

1 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 except 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 [3] 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) [7] 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 [3] 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 [3]. 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 [6], N3 parser powered by N3 NodeJS library is used. As well, same approach was used to build a turtle editor with syntax validation in [4].
- **Shape expressions approach :** in [5] 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 rephrase 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 actually most of editors of programming languages work, to alert on-the-fly syntax errors to the programmer, even before compilation .

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] 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>.
- [4] Niklas Petersen, Gökhan Coskun, and Christoph Lange. TurtleEditor: An ontology-aware web-editor for collaborative ontology development.
- [5] 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.
- [6] Miel Vander Sande. IDLab Turtle Validator @ONLINE . URL <http://ttl.summerofcode.be/>.
- [7] Karsten Tolle. Analyzing and Parsing RDF. Master’s thesis, 2000.