A NLG based error reporting system for production line machines

Frank Feulner, Robert Weißgraeber, Alexander Deutsch, Stefan Lasse
AX Semantics, Stuttgart, Germany
{firstname.lastname}@ax-semantics.com

Abstract

We present a toolchain for generating actionable realtime reports out of the error codes of shopfloor machinery. Output is generated in real-time and several languages. The NLG platform of AX Semantics is used to plan and realize the output.

1 Introduction

As we see industrial production becoming more and more connected and digitized, the need for machines and humans to communicate with each other is becoming more imminent, to enable a collaborative way of interacting. Integrating NLG technology into production lines enables machines to communicate errors and system states not only in a formulaic way (e.g. "error #XXX") or as a warning signal, but in the form of meaningful, actionable and user-individual error descriptions in correct language (Braun et al., 2015). The involved staff will be notified directly or even preventively via mobile devices, and will receive suggestions for repair or mitigation measures, taylored to their skillset and role in the production line. A precondition for this is to not only draw insights from the machines' error messages themselves, but also from continuously analyzing incident data generated by the staff over time. The configuration of AX Semantics is also trained to generate the error reports in several languages (in this example, German, English and Spanish) to improve the communication between machines and non-native German speakers.

2 Example

A big challenge was to integrate a whole production line, which usually contains a very heterogeneous combination of machines, varying strongly in age, ability to communicate and manufacturers. As an example, we used the training factory at the university of Reutlingen and connected several robotic manipulators (Universal Robots UR5 and UR10) as well as several older transport systems, 3D printers and other components to provide the heterogeneous environment as mentioned in figure 1.

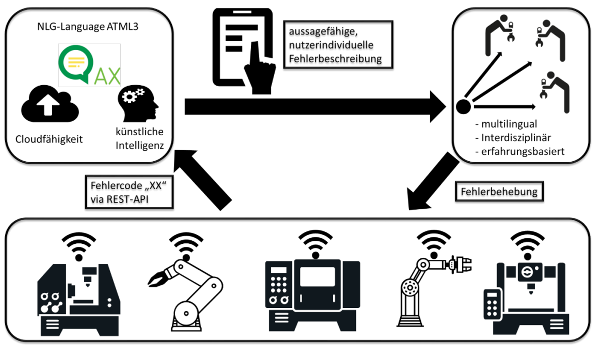

Figure 1: Information flow in a NLG automation for production lines.

3 Solution

To realize the toolchain, we built a database to systematically store possible error codes and the corresponding descriptions. The focus was on aggregating as many different styles of communication and types of errors as possible, to prepare the toolchain for a multitude of usecases. This so-called Virtual Repository is used to configure the AX Semantics platform to recognize and understand error reportings from the example toolchain. The components are connected with a REST-API. A simulator was programmed to facilitate configuring the AX Semantics NLG component. It will also be used to validate the output of the toolchain.

For the NLG component, we used our proprietary AX Semantics' NLG Cloud, which is a NLG webservice based on proprietary technol-

Workshop on Natural Language Generation for Human–Robot Interaction
at INLG 2018, November 8, 2018, Tilburg, The Netherlands.
http://purl.org/net/nlg-hri-workshop-2018/

ogy, providing a web-UI for configuration and a REST-API as well as webhooks for automation. The platform is able to realize stories and answers multilingually, providing up to 30 different languages (including German, English, French, Spanish, Finnic, Arabic, Turkish and Unified Chinese) (Weißgraeber and Madsack, 2017), while being completely topic-agnostic. Since it makes annotations semi-automatically in AX Semantics' NLG configuration language ATML3, based on the dependencies parsed from the user's input (Andor et al., 2016), it requires no technical skills for the person configuring and operating it and can be configured entirely on the customer's side. The whole roundtrip to generate a story takes about 200 ms.

The platform is configured to not only include information about the error itself and knowledge about the difficulty, duration and business impact of the repair, but also to conclude useful measures an recommendations for the staff involved in the repair process. (e.g. what parts are needed, links to the documentation, should the production line be switched off immediately, is it necessary to call the failing device's manufacturer, estimated cost of the incident, etc.)

All of these recommendations are designed to enable the staff to repair the machine as quickly and sustainably as possible.

After a successful repair, the staff is required to fill in a questionnaire to provide additional data about the nature of the defect, the actual procedure to fix it, the needed materials, time invest etc. This information is stored in an additional database and will be provided to the NLG component when a similar error occurs, which drastically improves the usefulness of the reports for a particular error and dynamically increases the amout of data available for the NLG component. Furthermore, additional information like the stock inventory for spare parts, prices, datasheets and repair manuals into the error report.

References

Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. 2016. Globally normalized transition-based neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2442–2452. Association for Computational Linguistics.

Daniel Braun, Ehud Reiter, and Advaith Siddharthan. 2015. Creating textual driver feedback from telemetric data. In *Proceedings of the 15th European Workshop on Natural Language Generation (ENLG)*, pages 156–165. Association for Computational Linguistics.

Robert Weißgraeber and Andreas Madsack. 2017. A working, non-trivial, topically indifferent nlg system for 17 languages. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 156–157. Association for Computational Linguistics.