# A working, non-trivial, topically indifferent NLG System for 17 languages

**Robert Weißgraeber** and **Andreas Madsack**
AX Semantics
[robert.weissgraeber|andreas.madsack]@ax-semantics.com

## Abstract

We present a fully fledged practical working application for a rule-based NLG system that is able to create non-trivial, human sounding narrative from structured data, in any language (e.g., English, German, Arabic and Finnish) and for any topic.

## 1 Introduction

Use cases for Natural Language Generation are abundant and vary widely from theoretically interesting to practically relevant. Topically limited systems are already being used in different areas like weather reports (Ramos-Soto et al., 2013) or financial analysis (Nesterenko, 2016). Our topically unlimited software has an abstraction layer for text planning, structure, semantics, and content that can be fitted to any kind of subject. In addition, this abstraction layer includes a cross-language abstraction as well, making it feasible to write texts in multiple languages at the same time, independent from data source language or grammatical differences in the output. This demonstrates a feature complete NLG system (Reiter et al., 2000) with a rule-based non-template approach (Van Deemter et al., 2005). Most important, this is available as a web-based tool for everyone including eLearning elements. A free sandbox license for playing around is available, as well as free licenses for educational use – https://cockpit.ax-semantics.com/signup.

## 2 Topicality Free NLG abstraction

The basic level of the realisation is based in a container, which implements the basic NLG notation for a phrase. The notation is written in Automated Text Markup Language (ATML3) syntax, which is an open specification.

Example:

```
Máte [property1,adj=yes,case=acc,
cardinal=property2].
```

This ATML3 expression will generate in *Czech: Máte jednu novou zprávu* (You have one new message) or *Máte sedm nových zpráv* (You have seven new messages) from the noun *zpráva*, the adjective *nový* and a numerical value coming from the data. Noun and adjective are provided by *property1* and the numerical value is taken from *property2* which abstracts the data itself. Both properties are supplied by the planner component.

Containers are used inside statements, which implement the content determination, together with story types and statement groups which are responsible for the document structuring. Lexical choices can be defined by the configuration in the property output, and are related to the data they are interpreting, removing the limitations of topicality inside of the NLG core. Data intake and interpretation is done in so-called properties, which relate the data to extractable meaning, and are then referenced inside the linguistics configuration. Together, these configurations build the text and logic ruleset as a training for a certain topic and text output. If cross-language output is desired, word-level and phrase-level lookups can be used to identify uninflected word pairs across languages, which will automatically be inflected correctly during realization of the phrases.

*Proceedings of The 10th International Natural Language Generation conference*, pages 156–157,
Santiago de Compostela, Spain, September 4-7 2017. ©2017 Association for Computational Linguistics

## 3 Components and Compilation workflow

The data intake for a text is one data document, which is combined with the ruleset to form the basis for the text production. The NLG core then interprets the data and ruleset together, and combines this with grammatical information about the selected output language and a lexicon for word-based grammar information.

## 4 NLG Core Capabilities

The core grammar module allows for all possible language features. Each distinct used language has a unique configuration that combines those features according to the required phenomena. This allows for adding "new" languages within a few days.

## 5 User Access

ATML3 is designed as a markup language that, in contrast to programming approaches, allows everyone to create the required configuration. No developer skills are required, making the system suitable to be used by any kind of user. A GUI is available as well. To decrease manual effort, the software combines NLP and data analysis features: (1) properties are automatically generated from the training data; (2) part-of-speech tagging is used to transform a manual written sample text to an ATML3 ruleset automatically.

## 6 Examples

The sentence "A camera, Wi-Fi and bluetooth are just a few of its features." in ATML3 will render the pluralization and the verb based on the number of features in the dataset (`DATA_Features`), handling conjunction, capitalization and articles as well:

```
[DATA_Features.all(),conj=and,
det=indef,id=subject] [G:verb=be,
grammar-from=subject] just [Text:a few;
On,true=LOGIC_more_than_one_features;
Alt:one] of its features.
```

will return in English:
"A camera, Wi-Fi and bluetooth are just a few of its features." OR "Bluetooth is just one of its features."

The same containers are used in Spanish:

```
[Text:Algunas;
On,true=LOGIC_more_than_one_features]
de sus características
[G:verb=ser,grammar-from=subject;Alt:Una]
[DATA_Features.all(),conj=y,id=subject]
```

"Algunas de sus características son cámara, WiFi y Bluetooth." OR "Algunas de sus características es Bluetooth."

## 7 Overcoming Limitations

As of now, no limitations exist as intrinsic or conceptual barriers, proven by (1) in use large scale text production in milliseconds, (2) the current 17 languages, and a wide range of topics from personalized communication, live dialogue systems or static text outputs. Practical limits relate to two categories: one being the feasibility in reference to effort by implementing a set of rules by the user to define inference and text output, the second one being the predictability of possible input data errors (Graefe, 2016).

These Limitations are already being solved by integrating the progress from other related fields from the language analysis side and machine translation improvements: An implementation for integrating NLP-based tools like POS-tagging allow for suggesting possible ATML3 rules, reducing the effort and error rate of human rule creation; and machine learning based toolchains for data analysis can be used to predict inference-oriented rules.

## References

Andreas Graefe. 2016. Guide to automated journalism.

Liubov Nesterenko. 2016. Building a system for stock news generation in russian. In *Proceedings of the 2nd International Workshop on Natural Language Generation and the Semantic Web (WebNLG 2016)*.

A. Ramos-Soto, A. Bugarin, S. Barro, and J. Taboada. 2013. Linguistic descriptions for automatic generation of textual short-term weather forecasts on real prediction data. In *International Conference on Flexible Query Answering Systems*.

Ehud Reiter, Robert Dale, and Zhiwei Feng. 2000. *Building natural language generation systems*, volume 33. MIT Press.

Kees Van Deemter, Emiel Krahmer, and Mariët Theune. 2005. Real versus template-based natural language generation: A false opposition? *Comput. Linguist.*, 31(1):15–24, March.