1 Introduction

- What is the subject of the study? Describe the economic/econometric problem.
- What is the purpose of the study (working hypothesis)?
- What do we already know about the subject (literature review)? Use citations: *Gallant [4] shows that... Alternative Forms of the Wald test are considered [3].*
- What is the innovation of the study?
- Provide an overview of your results.
- Outline of the paper:

  The paper is organized as follows. The next section describes the model under investigation. Section **??** *reviews the up-to-date research works. and Section 7 presents the results. Finally, Section 9 concludes.*
- The introduction should not be longer than 4 pages.

More and more the usage of RDF is increasing in many fields in computer science. RDF data representation helps in supporting the machine to perfrom the normally manual computation work in an automatic fashion. Moreover, the machines will be smarter to understand the data which is represented in RDF format.

The quality of RDF data needs to be ensure before proceeding of any further processing. Most of current parsers which focus on detection of the syntax error fail to detect more than one error, especially, of RDF data represented in Turtle or NTriples format.

This study was encouraged by the tremendous data representation of either Turtle and NTriples. Hence, the intention of the study to afford a user-friendly syntax checker or parser. Such parser or syntax checker should give all errors can be detect inside such data.

The thesis is organized as follows. The next section presents the problem description. Section 4 reviews the up-to-date research works. Section 6 describes the data set and Section 7 presents the results. Finally, Section 9 concludes

2 Motivation

This study was motivated by various scenarios which require verifying of valid RDF input and ensuring of RDF data quality. To mention one of them, let's consider an example where there is a collaboration system for machine learning exists. Of course, a valid data input to the system must be verified for further data processing. Most of the current existing systems which ensure syntax-error-free RDF data stop parsing at the first syntax error occurrence, as will be followed in Section 4.

This behavior of limiting the continuation of parsing when the first syntax error is found complicates our scenario. Assuming, the input RDF data contains, for example, 10 syntax errors. Normally what is happing, if an error found, it should be corrected by the user, then after correction data will be send back for re-verification. Imagine that the user will do such process for 10 times. what if then data contains hundreds of errors.

In seeking of finding a suitable solution for the explained scenario, this study has produce a software program that can detect almost syntax errors found in the input.

3 Problem Description

4 Literature Review

In order to validate RDF code, either by pasting URL where it exists or by uploading a file, almost the available tools and applications that we could find, will only give the first occurred error. Moreover, semantic developers and engineers will struggle in debugging their codes and they need alternative tools that could be more helpful. To the best of our knowledge, there is no comparable prior work regarding fault-tolerant tool to validate syntactically RDF serialization formats expect one that works only for RDF/XML format, the following text sheds light on this tool. The new proposed tool should feature prominently in listing of all errors included in the code.

This section reviews the related research works have been done and presents the current state of the art of RDF syntax validating. Despite the long record of RDF syntax validation research with many of theoretical models or practical tools, we can hardly find a research that describes the challenge of detection multiple syntax errors inside RDF code. During our journey of checking the existing tools that provide such validation service, The W3C RDF validation tool [2] was firstly checked, it is available online for parsing and validating RDF/XML codes. It uses the ARP parser of Jena [5] as a backend. However, it fails in detection of multiple syntax errors, the first error in the order will be only released. K. Tolle developed a Validating RDF Parser (VRP) [9] in his thesis, VRP is a Java-based build tool, and it validates RDF/XML code semantically and syntactically. Nevertheless, the validation service provided by VRP is limited to RDF/XML and does not support other RDF serialization formats, especially those formats which are structured in triples such as N3, NTriple, and Turtle. In this work, extension of syntax validation to other formats is planned.

The journey to check the existing tools that validate RDF serialization formats other than RDF/XML is continued. As previously stated, Jena RDF toolkit [5] offers validation service based on ARP parser. It can be used as a command-line program (standalone) or as an API within another application. Despite its ability to validate numerous RDF serialization formats, including RDF/XML, again, the first error is only reported. Some of the tools validating RDF formats use the following core techniques as a significant part of their implementations:

- **ARP-parser-dependable approach :** both W3C RDF validation tool [2] and RDF Validator and Converter [1] use ARP parser of Jena [5]. Moreover, the latter focuses more on triple-based serialization formats, validating them and converting from one format to another, where the former validates only RDF/XML format.

- **N3-parser-dependable approach :** N3 parser can also be used for syntax validation. In the online IDLab Turtle Validator [8], N3 parser powered by N3 NodeJS library is used. As well, same approach was used to build a turtle editor with syntax validation in [6].
- **Shape expressions approach :** in [7] a turtle parser was developed based on shape expressions. Shape expressions validates RDF through declaring of constraints on the RDF model, if the declared constraints are violated, then RDF is invalid. Furthermore, Shape expressions describes the RDF graph on regular expressions base.

When it comes to the application side, N3-parser-dependable approach can be fitted perfectly in the new tool, since it is built using Nodejs library. This can improve the performance of the tool, especially when it is validating a large RDF code. Moreover, the first two approaches are more expressive in explaining the syntax error and its location, where as, the tool used the third approach is less expressive.

In this research, our intention goes toward inventing a fancy tool that lists all syntax errors with an improved performance. The proposed tool can have a solution for the explained issue in either two ways:

- **Patching the output errors of parsers :** while reviewing the source codes of others' tools, an error event by an error handler will be emitted to show the first occurred error. An idea of looping inside the RDF code and fixing eachtime the first error can be suggested. Fixing the error can be by either deleting the triple made the error, removing or adding a punctuation, inserting a dummy IRI for an incorrect or missing one, etc, then reprase the RDF code again and again till the end of the code .
- **Parser Optimization :** this needs to review deeply the whole code of the parser and improve its method. The improvement should list all syntax errors that the parser can detect. Both parsers built with N3-parser-dependable approach or Shape expressions approach can be optimized to reach our goals while the optimization of the latter inherits more complexity.

To end this section, after describing the actual issue, reviewing the state of art of research works related to it, and finally presenting the possible solutions, we can say that both two solutions can solve the issue, but it seems to us that the second solution more efficient than the first, since this is the normal way how acually most of editors of programming languages work, to alert on-the-fly syntax errors to the programmer, even before compilation .

5 Method/Model/Theory

- How was the data analyzed ?
- Present the underlying economic model/theory and give reasons why it is suitable to answer the given problem.
- Present econometric/statistical estimation method and give reasons why it is suitable to answer the given problem.
- Allows the reader to judge the validity of the study and its findings.
- Depending on the topic this section can also be split up into separate sections.

The collection of the data to be used for the evaluation of the suggested approach were collected from different site. The random selection of data helps to reduce the introduced bias.

The coming days produce more and more data represented in different format. The focus of this study is on data of RDF triples. Before any further processing of data, the data quality must be ensured and verified. If no such verification is performed, it can leads to misleading result which affects decision taken by decision-makers, either or human of machines.

Several tools were founded in regards to checking syntax errors in RDF data, few of them as was mentioned in 4 proposed a formal procedure to find one than one syntax error rather than implement such procedure in their tools. The result of this study not only provide a formal process of searching syntactic errors of RDF code, rather

6 Data

- Describe the data and its quality.
- How was the data sample selected?
- Provide descriptive statistics such as:
  - time period,
  - number of observations, data frequency,
  - mean, median,
  - min, max, standard deviation,
  - skewness, kurtosis, Jarque–Bera statistic,
  - time series plots, histogram.
- For example:

	3m	6m	1yr	2yr	3yr	5yr	7yr	10yr	12yr	15yr
Mean	3.138	3.191	3.307	3.544	3.756	4.093	4.354	4.621	4.741	4.878
StD	0.915	0.919	0.935	0.910	0.876	0.825	0.803	0.776	0.768	0.762

Table 1: Some descriptive statistics of location and dispersion for 2100 observed swap rates for the period from February 15, 1999 to March 2, 2007. Swap rates measured as 3.12 (instead of 0.0312). See Table **??** in the appendix for more details.

- Allows the reader to judge whether the sample is biased or to evaluate possible impacts of outliers, for example.

7 Results

- Organize material and present results.
- Use tables, figures (but prefer visual presentation):
  - Tables and figures should supplement (and not duplicate) the text.
  - Tables and figures should be provided with legends.

    Figure 1 shows how to include and reference graphics. The graphic must be labelled before. Files must be in `.eps` *format.*

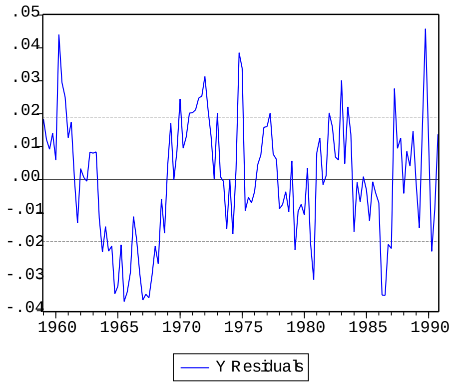

Figure 1: Estimated residuals from model XXX. ...

  - Tables and graphics may appear in the text or in the appendix, especially if there are many simulation results tabulated, but is also depends on the study and number of tables resp. figures. The key graphs and tables must appear in the text!

- Latex is really good at rendering formulas:

  Equation (1) represents the ACs of a stationary stochastic process:

$$f_y(\lambda) = (2\pi)^{-1} \sum_{j=-\infty}^{\infty} \gamma_j e^{-i\lambda j} = (2\pi)^{-1} \left(\gamma_0 + 2 \sum_{j=1}^{\infty} \gamma_j \cos(\lambda j) \right) \qquad (1)$$

  where $i = \sqrt{-1}$ *is the imaginary unit,* $\lambda \in [-\pi, \pi]$ *is the frequency and the* γ_j *are the autocovariances of* y_t.

- Discuss results:
    - Do the results support or do they contradict economic theory ?
    - What does the reader learn from the results?
    - Try to give an intuition for your results.
    - Provide robustness checks.
    - Compare to previous research.

8 Evaluation

9 Conclusions

- Give a short summary of what has been done and what has been found.
- Expose results concisely.
- Draw conclusions about the problem studied. What are the implications of your findings?
- Point out some limitations of study (assist reader in judging validity of findings).
- Suggest issues for future research.

References

[1] RDF Validator and Converter @ONLINE . URL http://rdfvalidator.mybluemix.net/.

[2] W3C RDF validation Service @ONLINE . URL http://www.w3.org/RDF/Validator/.

[3] T. S. Breusch and P. Schmidt. Alternative forms of the Wald test: How long is a piece of string. *Communications in Statistics, Theory and Methods*, 17:2789–2795, 1988.

[4] A. R. Gallant. *Nonlinear Statistical Models*. John Wiley & Sons, New York, 1987.

[5] Brian McBride. Jena: A semantic web toolkit. *IEEE Internet Computing*, 6(6):55–59, November 2002. ISSN 1089-7801. doi: 10.1109/MIC.2002.1067737. URL http://dx.doi.org/10.1109/MIC.2002.1067737.

[6] Niklas Petersen, Gökhan Coskun, and Christoph Lange. TurtleEditor: An ontology-aware web-editor for collaborative ontology development.

[7] Eric Prud'hommeaux, Jose Emilio Labra Gayo, and Harold Solbrig. Shape expressions: an rdf validation and transformation language. In *Proceedings of the 10th International Conference on Semantic Systems*, pages 32–40. ACM, 2014.

[8] Miel Vander Sande. IDLab Turtle Validator @ONLINE . URL http://ttl.summerofcode.be/.

[9] Karsten Tolle. Analyzing and Parsing RDF. Master's thesis, 2000.