

Purver, M., Shekhar, R., Pranjić, M., Pollak, S., and Martinc, M. (2021b). 24sata news article archive 1.0.

Qi, W., Yan, Y., Gong, Y., Liu, D., Duan, N., Chen,

J., Zhang, R., and Zhou, M. (2020). Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 2401–2410.

Qi, W., Gong, Y., Yan, Y., Xu, C., Yao, B., Zhou, B., Cheng, B., Jiang, D., Chen, J., Zhang, R., et al. (2021a). Prophetnet-x: Large-scale pre-training models for english, chinese, multi-lingual, dialog, and code generation. *arXiv preprint arXiv:2104.08006*.

Qi, W., Gong, Y., Yan, Y., Xu, C., Yao, B., Zhou, B., Cheng, B., Jiang, D., Chen, J., Zhang, R., Li, H., and Duan, N. (2021b). ProphetNet-X: Large-scale pre-training models for English, Chinese, multi-lingual, dialog, and code generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 232–239, Online, August. Association for Computational Linguistics.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, November. Association for Computational Linguistics.

Reiter, E. and Belz, A. (2009). An investigation into the validity of some metrics for automatically evaluating natural language generation systems. *Computational Linguistics*, 35(4):529–558.

Reiter, E. (2018). Hallucination in neural NLG. `https://ehudreiter.com/2018/11/12/hallucination-in-neural-nlg/`. Accessed: 2020-03-02.

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

Rush, A. M., Chopra, S., and Weston, J. (2015). A neural attention model for abstractive sentence summarization. In *EMNLP*, pages 379–389.

See, A., Liu, P. J., and Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Shen, S., Zhao, Y., Liu, Z., Sun, M., et al. (2016). Neural headline generation with sentence-wise optimization. *arXiv preprint arXiv:1604.01904*.

Shen, S.-q., Chen, Y., Yang, C., Liu, Z.-y., Sun, M.-s., et al. (2018). Zero-shot cross-lingual neural headline generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(12):2319–2327.

Šnajder, J. (2014). DerivBase.hr: A high-coverage derivational morphology resource for Croatian. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3371–3377, Reykjavik, Iceland, May. European Language Resources Association (ELRA).

Tang, Y., Tran, C., Li, X., Chen, P.-J., Goyal, N., Chaudhary, V., Gu, J., and Fan, A. (2020). Multilingual translation with extensible multilingual pretraining and finetuning. *arXiv preprint arXiv:2008.00401*.

Ulčar, M. and Robnik-Šikonja, M. (2020). Finest bert and crosloengual bert: less is more in multilingual models. *arXiv preprint arXiv:2006.07890*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Wei, J. and Zou, K. (2019). EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China, November. Association for Computational Linguistics.

Wen, T.-H., Gašić, M., Mrkšić, N., Su, P.-H., Vandyke, D., and Young, S. (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, September. Association for Computational Linguistics.

Wenzek, G., Lachaux, M.-A., Conneau, A., Chaudhary, V., Guzmán, F., Joulin, A., and Grave, E. (2020). Ccnet: Extracting high quality monolingual datasets from web crawl data. In *12th Conference on Language Resources and Evaluation (LREC 2020)*, page 4003–4012.

Yin, W., Hay, J., and Roth, D. (2019). Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3914–3923, Hong Kong, China, November. Association for Computational Linguistics.