

# Towards the Domain Agnostic Generation of Natural Language Explanations from Provenance Graphs for Casual Users

Darren P. Richardson and Luc Moreau

Electronics and Computer Science,  
University of Southampton, United Kingdom  
`dpr1g09@ecs.soton.ac.uk, l.moreau@ecs.soton.ac.uk`

**Abstract.** As more systems become PROV-enabled, there will be a corresponding increase in the need to communicate provenance data directly to users. Whilst there are a number of existing methods for doing this — formally, diagrammatically, and textually — there are currently no application-generic techniques for generating linguistic explanations of provenance. The principal reason for this is that a certain amount of linguistic information is required to transform a provenance graph — such as in PROV — into a textual explanation, and if this information is not available as an annotation, this transformation is presently not possible. In this paper, we describe how we have adapted the common ‘consensus’ architecture from the field of natural language generation to achieve this graph transformation, resulting in the novel PROVglish architecture. We then present an approach to garnering the necessary linguistic information from a PROV dataset, which involves exploiting the linguistic information informally encoded in the URIs denoting provenance resources. We finish by detailing an evaluation undertaken to assess the effectiveness of this approach to lexicalisation, demonstrating a significant improvement in terms of fluency, comprehensibility, and grammatical correctness.

## 1 Introduction

As organisations begin to understand the value of storing and utilising PROV data [13], they will increasingly find scenarios where it is useful to show that data to their users. Where resources allow, the best interfaces to this data will likely be bespoke creations, tailored to the specific needs of the application. However, we speculate that in many cases the resources will not be made available to take this approach, motivating the search for an application-generic way of communicating provenance to casual users.

In this vein, there are already a number of different ways for communicating PROV data to human users in formal [14], diagrammatic [5, 17], and linguistic forms [16]. The utility of these various approaches depends on a number of factors but, perhaps, most importantly the user and their familiarity with the intricacies of both PROV and the application context. For example, whilst it is

a very useful tool in a suitable context, it would not be appropriate to use the PROV-N notation to communicate with the vast majority of users. Likewise, the diagrammatic forms of representing PROV are also potentially inaccessible to many users who would perhaps have difficulty understanding mathematical graphs.

A competent speaker of a particular language, on the other hand, is presumably far more likely to understand a well-worded provenance explanation, than understand a diagrammatic representation in a format that they have not previously encountered. Linguistic interfaces are of further use in contexts where a visual interface might be inappropriate, unsafe, or illegal — such as when driving a vehicle.

The main contribution of this work is to extend the state-of-the-art with respect to natural language interfaces for provenance, showing significant improvements in terms of grammatical correctness, fluency, and comprehensibility. This is achieved by introducing techniques from the field of Natural Language Generation (NLG), and using URIs as a source of lexical information. We present an approach to transforming PROV graphs into natural language in an application-generic fashion, resulting in explanations which our evaluation shows are more accessible to a casual user.

After a brief review of related work in Section 2, we introduce the PROVglish architecture we have developed to accomplish this in Section 3, followed by a deeper explanation of how we extract the necessary linguistic information from the PROV graph in Section 4. We then continue by demonstrating the effectiveness of this approach in a formal evaluation in Section 5. Finally, in Section 6, we conclude with a brief summary of this work’s contributions followed by a consideration of possible avenues of future research.

## 2 Related work

PROV was standardised as a recommendation of the World Wide Web Consortium in 2013 [13]. The Provenance Working Group, which developed this recommendation, created a human-readable representation for PROV, PROV-N [14], and suggested a diagrammatic representation [17]. These representations — in particular PROV-N — were only intended for developers and implementors already familiar with the PROV data model, rather than for casual users of provenance. Since then, there has been little research published as to how to present provenance to users in a domain-generic way.

At present, interfaces are either being tailor-made for a particular application, such as on the website of *The Gazette*<sup>1</sup>, or by integrating the diagrammatic representations into the interface, such as in [23]. One notable exception is [5], in which Sankey diagrams are used to represent provenance in a process-centric way; this approach is probably far more accessible to a casual user than the

---

<sup>1</sup> *The Gazette* is the official public record of the United Kingdom. For an example of their provenance trail, see <https://www.thegazette.co.uk/notice/2184651/provenance>.

PROV Working Group’s diagrammatic approach, but is as yet unable to show provenance in an entity- or agent-oriented manner. Finally, in order to help users understand large-scale provenance datasets, an approach has been developed to generate summaries of a PROV graph [15], which can be displayed using a minimally adapted version of the PROV Working Group diagrams.

As for natural language interfaces, there exists a string-substitution templated approach to generating natural language explanations from provenance data [16], but this is application-specific due to the simple nature of the templates. The NLG research community, on the other hand, has developed more sophisticated approaches to generating texts from data. There are a number of NLG architectures described in the literature, with the two most well-known being the ‘consensus’ architecture [19], in particular the refined version [20], and the RAGS architecture [12]. The advantage of such architectures is that they allow for the sharing and reuse of architectural components, such as a realisation engine [4]. Within the field, there are also a number of common ways of evaluating NLG systems [11], often involving the use of humans comparing two example sentences across a number of different metrics [6].

There have been a number of attempts to apply NLG techniques to RDF datasets. Some use an ontology to annotate the linguistic information that is needed to perform the transformation of data to text [10], whilst others use a statistical, corpus-based approach [3]. Finally, there has been an attempt to exploit the linguistic information informally encoded in URIs [22], though this was only able to generate very short texts from up to six triples, due to the fact that there are very few constraints on what sorts of information an RDF document can include. PROV, on the other hand, contains a limited set of relationships, as well as information relating to the temporal ordering of events. Our earlier work [21], presented a technique that was able to choose a near-minimal set of templated sentences in a controlled natural language required to fully transform a provenance graph.

### 3 Generating explanations from provenance

We are aware of only one existing technique capable of transforming provenance graphs into text [16]. However, in that case, the approach taken is template-based, using simple string-substitution, and consequently is only as application-generic as the templates it is based on. Here there are two options: either, create templates that would work for all valid provenance graphs, based on the primitives of PROV; or, alternatively, use templates created for a specific application that can take advantage of the additional knowledge one has about the form and structure of the data to be transformed.

The latter, whilst able to produce the most natural-looking sentences, requires more development and maintenance than a general solution. On the other hand, the former, more general, solution has historically been limited to using sentences like: “*X* was derived from *Y*, which was a revision of *Z*.” Whilst this sort of language *might* be appropriate when talking about documents, in PROV

an entity can be almost anything, and it may not be appropriate to refer to all things using those terms. To illustrate, a person can be a `prov:Entity` (as well as a `prov:Agent`), and it is altogether less usual to refer to people as derivatives of their earlier selves in English, even if this is how they could be represented in a provenance graph.

The existing template-based PROV explainer [16] used a relatively simple string-replacement approach, where the template contained variables that were substituted by values each time the template was expanded. This has the advantage of being simple to implement, and efficient to execute, but makes generating orthographically correct sentences difficult for a number of reasons, such as number agreement or verb conjugation. In order to be able to generate explanations of more complex situations, we decided to explore the use of a more sophisticated template-based architecture drawing on the work of the natural language generation (NLG) research community.

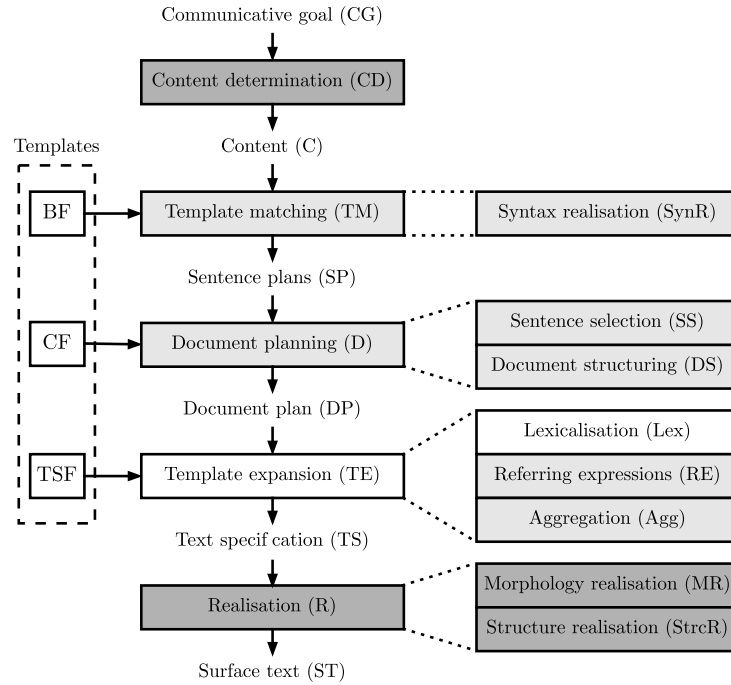


Fig. 1: Provenance explanation generation architecture, PROVglish. Modules are shown in the centre, with the component processes of those modules on the right. The template functions, and their mappings onto modules are shown on the left. Modules in dark grey occur outside of our system; those in light grey occur within our system; and those in white are the focus of the evaluation presented in this paper.

Figure 1 shows the components of our PROVglish architecture, which is based on the ‘consensus architecture’ [19,20]. The consensus architecture, whilst having its detractors [12], is so-called because it presents an attempt to describe how NLG systems are typically constructed, rather than prescribing a model that new systems should adhere to. Our design retains the core pipeline model of this architecture, starting with a communicative goal and resulting in a surface text, but splits and rearranges a number of the intermediate steps to better suit a template-based approach to generation.

One particular advantage of following a common architectural design such as this is that the task can be easily modularised, allowing for the use of off-the-shelf tools. In this case, specifically, it allows us to focus on the areas of NLG that are of particular interest from a provenance perspective, shaded in white in Figure 1. Those blocks shaded in dark grey, on the other hand, represent parts of the process that are not directly contributing to this research. For example, we were able to use a pre-existing off-the-shelf realisation engine (R) [4], capable of taking text specifications (TS) and producing orthographically-correct surface texts (ST). The first stage in the pipeline, content determination (CD), is where the application decides what provenance to communicate to the user. Because this will vary from application to application, we leave that to individual implementers, with our system providing all the stages from template matching (TM) to template expansion (TE).

### 3.1 Templates

In the PROVglish architecture templates are comprised of three functions:

*Bindings function (BF)* A function that takes the graph and returns a set of sets of bindings. Each set of bindings corresponds to a possible expression of a part of the graph in textual form — a possible sentence. The bindings themselves are values mapped to variable names that will later be used by the coverage function (CF) and text specification function (TSF). Because of the many possible combinations of sentences that could be used to express a graph, many more sets of bindings are generated than are actually necessary to do this.

*Coverage function (CF)* A function that returns the subgraph that can be inferred from, or is expressed by the template and a set of bindings — that is, what subgraph would be transformed into text by this template with this set of bindings. Using these coverage sets, we are able to ensure that every explanation generated by our system expresses the entire provenance graph that was passed to it.

*Text specification function (TSF)* A function that takes a set of bindings and returns the sentence as a text specification (TS). This is done by simple template string substitution, but unlike the existing technique allows the realisation engine to handle the much harder task of ensuring that the sentence is orthographically correct.

### 3.2 The process of explanation generation

*Template matching (TM)* In this stage, the bindings function of each template registered in the system is executed over the content provenance graph (C), returning a set of bindings for each pattern in the graph matched by the template. We call the combination of a template and a set of bindings a sentence plan (SP), because together they represent a possible sentence that can be expressed from the content graph (C).

*Document planning (D)* This stage is separated into two parts, sentence selection (SS) and document structuring (DS). Sentence selection is the process by which the list of sentence plans (SP) is whittled down to a smaller list; ideally one that is capable of completely expressing the graph with a minimal number of sentences. This is an example of the set-cover problem, and is solved using the greedy algorithm described in [21]. Document structuring (DS) is the process by which the chosen sentences are ordered into a coherent structure, either using a mathematical metrics-based approach, or by using a narrative-based approach such as Rhetorical Structure Theory [8] to guide the structuring.

*Template expansion (TE)* comprises three aspects of NLG, though in this work we only focus on the first two, leaving Aggregation (Agg) for further work. Lexicalisation (Lex) and referring expression generation (RE), on the other hand, form one of the most important contributions of this work. Lexicalisation, which will be discussed in more detail in Section 4 is the process by which the system decides how each concept in the provenance graph should be mapped onto English words in a general sense, while RE determines how a concept should be referred to in a particular context. For example, a person will commonly be lexicalised by their name, whereas in many sentences it is more appropriate to refer to them by a description, by their contextual situation, or by using some form of anaphoric reference. By choosing these terms, and substituting them into the template, we are able to generate text specifications to be passed into the realiser (R).

## 4 Extracting linguistic information from URIs

One of the major obstacles that is necessary to overcome when generating explanations from data is the matter of obtaining the linguistic information necessary to perform lexicalisation. More simply put: how does one know what to call anything? In RDF, there are ways to formally annotate this linguistic information [10], or one can even encode it less formally in features such as `rdfs:label`. However, making use of these features for explanation generation would not be application-generic, because these features are not guaranteed to be present in all PROV data. The only features that might contain linguistic information that are guaranteed to be present in a PROV dataset are the URIs that denote each resource.

It is worth noting that URIs, per the RFCs that define them, formally contain no linguistic information that would facilitate natural language generation, as they are intended to be treated opaquely [1]. However, many system developers have created systems that mint meaningful URIs, for a number of possible reasons, such as increasing code maintainability. This means that if one were able to understand how this linguistic information is often informally encoded in URIs, then it would be possible to exploit this information for the purposes of natural language generation. We are not the first to attempt to use URIs in this way [22], but previous attempts were met with limited success due to the fact that there are far fewer constraints on what an RDF document can say, and how it can say it. PROV, on the other hand, is a much smaller domain, limited by a number of constraints, leading us to hypothesise that we might have a greater degree of success. Our investigation has shown this to be the case.

We tested this hypothesis, using the University of Southampton Provenance Store, a PROV repository, as our source of provenance data. From these documents we extracted the URIs denoting all the PROV resources to use as our corpus of PROV URIs. Using this corpus, we were able to develop a regular expression that allowed us to split each URI into its linguistic tokens — this is not as trivial a task as might be expected, as there are a number of approaches people use to compose tokens, with CamelCase and snake\_case as just two examples. The expression we settled upon was able to correctly tokenise 96% of the URIs (2525 out of 2637 distinct URIs). The remaining 4% would have required a more complex system with an understanding of the English language. The regular expression we used was:

```
[0-9a-fA-F]{10,} | (?:(Mc|Mac)?[A-Z][a-z]+ | [A-Z]+s?(?![a-z]) | [a-z]+ | [0-9]+)
```

This expression is able to distinguish tokens that fall into the following categories:

- Hexadecimal numbers of at least 10 characters (this limit was introduced to prevent splitting English words like ‘feedback’, which is comprised mainly of the characters ‘a’ to ‘f’);
- Words beginning with a capital letter, including those with common Scottish name prefixes;
- Acronyms and their plurals;
- Lower case words; and
- Decimal numbers.

Having developed a technique to tokenise the URIs, it became necessary to understand the linguistic role played by each of the tokens. We used an off-the-shelf part-of-speech (POS) tagger to tag the URI tokens according to which role they would play in a sentence — for example, whether the token was a 2nd person present tense verb, or a superlative adjective, etc. Specifically, we used the maximum entropy POS tagger [18] trained on the Penn Treebank corpus [9], which came as the default tagger of the NLTK python library [2]. It was important to verify that the tagger would perform well with tokenised URIs because it was originally trained on standard English texts, and there was no

guarantee that it would work with the much shorter, less grammatically correct URI tokens.

However, the performance of the POS tagger was surprisingly good, with it being able to identify the correct specific tag (singular proper noun, present-tense 3rd person verb, superlative adjective, etc.) 62.7% of the time, and the correct class of tag (whether the token was a noun, verb, adjective, modifier, or a number) 92.3% of the time. This level of performance indicated that it would be possible to build generation rules using at least the classes of tags, if not the specific tags themselves.

## 5 Evaluation

Having used the knowledge gained in the investigations described in Section 4 to build templates that were able to exploit the linguistic information in URIs, we devised a human experiment, based on common examples in the literature [6,11] to evaluate the effectiveness of this approach relative to a system where this information was not exploited. Consequently, this experiment is centred around a number of sentence pairs generated automatically, where each sentence pair consists of one sentence generated using the URI lexicalisation technique, and an equivalent sentence that was generated without that additional linguistic information. For each sentence pair, the participant was asked to rank them across a number of dimensions: grammatical correctness, fluency of the language, and ease of comprehension. It should perhaps be noted that the KNIGHT study [6] had additional dimensions, but these were excluded because they relate to the structure of an explanation, whereas in this evaluation we were focussing on the performance of individual sentences within a document.

It was our hypothesis that the system exploiting the linguistic information in URIs should have performed better in terms of fluency and comprehensibility. However, it was our expectation that there would be no significant difference in terms of grammatical correctness, because the sentences that were generated without exploiting the linguistic information in URIs were guaranteed to generate grammatically correct sentences. (This is a consequence of the way the templates were constructed.)

We had 15 participants, each comparing 15 sentence pairs, resulting in the collection of 225 individual data-points for each condition (URIs exploited / URIs unexploited), and for each dimension (Grammatical correctness, Fluency, and Comprehension). Participants were not given definitions for each of the dimensions, but rather were asked to answer the questions with respect to their own perceptions of what those terms meant. The participants were drawn from our Electronics and Computer Science department, and the majority were unfamiliar with PROV or provenance annotation.

Table 1 shows the results of these experiments, and Figures 2a, 2b, and 2c clarify the distribution of responses. In addition to those results, when asked simply which sentence they thought was the better explanation, participants



	Grammar		Fluency		Comp.		Combined	
Metric	Exp.	Unexp.	Exp.	Unexp.	Exp.	Unexp.	Exp.	Unexp.
Mean	4.72	4.36	4.65	3.59	4.74	4.13	4.70	4.03
Standard Deviation	1.49	1.28	1.49	1.35	1.48	1.42	1.49	1.39
Difference	+0.36		+1.06		+0.61		+0.67	
<i>p</i> Value	$4.8 \times 10^{-5}$		$2.5 \times 10^{-15}$		$7.1 \times 10^{-7}$		—	
N	225						675	

Table 1: The results of the study across all sentence pairs and participants, broken down by dimension (Grammar, Fluency, and Comprehensibility), and further broken down by whether the linguistic information in the URIs was exploited or unexploited.

indicated the sentence with the URIs exploited 56.5% of the time, URIs unexploited 29.3% of the time, and neither the remaining 14.2% of the time.

We used a Mann-Whitney U-test [7] to determine the statistical difference between the condition where the linguistic information in URIs was exploited and the condition where it was not. Across all sentence pairs, and all participants, the system that exploited the linguistic information in URIs performed significantly better in all three dimensions than the system that did not. See figures 2a, 2b, and 2c. In each case, one can clearly see the difference in distributions from the graphs.

There are a couple of interesting points to be drawn from these results, that cannot be seen from the figures alone. Firstly, not a single participant gave significantly worse ratings for the sentences generated exploiting URIs, across any dimension, when averaged across all sentence pairs.

Additionally, the only sentence pair to perform significantly worse across all three dimensions was pair 8, where the system was confused by the fact that the word ‘step’ can be both a verb and a noun, resulting in a text specification that was further misinterpreted by the realiser, and where the following two sentences were generated:

*URIs exploited:* `Vote 1043 0 was executioned step at 2011-12-18T01:00:17+00:00.`

*URIs unexploited:* `‘/data/UpVote1043.0’ was generated by ‘/data/ExecutionStep652’ at 2011-12-18T01:00:17+00:00.`

By contrast, one of the sentence pairs that performed best was pair 12, with the following two sentences:

*URIs exploited:* `Derek illustrated chart 1.`

*URIs unexploited:* `‘/derek’ generated ‘/chart1’ by ‘/illustrate’.`

This example clearly demonstrates the impact of being able to extract linguistic information — and in particular, verbs — from URIs on the richness of the explanation generated.

With respect to our hypotheses, the data proves our expectations correct with only a single exception: we had anticipated that there would be no significant

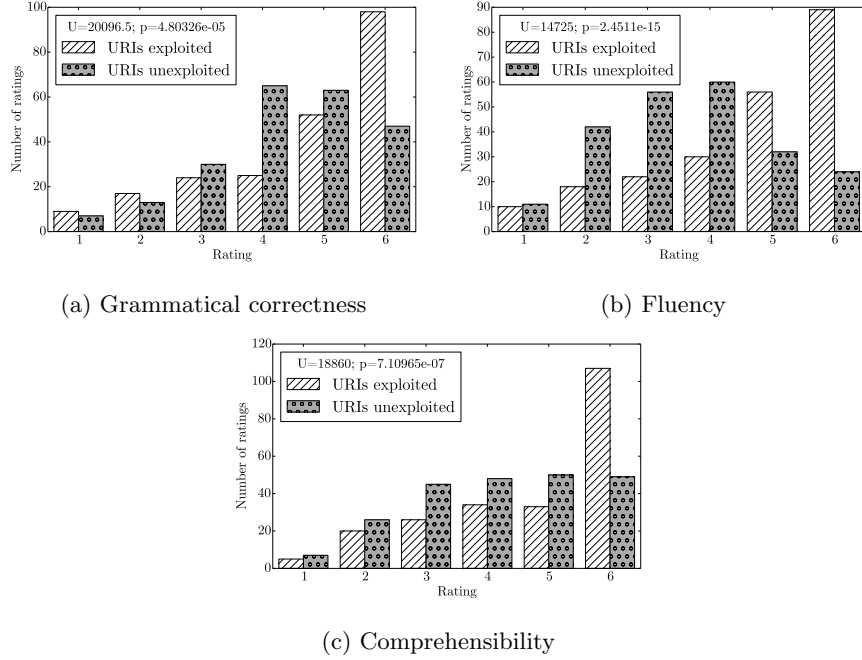


Fig. 2: Participant responses for each dimension, aggregated across all sentence pairs and participants. Legend shows U-value and p-value obtained from Mann-Whitney U Test.  $N = 225$

difference between the grammatical correctness of the sentences produced. In this regard, it would appear that our participants had a different expectation to us as to what makes a sentence grammatically correct. However, even in this case, the system that exploited the linguistic information in URIs still outperformed the system that did not, further validating the main thesis of this paper — that exploiting the linguistic information in URIs allows for the production of better explanations of provenance. With respect to that, the data is supportive on all counts.

## 6 Conclusions and Future Work

In this paper, we have shown how the use of more sophisticated architectures for natural language generation can be applied to the task of explaining provenance graphs to casual users. In addition to facilitating richer explanations at a lower development cost by using off-the-shelf components for surface realisation, when combined with the exploitation of linguistic information informally encoded in URIs, we were able to generate explanations that performed significantly better in a user evaluation. Our approach is application-generic, and can work with all valid provenance graphs.

At present, our system is only capable of generating single sentence explanations. However, many of the provenance graphs we might wish to communicate with a user are considerably larger than can be reasonably transformed into a single sentence. Consequently, we are now investigating the possibility of generating longer, multi-sentential explanations from larger PROV graphs, as well as from PROV graph summaries [15]. We are exploring the potential role of various narrative theories to the application of structuring these longer texts, with a goal of generating more engaging texts than can be achieved with more conventional, graph-metric-based approaches.

## Acknowledgements

Research was sponsored by US Army Research laboratory and the UK Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the U.S. Government, the UK Ministry of Defence, or the UK Government. The US and UK Governments are authorised to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon. The investigations and human experiment were subject to ethics approvals ERGO-FPSE-16722 and ERGO-FPSE-16731, and the source data used to generate the sentence pairs was drawn from the Southampton Provenance Store (<https://provenance.ecs.soton.ac.uk/store>). The research data can be found at <http://dx.doi.org/10.5258/SOTON/393255> and <http://dx.doi.org/10.5258/SOTON/393257>.

## References

1. Berners-Lee, T., Universal Resource Identifiers – Axioms of Web Architecture, technical note, World Wide Web Consortium, (1996). Available at: <https://www.w3.org/DesignIssues/Axioms.html>
2. Bird, S., Loper, E., and Klein, E., Natural Language Processing with Python. O'Reilly Media Inc, (2009).
3. Ell, B., and Harth, A., A language-independent method for the extraction of RDF verbalization templates, proceedings of the 8th International Natural Language Generation Conference, Philadelphia, PA, USA, (2014).
4. Gatt, A., and Reiter, E., SimpleNLG: A realisation engine for practical applications, proceedings of the 12th European Workshop on Natural Language Generation, Athens, Greece, pp. 90–93, (2009).
5. Hoekstra, R., and Groth, P., PROV-O-Viz — Understanding the Role of Activities in Provenance, in Provenance and Annotation of Data and Processes, Lecture Notes in Computer Science, Ludscher, B., and Plale, B., Eds, Springer, (2015).
6. Lester, J. C., and Porter, B. W., Developing and Empirically Evaluating Robust Explanation Generators: The KNIGHT Experiments, Computational Linguistics, vol. 23, no. 1, pp. 65–101, (1997).
7. Mann, H. B., and Whitney, D. R., On a Test of Whether One of Two Random Variables is Stochastically Larger than the Other, Annals of Mathematical Statistics, vol. 18, no. 1, pp. 50-60, (1947).

8. Mann, W. C., and Thompson, S. A., Rhetorical Structure Theory: Toward a functional theory of text organization, *Text*, vol. 8, no. 3, pp. 243–281, 1988.
9. Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A., Building a Large Annotated Corpus of English: The Penn Treebank, *Computational Linguistics*, vol. 19, no. 2, pp. 313–330, (1993).
10. McCrae, J., Spohr, D., and Cimiano, P., Linking Lexical Resources and Ontologies on the Semantic Web with Lemon, in *The Semantic Web Research and Applications, Lecture Notes on Computer Science*, pp. 245–259, (2011).
11. Mellish, C., and Dale, R., Evaluation in the context of natural language generation, *Computer Speech & Language*, vol. 12, no. 4, pp. 349–373, (1998).
12. Mellish, C., Scott, D., Cahill, L., Paiva, D., Evans, R., and Reape, M., A Reference Architecture for Natural Language Generation Systems, *Natural Language Engineering*, vol. 12, no. 1, (2006).
13. Moreau, L., Missier, P.: PROV-DM: The PROV Data Model. Recommendation of the World Wide Web Consortium, (2013). Available at: <http://www.w3.org/TR/prov-dm>.
14. Moreau, L., and Missier, P., PROV-N: The Provenance Notation. Recommendation of the World Wide Web Consortium, (2013). Available at: <http://www.w3.org/TR/prov-n>.
15. Moreau, L., Aggregation by provenance types: A technique for summarising provenance graphs, proceedings of Graphs as Models 2015 (An ETAPS’15 workshop), in *Electronic Proceedings in Theoretical Computer Science*, pp. 129–144, London, UK, (2015).
16. Packer, H., and Moreau, L., Sentence Templating for Explaining Provenance, proceedings of the 5th International Provenance and Annotation Workshop, *Lecture Notes in Computer Science*, vol. 8628, pp. 278–280 Cologne, DE, (2014).
17. PROV Working Group, PROV Graph Layout Conventions, technical note, World Wide Web Consortium, (2011). Available at: <https://www.w3.org/2011/prov/wiki/Diagrams>.
18. Ratnaparkhi, A., A Maximum Entropy Model for Part-Of-Speech Tagging, proceedings of the Conference on Empirical Methods in Natural Language Processing, New Brunswick, NJ (1996).
19. Reiter, E., Has a Consensus NL Generation Architecture Appeared, and is it Psycholinguistically Plausible?, proceedings of the The Seventh International Workshop on Natural Language Generation, Kennebunkport, ME, pp. 163–170, (1994).
20. Reiter, E., and Dale, R., *Building Natural Language Generation Systems*. Cambridge University Press, (2000).
21. Richardson, D. P., Moreau, L., and Mott, D., Beyond the graph: Telling the story with PROV and Controlled English, proceedings of the 2014 Annual Fall Meeting of the International Technology Alliance, Cardiff, UK, (2014).
22. Sun, X., and Mellish, C., Domain Independent Sentence Generation from RDF Representations for the Semantic Web, proceedings of the Combined Workshop on Language-Enabled Educational Technology and Development and Evaluation of Robust Spoken Dialogue Systems, Riva del Garda, Italy, (2006).
23. Toniolo, A., Wentao Ouyang, R., Dropps, T., Oren, N., Norman, T. J., Srivastava, M., Allen, J. A., de Mel, G., Sullivan, P., Mastin, S., and Pearson, G., Assessing the credibility of information in collaborative intelligence analysis, proceedings of the 2014 Annual Fall Meeting of the International Technology Alliance, Cardiff, UK, (2014).