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: 36 months

D5.3: Initial dynamic news generation technology (T5.2)

Executive summary

In this deliverable, we describe two approaches for dynamically deciding the order in which information is presented to the reader in automatically generated news texts. Both are intended to be employed as part of a larger text generation pipeline, described in deliverables D5.2 and D2.4. The first, based on an ensemble of heuristics, consumes a selection of available information that *can* be included in the news in an abstract, non-linguistic, format and generates a document plan of what information to actually express and using which structures. The second is a work-in-progress machine learning method for observing individual sentences and deciding the order in which they should be presented in the news text. Importantly for our context, this method only uses textual training data in the form of human-written (news) texts. Both methods are dynamic in the sense that they adapt the structure of the story to new information much more freely than traditional methods employed in real-world news automation systems, which are often based on extremely rigid document structures.

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:	**D5.3**
Deliverable full title:	**Initial dynamic news generation technology**
Deliverable short title:	**Initial dynamic news generation technology**
Document identifier:	**EMBEDDIA-D53-InitialDynamicNewsGenerationTechnology-T52-submitted**
Lead partner short name:	**UH**
Report version:	**submitted**
Report submission date:	**30/06/2020**
Dissemination level:	**PU**
Nature:	**R = Report**
Lead author(s):	**Leo Leppänen (UH)**
Co-author(s):	**Eliel Soisalon-Soininen (UH), Hannu Toivonen (UH)**
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
10/05/2020	v0.1	L. Leppänen (UH)	First draft (Sections 1-4, 6-7).
20/05/2020	v0.2	L. Leppänen (UH)	Second draft (Sections 1-4, 6-7).
22/05/2020	v0.3	E. Soisalon-Soininen (UH)	First draft of Section 5.
22/05/2020	v0.4	H. Toivonen (UH)	WP leader's comments.
22/05/2020	v0.5	L. Leppänen (UH)	Modifications to address WP leader's comments.
26/05/2020	v0.6	E. Soisalon-Soininen (UH)	Modifications to address WP leader's comments.
26/05/2020	v1.0	L. Leppänen (UH)	Submitted for internal review.
12/06/2020	v1.1	M. Purver (QMUL), S. Luz (UEDIN)	Internal review.
15/06/2020	v1.2	L. Leppänen (UH), E. Soisalon-Soininen (UH)	Modifications to address internal review.
15/06/2020	v2.0	L. Leppänen (UH)	Ready for quality control.
17/06/2020	v2.1	N. Lavrač (JSI)	Quality control.
24/06/2020	v2.2	L. Leppänen (UH)	Modifications to address QC feedback.
26/06/2020	final	L. Leppänen (UH)	Ready for submission.
30/06/2020	submitted	Tina Anžič (JSI)	Report submitted.

Table of Contents

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 Work Package 5 (WP5), which is concerned with *Natural Language Generation* (NLG). Natural language generation is a "subfield of artificial intelligence and computational linguistics that is concerned with the construction of computer systems that can produce understandable text in English or other human languages from some underlying non-linguistic representation of information" (Reiter & Dale, 1997; Gatt & Krahmer, 2018). More specifically, the focus of WP5 is on *automated journalism*, which concerns the automated generation of *news* texts (Dörr, 2015; Caswell & Dörr, 2018).

In order to support journalists and media companies in efficiently reaching as many demographics as possible, the objective of WP5 is to design and develop news automation systems that take data as input and produce a textual report in response; the methods designed in EMBEDDIA aim to be transferable across languages, transferable across domains, and transparent in their NLG process.

Following the requirements above, WP5 will develop a flexible, accurate, and transparent NLG system architecture that can be transferred to new domains and languages with minimal human effort; develop tools for creation of dynamically evolving content, incorporating narrative structure and user knowledge; and develop tools for creation of figurative language and headlines. The work package consists of three tasks (where this deliverable reports on work done in Task T5.2):

- Task T5.1, Multilingual text generation from structured data, adapts NLG technology for the requirements of news generation. The task develops 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, develops 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, makes the generated texts more varied and colourful by generating creative expressions, especially in headlines. We find similar terms and metaphors by finding analogous terms in different contexts using context-dependent embeddings. A special focus will be on cross-cultural metaphors.

In this deliverable, we report on the development relating to task T5.2 within the first 18 months of the project. An important factor in our work is the dynamic nature of the content generation. This means that rather than having a human predefine what a story looks like for some specific domain, we seek to dynamically determine an optimal structure based on the specific input data.

For more information on natural language generation in general, refer to Deliverable D2.4. For more information on application of NLG to news production, refer to Deliverable D5.2. Both deliverables are due concurrently with the present text.

We start by summarizing both non-technical research into the focus of this deliverable, the way news stories are structured in general (Section 2), as well as how documents are structured in NLG literature. Following this background, we briefly describe the context of our research, the EMBEDDIA news generation technology (Section 3). Next, we describe the work conducted within T5.2, namely an ensemble approach of structuring text dynamically in news generation (Section 4) and an machine learning approach that assumes only the existence of a textual corpus of news, rather than the existence of an aligned data-and-text corpus (Section 5). Section 6 provides an overview of our plans for evaluating the work, and section 8 provides our final conclusions and our plans for future work.

The approach described in Section 4 has been jointly developed with, and is also being used by, the NewsEye H2020 project, where it's used to structure texts describing analyses of historical newspapers.

2 Structure in news and automated journalism

The structure of news text – that is, the order in which the information is presented to the reader – varies significantly both between and within news domains. 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 journalists belief about the importance of the piece of information (Thomson, White, & Kitley, 2008).

According to Pöttker (2003), the entrenchment of this structure of news text goes back to late 1800s or early 1900s. Several reasons have been proposed in the literature for transitioning to such a structure of news, with common ones including a technological explanation based on the need to ensure that the most important part of the message was sent first over the unreliable telegraph to ensure that it – at the least – reached the editor, but the precise reasons for this transition are not clear to us. Irrespective of its origins, the structure has become so prototypical that it is held self-evident in the journalistic trade literature: "Every journalist knows how one writes 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 original Finnish).

At the same time, this method for structuring text is clearly not universal to all domains within news. For example in sports, while the story might start with a highlight of the game, the rest might be structured in a temporal order. In other words, the rest of the article might essentially describe the game as a story, with added analysis either interspersed or appended to the end of the text.

An interesting analysis of the structures employed in 'hard' news is presented by Thomson et al. (2008). The authors argue that the news article can be seen as consisting of a *nucleus* which represents the main point of the article and *satellites* that give context and additional information about the nucleus. In their analysis, Thomson et al. assign the role of the nucleus to the combination of the headline and the lead of the article, and describe the subsequent paragraphs as the nuclei. This interpretation matches well with the aforementioned pyramid model of news, as well as cultural artefacts such as the idiom '*bury the lede*,' meaning to hide the most important information (typically presented in the first sentences, known as 'lede' or 'lead') further in the article.

Thomson et al. (2008) analyze and identify several roles the satellites can have in relation to the nucleus. In their analysis, they identify that the satellites can *elaborate*, *reiterate*, describe *causes* or *consequences*, *contextualize* or provide *additional assessment*. An important observation on their part is that – as indicated by the term 'orbital' – these satellites are not necessarily required to be in any specific order with respect to the nucleus and are in fact often relatively freely reorderable without affecting the readability of the article.

It is notable that the relations identified by Thomson et al. (2008) are highly similar to those identified in the more general Rhetorical Structure Theory (RST) (Mann & Thompson, 1988). Rhetorical Structure Theory is a linguistic framework for describing text in general and goes into a more fine grained analysis than the theory of Thomson et al. (2008). Whereas Thomson et al. analyze newstext on the level of paragraphs of text, RST can be – and often is – applied to even individual phrases within sentences. As such, RST analyses are usually presented in tree-like diagrams similar to parse trees. In natural language generation, RST has taken a de-facto role as the main conceptual framework used to describe the structures of the documents, although recent works on neural models often ignore such theory.

At the same time, despite this standardization of the terminology and the analytical framework, the actual *methods* for structuring document contents in natural language generation have not – to our knowledge – resulted in any widely accepted algorithms that could be applied widely. This is very understandable given the domain specificity of the RST relations. The knowledge that a piece of information can, for example, be used to *justify* (a specific type of relation in RST) another requires significant semantic knowledge about the specific pieces of information and how they relate.

Similarly, even selecting the 'most important' piece of information to act as the nucleus (as the term is

used by Thomson et al.) requires seemingly domain-dependent analysis to rank the available pieces of information according to a domain-specific criteria for 'importance'.

It is perhaps for this reason that, to our knowledge, commercial approaches to automated journalism have largely been dependent on what we collectively call *story level templates*, where the structure of the story is hard-coded (albeit with often some conditional logic) together with the linguistics expressions used. On a broad level, this approach to document structuring is somewhat similar to a journalistic *Choose Your Own Adventure* book.[1] At the same time, we must acknowledge that our understanding of the commercially used news automation systems is limited to private discussions, often light on technical detail, with industry insiders and the few open source systems such as the open source ice hockey news generation system from Yleisradio (2018).

As most of the industry players are – understandably – keeping the details of their systems secret to guard any competitive advantage, it is possible that at least some industrial players are employing more complex approaches than that described above. At the same time, interviews with media insiders indicate that the systems employed are relatively 'classical,' (i.e. rule-based) (Sirén-Heikel, Leppänen, Lindén, & Bäck, 2019), which provides a contrast to where the academic state-of-the-art is heading.

Many academic works of late have focused on learning to structure and select content from data using machine learning methods but as a separate process from the actual text generation. Examples of these kinds of 'two-level' approaches are presented, for example, by Li and Wan (2018); Zhang, Zhong, Chen, Angeli, and Manning (2017); Puduppully, Dong, and Lapata (2019); Dou, Qin, Wang, Yao, and Lin (2018); Wiseman, Shieber, and Rush (2017). A prototypical example is provided by Puduppully et al. (2019), who describe a neural NLG system that is trained end-to-end (i.e. as a single system), but contains a separate sub-network for deciding what information should be expressed in the text. It is curious how the academic thinking has shifted from rule-based modular architectures to the opposite end of the spectrum, the end-to-end neural model, and seems now to be swinging back to more modular approaches, albeit again using neural models.

The dependence of even these modular neural methods on training data, however, severely limits their applicability to use in real-world newsrooms. While newsrooms have extensive archives of news text, these are almost never associated with the matching data that is 'behind' each piece of news text. Even the various statistical agencies around Europe provide little in the way of aligned corpora of text and data. For example, while Statistics Finland, the Finnish goverment agency in charge of producing the official national statistics, provides both structured data and natural language texts describing the data, the data provided is not historical. That is, while it is principle possible to align a text to a table (or multiple tables) of data, the state of the table at the time of writing the text is not available. In other words, the corpus would be limited to a historical text and the present state of the associated data tables.

The goal of EMBEDDIA Work Package 5 is to produce news automation technology that can be applied to many domains and situations. As such, our goal is to identify general methods for document structuring that can be similarly applied to many domains and situations. This means looking outside of both 'story level templates' (that can be hardly described as 'dynamic') and the massively data-driven approaches frequented in academia, which would not be *de-facto* applicable outside of very limited domains, such as sports and weather, where massive aligned datasets are available. The latter is especially troublesome, as it limits news automation from being applied to the myriad of potential news domains that have been traditionally too expensive to cover using human means, essentially creating a type of large-scale bootstrapping problem.

To better understand how the document structuring fits into the larger news automation architecture de-

[1] Also known by the generic term '*gamebook*,' in a Choose Your Own Adventure book the reader is presented with a section of text. After reading the section, they are presented with a choice of jumping to one of two or more continuing sections based on how they wish the storiy continued, thus presenting a branching narrative. The analogue to simple automated journalism stems from the way automated text can be structured as a sequence of decisions along the lines of 'if the home team won, start paragraph A. If the away team won, start with paragraph B. Otherwise start with paragraph C.'

veloped in EMBEDDIA task T2.3 and its implementations developed in Task T5.1, we next describe – in brief terms – the architecture and its implementation, especially noting how the document structuring fits into this larger context. Then, in the subsequent sections, we will describe our approach to conducting document structuring for news automation in a relatively general fashion.

3 The EMBEDDIA news generation technology

The work described in this deliverable must be interpreted as a part of the larger EMBEDDIA (news) text generation architecture that is being developed in both tasks T2.3 (see Deliverable 2.4 - "Multilingual language generation approach") and T5.1 (see Deliverable D5.2 - "Initial news generation technology"). As these deliverables together describe both the architecture at large (D2.4) and its two concrete implementations in the news domain (D5.2), we only include a large-scale overview of the architecture here. Please see the aforementioned deliverables for more details on the architecture at large and its specific implementation for news generation.

The EMBEDDIA news generation approach is based on a pipeline of components with dedicated responsibilities. This structure allows for the individual components to be modified and replaced without affecting the rest of the pipeline. As the domain and language specific aspects of the pipeline are largely segregated (with the part specific to both domain and languages further delegated to specific subcomponents), the system at large can be transferred to new domains and languages much more easily than applications based on a non-modular approach to NLG. At the same time, it is not dependent on large amounts of aligned data-and-text training data, which would preclude it from being used outside of the very few domains where such data is readily available, especially in the case of smaller newsrooms and languages.

For this work, the relevant parts of the architecture (see Figure 1) is the Document Planning component. This component receives as input a list of Message data structures, which in turn contain data structures called Facts. These are produced from the input data in the preceding steps of the pipeline and represent the whole total of all things the news text *could* contain. The data held in the Facts is to at least some degree domain-dependent, but for example in the case of the COVID-19 case study (see Deliverable D5.2) it contains the fields described in Table 1.

The output of the document planning procedure is a tree-structure, detailing the overall structure of the document: leaves correspond to Facts and the branches from the root node correspond to paragraphs.

Table 1: The fields of a Fact data structure in the COVID-19 case study (see D5.2). The hypothetical Fact states that between the first and the second of May, the the number of confirmed COVID-19 cases in Finland increased by 15.

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	2020-05-01:2020-05-02
`timestamp_type`	The type of the timestamp	date_span
`value`	A (usually) numeric value	15
`value_type`	A descriptor defining how the numeric value should be intepreted	Latest:Confirmed:DailyChange:Abs
`newsworthiness`	A non-negative number providing numeric estimate of the newsworthiness of the fact. Higher values indicate more newsworthy	1

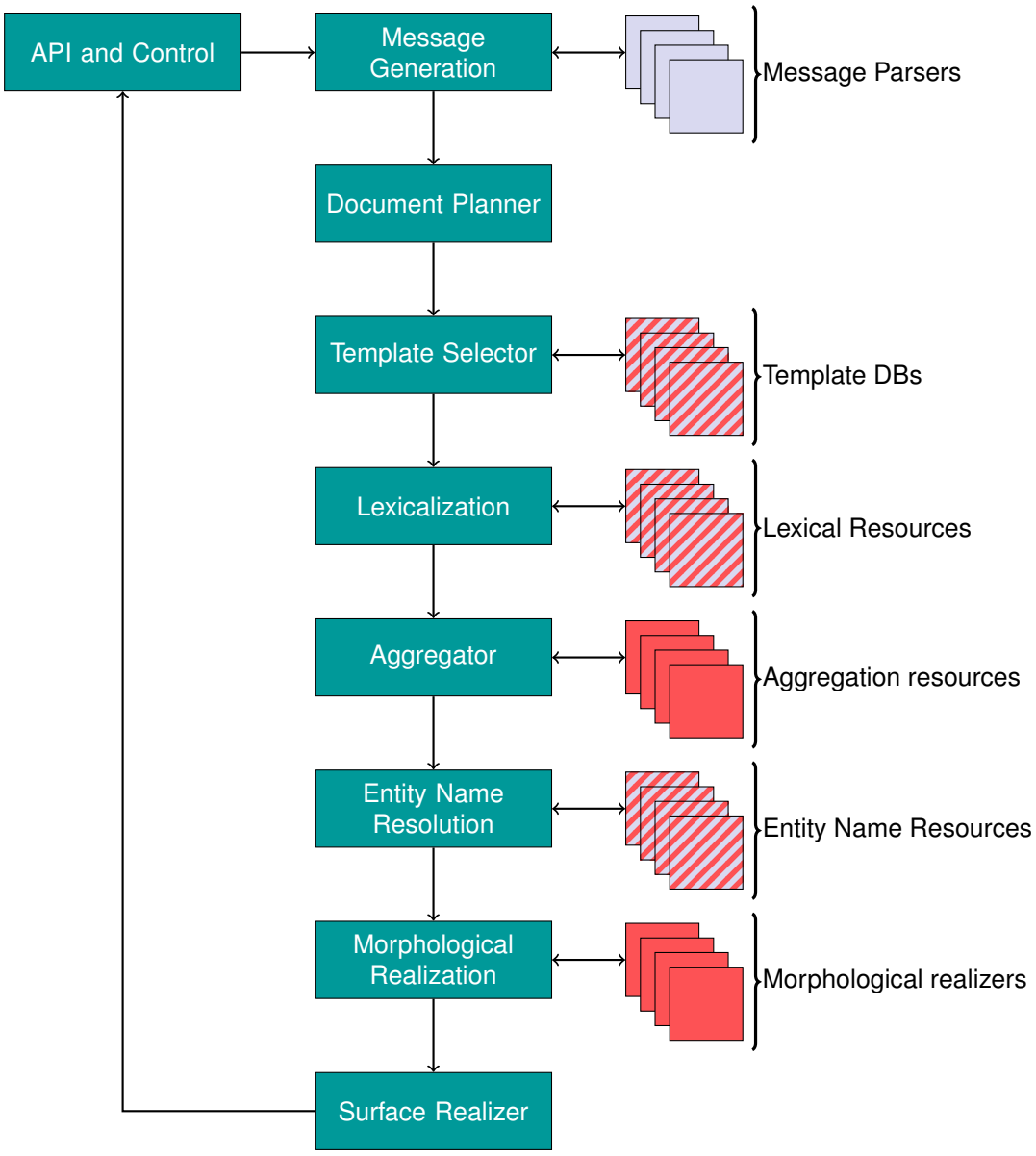

Figure 1: High-level architecture of the EMBEDDIA NLG technology. The overlapping boxes in the right-hand column indicate resources that vary based on either the language or the language and the domain. In this column only, the coloring of the boxes indicates whether the resource in question is dependent on the generation language (e.g. Morphological Realizers), the generation domain (e.g. Message Parsers), or both (hatched boxes).

This structure represents not only the *structure* in which the information is to be presented in the text, but various limits on the total length of the text also force the component to leave a significant bulk of the available Messages and Facts out of the plan.

It is also notable that further stages in the pipeline are allowed to further refine this document plan, for example by moving Messages to allow for more natural expressions and by condensing multiple Facts into a single Message. These considerations, however, are beyond the scope of this Deliverable.

4 Document planning for news generation

As noted above, the pertinent part of the natural language generation pipeline described in Section 3 – insofar as the present Deliverable is considered – is the Document Structuring component. In this Section, we describe a largely domain-independent ensemble method for structuring news content.

As described in Section 2, our task is to find a method that achieves three primary goals. First, the approach should dynamically structure news content in a largely domain-independent fashion without a human having to pre-define the structure of the story. Second, the method should construct paragraphs so that the first paragraph of the text is a *lede*, containing the most newsworthy aspect of the story. Third, further paragraphs should then be constructed so as to be thematically coherent within each paragraph and also to provide information that is related to the first paragraph. Our ensemble method, described below, seeks to fulfill these goals using a combination of various heuristics. These heuristics are to a large degree independent of any specific news domain, and as such the approach can dynamically adapt to changes in the input data.

The presented method is also dynamic in the sense that it does not depend on human-produced guidelines such as "start by discussing (sub)topic T_1, unless condition C_1 is true, in which case start with (sub)topic T_2", as is – to our knowledge – very common in non-academic approaches to NLG.

As noted in Section 3, each Message in the input provided to the document structuring component includes a numerical estimate of how newsworthy it is. Hereafter, we only assume that said values are non-negative and higher values indicate higher newsworthiness. The details of the actual computation conducted to arrive at this numeric estimate are not relevant for document structuring, but include for example a statistical estimate of the outlierness the piece of information, which acts as a proxy for how surprising the information is. Additional details are provided in Deliverable D5.2.

The messages, obtained as input to the document structuring component, describe what *could* be described in the generated news text before accounting for any practical limitations, such as the length of the resulting document. The goal of document structuring is to produce a tree-structure, where the root node corresponds to the document as a whole and 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. The mid-level structures of the produced tree thus correspond – in the abstract – to structures such as (longer) sentences and most importantly paragraphs.

A naïve approach to the document structuring process would be to greedily select messages in a descending order of newsworthiness until some length-based heuristic decides that each paragraph (and eventually the document) is of suitable length. Such an approach, however, would result in a document without any internal coherence, except perhaps by pure luck. As such, additional heuristics are needed to enforce a level of coherence into the document.

To enforce coherence, we adapt the terms '*nucleus*' and '*satellite'* from White (2005) and Thomson et al. (2008). However, whereas White and Thomson et al. use the terms in reference to *paragraphs* of the story, we observe that such structure is needed within the paragraphs as well. As such, we model each paragraph as consisting of a nucleus message (the main point of the paragraph) and associated satellite messages that provide additional information about the nucleus. This approach effectively

Algorithm 1 Pseudocode describing generation of the document plan from messages.

```
function GENERATEDOCUMENTPLAN(Messages)
    Root ← newDocumentPlanNode
    SelectedNuclei ← []
    while True do
        if reached maximum length or Messages = ∅ then
            return Root
        end if
        Node ← newDocumentPlanNode
        Nucleus ← SELECTNUCLEUS(Messages, SelectedNuclei)
        if Nucleus = null or Nucleus is not sufficiently newsworthy for inclusion then
            return Root
        end if
        Satellites ← SELECTSATELLITES(Nucleus, Messages)
        Node.children = [Nucleus]
        Node.children.extend(Satellites)
        SelectedNuclei.insert(Nucleus)
        Root.children.insert(Node)
    end while
end function
```

bridges between the observations of Thomson et al. and the rhetorical structure theory of Mann and Thompson (1988) by interpreting the recursive structures of the latter in terms of the former.

The procedure that creates the document plan is described as pseudocode in Algorithm 1. The generation progresses paragraph-by-paragraph, first selecting a suitable nucleus based on the previously selected nuclei, and then selecting suitable satellites for that nucleus to fill in the paragraph.

The system bases its decisions on three factors: the newsworthiness scores of the messages, thematic similarity and contextual similarity. Thematic similarity describes how similar the topics of two arbitrary messages are. This aspect captures, for example, that – all other things equal – it is intuitively more reasonable to follow a fact about a nation's spending on health care with other messages related to health care spending, than with a message about some unrelated topic such as football. Contextual similarity describes how similar the context of two arbitrary facts are. It captures the intuitive notion that it is more reasonable to follow a fact about Finland's spending on preventive medicine in 2020 with another piece of information about Finland in 2020, rather than about Austria in 1990. As noted above, the goal of combining these three factors is to produce a text that contains as newsworthy messages as possible, while also enforcing a level of coherence into the document.

In the case of the first paragraph, the `SelectNucleus` procedure simply selects the most newsworthy fact in the input as the *nucleus* of the first paragraph. This is a special case, as the two latter factors discussed above are not relevant: the first message *sets* the context and theme for the following content. Later nuclei are selected using a more complex process, described later-on.

To this nucleus, additional supporting facts or *satellites*, are added, as determined by the `SelectSatellites` procedure, described in pseudocode in Algorithm 2. These satellites are selected from among all available, so far unused, messages one-by-one. After each selection, the available, so far unused, satellites' newsworthiness scores are recalculated to reflect the satellites' similarity to both the previously selected satellite and the nucleus of the paragraph. That is, when selecting the first satellite, the similarity is measured against the nucleus only, whereas for the third satellite, the similarity is measured against the second satellite and the nucleus.

The similarity between two messages is a determined in the `ScoreBySimilarity` procedure as a combination of similarity of *context* and similarity of *theme*, as discussed above.

In the `ScoreBySimilarity` procedure, the message m's similarity to both the nucleus of the paragraph

Algorithm 2 Pseudocode describing how satellites are selected for a paragraph

```
function SELECTSATELLITES(Nucleus, Messages)
    SelectedSatellites ← []
    prev ← Nucleus
    while True do
        if maximum satellite count reached then
            return SelectedMessages
        end if
        for all m ∈ Messages do
            m.score ← SCOREBYSIMILARITY(m, prev, Nucleus)
        end for
        FilteredMessages ← FILTERBYNEWSWORTHINESS(Messages)
        if FilteredMessages = ∅ then
            if minimum satellite count reached then
                return SelectedSatellites
            else if Messages ≠ ∅ then
                FilteredMessages ← Messages
            else
                return SelectedSatellites
            end if
        end if
        NewSatellite ← arg max_{m∈FilteredMessages} m.score
        SelectedSatellites.append(NewSatellite)
        Messages.remove(NewSatellite)
        prev ← NewSatellite
    end while
end function
```

and the previously selected satellite are determined. The score of m is then set to m's newsworthiness value, weighted by the similarity values of m to both the nucleus and the previously selected satellite. The intuition behind this approach is to maximize the newsworthiness of the paragraph's contents, while also enforcing a certain level of coherence in the text. By continuously measuring against the previously selected satellite, the procedure allows for some *thematic drift* within the paragraph. That is, the theme of the paragraph can evolve over time. At the same time, the inclusion of the similarity measure against the nucleus also ensures that the theme does not drift *excessively* far from the original theme of the paragraph.

As mentioned above, the `ScoreBySimilarity` process considers two distinct types of similarity: *contextual* similarity and *thematic* similarity.

Two messages are considered to be more similar in *context* if they share the values of their underlying facts' fields related to the location and timestamp. For every field for which the two messages' field values are the same, a similarity value (initially 1) is multiplied by a weight. These weights are set per-field, which in turn enables the system to consider certain types of similarities to be more important than others for the purposes of document structuring. As such, the weights are a set of tuneable hyperparameters, and we expect to modify them based on experiments and feedback.

Two messages are considered to be similar in *theme* based on the `value_type` field. We assume here that the fields contain colon-separated hierarchies of labels describing how the `value` field is to be interpreted. For example, the field value `health:cost:hc2:mio_eur` would indicate that the number in the `value` field is the amount of money, measured in millions of euros, spent by some nation on rehabilitative care in some time period. Denoting the previous example as F_1, we can consider two other examples F_2 and F_3, where F_2 is `health:cost:hc2:eur_hab`, the cost of rehabilitative care as euros per inhabitant and F_3 is `health:cost:hc41:mio_eur`, the cost of health care related imaging services in millions of euros.

Intuitively, F_1 and F_2 are thematically closer than F_1 and F_3. We model this observation into a measure of similarity between two facts A and B as

$$sim(A, B) = \frac{2p(A, B)}{\ell(A) + \ell(B)} \tag{1}$$

where $\ell(A)$ is the length – in colon-separated units – of A's `value_type` field. That is, $\ell(F_1) = 4$. Similarly, $p(A, B)$ is the length – in colon-separated units – of the shared prefix between A and B's `value_type` fields. For example, $p(F_1, F_2) = 3$ whereas $p(F_1, F_3) = 2$. Applying the formula to various pairs of `value_type` fields, we observe the behavior shown in Table 2, which matches our intuition of the degree of similarity. Observe, for example, how sim(a:b:c, a:b:x) $<$ sim(a:b:c, a:b), which matches the intuition that, when changing topics, it is better to start the new topic with more general observations than highly specific ones, as a way of introducing the new topic.

Table 2: Examples of the behavior of the thematic similarity metric.

A	B	$\ell(A)$	$\ell(B)$	$p(A, B)$	$sim(A, B)$
a:b:c	a:b:c	3	3	3	1
a:b:c	a:b	3	2	2	0.8
a:b:c	a	3	1	1	0.5
g:e:f	a	3	1	0	0
g:e:f	a:b:c	3	3	0	0
a:b:c	a:b:x	3	3	2	$0.6\bar{6}$
a:b:c	a:x:y	3	3	1	$0.3\bar{3}$
a:b	a:x	2	2	1	0.5
a:b	a:x:y	2	3	1	0.4
a	x	1	1	0	0

Presented in terms of a string distance, the above metric is the fraction of the shared prefix out of the total length of the inputs – measured in the semicolon separated segments – but it can also be thought of as a measure of path similarity in a trie (prefix tree) of the segments of the `value_type` fields.

The above formulation of the similarity metric forces a similarity of zero for all pairs wherein the pairs have no shared prefix. This is somewhat undesirable, in that we would prefer to retain some concept of 'less completely different': it should be more acceptable in a complete topic transition to start with a more general message about the new topic than a highly specific one. In other words, we would expect that sim(g:e:f, a) $>$ sim(g:e:f, a:b:c). This can be achieved without modifying the calculation itself by prepending each label with a shared null prefix $\varnothing$. Described in terms of a trie, this is the same as adding a shared root node which all the otherwise separate tries are children of. The behavior of the similarity metric with this added null prefix is shown in Table 3.

As noted in Algorithm 2, the scores are recalculated after every new satellite has been selected to account for the effect of that satellites inclusion on the similarities. As such, the satellites can be thought of as forming a priority queue where the act of taking the first item in the queue always results in a recalculation of the priorities of the remaining elements. Satellites are added in this manner until the paragraph is considered full (by virtue of reaching a configurable maximum length) or the system runs out of 'sufficiently newsworthy' facts that pertain to the theme of the paragraph, as determined by the procedure `FilterByNewsworthiness`. We skip the details of the procedure, but note that 'sufficiently' newsworthy is defined in terms of both an absolute newsworthiness threshold as well as a relative threshold as a fraction of the nucleus' newsworthiness. Both aspects are controlled by tuneable hyperparameters.

The procedure also accounts for a tuneable minimal satellite count. If there are no more sufficiently newsworthy messages, but the minimal threshold has not been reached, the threshold of 'sufficiently' is relaxed so that even very unnewsworthy messages can be used if available, until the minimal paragraph

Table 3: Examples of the behavior of the thematic similarity metric with an added shared 'null' prefix ∅.

A	B	$\ell(A)$	$\ell(B)$	$p(A,B)$	$sim(A,B)$
∅:a:b:c	∅:a:b:c	4	4	4	1
∅:a:b:c	∅:a:b	4	3	3	0.85
∅:a:b:c	∅:a	4	2	2	$0.6\bar{6}$
∅:g:e:f	∅:a	4	2	1	$0.3\bar{3}$
∅:g:e:f	∅:a:b:c	4	4	1	0.25
∅:a:b:c	∅:a:b:x	4	4	3	0.75
∅:a:b:c	∅:a:x:y	4	4	2	0.5
∅:a:b	∅:a:x	3	3	2	$0.3\bar{3}$
∅:a:b	∅:a:x:y	3	4	2	0.57
∅:a	∅:x	2	2	1	0.5

Algorithm 3 Pseudocode describing how the next nucleus is selected.

```
function SELECTNUCLEUS(SelectedNuclei, Messages)
    if SelectedNuclei = ∅ then
        return arg max_{m∈Messages} m.fact.newsworthiness
    end if
    DiscussedThemes ← [PREFIX(n.value_type) | n ∈ SelectedNuclei]
    FilteredMessages ← [m ∈ Messages | PREFIX(m.value_type) ∉ DiscussedThemes]
    if FilteredMessages = ∅ then
        if |DiscussedThemes| > 1 then
            return null
        else
            FilteredMessages ← Messages
        end if
    end if
    NewNucleus ← arg max_{m∈FilteredMessages} m.fact.newsworthiness
    Messages.remove(NewNucleus)
    return NewNucleus
end function
```

length is reached. The minimal length can be ignored only if the messages completely run out during the generation process.

After the satellites have been selected, a new document plan node is constructed out of them and the nucleus, which is then added to the overall document plan. After this, the nucleus of the next paragraph is selected using the `NextNucleus` procedure. This procedure is described as pseudocode in Algorithm 3.

In terms of building the document, an important goal is to also maximize the overall coverage of the available, newsworthy data described to the user across the paragraphs: if the user of the system requests a news article about topics A, B and C, the text should reflect all of those to at least some degree. For this reason, whereas with the satellites we sought to maximize the semantic similarity between the satellites, the reverse holds for the nuclei. In other words, we want the different paragraphs to discuss as different things as possible.

To this end, Algorithm 3 seeks to enforce the requirement that each paragraph's nucleus must be a message about a so-far undiscussed theme by requiring that the selected nuclei do not share a prefix. This, however, causes a problem when **all** the results available for discussion about a single theme, i.e. they all start with a shared prefix. For this reason, we specifically allow that a previously discussed

analysis can be the nucleus of a new paragraph in the case where no other options are available.

This formulation allows us to naturally produce both texts discussing a wide variety of different factors present in the input data tables, as well as focused texts about a single data table. The behavior is driven by the system input: an input consisting of multiple themes naturally results in an overview-style text, whereas an input consisting of data about a single theme only results in an in-depth text.

A possible future improvement on this would be to base the length of prefix observed by `Prefix(·)` in Algorithm 3, so that in cases where all messages share the same first segment of their `value_type`, the system would automatically adjust so that the resulting text then contains a variety of *subthemes*. In other words, it would be desirable for `Prefix(·)` to dynamically determine the length of prefix to observe on the available Messages at the start of the content determination process.

We note that a thematic flow restriction, like the one used in satellite selection, could also be implemented in the nucleus selection procedure if the flow of the paragraphs turns out to be prohibitively incoherent. However, based on the observations of White (2005) regarding the exchangeable order of the paragraphs in news reports, we do not expect this to be necessary.

The planning then continues by selecting satellites for this nuclei, etc., until either a predefined maximum length, measured in paragraphs, is reached or there are no more sufficiently newsworthy nuclei to select. The term 'sufficiently' is defined as above with satellites.

As noted above, the output of the document planning procedure is a tree-structure, detailing the overall structure of the document: leaves correspond to Facts and the branches from the root node correspond to paragraphs. As described in Section 3, further stages then attach to these facts and messages phrase level templates and eventually realize them into natural language text.

An important caveat of the above approach to document planning is that it is still, fundamentally, greedy. At each stage of the planning process the algorithm selects the locally most suitable Message as the continuation of the story. While it would be more preferable to find a *globally* optimal document plan, an important factor in this case is the relatively high computational complexity of the document planning process, especially if it is to be used in a real time setting. To identify the globally optimal document plan containing k messages, the system would need to construct all document plans of size k (which number the k-permutations of n, where n is the total number of messages, i.e. $\frac{n!}{(n-k)!}$) and score each in linear time. The scoring would then also need to account for various k in an acceptable range. In other words, the number of document plans that needs to be evaluated to determine the globally optimal plan is too large to construct in a meaningful time.

While producing and evaluating *all* possible document plans to find a globally optimal selection seems, at this stage, unrealistic in all but the most limited domains, we suspect a beam-search approach could be incorporated into the process provided that a suitable method for normalizing for document plan length is determined. This normalization is required as simply maximizing the total newsworthiness would always result in maximally long documents. At the same time, many simple approaches such as maximizing the *mean* newsworthiness would results in minimally short documents.

5 Learning news structure from corpora

As already mentioned above, a significant problem for many neural approaches' practical usefulness is the lack of aligned training data from the journalistic process: only in very limited situations are newsrooms able to actually point out pairs of 'this dataset led to this article'. This means that machine learning based approaches that assume the existence of such data (e.g. Puduppully et al., 2019) are not suitable for use outside of a limited set of domains. At the same time, the newsrooms have extensive archives of the process outputs, i.e. the texts generated by the journalists. As such, we are highly interested in developing machine learning based methods for document structuring that are trained using textual corpora alone.

A potential solution is presented by learning a neural model that observes as input two distinct sentences, S_1 and S_2, and outputs which of the two sentences are more likely to come first in a news document. As such decisions are – to at least some degree – driven by the news values of the organization that produced the text, the model would then presumably also learn something about said news values. This method could then be expanded to predict which of the sentences is more likely to follow given a context C, where the context could either be the completely preceding text, the preceding sentence, or either the preceding or sentence-first nucleus.

We have recently begun to experiment with such neural models, using the Statistics Finland Text Corpus as training data. This dataset consists of statistical news articles in Finnish, Swedish, and English. In order to learn the relationship between pairs of sentences, we sample sentence pairs (S_1, S_2) from within news articles such that the associated label is $y = 1$ when S_1 precedes S_2 and $y = 0$ otherwise. Having trained a model on such pairs, its predictions on unseen sentence pairs should then be close to 1, when S_1 clearly should precede S_2, and close to 0, when S_2 should precede S_1. Then we can use the model to make comparisons between sentences, and sort a set of sentences into a likely coherent order using a suitable algorithm, e.g. beam search. This approach is similar to that of Chen, Qiu, and Huang (2016) and Agrawal, Chandrasekaran, Batra, Parikh, and Bansal (2016).

In our preliminary experiments, we train standard neural models such as multilayer perceptron (MLP) as well as convolutional (CNN) and recurrent neural networks (RNN). In order to feed sentences from raw text to neural models, we encode the sentences into vector or matrix representations, depending on the type of model. For example, for the MLP, a sentence is represented with one vector, while for the CNN and RNN, the representation is a matrix of concatenated token vectors. To obtain these vector representations, we use embeddings from the popular language models ELMo (Peters et al., 2018) and BERT (Devlin, Chang, Lee, & Toutanova, 2019), pre-trained on Finnish, Swedish, and English.

These models operate on inputs of single vectors or matrices. In order to feed sentence pairs (S_1, S_2) as input, a model is first applied separately to S_1 and S_2. This produces a vector representation for each, which we then merge into one vector e.g. by computing their absolute difference or with a bilinear transformation. This merged vector is then fed into one more hidden layer with dropout, whereafter the output value is produced with the sigmoid activation function.

Our preliminary results suggest that pairwise relationships between sentences can be learned from news corpora. For example, even simple MLP networks, using a bilinear transformation to merge sentence vectors initialised with ELMo, have achieved accuracies in the range of 55–65%, which is better than the 50% given by random choice for binary classification. These scores were obtained on a dataset of randomly sampled sentence pairs of which 50% were in the correct order (i.e. S_1 was before S_2 in the news article), and the remaining 50% were not. We anticipate that more complex models will yield better performance. However, at the time of writing, this line of work is still in its very early stages, and will continue with more extensive and conclusive experiments.

6 Evaluation method

Neither of the methods described above has yet been evaluated, but our initial qualitative analysis of the performance of the method described in Section 4 is promising. Figure 2 shows example output produced by the COVID-19 case study system described in Deliverable D5.2, which in our view has very good flow and coherence considering the fact that no predefined structures were imposed on the content plan beyond those described above.

As specified in Deliverable D5.1 – 'Datasets, benchmarks and evaluation metrics for multilingual text generation' – the evaluation of NLG systems and components is non-trivial. For content selection and document planning, the evaluation methods used in the literature are – to our knowledge – limited to human evaluations (where document planning is evaluated in the context of a complete NLG system) as well as evaluations based on aligned corpora of input data and text (E.g. Puduppully et al., 2019). When such aligned corpora are available, it is possible to either evaluate the system as a whole –

6826 total COVID-19 cases in Finland by yesterday

In Finland, there have been 6826 confirmed cases by yesterday. The number of confirmed cases increased by 0.7 percentage and 50 cases between the day before yesterday and yesterday. The number of confirmed cases increased by 3.9 percentage and 258 cases between the day before yesterday last week and yesterday.

There have been 316 confirmed deaths by yesterday. The number of deaths increased by 0.6 percentage and 2 cases between the day before yesterday and yesterday. The number of deaths increased by 3.3 percentage and 10 between the day before yesterday last week and yesterday.

Figure 2: Example of output from the COVID-19 case study system. The structure of the text is determined by the method described in Section 4. The initial newsworthiness values are obtained by outlierness analysis as described in Deliverable D5.2, and are weighed so that more recent events are more newsworthy. Within a specific time frame, no preference is otherwise given to any statistic over other, i.e. deaths are not held to be intrinsically more newsworthy than recoveries.

comparing the outputs of the system to the gold outputs – or alternatively evaluate the document planning in isolation by using information extraction tools to extract 'gold standard' document plans from the corpus.

Of these two approaches based on aligned corpora, the first suffers from the frustrating status of various automated evaluation metrics. For example, increasing evidence indicates the BLEU metric (Papineni, Roukos, Ward, & Zhu, 2002) that has long been a *de facto* standard is not suitable for scientific hypothesis testing in NLG research (Reiter, 2018). It has also been lately criticised in machine translation research (Mathur, Baldwin, & Cohn, 2020). While alternative metrics (see Deliverable D5.1) exist, automated evaluation metrics in general have been criticized as 'uninterpretable' and '[uncorrelated] with human judgements' (van der Lee, Gatt, van Miltenburg, Wubben, & Krahmer, 2019). It is our interpretation of the present status of NLG evaluation that automated metrics alone can not be used to evaluate scientific hypotheses, but rather that human testing must always be conducted.

The second corpus-based approach – based on extracting gold-standard document plans with information extraction methods – is based on the assumptions that the gold standard texts contain the full span of acceptable document plans, and that the noise introduced by the information extraction method is acceptably low. However, hard news present significant freedom in how the paragraphs of the story can be ordered (Thomson et al., 2008), which makes it very unlikely that any corpus would contain all acceptable document plans. These considerations, however, are all somewhat moot as the domains in which our works are conducted are such that no aligned corpora of data and output texts are available, making the use of automated evaluation metrics fundamentally untenable.

For the reasons described above, we intend to evaluate the developed methods by conducting a series of intrinsic human evaluations. In these tests, texts produced by a larger NLG system incorporating various combinations of the aforementioned algorithms – as well as a naïve baseline approach – will be evaluated by humans for subjective qualities such as text pleasantness and coherence. The scores for the various approaches (and combinations of approaches) can then be inspected for statistical differences, which can in turn be attributed to the document structuring approaches provided that the system variants are otherwise constant throughout the evaluation. A statistically significant improvement in the human judgements, when compared to a baseline approach, would indicate that the proposed methods are successful. While we intend to conduct these trial in multiple languages, we foresee difficulties in finding significant amounts of evaluators in smaller languages such as Finnish, Estonian etc. In such a case, we will conduct large-scale online human evaluations where possible and attempt to corroborate the findings with small-scale qualitative analyses conducted by native speakers – e.g. journalist employed by the project media partners – of the smaller languages.

7 Associated outputs

The algorithm described in Section 2 has been implemented in the two EMBEDDIA new generation systems described in Deliverable D5.2 ('EuroStat News Generation Technology' and 'COVID-19 News Generation Technology'). The source code repositories of these resources are at the present not public as they are undergoing live development while the work on Tasks T5.1 and T5.2 continues. We also intend to write a scientific publication on the work described herein. The source code repositories will be made public with a suitable license in the future after publication.

Description	URL	Availability
EuroStat news generation system (source code)	`https://github.com/EMBEDDIA/eurostat-nlg`	To become public
COVID-19 news generation system (source code)	`https://github.com/EMBEDDIA/covid-nlg`	To become public

8 Conclusions and further work

Task T5.2 has developed two methods for structuring text in the news context. The first, a method based on an ensemble of heuristics, forms a strong rule-based and largely domain-agnostic baseline. The second method – based on neural networks – is in its early stages. In the future, we will improve upon both methods and carry out more extensive experiments. It will be interesting to evaluate whether the best performance is obtained by one over the other, or whether the methods complement each other, resulting in the strongest performance when both are employed together. We will also continue to look for further possibilities in document structuring given the restrictions imposed by the news generation domain, i.e. the lack of aligned data-and-text corpora and extreme requirements for correctness. Most specifically, we are interested in how the cross-lingual and contextual word embeddings developed in Task T1.3 can be integrated further into the processes investigated herein.

The approach described in Section 2 has already been integrated into the two news generation systems' initial versions described in Deliverable D5.2. As such, it is also integrated into the WP6 media assistant. We also intend to use the algorithm described in Section 2 in WP3 in the context of the NLG system producing reports from online news comments. The approach decsribed in Section 5 is not yet integrated into any NLG system, but we intend to employ it in the D5.2 systems in the future as the approach matures. At the same time, it's applicability to the generation task in WP3 is limited by lack of available domain-specific training data.

As noted above, none of the aforementioned methods for document structuring have been evaluated as of yet. In the future, we will run intrinsic human evaluations of the developed methods in multiple languages, as the methods mature to a suitable point. The results of these evaluations will be described in a future deliverable.

We are also interested in identifying a method to augment the document structuring process so that it accounts for repetition and the effects of time by, essentially, extending the Gricean maxim of quantity across multiple news stories. In other words, a reader who consumes multiple news stories should not be told the same information in all of them. At the same time, the reader cannot be assumed to have a perfect memory, remembering everything they have previous read. On this point, we intend to draw inspiration from research into human learning and recall, most specifically the research investigating the degree to which humans forget pieces of information as a function of time (see Murre & Dros, 2015).

References

Agrawal, H., Chandrasekaran, A., Batra, D., Parikh, D., & Bansal, M. (2016). Sort story: Sorting jumbled images and captions into stories. *arXiv preprint arXiv:1606.07493*.

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

Chen, X., Qiu, X., & Huang, X. (2016). Neural sentence ordering. *arXiv preprint arXiv:1607.06952*.

Devlin, J., Chang, M.-W., Lee, K., & 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)* (pp. 4171–4186).

Dörr, K. N. (2015). Mapping the field of algorithmic journalism. *Digital journalism*.

Dou, L., Qin, G., Wang, J., Yao, J.-G., & Lin, C.-Y. (2018). Data2text studio: Automated text generation from structured data. In *Proc. 2018 conference on empirical methods in natural language processing.*

Gatt, A., & Krahmer, E. (2018). Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, *61*, 65–170.

Li, L., & Wan, X. (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* (pp. 1044–1055). Santa Fe, New Mexico, USA: ACL.

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

Mathur, N., Baldwin, T., & Cohn, T. (2020). Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. *arXiv preprint arXiv:2006.06264*.

Murre, J. M., & Dros, J. (2015). Replication and analysis of ebbinghaus' forgetting curve. *PloS one*, *10*(7).

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 of the association for computational linguistics* (pp. 311–318).

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 conference of the North American chapter of the Association for computational linguistics: Human language technologies, volume 1 (long papers)* (pp. 2227–2237).

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

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

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

Reiter, E., & Dale, R. (1997). Building applied natural language generation systems. *Natural Language Engineering*, *3*(1), 57–87.

Sirén-Heikel, S., Leppänen, L., Lindén, C.-G., & Bäck, A. (2019). Unboxing news automation: Exploring imagined affordances of automation in news journalism. *Nordic Journal of Media Studies*, *1*(1), 47–66.

Sulopuisto, O. (2018, March). Uutisia kortti kerrallaan. *Suomen Lehdistö*. (Available online `https://suomenlehdisto.fi/uutisia-kortti-kerrallaan/`)

Thomson, E. A., White, P. R., & Kitley, P. (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.

van der Lee, C., Gatt, A., van Miltenburg, E., Wubben, S., & Krahmer, E. (2019, October–November). Best practices for the human evaluation of automatically generated text. In *Proceedings of the 12th international conference on natural language generation* (pp. 355–368). Tokyo, Japan: Association for Computational Linguistics. Retrieved from `https://www.aclweb.org/anthology/W19-8643` doi: 10.18653/v1/W19-8643

White, P. (2005). Narrative impulse in mass-media 'hard news' reporting. *Genre and institutions: Social processes in the workplace and school*, 101–123.

Wiseman, S., Shieber, S., & Rush, A. (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`. GitHub.

Zhang, Y., Zhong, V., Chen, D., Angeli, G., & Manning, C. D. (2017). Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 conference on empirical methods in natural language processing* (pp. 35–45). Copenhagen, Denmark: ACL. doi: 10.18653/v1/D17-1004