International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:9, No:6, 2015

Natural Language News Generation from Big Data

Bastian Haarmann, Lukas Sikorski

Abstract**—**In this paper, we introduce an NLG application for the automatic creation of ready-to-publish texts from big data. The resulting fully automatic generated news stories have a high resemblance to the style in which the human writer would draw up such a story. Topics include soccer games, stock exchange market reports, and weather forecasts. Each generated text is unique. Ready-to-publish stories written by a computer application can help humans to quickly grasp the outcomes of big data analyses, save time-consuming pre-formulations for journalists and cater to rather small audiences by offering stories that would otherwise not exist.

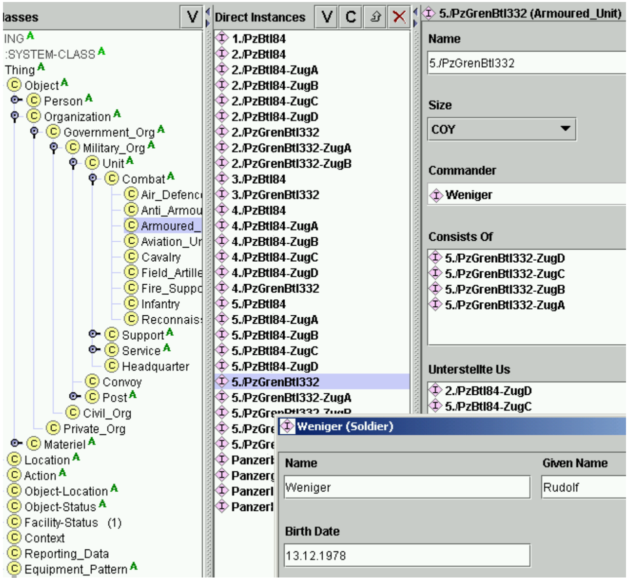
Keywords**—**Big data, natural language generation, publishing, robotic journalism.

I. Introduction

MANY media companies missed to seize the opportunities of society's digitalization in recent years and failed to install a financially attractive online appearance. This especially holds for print publishers such as publishing houses for newspapers and magazines. Hence, financial and human resources shrank while the need for an efficient automation of recurring writing tasks arose. However, audiences also demand coverage of events that may affect only rather small target groups. Writing more reports for fewer readers, however, is inconceivable. The natural language generation system we introduce can deliver a variety of automatically generated, data-based news stories about topics, i.e. regional or local sports coverage, articles about up-to-the-minute real estate price development, weather forecasts, or stock exchange market reports. The application is able to generate any thematic sort of text that is based on figures.

The system was developed by the Fraunhofer-Institute for Communication, Information Processing, and Ergonomics FKIE in order to demonstrate the feasibility of ready-to-publish soccer report generation in German from structured German and Turkish game data. The resulting texts are generated with the means of an ontology and can have various layouts and formats. The length of the stories is adjustable such that it is appropriate for any type of news, may it be a whole multi-page online-article or a tweet with only up to 140 characters.

In order to grasp the meaning of the facts given in the source data, the generation system needs background knowledge about the data's respective topic which the human reader already possesses in terms of empirical knowledge [1]. This knowledge has to be manually submitted to the generation system. However, the provision of knowledge in the form of simple lists or databases is difficult to manage and does not allow for the representation of hierarchies, properties or relations between lists' elements or database entries. Therefore, contemporary semantic applications make use of formal knowledge representations by using ontologies. An ontology is a collection of facts in which pieces of information are represented by so-called individuals and their respective properties and property values. The individuals are ordered hierarchically by super classes and subclasses. They might also be connected to each other by relations. Ontologies are usually domain-specific. For further reading on ontology basics the reader is referred i.e. to [2]. A sample picture of the outline of an ontology is depicted in Fig. 1.



5./PzGrenBtl332 (Armoured_Unit)

Name: 5./PzGrenBtl332

Size: COY

Commander: Weniger

Consists Of: 5./PzGrenBtl332-ZugD, 5./PzGrenBtl332-ZugC, 5./PzGrenBtl332-ZugB, 5./PzGrenBtl332-ZugA

Unterstellte Us: 2./PzBtl84-ZugD, 5./PzBtl84-ZugC

Weniger (Soldier)

Name: Weniger

Given Name: Rudolf

Birth Date: 13.12.1978

Fig. 1 Sample of a German ontology with individuals (purple), classes (yellow) and properties with data values as well as relations to other individuals (very right pane)

The grasp of relevant knowledge for a topic and the population of a respective ontology are usually time-consuming processes (cf. [3] for a contradicting view). They constitute the major part of the preparatory work to be carried out for the construction of a generation system.

II. Structure and Serialization

After the necessary background knowledge from representative data has been extracted and is available in the form of an ontology, the next steps are determining a) which kind of information is going to be packed together in one paragraph, b) in which order the paragraphs should be

Dr. Bastian Haarmann and Lukas Sikorski are with the Fraunhofer-Institute for Communication, Information Processing & Ergonomics FKIE, Fraunhoferstr. 20, 53343 Wachtberg, Germany (phone: +49-228-9435-722; fax: +49-228-9435-685; e-mail: bastian.haarmann@fkie.fraunhofer.de; lukas.sikorski@fkie.fraunhofer.de).

arranged, and c) which content should be contained in a paragraph. In the following, we will discuss these aspects.

A. Discourse Structure

The texts to be generated contain various kinds of information. They should be arranged under consideration of priority and content affiliation. One piece of information should at least be expressed by one sentence. However, it might also be expressed by more than one sentence. In order to determine the arrangement, a so-called discourse structure needs to be defined. The discourse structure determines the sections, their text spans, and the content they cover. Furthermore, the discourse structure contains conditions managing whether or not text spans are going to be produced.

So it can be set, for example, whether and under which conditions the text should have a section for a headline. The production of a headline can depend i.e. on the availability of a certain value in the input data. In addition, in every section it has to be determined which facts are expressed under which conditions. Moreover, within each section the order for these text spans should be set. This definition can either be declarative or a randomizer can be used if the arrangement of the text span within a section should vary or if text spans are arbitrarily exchangeable.

In distinction to the background knowledge which is supplied to the system by the ontology all information that comes from the data and has to be converted into text is called situational knowledge. This knowledge is either explicitly contained in the input data or can implicitly be calculated out of it. In the discourse structure, situational knowledge is grouped, positioned and enriched with respective output conditions. We will later discuss the question which facts are the most interesting and needs thus be expressed instead of others.

The discourse structure arranges the output text with respect to both order and content. It defines a layout determining which pieces of information belong together and form a text segment. Moreover, the order in which the pieces of information should be arranged within each section can be determined by the discourse structure as well. This order, however, can also be calculated by a randomizer. In addition, for every section one or more output conditions can be set in the discourse structure.

B. Sections

The sections model the paragraphs in the texts to be generated. A section contains a variable number of facts which can by itself consist of several sentences. Nevertheless, every section must express at least one fact. Thus, for example, the heading normally expresses exactly one fact (in the form of an entire sentence or a more or less reduced ellipsis). Other sections embrace more facts and group them to single units of meaning. Some units of meaning should not or cannot be expressed at all because the respective events or values are not given in the input data. For this case, output conditions can be formulated for each section so that a section will be verbalized only if all output conditions are fulfilled and all necessary data exist. An output condition can, i.e. allow for the existence or the height of one or more values in the input data.

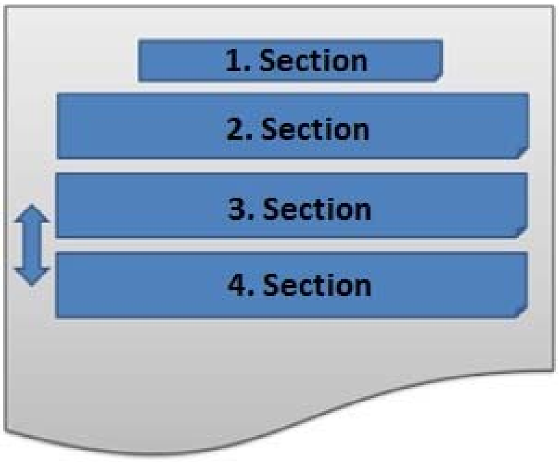


Fig. 2 Abstract pattern of a text’s discourse structure with both fixed and variable sections

Besides output conditions, separating characters can be defined for each section as well. The heading could be further separated from the text body than the body’s paragraphs are separated from each other, i.e. with two linefeed characters rather than one.

But serialization also deals with information arrangement in the form of main clauses and subordinate clauses as well as the assignment of syntactic roles such as subject, predicate, object and complements. However, since on the one hand the form cannot entirely be separated from its meaning and on the other hand it is basically about the implementation of grammatical rules, syntax will be discussed later.

C. Dramaturgy

Dramaturgical rules decide on the one hand the binary report worthiness of a fact and therefore whether something will be either verbalized or not. On the other hand these rules also qualitatively decide on the information value of a fact in comparison to other interesting facts within a section or the whole text. Dramaturgical rules therefore order the output text concerning its main topic and the other facts worthy to be reported. This is important for the choice of the headline and - if necessary - for the creation of the first paragraph, given that it delivers further details to the headlined fact.

It is possible that a fact is considered worthy to be reported as long as there are no more other interesting facts. With the occurrence of a more interesting piece of information a fact can lose its report worthiness and (in comparison to the information more interesting now) be classified as irrelevant. Facts that are rare or unexpected are usually dramaturgically higher valued and therefore considered more report worthy depending on the kind of text and its topic. As less report worthy are considered those facts that reoccur more or less regularly, are basically common and can be expected.

Typical examples of the influence of dramaturgical rules on the text generation can be found in weather reports. For example, the cloud density might be held report worthy for a region as long as no rare weather phenomenon such as a hurricane occurs which is then of course covered in the report instead making the information about the cloud density pretty much irrelevant. To give another example, the maximal temperature of the next day during summer is surely dramaturgically higher valued as the temperature at night. Hence, it typically will be expressed earlier in the report. Furthermore, more information may be connected to that piece of information, i.e. a comparison to the highest temperature of the previous day as well as a forecast for the whole week.

Vice versa, dramaturgical rules may also admit facts to be expressed which would ordinarily not appear in the text because of the absence of their report worthiness. This occurs i.e. if no further information exists in the input data that is interesting and should therefore be verbalized. An example for this can be the mention of a soccer game’s fouls in a match report in case no goals were scored and nothing else happened that could possibly be reported instead.

The dramaturgical rules also serve as an overview which pieces of information might be verbalized. In any case, it has to be checked in the input data whether an event which is subject of a rule to be applied exists. How this analysis works is described in detail in the following.

III. Provision of Data for Verbalization

At this point of the preparation of a text generation system it is already determined by dramaturgical rules which facts can possibly be verbalized and which priority each fact has. Furthermore the discourse structure lays out the order by which the facts shall be verbalized and the conditions under which this shall happen. Next, we will shortly describe the analysis of the facts given in the input data as well as how they are selected and stored. Subsequently, we discuss formal aspects of the verbalization.

The acquisition of the situational knowledge, in other words the information to be verbalized from the source data, is possible in various ways, cf. [4]. The optimal way to supply the data for verbalization to the generation system strongly depends on the use case. Also, the best solution depends on the data format and on the implementation environment of the generation system. Hence, the information flow we present here is to be considered as exemplary and shows only one of several possible solutions.

A. Fetching and Transferring Situational Knowledge

If the source data is given in different formats, it is necessary to convert the relevant information into a uniform format. For this purpose the XML format has proved itself. If a complete conversion of the data into XML is not possible or not desired, it is sufficient to select only those pieces of information to be verbalized and to transfer them into XML. The underlying XML schema should respect the previously formulated dramaturgical rules and should provide the data necessary for the execution of these rules. In addition, data necessary for the query of the conditions in the discourse planning must be selected and represented as well.

B. Variables as Storage

In the next step, concrete events and their metadata can be selected from the XML mass data and stored in predefined variables. Subsequently, implicit facts can be inferred or calculated from the existing explicit data.

For example, a set of input data might contain the final score of a soccer match for home and visiting team which is stored in a variable called $finalscore. However, it isn't explicitly stated in the data set that the winner is the team with the higher score and that the other team is the loser. Those implicit facts are calculated. The facts derived from implicit information must also be stored into predefined variables, e.g. in $winner and in $loser.

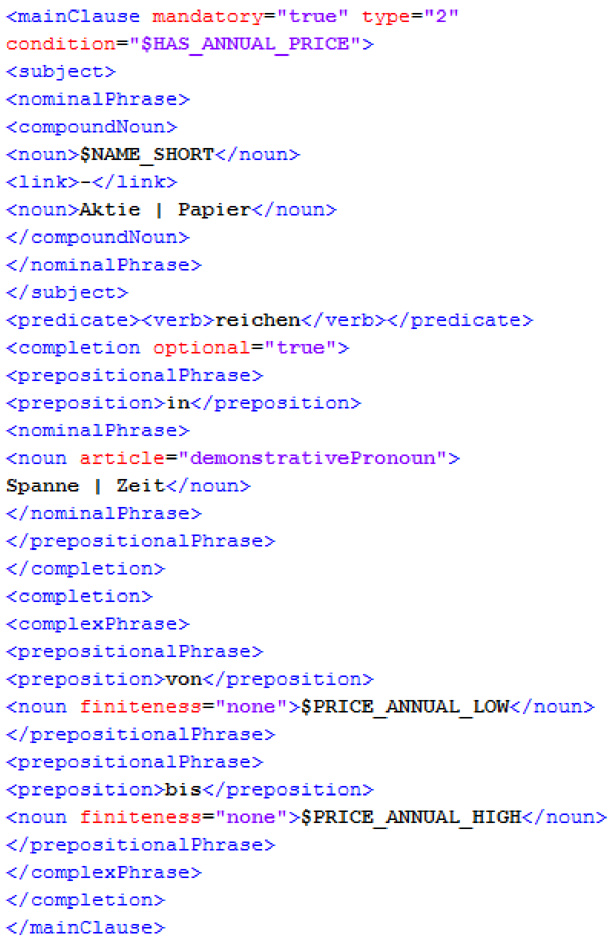
Besides, it should be distinguished between variables with Boolean values indicating the truth or untruth of a fact (e.g. $match_drawn = false) and variables with string values containing names (e.g. $winner). String variables therefore may at first get a fix value. However, this may ultimately be alternated by a variation manager. Thus, the name of the city Berlin obtains at first the string “Berlin”. Later it might be substituted with other expressions such as „the capital“. But this depends on other factors, i.e. entries for the respective entity in the ontology, the availability of other property values, dynamic form rules, previously used formulations and superior constraints. A detailed description of the word variation can be found in V.

IV. Semantic Templates

For every section there is a repertoire of templates which abstractly represents all potentially clauses. The abstraction basically exists in the form of a semantic structure of each clause with the least possible determination and the most possible exhaustion of degrees of freedom. Rigid syntactic structures for the sentence within the templates are avoided whenever possible and as far as acceptable for the reader. Hence, the semantic templates comprise of syntactic roles such as subject, object, completion, etc. Moreover, actors that have a variable name will be substituted with some suitable, predefined string variables in the templates (cf. III B).

A. Tagging Constituents

On the one hand, semantic templates constitute the semantic content of a sentence and are adjusted to the least possible syntactic determination. On the other hand, they may also contain conditions that determine if a sentence contains a specific constituent (resp. a specific formulation) or determine which constituents can be expressed by a randomly chosen pattern. In Fig. 3 above the role completion for example has the parameter optional=”true”. This means that it may or may not be verbalized together with the rest of the template. The allocation of syntactic roles to semantic actors enforces the syntactically correct form according to grammatical rules (cf. section VI “Linguistic Conversion”). Constituents are modeled as phrases that might be complex. Syntactic subjects and objects are verbalized as noun phrases. Completions and extensions are mostly expressed as prepositional phrases. Free adverbs are possible as well. Syntactic order is managed both by sentence construction plans which enforce grammatical correctness and by a randomizer which determines the output order out of the total of all allowed variants.



```
<mainClause mandatory="true" type="2"
condition="$HAS_ANNUAL_PRICE">
<subject>
<nominalPhrase>
<compoundNoun>
<noun>$NAME_SHORT</noun>
<link>-</link>
<noun>Aktie | Papier</noun>
</compoundNoun>
</nominalPhrase>
</subject>
<predicate><verb>reichen</verb></predicate>
<completion optional="true">
<prepositionalPhrase>
<preposition>in</preposition>
<nominalPhrase>
<noun article="demonstrativePronoun">
Spanne | Zeit</noun>
</nominalPhrase>
</prepositionalPhrase>
</completion>
<completion>
<complexPhrase>
<prepositionalPhrase>
<preposition>von</preposition>
<noun finiteness="none">$PRICE_ANNUAL_LOW</noun>
</prepositionalPhrase>
<prepositionalPhrase>
<preposition>bis</preposition>
<noun finiteness="none">$PRICE_ANNUAL_HIGH</noun>
</prepositionalPhrase>
</complexPhrase>
</completion>
</mainClause>
```

Fig. 3 Example for a simple semantic template

The German language is – in contrast English – a language with quite free constituent order, especially with respect to constituent indicating time and place. Nevertheless, for some constituents we cannot completely waive a certain control of their position in the sentence. Certain completions are only acceptable at the end of the respective sentence, depending on semantics and value of information in comparison to other completions. This is i.e. the case in Fig. 3 (last complex phrase). Although the phrase "of X to Y" is grammatically correct for prepositional phrases in all permissible positions it is nevertheless accepted only either at the very beginning or at the very end of the sentence. In order to be able to control this phenomenon, there is a so called extension which can be used once per sentence and which is handled by the system in the same way as a completion but is always placed at the end of a sentence. Certain markings (not in the figure) can also enforce a sentence-initial realization. In German, a sentence initial prepositional phrase triggers the positional exchange of subject and verb.

Phrases are formed and declared in accordance with dependence grammar (Tesnière 1959). The noun between the tags <noun> and </noun> determines the noun phrase according to the grammatical meta-information (i.e. gender, inflection etc.) as stored in the ontology. Nominal phrases take the grammatical case as required by the preposition. All grammatical meta-information comes from the ontology. Determiners don't have to be indicated. All noun phrases are per default generated with a finite determiner given that nothing else is indicated. Determiners which are non-finite or absent can be enforced with the help of the attribute "finiteness". A determiner can also be replaced with a possessive pronoun (“his”), a demonstrative pronoun (“this”) or an indefinite pronoun (“some”) under certain conditions. Numerals are not yet implemented.

In addition, the templates may contain conditions under which they are valid. These conditions depend on Boolean variables (this can i.e. be seen in the second line in Fig. 3: condition = "HAS_ANNUAL_PRICE").

B. Competitive Templates

Templates which can express the same information by a different wording are assigned to the same type. The type of a template is indicated by an ongoing ID (Fig. 3, line 2: type=”2”). Basically, semantic templates of the same type can substitute each other (paradigmatic substitution). During generation, the system consecutively collects templates which can satisfy all conditions for expressing a fact to be verbalized. One of those templates is randomly selected from this set. Once being chosen, templates can be blocked for a customizable number of consecutive sentences. In addition, it is also possible to block a chosen verb for a customizable number of sentences in order to avoid unnatural repetitions.

C. Alternation

String variables (Fig. 6, line 6: $NAME_SHORT) as well as defined words can be enriched with alternative words or alternative string variables. The randomizer selects one option from the set of alternatives (marked by the separator "|" such as i.e. in Fig. 3, line 8: share | paper (“Aktie” | “Papier”) or line 19: span | time (“Spanne” | “Zeit”) at generation time. It is furthermore possible that every word has synonyms defined in the underlying ontology which also can be chosen as an alternative (e.g. “share” for “stock”).

V. Factors Influencing the Variation of Content

During the generation process the system must often choose among alternatives in order to allow variation. However, there are several factors that can specifically control the choice or exert an influence on it. The influence factors are explained in the following paragraphs.

A. Randomness

In the case of equal alternatives of equal interest a randomizer decides which alternative is chosen. This happens equally distributed leading to all equal alternatives being chosen equally often. Randomness is the most obvious one. It is nevertheless sometimes the last instance of the choosing mechanisms. It becomes effective either if several construction plans, templates or words equally compete for the choice or if the constituents or whole templates have a binary optional condition (optional=”true”). In this case they are either randomly output or not.

B. Content Conditions

Certain templates are not to be chosen in any case. Templates can contain content conditions which must be fulfilled for the template to be part of the choice set. Templates which do not satisfy the conditions are early excluded from the choice set. For example, in a weather report all templates for the generation of sentences about rain will be suppressed if the weather is sunny.

C. Verbal Context

It is taken into account what was generated previously. Templates and verbs may be blocked for a certain amount of sentences after they have been used. By this, it is prevented that one and the same pattern or verb appears too often. The text thereby becomes more diverse. Linguistic monotony is prevented. These suppressions are light, meaning they do not take effect if no other verb or template can be chosen and the sentence could otherwise not be generated.

D. Predefined Parameters

Some alternatives are ruled out by parameters in a configuration file. For example, it can be controlled in which tense the text is basically written. For many texts past tense is usual such as for daily reports about stock market values. Anyway, if the events to be verbalized lie in the future the present tense is necessary. The strictly defined tense is a standard value and can be overwritten for every single template verb. This way, it is possible to generate single sentences or sections in a different tense. Other examples for predefined parameters are the presence or absence of a heading or the issue of tense relations (absolute date or a statement relatively to the generation time, e.g. “yesterday”).

E. Constraints

With the help of an ontology, constraints can be included. This makes it possible to define rules which involve an entity’s property values in the name-forming of synonyms. Such constraints permit to use for example “the German capital” or “the French capital” instead of the entities “Berlin” or “Paris” by choosing the adjective form of the country’s name from the ontology and insert it into the expression. The whole construction then substitutes the regular entity name. This is the case if the respective city is declared as a capital in the ontology in contrast to all other cities of the country and a corresponding condition for the forming rule exists which queries the ontology. In addition, there are defined synonyms for many words in the ontology which compete automatically with the standard term. This rule is also a constraint.

VI. Linguistic Conversion

After the situational knowledge is extracted from the input data and templates are selected, the generating the expression is next. During this step, a number of grammatical rules are to be obeyed. For further reference, see [5] and [6]. But there are also other factors of influence on the form of verbalization which will be discussed in the following.

A. Diction

If a template has been selected from the pool of those appropriate, all of its variables and nouns will be looked up in the ontology during the conversion process. In the ontology there may be synonyms, so that in the next step it is decided which terms and nouns are substituted by synonyms.

B. Inflection and Phrase Construction

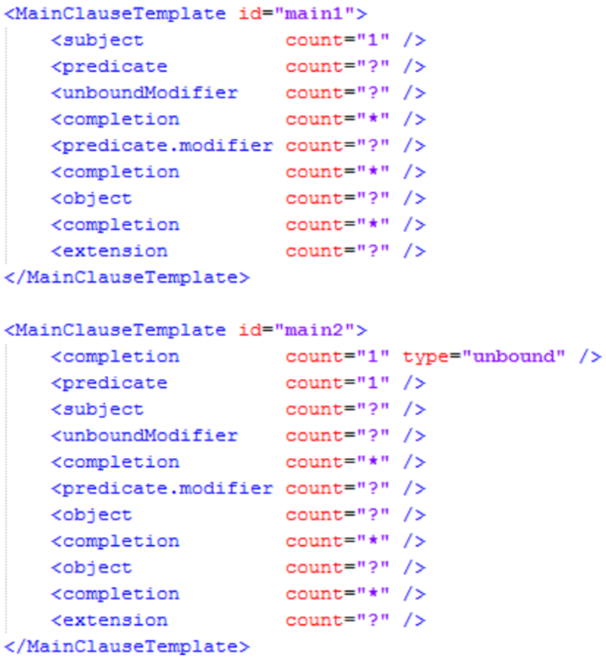
Every content word and every word governing others is stored in the ontology together with its grammatical meta-information. For words that are not available in the ontology there are default values concerning their grammar. In order to transform the structure of a template into a "surface structure"[1], the first step is to inflect content words during the construction of phrases. The necessary information for inflection is provided by the ontology. Nouns for example are inflected with respect to case, gender, and number. Verbs are classified according to being either regular or irregular, have diverse tense forms, case government etc. The inflection component inflects the words according to the provided rules (such as tense morphemes) and the grammatical information from the ontology. The phrase construction component assigns the inflected words to a phrase in their proper order and applies other word forming processes, i.e. a contraction (“do” + “not” results in “don’t”).

C. Sentence Construction

After the phrases of the sentence are constructed they will be put together to main clauses and subordinate clauses. In order to do so, there are various syntactic templates in the system which can be used for serialization of phrases. All syntactic information provided is taken from [9]. Basically, in case of the German main clause, its construction only depends on the differentiation between an inverted main clause and a main clause with subject on first position and verb on second position. The inversion, meaning the positional permutation of subject and predicate, only occurs whenever a subordinate clause, some adverb or a phrase that is not the subject occupies the initial position. For prepositional phrases which are completion constituents there are several positions to be taken. However, multiple occurrences of a specific type of constituent in the same position might be allowed while this does not hold for other constituents. This is governed by the respective syntactic template: every component of a syntactic template contains an attribute “count”, whose value expresses the component’s allowed occurrence in the form of an

[1] Though this step reminds of transformations in terms of [7] and [8] it doesn’t mean that a component of the transformation grammar or the theory of government and binding would be used for it.

operator indicating either mandatory once (1), optional once (?) or arbitrary often (*), see Fig. 4.



```
<MainClauseTemplate id="main1">
    <subject              count="1" />
    <predicate            count="?" />
    <unboundModifier      count="?" />
    <completion           count="*" />
    <predicate.modifier count="?" />
    <completion           count="*" />
    <object               count="?" />
    <completion           count="*" />
    <extension            count="?" />
</MainClauseTemplate>

<MainClauseTemplate id="main2">
    <completion           count="1" type="unbound" />
    <predicate            count="1" />
    <subject              count="?" />
    <unboundModifier      count="?" />
    <completion           count="*" />
    <predicate.modifier count="?" />
    <object               count="?" />
    <completion           count="*" />
    <object               count="?" />
    <completion           count="*" />
    <extension            count="?" />
</MainClauseTemplate>
```

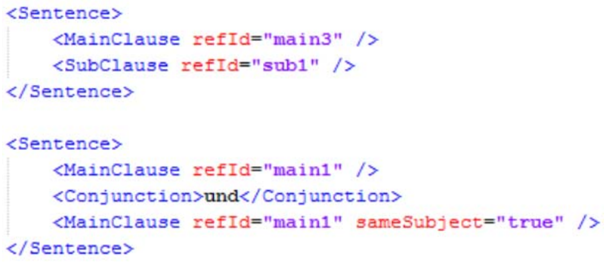
Fig. 4 Example for a simple semantic template

Only those syntactic templates are admitted for the sentence construction that can accommodate all available constituents of the semantic templates. Therefore, the generation system can easily distinguish between main- and subordinate clauses because subordinate clauses come with a subjunction which can't be represented by any main clause template.

After the construction of clauses they might be put together to form complex sentences. The joining of a main and a sub clause forms a hypotaxis whereas the joining of two main clauses forms a parataxis. In our system, the latter happens only in case both main clauses have the same subject (Precisely: The same noun that forms the subject's noun phrase or one of its synonyms). These concatenations are made by choosing one of the existing syntactic templates that can represent the respective clauses (cf. Fig. 5). If for a main clause there is neither a fitting sub clause nor another main clause with an identical subject, only the simple main clause will be generated.

VII. Factors of Influence on Formal Variation

Concerning the way how something is expressed, the generation system also tries to make for linguistic variation. Hence, there are some factors of influence – even if only a few – for the formulation of the content. These are explained in the following.



```
<Sentence>
    <MainClause refId="main3" />
    <SubClause refId="sub1" />
</Sentence>

<Sentence>
    <MainClause refId="main1" />
    <Conjunction>und</Conjunction>
    <MainClause refId="main1" sameSubject="true" />
</Sentence>
```

Fig. 5 Syntactic rule for a construction of a hypotaxis and a parataxis

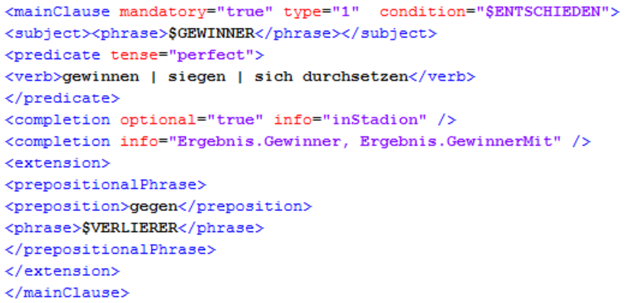
A. Sentence Construction Templates

At every position that is grammatically allowed, the system tries to vary the resulting expressions. The positional exchange of constituents is controlled by the aforementioned syntactic templates (cf. chapter VI C) that on the one hand formally realize the inversion of main clauses and on the other hand arrange clauses to complex sentences. The shifting of prepositional phrases to different positions within a clause is another variation.

B. Compression and Outsourcing of Information

Not every fact must exactly correspond to one single sentence. Some facts can be expressed within sentences covering another fact, e.g. as a completion in the form of a prepositional phrase. Of course, the same fact can be expressed in a proper sentence for itself. Very often this holds for facts about locations or points in time. As a result, the number of generated sentences can vary although the number of expressed facts is fix.

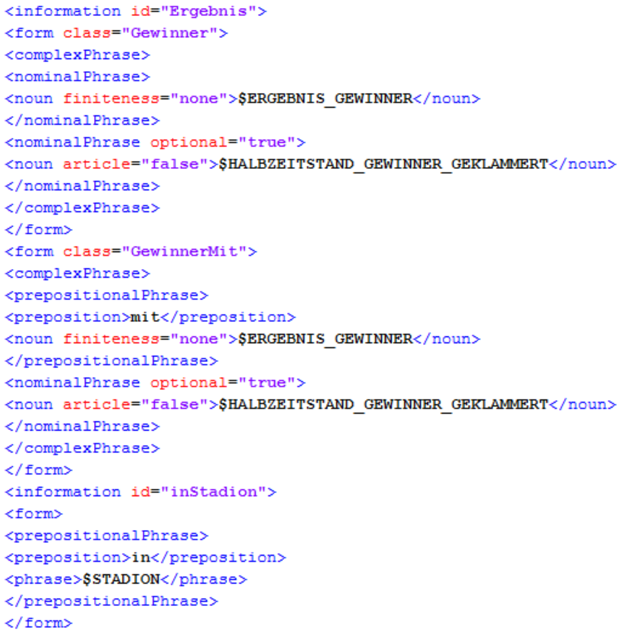
Technically, information (a fact) is labeled to be either mandatory or optional with respect to a sentence by the attribute "mandatory". This attribute occurs in the template of the sentence. Its value is Boolean. The value "false" means that the template must be an independent sentence, the value "true" however indicates that the template can either be an independent sentence or the complement of the preceding sentence (cf. Fig. 6).



```
<mainClause mandatory="true" type="1"  condition="$ENTSCHIEDEN">
<subject><phrase>$GEWINNER</phrase></subject>
<predicate tense="perfect">
<verb>gewinnen | siegen | sich durchsetzen</verb>
</predicate>
<completion optional="true" info="inStadion" />
<completion info="Ergebnis.Gewinner, Ergebnis.GewinnerMit" />
<extension>
<prepositionalPhrase>
<preposition>gegen</preposition>
<phrase>$VERLIERER</phrase>
</prepositionalPhrase>
</extension>
</mainClause>
```

Fig. 6 Semantic template with reference on outsourced units of information

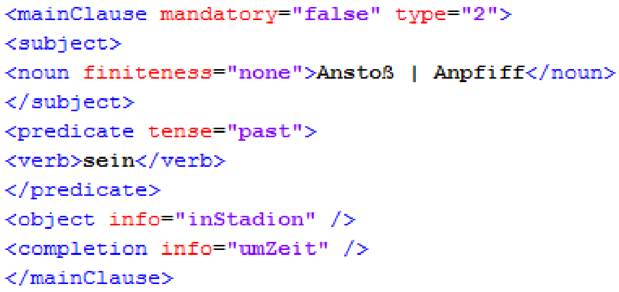
Furthermore, each template can include outsourced facts expressed by prepositional phrases (cf. Fig. 7). It can also be declared within the superior units of information that more than one variation is realizable. All possible variations' names can be separated by a comma and given as an attribute value.



```
<information id="Ergebnis">
<form class="Gewinner">
<complexPhrase>
<nominalPhrase>
<noun finiteness="none">$ERGEBNIS_GEWINNER</noun>
</nominalPhrase>
<nominalPhrase optional="true">
<noun article="false">$HALBZEITSTAND_GEWINNER_GEKLAMMERT</noun>
</nominalPhrase>
</complexPhrase>
</form>
<form class="GewinnerMit">
<complexPhrase>
<prepositionalPhrase>
<preposition>mit</preposition>
<noun finiteness="none">$ERGEBNIS_GEWINNER</noun>
</prepositionalPhrase>
<nominalPhrase optional="true">
<noun article="false">$HALBZEITSTAND_GEWINNER_GEKLAMMERT</noun>
</nominalPhrase>
</complexPhrase>
</form>
<information id="inStadion">
<form>
<prepositionalPhrase>
<preposition>in</preposition>
<phrase>$STADION</phrase>
</prepositionalPhrase>
</form>
```

Fig. 7 Outsourced units of information with formal variation

Facts can be only expressed once per section. This is guaranteed by a superior constraint which prevents the generation of informatively redundant sentences. Thus, the outsourced information can be realized either in mandatory semantic templates as a completion or by an optional template as a proper sentence (cf. Fig. 8). This way, the system alternately produces sentences with different length and density of information.



```
<mainClause mandatory="false" type="2">
<subject>
<noun finiteness="none">Anstoß | Anpfiff</noun>
</subject>
<predicate tense="past">
<verb>sein</verb>
</predicate>
<object info="inStadion" />
<completion info="umZeit" />
</mainClause>
```

Fig. 8 Optional template able to realize location information ("inStadion") and time information ("umZeit") in a proper sentence

VIII. Conclusion

The automated generation of natural language texts is especially recommended for expressing information extracted from data which is existent in a highly structured form and which refers to a limited topic of a specific domain. Text generation in the way presented in this paper is recommendable wherever the inherent meaning of a huge amount of data has to be presented to a human user in a compact way. Especially, the meaning of figures is otherwise only implicitly accessible to humans by a laborious structuration, e.g. by transforming the figures into several extensive tables. While the intended statement of graphical data representation remains implicit and is up to the beholder's interpretation, a textual condensate can explicitly state the meaning of large and recurring figural data and thus provides a quick and easy access to insights for humans.

References

[1] N. Bouayad-Agha, G. Casamayor & L. Wanner: Content selection from an ontology-based knowledge base for the generation of football summaries. ENLG 2011 Proceedings, pp. 72-81. Nancy, FR. 2011.

[2] N. Effingham: *An Introduction to Ontology*. Wiley-Blackwell: Hoboken, NJ, USA. 2013.

[3] T. Hoppe: *Messung des Nutzens semantischer Suche*. In: B. Humm, A. Reibold & B. Ege (ed.): Corporate Semantic Web. Berlin: Springer. 2015.

[4] E. Reiter & R. Dale: Building Natural Language Generation Systems. Cambridge University Press. Cambridge, UK. 2000.

[5] M. Bollmann: Adapting SimpleNLG to German. Proceedings of the 13th European Workshop on Natural Language Generation (ENLG), pp. 77-81. Nancy, FR. 2011.

[6] A. Gatt & E. Reiter: SimpleNLG: A realization engine for practical application. Proceedings of the 12th European Workshop on Natural Language Generation (ENLG), pp. 72-81. Athens, GR. 2009.

[7] N. Chomsky: Syntactic Structures. Mouton. Den Haag, NL. 1957.

[8] N. Chomsky: Lectures on Government and Binding. Mouton de Gruyter. Berlin, DE. 1993.

[9] L. Tesnière: Eléments de syntaxe structurale. Klincksieck. Paris, FR. 1959.