Semantic Web Technology Research Proposal

Team members: Hennie Veldthuis, Rémon Kruizinga, Daphne Groot

October 15, 2017

1 Description of task/Research Question

The aim of this project is to create a program that can generate sentences that accurately convey the content of one or two triple sets. When two triples would be used, the data of the triples will be combined and used to generate a single sentence.

2 Background Literature

The task of generating text based on factual knowledge in the form of triples <object, relation, subject> is known as a difficult task. In previously done research, many different approaches have been used to solve the task. One of the most common approaches is to use a corpus of natural language sentences aligned with knowledge base facts in order to formulate the right text. In the research of Mrabet et al. (2016), the approach was a crowd-sourcing method whereas each contributor made their own sentence simplification of existing corpora, aligned with the factual knowledge. A difficulty of this rule-based approach is the lack of robustness, since it heavily relies on the known facts. A solution that the contributors of the papers use is to train a system that would perform the same task efficiently, which depends on the size of the training dataset.

For our research we are planning to use a rule-based approach and focus on simple sentence structures.

3 Data set

The data used for this final project is made available from the WebNLG challenge (Gardent et al., 2017). and was send to use by our lecturer dr. G. Bouma. There are different sort of files, which has to do with the number of triplets provided per entry. To start, we will only use the data set which consist of one triple per entry. An example of how this data looks like can be seen in Figure 1. After creating a program with the one triple file, the program might be extended to the two triple files.

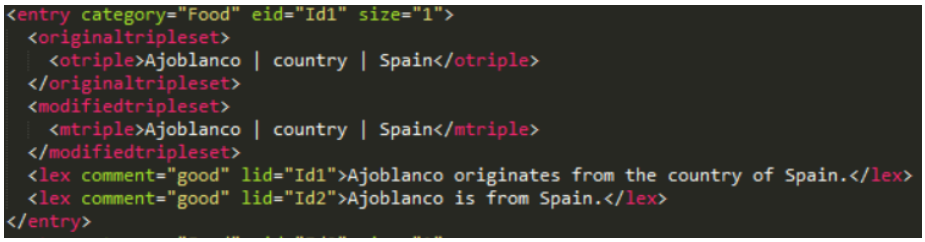

Figure 1: 1 Triple Example

As can be seen in Figure 1, each entry is encapsulated in and <*entry*> tag, which has three attributes: the DBpedia category, the entryID and the size of the triple set. Every entry also has three subsections, tags which fall under the <*entry*> tag. These are <*originaltripleset*>, <*modifiedtripleset*> and <*lex*>.

The <*originaltripleset*> contains a representation of how the triple was extracted from DBpedia, the <*modifiedtripleset*> contains the modefied triples. The modifications mainly consist of clarifying the triples, e.g. by renaiming unclear properties. The <*lex*> tag contains the representations of the natural language text which corresponds to the triples. Each <*lex*> tag is made up from two attributes: a comment and a lexicalisation ID. The comments have by default the value 'good', except for some cases where the value is 'toFix'. This 'toFix' means that the corresponding lexicalisation did not exactly matched a triple set.

In this project, we will first of all focus on the triple sets that belong to the category *Food*, which might later be extended to more categories.

4 Techniques and Tools

The program used to convert the triple sets into plain, readable text will be Python 3. In the Python program the XML-tree functionality will be used to extract all the information needed from the triple sets. The created program can eventually run in a Linux environment, as well as a Windows environment using the Windows Bash.

5 Evaluation

The evaluation will be done in two ways. We first check if the sentences occur in the data set. For each triple set the data set contains three or four examples of correct sentences. (In the xml files these can be found by searching for 'comment="good"'.) If our generated sentences occur in the data set, it is deemed correct. The remaining sentences will be manually evaluated because a sentence may still be correct even though it was not present in the data set. The manual evaluation will be done by three different people. In case of disagreement, the majority wins.

Once all sentences have been assessed, we can calculate the precision, recall, F-score and accuracy to determine how well our program can generate sentences out of triples.

References

Gardent, C., A. Shimorina, S. Narayan, and L. Perez-Beltrachini (2017, August). Creating training corpora for micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada. Association for Computational Linguistics.

Mrabet, Y., P. Vougiouklis, H. Kilicoglu, C. Gardent, D. Demner-Fushman, J. Hare, and E. Simperl (2016). Aligning texts and knowledge bases with semantic sentence simplification.