Using Very Large Scale Ontologies for Natural Language Generation (NLG)

Hermann Bense
textOmatic AG, Dortmund, GERMANY,
hermann.bense@textOmatic.ag

Abstract. Ontology-based natural language generation can be a determinative factor for the digitalization in the publishing, media and content production industry. Based on the technology presented here, in the foreseeable future the amount of generated news will exceed that of news written by human authors. In future, publicly available data in the domains of weather, sports, finance, traffic, events or open data from sources like Wikipedia, dbpedia, YAGO etc. will be combined to create hyper-personalized news streams. Thousands of product descriptions for online shops can be generated as unique texts in many languages day by day.

Keywords. Automated Content, Robot-Journalism, Computer Linguistics, Hyper-Personalisation, Ontology-based Text Generation, SaaS, Scaling in the Cloud

1. Introduction

The publishing industry has a fast increasing demand for unique and highly up-to-date news. Online shops need thousands of product descriptions in multiple languages. Then, there is a vast amount of data steadily raised in the domains of weather, finance, sports, events, traffic, and products. However, there are not enough human editors to write all the stories and descriptions buried in these data. As a result, the automation of text writing is demanded. In Bense, H., Schade, U. (2015), we presented a Natural Language Generation (NLG) approach, which is used for automatic text generation. Data that is available in a structured form like tables or XML/JSON-formats are transformed into news, stories, reports and product descriptions.

2. Automated Text Generation using Ontologies

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). 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 application example is Focus Online (2017), one of the biggest German online news portals. They publish around 30,000 automated weather reports with three days forecasts for each German city every day. Another example for high-speed and high-volume journalisms is Handelsblatt (2017). Based on the data of the German stock exchange, stock reports are generated for the indexes DAX, MDax, SDax and TecDax

every 15 minutes. In each trading day more than 1,000 reports are generated consisting of 300 to 700 words. These reports contain information on share price developments and correlate them to past data like 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 like Google, Bing etc. As a consequence media outlets profit from more page views and revenues from affiliate marketing programs. The finance reports for handelsblatt.com rank top on Google page one for names of companies out of the German DAX (Deutscher Aktien Index) like SAP, e.on, Lufthansa etc., though the lists of search results show tenth of millions entries. For the purpose of SEO (Search Engine Optimization) also meta-tags based on schema.org structures are generated into the documents.

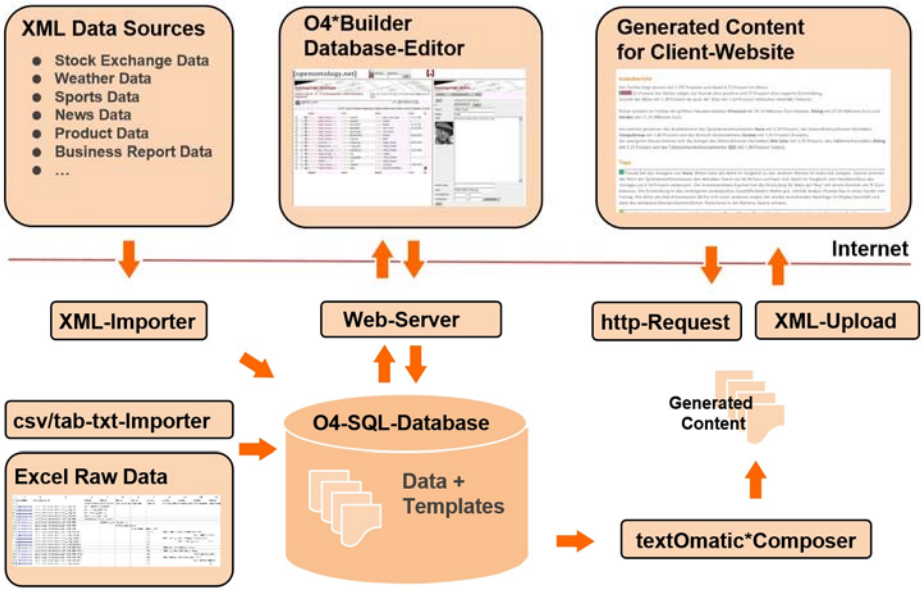

Figure 1: textOmatic*Composer System Architecture and Components

From the mere 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 seconds, and running the text generation tools in Google Cloud Platform or in Amazon AWS adds arbitrary scalability since the majority of the reports can be generated in parallel. The growth rate of the ontology for the generation of stock reports is about 90 million triples per year. The generated reports are provided to the customers mainly in XML-format as Software as a Service (SaaS). The ontologies which we currently have built, contain billions of triples. The excerpt from the English Wikipedia has about 250 million triples.

3. Text Generation Framework and Technology

The textOmatic*Composer (2017) text generation framework is shown in figure 1. The central component is the O4-SQL-database which stores ontology data in triple form

as well as the templates for the text generation. The data for the text generation process is obtained from many sources and mainly provided as XML-streams based on XSD-definitions. Also imports from csv-files and Excel-tables are possible. For each data source we have set up a different ontology following the conceptual modeling and naming conventions defined in Bense, H. (2014). These ontologies can be jointly accessed in one text generation application. Thus, it is possible to enrich the generated content with background information e.g., countries, cities, famous persons, etc. In the case, that complementing data is available from different providers, the conceptual models are amalgamated into domain-specific super-set models, which contain all classes and properties of the different source ontologies. Some providers of weather data supply data for pollination and ozon levels, while others do not. This enables to generate reports with richer and more precise contents, combining data from different sources. Since the internal data model of the O4-SQL databases consist of only one relation which stores the (s,p,o)-triples, changes to the conceptual models of the ontologies can be applied easily on-the-fly.

Our software works on cloud-based server platforms. The generation of texts of several hundred words takes no longer than a few seconds. In the Google and Amazon Clouds we have done benchmarks which show that we can generate with an average rate of several thousand texts per second. There are two different SaaS-methods (Software as a Service) to access the generated texts (Figure 1). Either generated texts can be retrieved by a http-request or they can be pushed with ftp-file transfers to the client site (XML-Upload). Mainly for the generation of product descriptions, customers also use in-house installations of the generation framework.

The Text Composing Language TCL which we developed, is a programming language specially designed for generating natural language texts. A TCL program is called a template and integrates HTML/XML/Javascript and PHP-Syntax. Van Deemter, K., Krahmer, E., Theune, M. (2005) argue, that systems *that call themselves template based can, in principle perform all NLG tasks in a linguistically well-founded way and that more and more actually systems of this kind deviate dramatically from the stereotypical systems that are often associated with the term* ***template***. Templates can be arbitrarily nested like subroutines. The templates and the ontological knowledge is stored in RDF-triple stores which have been implemented in MySQL. The data can be accessed via query interfaces in three different layers of expressive power. The top layer provides a description logic type of 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. MySQL has been chosen as underlying database system, because it is widely used and gives developers also the option to access ontologies with SQL.

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)`**. For instance, **`getObjects(>Pablo_Picasso,*)`** would retrieve all data and object properties of the painter Pablo Picasso. **`getSubjects (.PlaceOfBirth, Malaga`**) would return the list of all subjects, who were born in *Malaga*. Here is the example of a small TCL-program*:*

```
[[LN  = get(>Pablo-Picasso,.LastName,*)]]
[[PoB = get(>Pablo-Picasso,.PlaceOfBirth,*)]]
$LN$ was born in $PoB$.
```

This rather trivial TCL program creates the output: "*Picasso was born in Malaga*". Non trivial TCL programs comprise hundreds of templates including rules for the derivation of conclusions and the generation of instructions.

4. The Hyper-Personalization of News Feeds

The upcoming trend in the media industry is hyper-personalization. The purpose is to create personalized news streams for individuals. Within our Google DNI (2017) funded project 3DNA.agency (2017), an approach has been initiated to offer such a service in multiple languages. A user is immediately informed by e-mail or push news in apps if a specific share exceeds a given threshold, or when the next soccer game of her/his favorite team begins. In addition, she/he is informed about the weather conditions expected during the game and about all traffic jams on the way from home to the stadium. With hyper-personalization publishing companies and news portals will be able to provide their readers new service offerings resulting in a higher customer retention.

5. Demo

In the demo session we present examples for the generation of texts from the domains of finance, weather and sports and explain their impact on SEO. Another important component of the text generation framework is the dictionary ontology, which contains thousands of entries for different languages along with lists of synonyms. Therefore we can also show how ontologies and templates can be cooperatively developed in many languages by different programmers and translators using web-based tools.

References

[1] Bense, H. (2014), The Unique Predication of Knowledge Elements and their Visualization and Factorization in Ontology Engineering, in Pawel Garbacz, Oliver Kutz, 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 B. Humm, A. Reibold, & B. Ege, Corporate Semantic Web. Berlin: Springer.

[3] Van Deemter, K., Krahmer, E., Theune, M. (2005), Real versus Template-Based Natural Language Generation: A False Opposition?, Computational Linguistics, Vol. 31, No. 1, 2005

[4] textOmatic*Composer (2017) Summary of the Text Generation Technology of the textOmatic*Composer http://textomatic.ag/english/Technology/ Accessed 10 Sept 2017

[5] Focus Online (2017) Generated Weather Report with 3 Day Forecast for Dortmund on Sunday, http://www.focus.de/regional/wetter-dortmund-aktuelle-wettervorhersage-und-3-tages-vorschau-10-09-2017_id_6774456.html Accessed 10 Sept 2017

[6] Handelsblatt (2017) Stock Exchange Reports for DAX-Companies (Deutscher Aktien Index) http://www.handelsblatt.com/finanzen/maerkte/marktberichte/ Accessed 10 Sept 2017

[7] 3DNA.agency (2017) Multi-Language Website of the Google Sponsored project Data Driven Digital News Agency http://3dna.news/en/ Accessed 10 Sept 2017

[8] Google DNI (2017) Digital News Initiative: A collaboration between Google and News Publishers in Europe https://digitalnewsinitiative.com/ Accessed 10 Sept 2017