ConceptRDF: An RDF presentation of ConceptNet knowledge base

Erfan Najmi*, Zaki Malik*, Khayyam Hashmi* and Abdelmounaam Rezgui†
*Department of Computer Science, Wayne State University, Michigan, USA
E-mail: {erfan,khayyam,zaki}@wayne.edu
†Department of Computer Science & Engineering, New Mexico Tech, New Mexico, USA
E-mail: rezgui@cs.nmt.edu

***Abstract*—The exponential growth of information on the Web has prompted the introduction of new technologies such as Semantic Web and Common Sense knowledge bases. While there are different models available to present information, RDF, as the cornerstone of Semantic Web technologies, has a dominant place for formats suitable both for human interactions and machine understanding. In this paper we present ConceptRDF, a conversion of ConceptNet, as one of the largest common sense knowledge bases available for public use, to RDF/XML format, suitable for use in different fields of information retrieval.**

I. Introduction

Semantic Web paradigm, has gained a considerable amount of traction in recent years. On the other hand, the expansion of internet, without an agreed upon mechanism to organize and retrieve information has introduced different approaches for organizing and retrieving information. To simplify the retrieval process from the textual body of the Web, many research groups have introduced different knowledge bases which have gained popularity based-on the extent of information presented in them, the simplicity of information retrieval from them and the maintenance and support which has been provided for them.

An important issue which prohibits the flow of information to different knowledge bases is lack of consensus in their syntaxes and the diversity of their information presentation. One of the solutions for the first part, as many research groups have worked on, is to create a unified upper level ontology, which provides simplistic syntaxes and follows clear rules for presenting relationships and entities in any knowledge field. The other approach is to convert different knowledge bases to semantic Web format using different ontologies to add more functionalities which can make use of these knowledge bases in semantic Web field. In this paper we follow the second approach to present our methodology and steps for converting ConceptNet [12] [21], one of the biggest common sense knowledge bases available, to RDF/XML [11] format. We believe that RDF model, as the cornerstone of Semantic Web, combined with other related technologies (such as SPARQL [19] for querying over RDF, OWL [13] to create and unify ontologies) can be the technology to be used to integrate all the data available in different knowledge bases and create a unified source of information. It is noteworthy that while we present RDF/XML format, it is a straightforward process to convert it to any of the other RDF formats such as TTL.

ConceptNet is one of the major knowledge-bases which has gained popularity due to its extensive knowledge and periodic updates. In addition to the mentioned benefits, it is also available for public use in CSV and JSON [5] format and through its Web site and API. Converting this knowledge base to RDF has its unique challenges which arise from the complexity of its edges and deep hierarchy of the presented information in it. The feasibility and benefits of this conversion have been discussed previously in [8], but here we present the actual conversion steps and limitations. The work presented in this paper is an expansion of our previous work in ConceptOnto [16], in which we described the steps to create an upper ontology based on relations in ConceptNet.

In the following sections, first we review some of the different works in presenting large knowledge bases in RDF/XML and similar formats (Section II). Then in section III, we describe the structure of ConceptNet both in JSON and in CSV format and analyze different parts of information presented in it. Section IV gives an overview of the OWL ontology which has been created from properties and classes of ConceptNet and used in the conversion process. The process and the methodology of the conversion is presented in section V. This section also provides the main limitations of converting the data from JSON or CSV to RDF and provides solution for them to some extent (Subsection V-A). We believe this conversion is beneficial for different use cases for researchers and normal users, which we have described in section VI. The final section (Section VII) of this paper presents a conclusion and discusses our future work directions.

II. Other Semantic knowledge bases

Presentation of different knowledge bases in human and machine readable formats has a long history. The traditional format for this purpose has been comma or tab separated (CSV and TSV) versions of the data. Even for many of the other data-sets which are available in newer formats such as turtle or RDF, the CSV or TSV formats are available as native implementations. A good example of works only implemented in CSV is NELL [4] (Never Ending Language Learning). In NELL, the data gathering approach has two phases. In the first phase, the approach extracts data from Web

978-1-4673-8614-2/16/$31.00

and other resources; and in the second phase it measures the confidence level on correctness of the information gathered in the previous step. What makes NELL a special case of information gathering is its automatic and gradual expansion over time. The gathered information is available on weekly iterations for public use in CSV format.

Freebase [2] chooses a different approach for presenting the information by using graphs. The graph data structure provides the topics as nodes and the relations as links between them. This implementation decision, with introduction of MQL (MetaWeb Query Language) [7] and use of folksonomies, collaboratively created tags to describe contents, instead of traditional ontologies has created an eco-system for information retrieval and expansion on Freebase. Finally it is noteworthy that after the acquisition of MetaWeb (the company behind FreeBase) from Google, the graph is currently used in Google Knowledge Graph.

While RDF, CSV and TSV are some of the major formats for representing semantic knowledge in large data-sets, there are other formats in use for the same purpose with some extent of support in industry and academia. One of these formats is Turtle [1]. Turtle, Terse RDF Triple Language or TTL, a sub language of RDF, presenting the information in a format similar to SPARQL compared to the triple implementation of RDF. An example of data-sets presented in Turtle format is YAGO [22]. YAGO (Yet Another Great Ontology), currently in its second version [9], extracts information from Wikiepedia, WordNet [6] and GeoNames [24] and stores them in Turtle format. While this data-set is mainly presented in TTL format, it is also available in TSV and in various endpoints (APIs and SPARQL).

The mentioned knowledge bases are mainly focused on gathering facts in different contexts, similar to encyclopedias. On the other hand, there are some works focused on presenting description and meaning for different words, similar to dictionaries. A classic work in this field is WordNet. WordNet organizes words in different synsets providing not only meaning but also many other linguistic relations between words. The latest version of WordNet is currently version 3.1 which is available for download and public use. Regarding semantic Web, version three of WordNet is available in RDF format [14]. Another work similar to WordNet, but with addition of European languages is BabelNet [17]. Other than including BabelNet synsets for different terms, this data-set includes related images for each term to clarify the concepts further. This data-set is also available for download in RDF and TTL n-triples.

In addition to considering the meaning or the factual perspective of information, there are some works focusing on other aspects of knowledge. For example SenticNet [3] is a knowledge base which mainly focuses on sentiment values of the knowledge. For each piece of information, SenticNet mainly assigns polarity values to different terms extracted from ConceptNet, or in more recent versions (currently the third version is in beta) other common knowledge bases. Another sub-work of SenticNet is IsACore, provided in CSV format, specifically consisting of IsA relations, or *subPropertyOf* relations in RDF.

III. CONCEPTNET STRUCTURE

One way to see ConceptNet is as a graph in which each concept or assertion is a node and edges (in graph context and not in the context of ConceptNet, as described later) are relationships connecting them. There are two output formats available to download ConceptNet; one is a normal CSV file, outputting one line for each relation, separated with comma and tab. The second format is JSON which on the perspective of information presented is the same as CSV but with better human readability. Considering that the information presented in these two formats are practically the same, we describe the data-set with disregard to the presented format.

The main components of each line of information is a triple of subject, object and predicate. While in most cases the subject and the object are concepts, there are other cases which need clarification. In the following we first analyze a general concept relationship line in the data-set and later on we focus on special cases of information presentation in ConceptNet. The line (as shown in Figure 1.A) in the CSV file begins with the triple presented inside an assertion tag. The tag marks in ConceptNet URL are separated with a /. Each line in the ConceptNet data-set presents an edge. Each edge, identified by its ID following */e/* tag, consists of different parts as follows. Assertions, shown as */a/*, are the general knowledge presented in ConceptNet by marking the relation name, its beginning and the end of it. Every concept in the data-set follows a */c/* tag, presenting a term or a phrase. Relations, as described in subsection IV-B, are presented using */r/*. It is important to note that these relations are language independent and the concepts on both sides of them can be in different languages. */d/* marks the data-set the edge has been extracted from, defined in ConceptNet context as a large source of knowledge which can be downloaded as a unit. Similar to the */d/* tag, there is */s/* to present the source of the information presented in the assertion. Currently ConceptNet has four source of knowledge. The first source is *contributor*, meaning an individual which has added the knowledge to the data-set. *Activity*, a knowledge collection task presented to a user to collect the knowledge (such as a game). *Rule*, an automatic rule to extract knowledge from different resources to the desirable format. And *site*, a knowledge base extracted from a Web site. The ctx tag shows the context of the relationship, e.g. */ctx/all*. Finally */and/* and */or/* marks the conjunctions and disjunctions of sources.

Concepts in ConceptNet can have up to four parts. As mentioned previously the first portion of any concept URI is */c/*. This tag normally follows by a two letter language mark, e.g. *en* for English, *ja* for Japanese. The third part is the concept itself which has been normalized by lemmatizer available in *conceptnet5.language* package as part of ConceptNet code repository [1]. The concept then follows by a letter marking the part of speech tagging of the word (e.g. *v* for verb or *n* for

[1]Latest version at https://github.com/commonsense/conceptnet5

(A) /a/[/r/Antonym/, /c/en/ability/n/possession_of_the_ qualities_ required_ to_ do_ something_ or_ get_ something\ _done/, /c/en/inability/ n/lack _of _ability_to_do_ something/] /r/Antonym/c/ en/ability /n/possession _ of_ the_ qualities _ required_ to_ do_ something _ or_ get_ something_ done /c/en/inability/n/lack_ of_ ability_ to_ do_ something /ctx /all 1.5849625007211563 /s/wordnet/3.0 /e/ 3ca0f777df726b705e90ec84381f88e0b67d916c /d/wordnet/3.0

(B) {"rel": "/r/Antonym", "dataset": "/d/wordnet/3.0", "source_ uri": "/s/wordnet/3.0", "start": "/c/en/ability/n/possession_ of_ the\ _ qualities_ required_ to_ do_ something_ or_ get_ something_ done", "surfaceText": null, "features": ["/c/en/ability/n/ possession_ of_ the_ qualities_ required_ to_ do_ something_ or_ get_ something_ done /r/Antonym -", "/c/en/ability/ n/possession_ of_ the_ qualities_ required_ to_ do_ something_ or_ get_ something_ done - /c/en/inability/n/lack_ of_ ability_ to_ do_ something", "- /r/Antonym /c/en/inability/n/lack_ of_ ability_ to_ do_ something"], "sources": ["/s/ wordnet/3.0"], "context": "/ctx/all", "uri": "/a/[/r/Antonym/, /c/en/ability/n/possession_ of_ the_ qualities_ required_ to_ do\ _ something_ or_ get_ something_ done/,/c/en/inability/n/lack_ of_ ability_ to_ do_ something/]", "license": "/l/CC/By- SA", "weight": 1.5849625007211563, "end": "/c/en/inability/n/lack_ of_ ability_ to_ do_ something", "id": "/e/ 3ca0f777df726b705e90ec84381f88e0b67d916c"}

(A) A sample relation in CSV format. (B) A sample relation in JSON format.

Fig. 1. Sample line from ConceptNet data-set

noun) and the last part of the URI is the sense of the concept, if available (generally available for the knowledge extracted from WordNet).

Description of the edge presented in Figure 1 in plain English would be “ability, as a noun, with the meaning of possession of the qualities required to do something or get something done, is antonym of inability, as a noun, meaning lack of ability to do something. The context of this edge is all (currently all the contexts are “all” in ConceptNet version 5, but we store the context in the case of compatibility with future versions). The source and the data-set of this knowledge is WordNet version 3.0. This edge has an ID of 3ca0f7...b67d916c and the weight of this edge is 1.584...11563.” The same line of information in JSON is presented in Figure 1. The JSON presentation stores the same information while providing better readability. JSON format guaranties a presentation which is both suitable for machines and humans. The down side of this presentation is the space consumption of all the extra tags and labels which makes processing this information for any text analyzer more difficult. A partial view of the final result, without the modifications implemented in later sections of this work is as follows.

```
<rdf:Description
rdf:about=
    "http://conceptnet5.media.mit.edu/
    web/c/en/ability">
    <COnto:hasDataset>wordnet</COnto:hasDataset>
    <COnto:hasSource>wordnet</COnto:hasSource>
    <COnto:hasContext>all</COnto:hasContext>
    <COnto:hasPOS>n</COnto:hasPOS>
    <COnto:hasSense>
    lack_of_ability_to_do_something
    </COnto:hasSense>
    <COnto:hasAntonym rdf:resource=
    "http://conceptnet5.media.mit.edu/
    web/c/en/inability"/>
</rdf:Description>
```

IV. CONCEPTNET ONTOLOGY

For the purpose of converting ConceptNet to RDf/XML (for simplicity we call RDF/XML format RDF) format we make use of our previous work, ConceptOnto presented in [16]. While this ontology is mainly created using the properties and classes of ConceptNet, it can be used as any other upper ontology in a larger scale to convert other knowledge bases. In the following two subsections we first discuss the classes in ConceptNet and ConceptOnto (Section IV-A), and then in Section IV-B we discuss the relations implemented in ConceptOnto and some additional relations.

A. classes

There are four main classes of objects in ConceptNet as implemented in ConceptOnto. Compared to many other ontologies, these classes are more general as appropriate for a data-set mainly focused on common sense knowledge. We provide a brief description of these classes in the following.

Noun Phrase (NP) is by far the most common entity in *ConceptNet*. These phrases normally consist of one or more main noun as the root and one or more other parts to clarify the noun. For example for the relation “isDefinedAs”, for any triple, the object and subject are *NP*. *Verb Phrase (VP)* is any phrase which has a verb as its root. These verbs can be preceded or succeeded by any other word. An example of a *VP* is the property “isCapableOf” with the triple “bike isCapableOf moving_forward”, *moving_forward* is a verb phrase with the root of the verb moving followed by the adjective “Forward”.

Adjective Phrase (AP) is the general term used specifically in range and domain of properties such as “hasProperty”. Any *AP* can consist of a set of words which in turn can have multiple adjective, noun or even verbs. For example in triple “bike hasProperty common_In_Asia”, common_In_Asia while starts with an adjective, followed by a noun to complete the concept.

Terms are the last class of entities in *ConceptNet*. Terms are general phrases used in “isDerivedFrom” or “isTranslationOf” which shows the relation between two phrases when one derives from the other one or the phrases are equal in different languages. For example in the triple “begin isDerivedFrom start” subject and object are verbs while in “earth_science isDerivedFrom earth” the object is a NP and the subject is a noun. This generality of concept in Terms make all the other classes subclass of this class.

B. Object and data properties

Data properties in ConceptNet are the main portion of the ontology used for conversion of ConceptNet to RDF. While this is the main purpose of this ontology, we believe that it can be used like any other upper level ontology. The following steps are mainly to modify the properties and prepare the ontology for the mentioned purpose. The first addition to the properties of ConceptOnto is to modify the naming of the properties by adding *is* and *has* to them, based on objective or subjective meaning of them. This naming methodology defines the difference between having a specific property versus an entity residing in another entity. While we have not found this extensive use of this methodology in similar works, it has been suggested in different references such as [10].

The second change to ConcpetNet properties is the addition of reverse properties. As OWL language does not have the ability to extract explicit relations based on the existing relations, this addition will allow new reasoning and information retrieval from the system. The next addition to the relations is to add properties such as symmetric, transitive and reflexive. The same reasoning as the previous change applies to this addition as well.

Finally to check, expand and extract implicit relations in ConceptOnto two different reasoners have been evaluated. After through analysis of FaCT++ [23] and HermiT [20], we chose HermiT to make the final modifications to ConceptOnto such as using the equivalent relations for expanding the inverse relations. More detailed and further explanation of the ontology can be found in our previous work [16].

V. ConceptRDF conversion process

As mentioned in Section III, ConceptNet data-set is available in JSON and CSV format. The JSON format has extra labels and tags to increase its readability for humans, which makes it larger in volume which in turn makes it computationally more expensive. For this reason, we decided to process the CSV files to generate the RDF files. The process of converting the files to RDF is straightforward. This process consists of reading the files line by line, create tokens from them, parse the tokens and extract the information. Algorithm 1 shows the algorithm used for this process. However, as much as this process is thorough, there are issues need addressing which we discuss in the next section. The result of this conversion, alongside the ConceptOnto ontology, is available for public use on our Web site [2] [3].

```
read file;
read line;
separate by comma;
while Tokens available do
    find /r/;
    extract the relation as predicate;
    find the first concept;
    read the subject;
    if subject then
        if part of speech available then
            extract subject pos;
        end
        if sense available then
            extract sense;
        end
    end
    if object then
        if part of speech available then
            extract object pos;
        end
        if sense available then
            extract sense;
        end
    end
    extract context;
    extract data-set;
    extract source;
    extract weight;
    extract edge ID;
end
```

Algorithm 1: Extracting information from a CSV edge algorithm

A. Limitations

The issued for converting ConceptNet to RDF are two folded. The first issue is with the logical disambiguation between formats like JSON and CSV to RDF. This issue is more fundamental and has more importance compared to the other problem. The second issue is more up to case by case basis related to specific relations which have further complexity than a normal triples.

Regarding the first problem, in the ConceptNet official blog it mentions [4] the main reason for preference of JSON over RDF:

> ConceptNet is not RDF
> I have sometimes been asked, given that ConceptNet is fundamentally a graph, why it isn’t published in an RDF-based format. RDF is a very general representation of

[2]http://score.cs.wayne.edu/ConceptOnto.owl
[3]http://score.cs.wayne.edu/result
[4]https://github.com/commonsense/conceptnet5/wiki/Linked-Data-and-the-Semantic-Web

> graph data, and yet it doesn't quite cover the information that ConceptNet needs to convey.
> Much of the information in ConceptNet is expressed as properties of its edges. In RDF, edges simply exist; they don't have properties. Additionally, all edges in RDF have to be considered incontrovertibly true, regardless of what source they came from, because they don't preserve any information about their sources.
> If you want to be able to talk about an edge, you need to reify it by turning it into a node and connecting it with a different kind of edge. This representation of ConceptNet would be difficult to create and even more difficult to work with. Instead, representing the edges of ConceptNet as JSON structures (see JSON streams) makes the information in it easily accessible in a variety of programming languages.

The main point in this discussion is the complexity of presenting the diversity of information in edges in RDF form. This information can be divided into three parts, respectively subject, predicate and object. In creating an RDF triple for the main triple in each line, we can expand the information presented for the subject by creating multiple triples for the same subject. The problem arises for creating triples with objects of the original triple as subject. This limitation can be solved by assigning IDs to objects and create triples describing them separately. While this solution answers the problem in hand, it complicates the data-set to the extent that retrieving required information (using SPARQL, as described in section VI, or any other method) will be lengthly and complicated. To this extent, and based on the fact that the main piece of information presented for the object is the meaning of them (extracted from WordNet), we have decided against this approach.

Another approach to solve this problem is to separate the properties which are related to the subject of the triple from the ones related to the object of the triple. In the case of ConceptNet, we can do this by addition of one relation, to add a similar property to hasSense; namely replacing hasSense relation with subjectHasSense and objectHasSense. As a side note, it is necessary to mention that the only case that this addition is useful for is in converting the information extracted from WordNet, but because of the extendability of this knowledge-base (such as adding BabelNet synsets to the data-set) this is a useful addition and future approach. While this approach has better presentability for humans, considering the way SPARQL queries are presented, it complicates the creation and readability of the queries. We try to clarify this point further by an example. For the edge we presented in section III (ability hasAntonym inability), if we try to extract the sense of inability we run the following SPARQL query, in which the pivot point of the query is still the subject (ability) while we are looking for information on the object (inability).

```
SELECT ?sense
WHERE {
 :ability conto:objectHasSense ?sense;
 :ability conto:hasAntonym :inability
}
```

In this instance, and similar relations with transitive property, we can use this property to run the query for the opposite direction of the relation and run the same query for the object which simplifies the process to some extent; but for any other relation this complexity still exists. Another main limitation for converting JSON to RDF format (the second issue as discussed earlier) is the TranslationOf (isTranslationOf in ConceptOnto) property. In the context of ConceptNet most instances of this property have different triples as subject and object. The issue with this proposition is RDF does not have the capacity to inquire triples inside other triples. For example the following triple has two triples as its subject and object (dog IsA animal in English and Japanese).

(犬 IsA 動物) TranslationOf (dog IsA animal)

Analysis of instances of TranslationOf shows that the triples implemented in both object and subject of this relation are both presented in other lines. As any triple which exists in ConceptNet has an ID (edgeID), we can use these IDs to create new triple consist of subject edge ID, the relation isTranslationOf and the object triple edge ID. Using this approach the mentioned relationship changes to the following (edge IDs have been shortened for convenience of formatting)

```
/e/e1...5c1 isTranslationOf /e/5c6...0ea9
```

Another approach for this issue in RDF model is to use RDF reification. Reification is usually used to create statement describing another statement. In this case, we believe reification is a better option considering that having all the details in regard of the statement in one place is more self explanatory and makes the retrieval of the information easier. While this approach mostly solves the mentioned problem, there is still a lot of discussion [18] on the use of reification and its complexity, which we consider to pursue for our future works.

VI. Use cases

There are multiple use cases for Semantic Web data representation such as in Question Answering systems, Sentiment Analysis or any similar research topic in which the research tries to make sense of common sense knowledge which is hard to represent in any other format except an interconnected web of information. In this section we present two examples of the possible use cases of ConceptRDF.

For the first example we follow the directions in our previous work [15] to answer a simple question using ConceptRDF. In that work we mentioned the need for a large RDF knowledge base as a pre-requirement for our approach. While the approach has an acceptable performance on a simulated data-set, we believe that with the extended ConceptNet knowledge base and a possible additionother similar data-sets it can improve exponentially. We try to provide an example which not only shows the usefulness of our approach, but also shows the limitation of current status of ConceptNet for the purpose of QA system. The question we consider is "Who is Bill Clinton?". The translation of this question in SPARQL is shown in the following.

```
SELECT ?object ?weight
WHERE {
 Bill_Clinton conto:IsA ?object;
```

```
 Bill_Clinton conto:hasWeight ?weight
} ORDER BY DESC(?weight)
```

It is noteworthy that this query orders the results by their edge weight, provide the results with the highest confidence as the first answer. Also we can see that lack of temporal information, in many cases can cause wrong information to be retrieved from the data. In this case the result of the query is the triple "$Bill\_Clinton$ isA $president\_of\_unite\_state$" which with temporal consideration is not true (Bill Clinton is a former president of the Unites States would be a better answer). Finally the normalization to the concepts ConceptNet has changed United States to unite state which is not the same answer the user is looking for.

In the field of sentiment analysis, the knowledge presented in ConceptNet can be used as an intermediary information resource. Because the ConceptNet is a knowledge base based on common sense, it is rich on information regarding emotions, their states and causes. A logical follow through of the emotions can start by using IsA relation, extract the emotions by using the following query, then by using causes relation, extract the words and concepts which can be used to find relations between emotions, events, feelings or any other concept.

```
SELECT ?emotionNoun ?cause
WHERE {{
SELECT DISTINCT ?emotionNoun WHERE {
        ?subject conto:IsA emotion;
      }}
    OPTIONAL {
     ?cause conto:causes ?emotionNoun
      }
}
```

The result of this query returns emotions such as love, happiness and fear as result. Also in search of emotion without limitation of IsA relation, if there is an intelligence system it can find other details regarding emotions, for example people have emotions (people HasA emotion) or computers do not have emotions (computer NotHasA emotion).

VII. Conclusion and Future Work

In this paper we presented ConceptRDF, which is representation of ConceptNet knowledge base in RDF format, along with a thorough process and steps which are required to commit this conversion. In different sections of this paper we review some of the related work and some details from our previous work regarding the ontology which we use for the basis of this conversion, then provide the steps required for the conversion. Then we discussed some of the many use cases of the new knowledge-base to give a proof of concept of the usability of the conversion.

Part of the benefit of our process is the ease of expanding it with any other knowledge base. We have already selected some candidates for this purpose such as IsACore to convert to RDF as the next step. To expand the work further we consider creating an automatic process to expand the knowledge base by using machine learning and pattern matching approaches.

References

[1] David Beckett, Tim Berners-Lee, and Eric Prudhommeaux. Turtle-terse rdf triple language. *W3C Team Submission*, 14, 2008.

[2] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM, 2008.

[3] Erik Cambria, Robert Speer, Catherine Havasi, and Amir Hussain. Senticnet: A publicly available semantic resource for opinion mining. In *AAAI Fall Symposium: Commonsense Knowledge*, volume 10, page 02, 2010.

[4] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka Jr, and Tom M Mitchell. Toward an architecture for never-ending language learning. In *AAAI*, volume 5, page 3, 2010.

[5] Douglas Crockford. The application/json media type for javascript object notation (json). 2006.

[6] Christiane Fellbaum. *WordNet*. Wiley Online Library, 1998.

[7] D Flanagan. Developing metaweb-enabled web applications. *Metaweb Technologies*, 2007.

[8] Marco Grassi and Francesco Piazza. Towards an rdf encoding of conceptnet. In *Advances in Neural Networks–ISNN 2011*, pages 558–565. Springer, 2011.

[9] Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, and Gerhard Weikum. Yago2: a spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*, 194:28–61, 2013.

[10] Matthew Horridge, Holger Knublauch, Alan Rector, Robert Stevens, and Chris Wroe. A practical guide to building owl ontologies using the protégé-owl plugin and co-ode tools edition 1.0. *University of Manchester*, 2004.

[11] Graham Klyne and Jeremy J Carroll. Resource description framework (rdf): Concepts and abstract syntax. 2006.

[12] Hugo Liu and Push Singh. Conceptneta practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226, 2004.

[13] Deborah L McGuinness, Frank Van Harmelen, et al. Owl web ontology language overview. *W3C recommendation*, 10(10):2004, 2004.

[14] Sergey Melnik and Stefan Decker. Wordnet rdf representation, 2001.

[15] Erfan Najmi, Khayyam Hashmi, Fayez Khazalah, and Zaki Malik. Intelligent semantic question answering system. In *Cybernetics (CYBCONF), 2013 IEEE International Conference on*, pages 255–260. IEEE, 2013.

[16] Erfan Najmi, Khayyam Hashmi, Zaki Malik, Abdelmounaam Rezgui, and Habib Ullah Khan. Conceptonto: An upper ontology based on conceptnet. In *Computer Systems and Applications (AICCSA), 2014 IEEE International Conference on*. IEEE, 2014.

[17] Roberto Navigli and Simone Paolo Ponzetto. Babelnet: Building a very large multilingual semantic network. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 216–225. Association for Computational Linguistics, 2010.

[18] Vinh Nguyen, Olivier Bodenreider, and Amit Sheth. Don't like rdf reification?: making statements about statements using singleton property. In *Proceedings of the 23rd international conference on World wide web*, pages 759–770. International World Wide Web Conferences Steering Committee, 2014.

[19] Eric PrudHommeaux, Andy Seaborne, et al. Sparql query language for rdf. *W3C recommendation*, 15, 2008.

[20] Rob Shearer, Boris Motik, and Ian Horrocks. Hermit: A highly-efficient owl reasoner. In *OWLED*, volume 432, 2008.

[21] Robert Speer and Catherine Havasi. Conceptnet 5: A large semantic network for relational knowledge. In *The Peoples Web Meets NLP*, pages 161–176. Springer, 2013.

[22] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A Core of Semantic Knowledge. In *16th international World Wide Web conference (WWW 2007)*, New York, NY, USA, 2007. ACM Press.

[23] Dmitry Tsarkov and Ian Horrocks. Fact++ description logic reasoner: System description. In *Automated reasoning*, pages 292–297. Springer, 2006.

[24] Marc Wick. Geonames, 2011.