An Experiment on “Free Generation” from Single RDF triples

Xiantang Sun
Department of Computing Science, University of Aberdeen
Aberdeen, UK, Ab24 3UE
xsun@csd.abdn.ac.uk

Chris Mellish
Department of Computing Science, University of Aberdeen
Aberdeen, UK, Ab24 3UE
cmellish@csd.abdn.ac.uk

Abstract

This paper introduces our domain independent approach to “free generation” from single RDF triples without using any domain dependent knowledge. Our approach is developed based on our argument that RDF representations carry rich linguistic information, which can be used to achieve readable domain independent generation. In order to examine to what extent our argument is realistic, we carry out an evaluation experiment, which is the first evaluation of this kind of domain independent generation in the field.

Introduction

In the Semantic Web, both instance data and ontological data[1] are represented as graphs based on the Resource Description Framework (RDF) (W3C 2004). In order to facilitate non-technician users to access the knowledge and information coded in RDF, we are eventually aiming at developing a domain independent approach to presenting RDF graphs in natural language, which can greatly reduce the cost of applying NLG techniques to various RDF domains (e.g., medical RDF data and chemical RDF data). In this paper we introduce our domain independent approach to generating phrases or sentences from single RDF triples[2] without using any domain knowledge but only generic linguistic knowledge sources. This contrasts with almost all existing work generating natural language from ontologies, which assumes the existence of domain-dependent lexicons. This work is a key part of our final system because generation from single RDF triples, which are the atomic units of RDF graphs, is the foundation and also the first step for any further generation from larger RDF graphs. In order to examine to what extent the linguistic structures can be used to achieve domain independent generation from single RDF triples, we have built a generation system, *Triple-Text (TT)*, and here we compare TT’s generation with human experts’ generation in an evaluation experiment.

1 Linguistic structures in ontologies

Let’s start with an example of a triple, which consists of a subject, a predicate (also known as a property) and an object. The triple *(LongridgeMerlot HasMaker Longridge)*, may be realised as, *LongridgeMerlot has a property ‘HasMaker’ with a value ‘Longridge’,* if this triple is viewed as a pure logical representation. But there could be a much better way of realising it, if the implicit linguistic information embedded in the triple could be correctly recognised and exploited:

Longridge Merlot has a maker, Longridge.

More interestingly, we find that the generation process in fact does not require understanding the real semantics of the terms, *‘Longridge’, ‘Merlot’, ‘has’, ‘maker’* and *‘Longridge’*. That is, the words embedded in the names of these terms could form most of the final sentence and they have been already placed in a sensible way by their creators. This implies that what is required to make the final sentence is to just to fill in “missing” words, e.g., determiners. Obviously, the name of the property in the triple plays the

[1] Ontological languages are developed based on the RDF syntax, so ontological data are still RDF graphs.

[2] Generation from RDF triples in our case means only presenting the information in the triples, rather than explaining the information.

most crucial role in the generation.

As our earlier work (Mellish and Sun 2005) (Sun and Mellish 2006) shows, RDF representations indeed contain rich linguistic information. In our corpus of 882 OWL files which include 37260 class names and 1218 property names, only 14% of class names consist totally of meaningless strings and 97% of the properties' names fully or partially consist of natural words. Our further analysis has shown that the properties may be classified into 6 categories based on their "patterns". 37.2% of the properties start with '*has*', 12.3% start with '*is*', 11.3% end with a preposition, 18.0% are single words, 12.5% have two words, and 8.3% have more than two words. In each category, we further define some specific patterns[3], for instance those shown in figure 1. In total, we define 23 patterns in the six categories.

Patterns of 'has'	**Examples**
'has'+…+noun	*hasColor, hasName*
'has'+…+preposition	*hasExposureTo, hasShapeAnalagousTo*
'has'+…+adj	*hasTimeClose, hasTimeOpen*

Figure 1: examples of the patterns for '*has*'

2 Generating single sentences from RDF triples

Based on the patterns found from the corpus, we developed a system (TT) to generate single sentences from single RDF triples without any domain dependent knowledge. The key idea here is to construct VPs from the properties of input triples and treat the subjects and objects of the triples as domain dependent terms, which are simply seen as proper nouns e.g., *(John Surname Murphy)* à "*John has a surname Murphy.*" So the main part of the system is about constructing proper VPs from properties. In the process of constructing VPs, every property is tokenised into a sequence of units, e.g., *hasEmail* is separated into *has* and *email*. Then, the property is classified into one of our 6 categories, and in this case *hasEmail* belongs to the *'has'* category. In each category, we have a set of rules to construct VPs from input properties. A rule's LHS is a pattern and the RHS is a corresponding linguistic form (VP) for the pattern. For example, we have a rule like,

LHS: 'has' + [unit] + [noun]* à
*RHS: 'has'+det**[4]*+'units'+'noun'*

which can construct a VP, *has an email*, for the property, *hasEmail*. Assuming that TT is given a triple, *(Peter, hasEmail, X@Y.com)*, then the generated single sentence is, *Peter has an email, X@Y.com*, where the subject and the object are treated as proper nouns without any analysis. In the case of input that our rules cannot cover, TT outputs a kind of "RDF flavoured" text, e.g., TT generates, *abg has a property k34 with value 377*, from the triple *(abg k34 377)*.

3 Evaluation

In order to examine to what extent our domain independent generation is realistic in terms of syntax, comprehensibility and overall quality, we carried out an experiment to compare TT's generation with human experts' generation and pure "RDF flavoured" generation (the "RDF generator"), as our baseline (as in the example shown at the end of section 2).

3.1 Experiment design

We applied a "Two-Panel" methodology to compare the three kinds of generation, which is similar to the methodology applied in KNIGHT[5] (Lester and Porter 1997). The Two-Panel evaluation methodology can be used to empirically evaluate NLG generation by comparing computer generation with human generation. We take *Computer Blindness* as a central principle through the experiment in order to guarantee the integrity of the evaluation results. This means that our judges do not know that any texts are generated by computer..

[3] We use QTAG (Mason 2003) to recognise the parts-of-speech (POS) of the units extracted from the property names. For example, QTAG can recognise '*has*' and '*colour*' (from '*hasColour*') as a verb and a noun. We achieved 99.2% accuracy of POS recognition on our corpus using QTAG with the assistance of some manually-added rules.

[4] We simply add an indefinite determiner if the '*noun*' is not in its plural form.

[5] In KNIGHT only the system's generation and human generation are compared.

There are four steps in our methodology:

- randomly selecting 90 triples with different properties and then generating from them with TT and the "RDF generator" (we took the 90 triples from ontologies collected in an knowledge engineer's ongoing project);
- arranging two panels consisting of RDF experts and PhD students whose areas are irrelevant to computing. NB the RDF experts in the first panel did not know the domains that the 90 triples were from, in order to make their generation "domain independent" like TT;
- asking panel 1 to manually generate 90 short sentences from the triples;
- evaluating all generations by panel 2.

3.2 Experiment

The source RDF data for the experiment were collected from 7 domains in order to test TT's performance in general. The input data were not known to us until we started the experiment. 90 different single RDF triples were randomly collected from the data and input to TT and the "RDF generator". We avoided having 2 sentences with the same property. Now we had 90 sentences from TT and another 90 sentences from the "RDF generator". We invited 3 RDF experts (2 PhD students and a Post-Doc) for panel 1, 6 law PhD students for panel 2. Each of the experts in panel 1 was asked to present 30 different triples from the 90 triples in natural language. Panel 2 judged the generation in terms of syntax, comprehensibility, and overall quality. Panel 2 was given mixtures of the generations from TT, the "RDF generator" and the experts, but they were not shown the source triples and did not know that there were computers involved in the experiment. Each judge was given *90* short sentences and asked to judge them in terms of syntax, comprehensibility, and overall quality by choosing between possible options.

- Syntax: we asked *"does the sentence have any grammar mistakes?"* and gave five options, *A) all wrong B) basically wrong C) some mistakes but understandable D) minor mistakes E) no mistakes*
- Comprehensibility: we asked "do you understand what the sentence says?" and gave options, from *not at all, a little bit, some of it, understand most of it,* and *understand it all*
- Overall quality: we asked "*Do you like the way the sentence is written?"* and gave options from *not at all, a little bit, generally ok, good* and *excellent.*

When we distributed these sentences to the judges, we followed the four principles that applied in KNIGHT's evaluation. They are

- System-Human division: Each judge in panel 2 received 90 different sentences from TT, the "RDF generator" and experts in random order (30 of each).
- Domain Division: Each judge in panel 1 and panel 2 received sentences that were approximately evenly divided among the domains which the 90 RDF triples were from.
- Single-generation restriction: No judge in panel 2 received more than one sentence from the same RDF triple.
- Multijudge Stipulation: Each sentence is judged exactly twice in order to obtain relatively unbiased judgements.

3.3 Experiment results

After the experiment, for each triple we had 3 sentences (generated from TT, the "RDF generator" and panel 1) judged by panel 2. Then, we had three samples of quantitative data of the judges' opinions of each dimension by mapping options A-E onto 1-5 (a sample of TT, a sample of the "RDF generator" and a sample of the experts' text). The two-tailed Standard T-test is a good way to detect differences between these samples if the differences exist (as in KNIGHT). We compared TT with the experts, TT with the program and the program with the experts. Here are the results for the means[6] and the differences and their significance[7] (tables 1, 2, 3 and 4).

[6] ± in table1 stands for the standard error.

[7] The t-tests used in our case are unpaired, two-tailed. The results are reported for a 0.05 level of confidence. Significance does not depend on whether we apply a multiple test correction.

Generator	Syntax	Comprehensibility	Overall
RDF gen.	2.44±0.09	2.01±0.09	1.81±0.08
TT	3.14±0.12	2.76±0.12	2.29±0.11
Experts	3.3±0.12	2.88±0.12	2.4±0.11

Table1: Means

RDF gen. VS TT	Syntax	Comprehensibility	Overall
Difference	0.70	0.75	0.48
Significance	3.72E-06	1.89E-06	1.89E-06
Significant?	Yes	Yes	Yes

Table2: Differences and significance

RDF gen. VS Expert	Syntax	Comprehensibility	Overall
Difference	0.86	0.87	0.59
Significance	2.42E-08	1.16E-08	6.26E-06
Significant?	Yes	Yes	Yes

Table3: Differences and significance

TT VS Expert	Syntax	Comprehensibility	Overall
Difference	0.16	0.12	0.11
Significance	0.35	0.47	0.46
Significant?	No	No	No

Table4: Differences and significance

As shown in table 1, experts score the highest and the “RDF generator” scores the lowest in every dimension. TT’s performance is worse than but close to the experts’, however neither of them scores very high. Indeed, both of them score less than 2.5 in overall quality. The reason for the low scores is probably that the data for the test contained many domain dependent terms, which the readers did not understand or felt were “odd”, e.g., *area125*. In syntax and comprehensibility, both experts and TT achieve an “average” level. However, it seems that the readers do not like the “RDF flavoured” text. We talked with the readers about the texts after the experiment and found out that the “odd” domain dependent terms lowered readers’ scores, though the readers understood most of the texts. According to table 2 and table 3, both experts and TT differ significantly from the “RDF generator”. According to table 4, we could not find a significant difference between the experts and TT. This does not indicate that TT is as good as the experts because a bigger sample may show a significant difference. As an example of where TT is not as good as the humans, one of our experts writes “*The industry of the North is manufacturing sector.” from* (*North industryOfArea manufacturing_sector)*, which is more “natural” than TT’s generation, *“North is the industry of area manufacturing sector.”* So we may only say that TT’s performance is to some extent close to the experts. On the other hand, the fact that we were with this sample able to show a significant difference between the other pairs gives us some confidence in the adequacy of TT’s output.

Conclusion

Our corpus analysis and the evaluation experiment show that there is an opportunity to achieve adequate quality domain independent generation from RDF data. Our future work will focus on generating from larger RDF graphs.

Acknowledgements

Our thanks go to EPSRC, who funds our research project through grant GR/S62932/01.

References

J. C. Lester, and B. W. Porter. (1997) Developing and empirically evaluating robust explanation generators: The Knight Experiments. *Computational Linguistics* 23(1):65–101.

O. Mason, (2003). *Qtag 3.1*. Department of English, School of Humanities, University of Birmingham,http://web.bham.ac.uk/O.Mason/software/tagger/

C. Mellish and X. Sun. (2006). The Semantic Web as a Linguistic Resource. *Knowledge Based Systems* 19, pp 298-303.

W3C, (1999). "Resource Description Framework (RDF) Model and Syntax Specification.", http://www.w3.org/TR/PR-rdf-syntax/.

X Sun and C. Mellish. (2006). Domain Independent Sentence Generation from RDF Representations for the Semantic Web. Combined Workshop on Language-Enabled Educational Technology (ECAI'06), Italy.