Generating Natural Language Texts

6

Hermann Bense, Ulrich Schade, and Michael Dembach

Key Statements

1. Natural Language Generation (NLG) allows the generation of natural language texts to present data from a structured form.
2. NLG is a founding technology for a new sector in the publishing industry.
3. Natural language texts generated automatically are of medium quality with respect to cohesion and coherence. Their quality can be improved, however, by applying methods from the cognitive process of language production, e.g. the mechanisms determining sequencing of phrases which exploit information structure in general and the semantic roles of the phrases, in particular.
4. The number of automatically generated news articles will exceed those written by human editors in the near future since NLG allows hyper-personalization, the next key factors for reader contentment.

6.1 Introduction

The publishing industry has a rapidly increasing demand for unique and highly up-to-date news articles. There are huge amounts of data continuously being produced in the domains of weather, finance, sports, events, traffic, and products. However, there are not enough

H. Bense (✉)
TextOmatic AG, Dortmund, Germany
e-mail: hermann.bense@textomatic.ag

U. Schade · M. Dembach
Fraunhofer-Institut FKIE, Wachtberg, Germany
e-mail: ulrich.schade@fkie.fraunhofer.de; michael.dembach@fkie.fraunhofer.de

T. Hoppe et al. (eds.), *Semantic Applications*,
https://doi.org/10.1007/978-3-662-55433-3_6

human editors to write all the stories buried in these data. As a result, the automation of text writing is required. In Bense and Schade [2], we presented a Natural Language Generation (NLG) approach which is used for automatically generating texts. In these texts, data that is available in structured form such as tables and charts are expressed. Typical examples are reports about facts from the mentioned domains, such as weather reports.

Currently, texts generated automatically are correct, but of medium quality and sometimes monotonous. In order to improve the quality, it is necessary to recognise how semantics in general and information structure in particular are implemented in good texts. In order to illustrate this we need to compare the NLG approach to the cognitive process of language production. Therefore, Sect. 6.3 sketches the cognitive process. In Sect. 6.4, we will discuss what kinds of texts can be successfully generated automatically. Some technical aspects, especially those by which background knowledge can be exploited for the generation, will be presented in Sect. 6.5. In Sect. 6.6, we discuss methods we are currently developing to further enhance the quality of generated texts in terms of cohesion and coherence. These methods exploit insight from the cognitive process of language production as well as the linguistic theory of "topological fields". We summarise our results in Sect. 6.7 and give an outlook to hyper-personalization of news, the next trend in NLG.

6.2 The Cognitive Process of Language Production

In 1989, Prof. Dr. Willem J.M. Levelt, founding director of the Max Planck Institute for Psycholinguistics in Nijmegen, published "Speaking: From Intention to Articulation" [11]. This influential monograph merged the knowledge from multiple insights on the cognitive process of language production into one consistent model. Levelt's model is based on models by Karl Bühler [4], Victoria Fromkin [6], Merrill Garrett [7], and J. Kathrin Bock [3], and includes insights on monitoring and error repair elaborated by Levelt himself [10]. It incorporates important advancements about the structure of the mental lexicon [9] and the process of grammatical encoding [8] by Gerard Kempen, as well as equally important advancements about the process of phonological encoding by Gary S. Dell [5]. To this day, Levelt's model provides the base for research on language processing. Levelt, together with his co-workers (among them Antje S. Meyer, Levelt's successor as director at the MPI in Nijmegen, Ardi Roelofs, and Herbert Schriefers) contributed to that research by examining the subprocess of lexical access, see for example Schriefers, Meyer and Levelt [15] and Levelt, Roelofs and Meyer [12].

To compare NLG approaches to the cognitive process of language production, it is of specific importance to take a look at Levelt's breakdown of language production into subprocesses, a classification that is still widely accepted in the field. Levelt distinguishes preverbal conceptualization, divided into macro planning and micro planning, from the linguistic (and thus language dependent) processes of formulation, divided into grammatical encoding and phonological encoding, and articulation, the motoric subprocesses of speaking (and writing). Speaking (and of course also writing) is triggered by an intention. The speaker acts by speaking in order to inform the listener about something, to manipulate the listener to do something, or to convince the listener that he or she will do something. Conceptualization in general, and macro planning in particular, starts with that intention. Considering the intention, the process of macro planning determines the content of the next part of an utterance, i.e., the content of the next sentence. To do so, macro planning exploits different kinds of knowledge at the disposal of the speaker. This knowledge includes encyclopaedic knowledge, e.g., that Robert Lewandowski is a Polish star striker (encyclopaedic knowledge about soccer) and discourse knowledge, e.g., what had already been mentioned, who is the listener, what is the background of the dialog and more. Micro planning takes the determined contents and compresses it into a propositional structure which is coined the “preverbal message” by Levelt. According to Levelt, the preverbal message still is independent from language.

The transformation of the preverbal message into the target language is the task of formulation, the second major subprocess of language production. First, for each concept which is part of the preverbal message, a lexical entry is determined. For example, the concept of a share may trigger lexical entries like “share” or “stock”. A competition process then decides whether “share” or “stock” will be used in the resulting expression. The selected entries will be expanded into corresponding phrases, e.g. “the share”, in parallel. To achieve this, a procedure inspects the preverbal message in order to determine the specific forms of those phrases. For example, in a noun phrase, a decision has to be made as to whether a determiner is needed and if so, whether the determiner has to be definite or indefinite, whether the noun is singular or plural, and whether additional information has to be incorporated, e.g. in the form of adjectives. In some cases, the noun phrase can even be expressed in the form of a single personal pronoun. Starting with the first concept for which the corresponding phrase is completed, formulation’s subprocess of grammatical encoding starts to construct a sentence in which all the phrases are integrated. Of course the concept that represents the action in the message is transformed not into a phrase but into the sentence’s verb group. In order to execute the process of grammatical encoding, speakers use all their knowledge about the target language, their vocabulary and their grammatical expertise.

The result of grammatical encoding can be seen as a phrase structure tree, with words as terminals. The representations of these words (lemmata) become the subject of the formulation’s second subprocess, which in case of speaking is coined phonological encoding. This process transforms the words into their sequence of phonemes (or letters in the case of writing). Phonological encoding taps into the speaker’s knowledge about how to pronounce (or spell) a word. Finally, the articulation process takes over and generates overt speech (or written text).

6.3 Automated Text Generation in Use

The main areas for automated text generation are news production in the media industry, product descriptions for online shops, business intelligence reports and unique text production for search engine optimization (SEO). In the field of news production vast amounts of data are available for weather, finance, events, traffic and sports. By combining methods of big data analysis and artificial intelligence, not only pure facts are transferred into readable text, but also correlations are highlighted.

A major example is focus.de, one of the biggest German online news portals. They publish around 30,000 automated weather reports with 3 days forecast for each German city each day. Another example for high-speed and high-volume journalism is handels-blatt.com. Based on the data of the German stock exchange, stock reports are generated for the DAX, MDax, SDax and TecDax indexes every 15 minutes. These reports contain information on share price developments and correlate it to past data such as all time highs/lows, as well as to data of other shares in the same business sector.

An important side effect resulting from publishing such big numbers of highly relevant and up-to-date news is a considerably increased visibility within search engines such as Google, Bing etc. As a consequence, media outlets profit from more page views and revenues from affiliate marketing programs.

From the numbers of published reports it is clear that human editors are not able to write them in the available time. In contrast, automated text generation produces such reports in fractions of a second, and running the text generation tools in cloud based environments adds arbitrary scalability since the majority of the reports can be generated in parallel. Thus, in the foreseeable future, the amount of generated news will exceed that of news written by human authors.

6.4 Advanced Methods for Text Generation

In this section we will sketch a semantic approach to augment our generation approach. The base functionality of our tool, Text Composing Language (TCL), used for text generation has already been described in Bense and Schade [2]. In short, TCL is a programming language for the purpose of generating natural language texts. A TCL program is called a template. A template can have output sections and TCL statements in double square brackets. The eval-statement enables calls to other templates as subroutines.

The semantic expansion we want to discuss here aims at adding background knowledge as provided by an ontology. This corresponds to the exploitation of encyclopaedic knowledge by the cognitive "macro planning". The ontological knowledge for TCL is stored in a RDF-triple store[1] which has been implemented in MySQL. The data can be accessed via query interfaces on three different layers of abstraction. The top most layer provides a kind

[1] https://en.wikipedia.org/wiki/Triplestore.

of description logic querying. The middle layer, OQL (Ontology Query Language), supports a query interface which is optimized for the RDF-triple store. OQL queries can be directly translated into MySQL-queries. Triples are of the form (s, p, o), where s stands for subject, p for property and o for object. The basic OQL-statements for the retrieval of knowledge are getObjects (s, p) and getSubjects (p, o), e.g. getObjects (‘>Pablo_Picasso’, ‘*’) would retrieve all data and object properties of the painter Pablo Picasso, and getSubjects (‘.PlaceOfBirth’, ‘Malaga’) would return the list of all subjects, who were born in *Malaga*. According to the naming conventions proposed in Bense [1], all identifiers of instances begin with the >-character, those for classes with the ^-character, data properties with a dot and names of objects properties start with <>. TCL supports knowledge base access via the get(s,p,o) function. Depending on which parameters are passed, internally either *getObjects* or *getSubjects* is executed, e.g. getSubjects (‘.PlaceOfBirth’, ‘Malaga’) is equivalent to get (‘*’, ‘.PlaceOfBirth’, ‘Malaga’). An example for a small TCL program is:

```
[[ LN = get (‘>Pablo_Picasso’, ‘.LastName’, ‘’)]]
[[ PoB = get (‘>Pablo_Picasso’, ‘.PlaceOfBirth, ‘’)]]
$LN$ was born in $PoB$.
```

This TCL program creates the output: “Picasso was born in Malaga”.

The graph in Fig. 6.1 shows an excerpt of the knowledge base about a soccer game. Instances are displayed as rounded rectangles with the IDs of the instances having a dark green background colour [1]. The data properties are shown as pairs of attribute names and their values. The named edges which connect instance nodes with each other represent the object properties (relationship types) between the instances, e.g., <>is_EventAction_of and <>is_MatchPlayerHome_of. The schema behind the example data contains classes for ^Teams (‘T_’), ^MatchFacts (‘MF_’), ^MatchEvents (‘ME_’), ^Player (‘P_’), ^Match_PlayerInfo (‘MP_P_’), ^Stadium (‘STD’) and ^City (‘CIT’). The match is connected to its teams via (>MF_160465, <>HomeTeam, >T_10) and (>MF_160465, <>AwayTeam, >T_18). All match events are aggregated to the match by the object property <>is_EventAction_of. The inverse of <>is_EventAction_of is <>EventAction. An event action has a player associated with an ^Match_PlayerInfo instance, e.g., by <>MatchPlayerScore in the case the player scoring a goal, or an assisting player is connected to the event using the object property <>AssistPlayer. Each ^Match_PlayerInfo instance is associated with a player via the <>Player relationship type. Finally, each team has a stadium (<>Stadion, inverse object property: <>ist_Stadion_von) and each stadium has an associated city (<>ORT).

The data model behind the application for the generation of premier league soccer match reports is much more complex, but the small excerpt gives a good impression of the complexity it deals with. Accessing the information needed for generating text output for a report can be a cumbersome task even for experienced database programmers. The following explains the implementation of a method that can quickly retrieve information out of these graphs. In principle, the terms sought can be easily derived even by non-programmers, by following the path from one instance in the knowledge graph to the

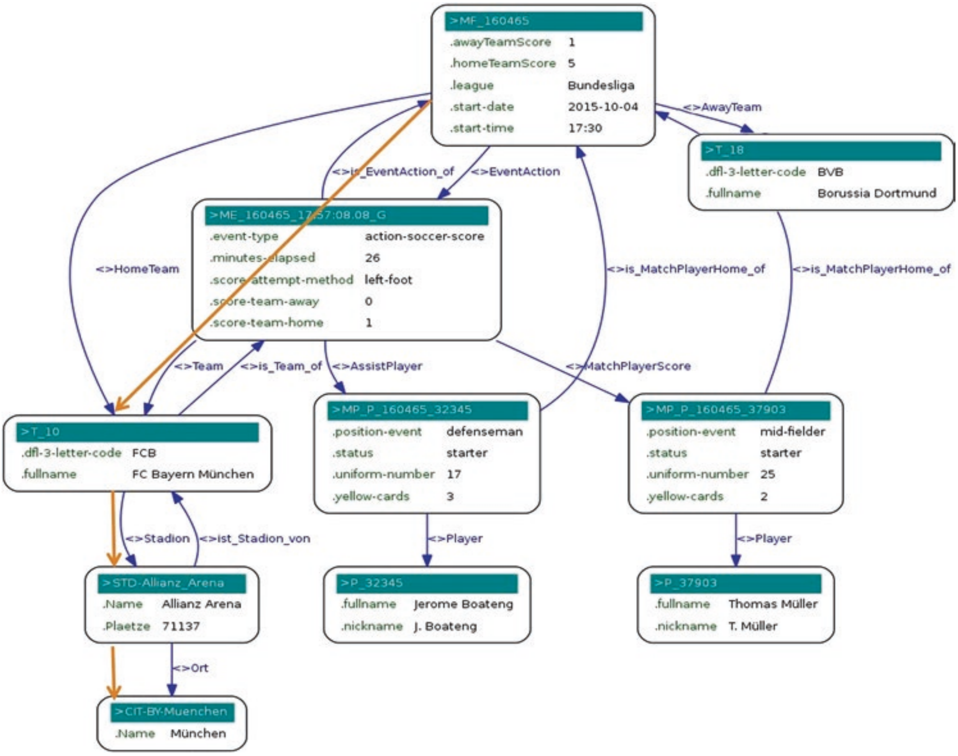

Fig. 6.1 Ontological knowledge to be exploited for generation of soccer reports

targeted instance, where the needed information is stored. The path (the orange arrows in Fig. 6.1) starting at the instance node of the match **>MF_160465<>Hometeam<>Stadion<>Ort.Name** follows the chain of properties **<>Hometeam<>Stadion<>Ort.Name** to give access to the name of the city where the event takes place. A property chain is the concatenation of an arbitrary number of object property names, which can be optionally followed by one data property name, in this case **.Name**.

In TCL, templates can be evaluated on result sets of OQL queries. The query getObjects ('>MF_160465', '<>HomeTeam') positions the database cursor on the corresponding triple of the knowledge base. In a template, the values of the triple can be referenced by the term **S**. Beyond this, the TCL runtime system is able to interpret property chains on the fly. Therefore it is possible to have the following declarations as part of a template header:

```
STRT = $S.start-time$
DTE = $S.start-date;date(m/d/Y)$  /* formatted in English date format
STDN = $S<>HomeTeam<>Stadion.Name$
CTYN = $S<>HomeTeam<>Stadion<>Ort.Name$
```

Then the template ***"The game started on DTE at $STRT$ o'clock in the $STDN$ in $CTYN$."*** generates for the sample data the output ***"The game started on 10/4/2015 at 17:30 o'clock in the Allianz Arena in München."***.

Internally, an automatic query optimization is applied for property chains. The processing of property chains is an iterative process, where initially the subject is retrieved together with its first property. The resulting object becomes the new subject, which is then retrieved in combination with the second property and so forth. Each retrieval is realized by an SQL-SELECT. The length of the property chain determines, how many queries have to be executed. Therefore, starting from the match instance, four queries are needed to retrieve the name of the city where the match takes place. The query optimizer takes the complete property chain and internally generates and executes a nested SQL-Query. Performance benchmarks have shown that when property chains are used, execution time can be significantly reduced.

6.5 Increasing Quality by Exploiting Information Structure

In this section, we will discuss how to increase the quality of the generated texts by exploiting the semantic principle of information structure. The selection of another lexical entry for a second denotation of a just mentioned concept (e.g., in order to denote a share, the term "stock" can be used in English; in German "Wertpapier" can substitute "Aktie") increase readability and text quality. In the cognitive process of lexical access, this principle is incorporated naturally as used items are set back in activation and have to recover to show up again. Sometimes, the same holds for grammatical patterns: consecutive SPO sentences feel monotonous. We will discuss an approach to automatically vary sentence patterns below. With this approach, we make available a set of grammatical patterns that can be used to generate the next expression. Having this set available, we can prune it semantically to emulate information structure. In order to clarify what is meant by "information structure" from the perspective of the cognitive process, we will shortly discuss its lexical counterpart. In the Levelt model, the concepts of the preverbal message are annotated according to their "availability" (whether they have already been mentioned before). This might lead to the selection of a different lexical entry as discussed. Alternatively, complex nouns in noun phrases can be reduced to their head ("Papier" instead of "Wertpapier"). Noun phrases even can be reduced to the corresponding personal pronoun, if the respective concept is in "situational focus". For example, "Robert Lewandowski has been put on in minute 62. Robert Lewandowski then scored the goal to 2-1 in minute 65" can and should be substituted by "Robert Lewandowski has been put on in minute 62. He then scored the goal to 2-1 in minute 65" in order to generate a cohesive text. In Bense and Schade [2], we already discussed an algorithm that can handle these kinds of cases. In addition to this, noun phrases that are constituted by a name and that in principle can be reduced to a pronoun, can also be substituted by another noun phrase that expresses encyclopaedic knowledge. Considering again the “Robert Lewandowski” example, the second occurence of his name in the original text could be substituted with “The Polish national player” which adds information and makes the whole expression more coherent [14].

We also developed a program that generates the possible variations for given sentences. In doing so we make use of the fact that, especially in German and English, the word order is determined by certain rules and structures. Phrases have already been introduced. There are a few tests at hand which help clarify whether a certain group of words form a phrase or not. One of these tests, the permutation test, checks if the words in question can be moved only as a whole. In example (2) the word sequence “in the 65th minute” is moved. The result is a correct sentence, so the sequence is a phrase. In (3) only “65th minute” is moved. The result is not grammatically correct as indicated by an *. Thus, “65th minute” is not a phrase on its own.

(1) Lewandowski scored in the 65th minute.
(2) In the 65th minute, Lewandowski scored.
(3) *65th minute Lewandowski scored in the.

The mobile property of phrases is used to determine the variants of a sentence, but in order to do so, another linguistic concept must be taken into consideration.

For practical reasons, the German language is the most important to us and its word order can be described quite conveniently with so called topological fields (a good description can be found in Wöllstein [16]. Similar approaches hold for most other Germanic languages, e.g., Danish, but not for English. The topological field approach separates a sentence into different fields corresponding to certain properties. Three basic types are distinguished using the position of the finite verb as the distinctive characteristic. The types are illustrated by the three sentences in Table 6.1. In V1-Sentences, the finite verb is the first word of the sentence and builds the so called Linke Klammer (left bracket), which – together with an optional Rechte Klammer (right bracket), built by the infinite part of a complex predicate – surrounds the Mittelfeld (middle field; contains all the other parts of the sentence). This type of sentence corresponds mostly with the structure of questions. In the case of V2-Sentences, the finite verb is preceded by exactly one phrase – the verb therefore occupies the second

Table 6.1 The German sentence types illustrated by examples – the example sentences translate to “Did Lewandowski run 100 Meter?”, “Lewandowski ran 100 meter because…” and “… because Lewandowski ran 100 meter”, respectively

Type	Vorfeld (prefield)	Linke Klammer (left bracket)	Mittelfeld (middle field)	Rechte Klammer (right bracket)	Nachfeld (final field)
V1		Ist	Lewandowski 100 Meter	gelaufen?	
V2	Lewandowski	ist	100 Meter	gelaufen,	weil…
VL		weil	Lewandowski 100 Meter	gelaufen ist.	

spot – in the Vorfeld (prefield). The rest of the sentence is the same as the V1-Sentence, except for the addition of the Nachfeld (final field), which can be found after the Rechte Klammer (right bracket) and mostly contains subordinate clauses. This type mostly corresponds with declarative sentences. Finally, there is the VL-Sentence (verb-last sentence) and as the verb is in last position, its construction is a bit different. Vor- and Nachfeld are not occupied, and a subjunction fills the Linke Klammer while the whole predicate is in the Rechte Klammer. The Mittelfeld is again filled with the rest of the sentence.

The different properties of the fields are numerous enough to fill several books. Here, two examples shall be sufficient to demonstrate in which way we make use of which principles. We will focus on V2-Sentences because their relative abundance makes them the most important sentences to us. The sentences (4)–(6) all have the same proposition – that yesterday Lewandowski ran 100 meter – but we will not translate all the variations being discussed, because rules of word order can be very language specific, i.e. one word order might be wrong in two languages for two different reasons. Trying to imitate the different word orders of German might therefore lead to false analogies in the reader's perception. A prominent property of the Vorfeld – and an important difference to its English equivalent – is its limitation to only one phrase. The following sentence, which has an additional "gestern" ("yesterday") is incorrect because two phrases occupy the Vorfeld:

(4) *Lewandowski gestern ist 100 Meter gelaufen.

The properties of the Mittelfeld mostly concern the order of its phrases. The subject – in this example "Lewandowski" – is mostly the first element in the Mittelfeld, if it doesn't occur in the Vorfeld already. Therefore, example (6) is grammatically questionable as indicated by a '?' while (5) is correct.

(5) Gestern ist Lewandowski 100 Meter gelaufen.
(6) ?Gestern ist 100 Meter Lewandowski gelaufen.

Interesting to us is the fact that the limitation of the Vorfeld actually concerns the concept of the phrase and not just a few select words. The following sentence is absolutely correct in German:

(7) Der in Warschau geborene und bei Bayern München unter Vertrag stehende Fußballspieler Robert Lewandowski ist gestern nur 100 Meter gelaufen. ("The soccer player Robert Lewandowski, who was born in Warsaw and is under contract at Bayern München, has run 100 meters yesterday.")

This shows that phrases and topological fields are not just concepts invented by linguists in order to describe certain features of language more accurately, but reflect actual rules, which are acquired in some form and used during speech production. Therefore, we want to use these rules for the generation of texts as well.

This means that, under the rules described by Vorfeld and Mittelfeld, sentence (8), the German translation of sentence (2), has 2 valid variations; sentences (9) and (10).

(8) Lewandowski schoss das Tor in der 65. Minute.
(9) In der 65. Minute schoss Lewandowski das Tor.
(10) Das Tor schoss Lewandowski in der 65. Minute.

Up to this point the argumentation has been exclusively syntactical. This might lead to the conclusion that sentences (8), (9), and (10) are equivalent. However, the fact that things are a bit more complex becomes obvious when semantics or, more precisely, information structure is taken in consideration. In the cognitive process of language production, concepts are annotated by accessibility markers, as we already mentioned when discussing variations of noun phrases. In the production process, accessibility markers also represent additional activation. This means that a concept with a prominent accessibility marker will most probably activate its lexical items faster. The corresponding phrase therefore has a better chance to appear at the beginning of the sentence to be generated. From a formal linguistic point of view, this is the meaning of "information structure", expressed by the formal concepts "Theme-Rheme" and "Focus" [13]. The terms theme and rheme define a sentence by separating known information (theme) and new information (rheme), where the theme would normally precede the rheme. Following this concept, variant (8) would be chosen, if the information that a goal has been scored is already known. The focus is a way to stress the important information in a sentence. It mostly coincides with rheme, but this is not necessarily the case. The concept only works in combination with the concept of an unmarked sentence, where the structure is being changed to emphasize certain elements. One could argue that variant (8) is such an unmarked sentence, because it follows the order of subject-predicate-object. Variant (10) differs from that order and, by doing so, pushes "das Tor" ("the goal") into the focus. This variant could be used as contrast to another action of Lewandowski, e.g. a foul. Currently, we are working on automatically determining the best choice from the available set of sentences.

6.6 Recommendations

Automatic generation of texts is worth considering if the purpose the text is given and simple. It is best employed to present data that is available in a structured form, e.g. in a table. Automatic generation of texts is profitable if such a presentation of data is in demand and needs to be repeated regularly.

In order to generate the texts, it is sufficient to make use of templates. Smart variations are fine and necessary but high literacy is not needed and beyond the scope of automatic generation.

In order to increase the quality of texts, human language production strategies can and should be exploited. This includes linguistic means specific for the target language and the use of (simple) ontologies for the representation of knowledge, see also Hoppe and Tolksdorf (Chap. 2).

6.7 Summary

In recent years natural language generation has become an important branch in IT. The technology is mature and applications are in use worldwide in many sectors. It is part of the digitalization and automation process which in traditional manufacturing is denoted Industry 4.0. The focus of this article so far was to show what can be expected in terms of generating more sophisticated texts. We advocated for the use of semantics to do so. In combination with an ontology as the knowledge base, the integration of reasoners will allow the derivation of automatically inferred information into the text generation process. The concept of property chains is essential for making this kind of retrieval fast enough. We also have shown how information structure can be used to vary the lexical content of phrases and to find the variation of a sentence that best captures the flow of information and thus contributes to enhancing the quality of generated texts in terms of cohesion and coherence.

6.8 The Next Trend: Hyper-Personalization of News

The upcoming trend in media industry is hyper-personalization. To date, most of the news articles are written for a broad audience. The individual reader has to search and select the news that is relevant to her/him. Though many apps already provide news streams for specific domains such as weather, sports or events, none of them create a personalized news stream. In the Google funded project 3dna.news, a novel approach has since been offered as a service in multiple languages. A user is immediately informed by e-mail or WhatsApp if, for example, a specific share she/he is interested in exceeds a given threshold, or when the next soccer game of her/his favorite team begins. In the latter example, she/he is also informed about relevant related information such as the weather conditions expected during the game and about all traffic jams on the way from their home to the stadium.

With hyper-personalization, publishing companies and news portals will be able to provide their readers with new service offerings resulting in a higher customer retention. The news consumer can tailor a subscription to their personal demands and gets the relevant information promptly. Hyper-personalization will also create a new opportunity for in-car entertainment. Currently, radio stations produce one program for all of their listeners. In the future it will be possible to stream the news individually to each car. Generated news will run through a text-to-speech converter and be presented to the driver as an individual radio program. This would also be applicable to in-home entertainment. Amazon's Alexa will allow the user to interact with text generation systems to respond to demands such as: "Alexa, give me a summarized report on the development of my shares!" or "Alexa, keep me informed about the important events of the soccer game of my favorite team!".

However, hyper-personalization potentially increases the danger of "echo chambers" disrupting societies. In addition to this, the resources needed to offer such services are tremendous. The number of news articles that have to be generated is on a much larger scale compared to general news for a broad audience. Also, the news generation process has to run continuously because events triggering the production of a new text could happen any time.

References

1. Bense H (2014) The unique predication of knowledge elements and their visualization and factorization in ontology engineering. In: Garbacz P, Kutz O (eds) Formal ontology in information systems, proceedings of the eighth international conference (FOIS 2014), Rio de Janeiro, Brazil, Sept. 22–25, 2014. IOS Press, Amsterdam, pp 241–250
2. Bense H, Schade U (2015) Ontologien als Schlüsseltechnologie für die automatische Erzeugung natürlichsprachlicher Texte. In: Humm B, Reibold A, Ege B (eds) Corporate Semantic Web. Springer, Berlin
3. Bock JK (1982) Toward a cognitive psychology of syntax: information processing contributions to sentence formulation. Psychol Rev 89:1–47
4. Bühler K (1934) Sprachtheorie: Die Darstellungsfunktion der Sprache. G. Fischer Verlag, Jena
5. Dell GS (1986) A spreading-activation theory of retrieval in sentence production. Psychol Rev 93:283–321
6. Fromkin V (1971) The non-anomalous nature of anomalous utterances. Language 47:27–52
7. Garrett M (1975) The analysis of sentence production. In: Bower G (ed) Psychology of learning and motivation, vol 9. Academic, New York, pp 133–177
8. Kempen G, Hoenkamp E (1987) An incremental procedural grammar for sentence formulation. Cogn Sci 11:201–258
9. Kempen G, Huijbers P (1983) The lexicalization process in sentence production and naming: indirect election of words. Cognition 14:185–209
10. Levelt WJ (1983) Monitoring and self-repair in speech. Cognition 14:41–104
11. Levelt WJ (1989) Speaking – from intention to articulation. MIT Press, Cambridge, MA
12. Levelt WJ, Roelofs A, Meyer AS (1999) A theory of lexical access in speech production. Behav Brain Sci 22:1–75
13. Musan R (2010) Informationsstruktur. Universitätsverlag Winter, Heidelberg
14. Nübling D, Fahlbusch F, Heuser R (2015) Namen: Eine Einführung in die Onomastik (2 Ausg.). Narr, Tübingen
15. Schriefers H, Meyer AS, Levelt WJ (1990) Exploring the time course of lexical access in language production: picture-word interference studies. J Mem Lang 29:86–102
16. Wöllstein A (2014) Topologisches Satzmodell (2. Ausg.). Universitätsverlag Winter, Heidelberg