Extraction of RDF Statements from Text

Jose L. Martinez-Rodriguez[1(✉)], Ivan Lopez-Arevalo[1], Ana B. Rios-Alvarado[2], Julio Hernandez[1], and Edwin Aldana-Bobadilla[1]

[1] Cinvestav Tamaulipas, Victoria, Mexico
{lmartinez,ilopez,nhernandez,ealdana}@tamps.cinvestav.mx
[2] Faculty of Engineering and Sciences, UAT, Victoria, Mexico
arios@uat.edu.mx

Abstract. The vision of the Semantic Web is to get information with a defined meaning in a way that computers and people can work collaboratively. In this sense, the RDF model provides such a definition by linking and representing resources and descriptions through defined schemes and vocabularies. However, much of the information able to be represented is contained within plain text, which results in an unfeasible task by humans to annotate large scale data sources such as the Web. Therefore, this paper presents a strategy for the extraction and representation of RDF statements from text. The idea is to provide an architecture that receives sentences and returns triples with elements linked to resources and vocabularies of the Semantic Web. The results demonstrate the feasibility of representing RDF statements from text through an implementation following the proposed strategy.

Keywords: Semantic Web representation · RDF representation · Entity linking · Relation extraction · RDF statements

1 Introduction

The Semantic Web refers to an extension of the traditional Web, which has an important goal of providing a formal data representation that enables the sharing and reuse of information by people and applications [1]. This goal is being addressed by varied standards and protocols, such as the Resource Description Framework (RDF) and the Linked Open Data (LOD) principles [2]; on which the data are represented through basic units of information called RDF triples, each one composed of *Subject-Predicate-Object* elements. In consequence, the data are organized into a knowledge graph where the nodes correspond to information resources (such as real-world objects, *aka* "entities") and edges to descriptions (that adopt formal vocabularies or ontologies[1]) or relationships between such resources. Additionally, every resource (node/edge) must be individually identified through Internationalized Resource Identifiers (IRI) and retrieved (dereferenced) via the HTTP protocol to provide more information of the resource through the Internet (as is done on the traditional Web).

[1] An ontology defines the concepts, terms, classes, taxonomies, and rules of a domain [11].

B. Villazón-Terrazas and Y. Hidalgo-Delgado (Eds.): KGSWC 2019, CCIS 1029, pp. 87–101, 2019.
https://doi.org/10.1007/978-3-030-21395-4_7

The Semantic Web information representation usually follows a process focused on extracting knowledge elements[2] to later associate them with (unambiguous) identifiers (IRIs) based on ontology descriptions and standards of the RDF model. In consequence, the information represented through RDF triples can be used to create or enrich a Knowledge Base (KB) that can be queried using the SPARQL language[3]. Hence, in order to represent plain text sentences[4] as RDF triples, relevant elements should be extracted from text (e.g., named entities and their relationships) to then associate them with the elements of the RDF triple. For instance, the sentence "*Ciudad Victoria is a town located in the state of Tamaulipas*" can be represented by RDF triples in two ways as depicted in Fig. 1. While the option (a) uses a binary statement (two resources linked by a property), the option (b) relies on an n-ary statement (more than two resources linked to various properties) [15].

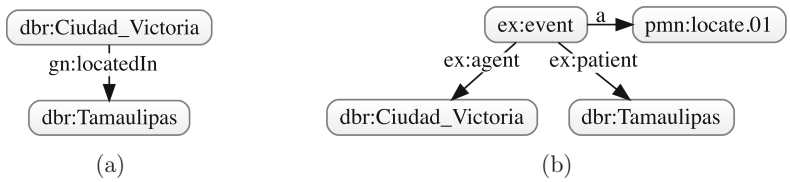

Fig. 1. Example of a sentence represented through RDF triples. (a) Indicates a binary statement; (b) Indicates an n-ary statement.

From the Fig. 1, both representations are used to describe the same idea within the sentence. In this way, binary statements link two resources (or a resource and a literal value) through a property from a KB. On the other hand, n-ary statements allow a resource to be linked to one or more resources and/or literal values. n-ary statements are useful for describing particular events/situations involving diverse actors; for example, in a product sale we may find actors such as the buyer, seller, and the product. However, it is often difficult to know the exact role of the actors in a sentence. Hence, in the example of Fig. 1b the actors of the event (`ex:event`) are denoted by semantic roles[5], in which the *causer* of the event (or action) is denoted by the property `ex:agent`, the *undergoer* by the property (`ex:patient`), and the type of event by the letter

[2] In this context, knowledge elements refer to Conceptual Knowledge [22] in terms of things or concepts and the way they are related to each other with the support of an ontology.

[3] https://www.w3.org/TR/sparql11-overview/. All URLs in this paper were last accessed on 2019/04/15.

[4] Different to formatted text, plain text does not contain any style information or graphical objects and refers to only readable characters.

[5] Semantic roles identify the participants in an event guided by a verb and its underlying relationship [13].

`a` or `rdf:type`, to mention a few. Note that we model n-ary statements according to the reification options presented by Hernández *et al.* [15], where a relation is modeled through a resource instead of a property, which can be annotated with meta-information. Note that particular implementations (through tools or strategies) of the tasks involved within the proposed methodology depends on the modeler decisions according to the type of addressed statement (i.e., binary or n-ary).

According to the previous example, the representation of sentences as RDF triples faces varied difficulties and challenges to detect the two main elements of a statement: named entities and their semantic relations. Moreover, in the context of the Semantic Web, such elements must be associated with resources and properties from an existing ontology (or KB) respectively. Therefore, this paper proposes a methodology for the extraction and representation of RDF statements from text. Particularly, binary and n-ary RDF statements from plain text sentences. The aim is to provide a way to represent such statements through a methodology that encompasses the interaction of diverse tasks from areas such as Information Extraction (IE) and Natural Language Processing (NLP). As a proof of concept, we present a strategy for the extraction and representation of n-ary statements from plain text, where specific tools and strategies are configured and implemented in order to fulfill the tasks presented in the proposed methodology. This implementation is useful for presenting an initial evaluation of the proposal, which involves the participation of human judges on topics such as news and tourism.

2 State of the Art

General strategies and recommendations for the information representation on the Semantic Web have been performed so far. In this regard, Bizer and Heath [14] described the Linked Data operation sequence for publishing semantic information. Their architecture is organized into three stages that receive distinct input sources. First, a preparation stage parses structured text or processes unstructured data through NLP tools. The second stage is intended to extract and store entities and parsed elements obtained from the text. Finally, the information is published using a web server. The difficulty of this approach relies on the lack of association of resources with Semantic Web resources.

Another representation strategy is provided by the FOX (*Federated knOwledge eXtraction*) framework[6], which generates RDF data by using Named Recognition (NER), Keyword Extraction (KE) and Relation Extraction (RE) algorithms within an architecture composed of three layers: Automatic Learning layer, for training a module with the best-performing tools and categories; Controller layer, to coordinate information and parsing tools; and Tools layer, containing a repository of tools such as NLP services and data mining algorithms.

Similarly, Auer *et al.* [3] identified three branches for extracting features used for representing RDF triples from unstructured text: NER to extract entity labels

[6] FOX framework. http://aksw.org/Projects/FOX.html.

from text, KE to recognize central topics, and RE to extract properties that link entities. Moreover, authors also state that a disambiguation task is necessary to obtain adequate URIs for every resource within the extracted RDF triples. This task is conducted employing entity matching over KBs like DBpedia[7] or FreeBase[8]. Along these lines, based on the previous steps, approaches such as [5, 10,17,25] obtain entities through NLP tools, apply morphosyntactic analysis and lexical databases like WordNet[9] to extract and disambiguate elements from text using existing vocabularies.

On the other hand, according to the type of extraction, we distinguished three groups of approaches that extract RDF statements from text using NLP and/or machine learning techniques. First, discourse-based approaches [5,12] employ a framework for describing lexical meaning in terms of a set of predicates (Frames) and their arguments (Frame Elements)[10], where the elements are directly mapped to properties of a KB through n-ary statements. Second, pattern-based approaches [9] generate patterns (pattern induction) that describe conventional relations from the text, where properties could be directly mapped to a KB or obtained by semantic similarity matching. Third, machine learning-based approaches [4] use semantic and syntactic annotations together with information from a KB to obtain features used for training a machine learning algorithm (mainly a supervised strategy), where properties of binary relations are directly associated through the training data.

The above approaches are consistent regarding the stages such as the extraction of features from text, named entities, and semantic relations. However, such approaches provide only a brief overview of basic architectures to extract and publish information as RDF statements, which do not state stages for the association of relations with properties and the organization of elements that should be part of the final statements. Thus, the following section provides the proposed methodology containing the tasks and components involved in the representation of RDF statements on the Semantic Web.

3 Methodology

This section presents the proposed methodology for the representation of RDF statements from text. It consists of the architecture presented in Fig. 2, which is composed of three main stages: Data Layer, Knowledge Extraction Layer, and Representation Layer. Such stages cover several tasks and components involved in the representation of statements on the Semantic Web such as the diverse types of input data (domains), representation structures (RDF reification), and representation formats.

A description of the stages within the proposed architecture is presented in the following subsections.

[7] https://wiki.dbpedia.org.
[8] https://developers.google.com/freebase/.
[9] WordNet is a lexical database for English http://wordnet.princeton.edu.
[10] From a First Order Logic perspective, the predicate of a sentence corresponds to the main verb and any auxiliaries surrounding it.

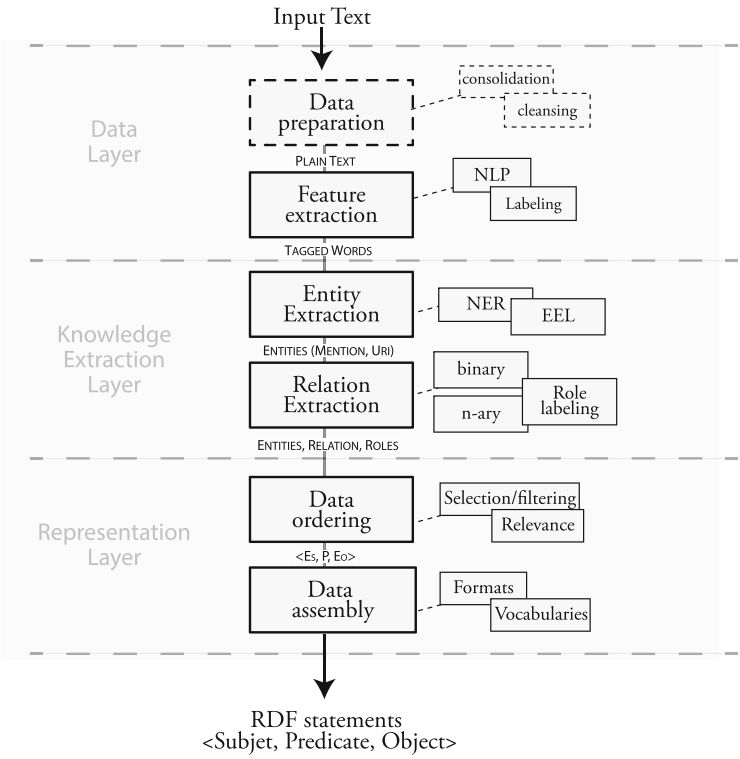

Fig. 2. Architecture for the representation of RDF statements from text

3.1 Data Layer

The first stage of the architecture refers to operations performed over the initial data in order to clean and obtain features used in further steps. In this sense, we only consider unstructured text (plain text) as input data to the architecture[11]. This stage is composed of the following steps:

– Data preparation. The input data might be given in diverse formats (e.g., PDF, Word). The purpose of this step is to merge the diverse sources that could be presented for the architecture in order to get a homogenized data source. Moreover, the input data often contains superfluous annotations that are not useful for the proposed strategy. For example, formatting tags (e.g., HTML). In this sense, this step is aimed at applying cleaning and parsing operations over the input text, so that plain text is finally obtained. Note that this step is optional because the text might be provided as plain text.

[11] Although the architecture only admits plain text as input data, there are several types of data that could be considered such as structured data (e.g., databases, tables), images, or raw data (e.g., data from sensors).

– Feature extraction. As previously mentioned, we consider plain text sentences as input data, which means that it does not include features or boundaries describing the elements that can be extracted. Therefore, this step is aimed at identifying and tagging features from the input data. For example, Part of Speech (POS) tags, dependency tree structures that denote groups of nouns, among others. This step is particularly important to facilitate the process of the following step for the obtention of relevant items from text.

3.2 Knowledge Extraction Layer

The strategy followed in this work seeks to obtain RDF binary and n-ary statements. Thus, the two most relevant elements involved in such statements are named entities and their semantic relation. Therefore, this stage is aimed at extracting these elements from the input text. In this regard, the idea is to get support from the *Data Layer* stage by providing structures and tags used in the organization and extraction of the required elements. The steps involved in this stage are as follows:

– Entity Extraction. The purpose of this step is to extract Named Entities from text. Thus, this process involves the detection of those nouns that can be described according to their type (i.e., Person, Place, etc.) and the resources representing them from a real-world perspective; that is, linking a found entity[12] mention to a resource from a KB such as Wikidata[13] or DBpedia. This complete task is known as *Entity Extraction and Linking* (EEL). The output of this step consists of a set of Entities, in which each element contains the mention and its URI.
– Relation Extraction. A subsequent and important process is to detect how those previously found entities are related to each other. Thus, this step is aimed at detecting semantic relations between entities from text. However, as presented in the introduction, there are different types of statements according to the level of detail and context that needs to be extracted; binary and n-ary relationships. While the former lead to declare two resources joined by a property, the latter involves an exhaustive analysis to get several resources involved in the same idea and the correct relationship among them[14]. Together with the set of entities, the output of this step contains the identified relation between entities and the roles that denotes the activity of entities (e.g., the causer of an action).

3.3 Representation Layer

Once the entities and their relationships are extracted from the input text, the next task is to order such elements to make the final representation of statements using standards of the Semantic Web. This stage involves the following steps:

[12] In this work, we indistinctly refer to named entities as only entities.
[13] www.wikidata.org.
[14] This process is often supported by the Semantic Role Labeling task, which helps to determine the role or action performed by an entity within a statement.

– Data ordering. The previous stage considers the identification of entities and their role on the statement –as performed by Semantic Role Labeling (SRL)–. Thus, this step is in charge of determining the correct position of an entity within the final RDF triple. That is, positioning an entity as the subject/object within a binary statement or as the agent/patient within an n-ary statement (as described in Fig. 1b). In any case, the relationships on both statements must be linked to properties in a KB, on a process known as *property selection*. Thus, a property can be obtained from a KB by an entity matching comparison [9] or by direct string mappings [12], to mention two. Moreover, the property selection is often accompanied by a score that measures the level of matching between the predicate of a semantic relation and a property from a KB. Thus, this step also involves filtering irrelevant statements according to a score or function. Note that the complete process of extracting semantic relations and linking the components to resources and properties from a KB is known as *Relation Extraction and Linking* (REL) [21]
– Data assembly. This final step involves the formatting of statements. Therefore, it should be able to export the information on diverse formats (e.g., Turtle, XML/RDF). In some cases (e.g., n-ary statements), the representation also involves the declaration of descriptions through defined vocabularies. For example, including provenance information that allows the represented data to be evaluated according to the original data.

The next section provides the implementation of a version of the proposed methodology, where n-ary statements are extracted and represented from text.

4 Implementation Focused on n-ary RDF Statements

This section presents the steps followed for the implementation of a version of the proposed methodology. This is because several IE and NLP approaches are involved in the process, which can be replaced by others that cover the same purpose. Although such a methodology provides the steps involved in the representation of either binary and/or n-ary statements, we only cover the extraction and representation of n-ary statements (see Fig. 1), which is useful for describing the resources and their performed role within an action/event stated in a text sentence (we plan to include a strategy to extract binary statements in a future work). In this regard, the implementation was developed as a Java application. Thus, some internal configuration details of the Information Extraction and NLP tools and services used by the application are provided in this section with respect to the architecture depicted in Fig. 2.

Data layer. Although we assume that the input data is given as plain text that does not require further cleaning operations, the following tasks were applied:

– Feature extraction. This step is intended to perform NLP tasks through the Stanford CoreNLP tool [19], where models for English[15] were used in the

[15] Stanford CoreNLP models https://stanfordnlp.github.io/CoreNLP/.

configuration. Hence, this step performs the following tasks: tokenization of words, sentence segmentation, Part of Speech (POS) tagging, and structural parsing (*constituency tree* parsing). Additionally, we also performed a strategy to expand language contractions; for example, converting words such as *aren't* into *are not*.

Knowledge Extraction Layer. We performed tasks for the extraction of entities and semantic relations as follows:

- Entity Extraction. Entities were extracted and linked to a KB by following the strategy presented in [20], where four EEL systems were configured (DBpedia Spotlight, TagMe, Babelfy, and WAT) and integrated into an ensemble-like system. Additionally, it was developed a Java module that takes as input a sentence (and its constituency tree extracted by the feature extraction step) and the entities extracted in the EEL step to return entities grouped by noun phrases (NP); which is intended to preserve the coherence of ideas (in order to not decompose entities that belong to the same unit of information).
- Relation Extraction. The extraction of semantic relations was performed through the OpenIE tool ClausIE [8], which was configured using default parameters to obtain only binary relations. Additionally, Mate-Tools[16] was used for obtaining semantic roles (SRL) associated with entities and predicates (verbs) of the identified semantic relations. In this regard, predicates and arguments provided by Mate-Tools are based on annotations of the lexical resource PropBank. The data models used by Mate-Tools for internally parsing, lemmatizing and tagging were the CoNLL2009 models for English[17]. Note that, although we extract binary relations, the final representation results in n-ary statements by representing the thematic roles (i.e., agent, patient, predicate) and additional elements (e.g., original sentence).

Representation Layer. The final representation of entities and relations was as follows:

- Data ordering. Entities in the input sentence were selected according to the roles detected by the SRL tool (Mate-Tools). However, if the role is not identified, the entity near to the verb (within the semantic relation) is selected. Moreover, to obtain an identifier for the event/action expressed in the semantic relation, we leverage the predicate sense identified by Mate-Tools to perform a SPARQL query over the Premon KB[18], requesting the resource with the label (`rdfs:label`) matching the identified predicate sense. For such purpose, a Jena[19] module was implemented, using the SPARQL 1.1 syntax.

[16] MatePlus https://github.com/microth/mateplus.
[17] Data models downloaded from https://code.google.com/archive/p/mate-tools/downloads.
[18] https://premon.fbk.eu/query.html.
[19] Jena https://jena.apache.org.

– Data assembly. A Jena module was implemented for organizing all event-based information obtained from sentences and documents throughout the pipeline. In other words, this step represents events that contain a predicate and its arguments (*Agent* and *Patient*), which are represented by an n-ary reification model using the TriG[20] format.

5 Evaluation

This section presents the evaluation of the method for the representation of RDF statements. In this sense, two types of evaluation were performed, quantitative and qualitative. First, we obtained the total number of RDF triples represented by the method (including entities and relations). Second, we evaluated the precision of such data. The experiments were performed over a computer with OS X Yosemite, Intel Core i5, and 8 GB RAM. The next section provides details of the datasets used for the experiments.

5.1 Datasets

The experiments were performed over three datasets:

– IT news. It contains 605 documents regarding the IT domain manually extracted from sites such as DailyTech[21] and ComputerWeekly[22].
– LonelyPlanet[23]. It consists of 1801 webpage documents containing descriptions of places such as countries, cities, and so on. The HTML content of the retrieved webpages have been cleaned and converted to plain text, it contains over one million of tokens. This dataset was used by Cimiano [7] for the ontology learning task.
– BBC news. This dataset contains 2225 documents extracted from the BBC news website[24] corresponding to stories categorized in five topics: business, entertainment, politics, sport, and tech.

A description of the datasets used for the experiments is presented in Table 1. Note that the BBC dataset was divided according to document topics.

5.2 RDF Quantitative Evaluation

This section presents the quantitative experiment of the RDF n-ary representation produced from the three datasets. The aim of this experiment is to analyze the information that can be extracted from text and represented as RDF triples

[20] https://www.w3.org/TR/trig/.
[21] https://dailytech.page.
[22] https://www.computerweekly.com.
[23] The LonelyPlanet dataset was originally downloaded by Martin Kavalec from the site http://www.lonelyplanet.com/destinations.
[24] http://mlg.ucd.ie/datasets/bbc.html.

Table 1. Description of datasets used for RDF representation experiments.

Dataset	Domain	Documents	Sentences
IT news	News	605	12015
LonelyPlanet	Tourism	1801	16540
BBC	Business	510	5988
BBC	Entertainment	386	4482
BBC	Politics	417	5902
BBC	Sport	511	6514
BBC	Tech	401	6901
Total		4631	58342

by following the proposed representation approach. This experiment consisted of the execution of the strategy described in Sect. 4 over the three selected datasets (IT news, LonelyPlanet, and BBC news). Hence, for this analysis, every document was submitted to tasks such as entity extraction, relation extraction and ordering until RDF n-ary statements were represented.

The results obtained through the execution of the proposed representation approach are presented in Table 2, where the column *Documents* refers to the number of documents on each dataset, *Rep. Sent.* refers to the represented sentences (with at least one event), *Entities* refers to the extracted and linked entities, *Relations* refers to the extracted relations, *Events* refers to the number of events represented in an n-ary fashion (every event contains elements such as Agent, Patient, predicate sense, location, and so on), and *Triples* refers to the total number of represented triples.

Table 2. Result of the RDF triple representation on the three datasets.

Dataset	Rep. Sent.	Relations	Triples	Events	Entities
IT news	4262	41190	89486	7606	20536
LonelyPlanet	4312	37657	63553	5059	12014
BBC (business)	3181	17651	51561	4451	10352
BBC (entertainment)	1984	12930	32456	2747	6928
BBC (politics)	2850	18499	46217	4057	9770
BBC (sport)	2538	19023	40184	3498	8935
BBC (tech)	3213	19376	50956	4451	8894
Total	**22340**	**166326**	**374413**	**31869**	**77429**

Discussion. According to the results presented in Table 2, it has to be noticed that the input data was not completely represented as RDF. The proportion

of represented information as RDF statements is shown in Table 3, where the second column indicates the ratio of represented sentences regarding the original ones, and the third column represents the ratio of semantic relations represented as events. These facts are produced because diverse NLP strategies are involved in the representation strategy. Hence, the following facets regarding the representation strategy can be mentioned:

- Feature extraction. Although the NLP tool's accuracy has improved during the years [23], elements in the text are often difficult to segment or annotate, particularly large sentences or text with typos.
- Recognition. Mentions of entities and relations need to be found. However, such tasks often depend on the segmentation and annotation of words provided by the previous aspect, which in turn, might produce wrong element extractions. Moreover, while OpenIE tools do not depend on a particular domain to obtain semantic relations, not all types of relationships are covered by the rules and patterns employed by such tools. On the other hand, EEL systems used for providing entities from text are often associated with a domain and/or KB. In consequence, the proposed method can represent only those sentences that contain semantic relations in which entities appear both in subject and object.
- Representation. The representation only covers RDF triples with object properties. That is, only those relationships containing resources (named entities) in *subject* and *object* are represented in RDF (e.g., `dbr:New_York_City rdf:type dbr:Location`). Although literal values can be assigned to the object of an RDF triple (e.g., `dbr:New_York_City rdfs:label"New York"`), such cases are not within the focus of this work.

Table 3. Ratio of represented information.

Dataset	Ratio Sent./R. Sent.	Ratio Rel./Events
IT news	35.47	18.47
LonelyPlanet	26.07	13.43
BBC (business)	53.12	25.22
BBC (entert.)	44.27	21.25
BBC (politics)	48.29	21.93
BBC (sport)	38.96	18.39
BBC (tech)	46.56	22.97
Total	38.29	19.16

Given the previous facets and results, it can be observed that, from the total of input sentences of the three datasets, only `38.29%` was represented as RDF

statements and only 19.16% of the extracted relations was represented using RDF events. The most affected dataset (regarding the level of representation) was LonelyPlanet, which contains text on the tourism domain. Note that such a dataset is in English but some elements such as names of things and places are expressed in diverse languages (e.g., Spanish, African-based) around the world that can difficult the recognition of entities. On the other hand, the BBC dataset corresponding to the business domain obtained the higher proportion of information represented. This is due to the kind of data from the domain, which contains several relations between well-known companies/organizations, people and places.

5.3 RDF Qualitative Evaluation

After counting the number of represented RDF triples, the following step consisted of evaluating the quality of such triples. Hence, this subsection presents a qualitative evaluation based on the strategy proposed by Dutta *et al.* [9], in which a set of triples is presented to a human judge for evaluation. Thus, every element of the triple must be marked as correct (including the semantic relation) to deem the whole statement as precise. Details of this evaluation were as follows:

- This evaluation was conducted by four human judges from an IT-based engineering college. Given that the proposed representation method processes English sentences to represent events, the judges must have (at least) an intermediate level of English (e.g., to read and understand news in English). Likewise, judges have notions of the terminology and structure used for the RDF representation (e.g., RDF triples, thematic roles).
- A total of 50 events were randomly selected from the IT-news dataset (presented in Subsect. 5.1). Each event contains triples for describing elements such as Agent, Patient, Action/Predicate, Semantic Relation, and the original sentence.
- The selected events were presented to the judges via a web application, where every element of the event had to be judged as "Correct" or "Incorrect".

We obtained the precision values for the events evaluated by the four judges. The values obtained for each element were: Agent (0.72), Predicate/Action (0.89), Patient (0.64), Semantic Relation (0.82), Total (0.51). Note that the Total value refers to those cases where the event is marked as correct for all its elements. Additionally, the results of the agreement among judges were obtained through the kappa score [24] as follows: Agent (0.45), Predicate (0.65), Patient (0.31), Semantic Relation (0.70), and Total (0.38). This evaluation demonstrates that human judges depict a fair to a moderate agreement that the represented data is not given by chance [18].

5.4 Discussion

Although the evaluation of the accuracy of represented sentences could sometimes be guided by the subjectivity of the judges, the obtained results can be

also influenced by aspects such as the complexity of the evaluated sentences, the dependency of NLP and IE tools (that could be inaccurate), and the understanding of concepts by the judges. However, the recognition of entities for the Agent and Patient demonstrates encouraging results in comparison to other approaches with similar purpose [12]. Moreover, the implementation also demonstrates the capability of the methodology to cover the stages needed for the representation of n-ary statements. It is worth mentioning that there is a lack of gold standard datasets that limits a fair comparison regarding existing works. Thus, we plan to include a more consistent evaluation under a scenario that considers diverse approaches, domains, and type of extractions.

6 Conclusions

The information represented on the Semantic Web has been used in tasks related to question answering [26], semantic annotation [6], and information retrieval [16], to mention a few. Thus, the main motivation of this research work is to formally represent unstructured data on the Semantic Web in order to support the consumption and dissemination of information through the integration of tools from areas such as Information Retrieval (IR), Information Extraction (IE), Machine Learning, Natural Language Processing (NLP), among others. This paper presented a methodology for the representation of RDF binary and n-ary statements. This is based on the steps followed by general Relation Extraction and Linking (REL) approaches for obtaining named entities and relations to then link them using data and standards of the Semantic Web. As a proof of concept, we presented an implementation of the proposed methodology for the representation of RDF n-ary statements from plain text. The experiments demonstrate the feasibility of the proposed architecture for the representation of statements in terms of the number of represented triples and the factors influencing their quality. Moreover, we also noted that diverse standards and scenarios are needed for the evaluation of these types of representation approaches.

Acknowledgments. This work was funded in part by the Fondo SEP-Cinvestav, Project No. 229. We would like to thank the reviewers for their comments on this paper.

References

1. Antoniou, G., Groth, P.T., van Harmelen, F., Hoekstra, R.: A Semantic Web Primer, 3rd edn. MIT Press, Cambridge (2012)
2. Auer, S., Bryl, V., Tramp, S. (eds.): Linked Open Data - Creating Knowledge Out of Interlinked Data - Results of the LOD2 Project. LNCS, vol. 8661. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-319-09846-3
3. Auer, S., Lehmann, J., Ngonga Ngomo, A.-C., Zaveri, A.: Introduction to linked data and its lifecycle on the web. In: Rudolph, S., Gottlob, G., Horrocks, I., van Harmelen, F. (eds.) Reasoning Web 2013. LNCS, vol. 8067, pp. 1–90. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39784-4_1

4. Augenstein, I., Maynard, D., Ciravegna, F.: Distantly supervised web relation extraction for knowledge base population. Semant. Web **7**(4), 335–349 (2016). https://doi.org/10.3233/SW-150180
5. Augenstein, I., Padó, S., Rudolph, S.: LODifier: generating linked data from unstructured text. In: Simperl, E., Cimiano, P., Polleres, A., Corcho, O., Presutti, V. (eds.) ESWC 2012. LNCS, vol. 7295, pp. 210–224. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30284-8_21
6. Chabchoub, M., Gagnon, M., Zouaq, A.: Collective disambiguation and semantic annotation for entity linking and typing. In: Sack, H., Dietze, S., Tordai, A., Lange, C. (eds.) SemWebEval 2016. CCIS, pp. 33–47. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-319-46565-4_3
7. Cimiano, P.: Ontology Learning and Population from Text - Algorithms. Evaluation and Applications. Springer, Heidelberg (2006). https://doi.org/10.1007/978-0-387-39252-3
8. Del Corro, L., Gemulla, R.: Clausie: clause-based open information extraction. In: International Conference on World Wide Web, pp. 355–366. ACM (2013). https://doi.org/10.1145/2488388.2488420
9. Dutta, A., Meilicke, C., Stuckenschmidt, H.: Enriching structured knowledge with open information. In: Gangemi, A., Leonardi, S., Panconesi, A. (eds.) World Wide Web Conference (WWW), pp. 267–277. ACM (2015)
10. Exner, P., Nugues, P.: Entity extraction: from unstructured text to DBpedia RDF triples. In: The Web of Linked Entities Workshop (WoLE 2012), pp. 58–69. CEUR-WS (2012)
11. Fensel, D., et al.: Enabling Semantic Web Services. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-34520-6
12. Gangemi, A., Presutti, V., Recupero, D.R., Nuzzolese, A.G., Draicchio, F., Mongiovì, M.: Semantic web machine reading with FRED. Semant. Web **8**(6), 873–893 (2017). https://doi.org/10.3233/SW-160240
13. Gildea, D., Jurafsky, D.: Automatic labeling of semantic roles. Comput. Linguist. **28**(3), 245–288 (2002). https://doi.org/10.1162/089120102760275983
14. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space. Synthesis Lectures on the Semantic Web. Morgan & Claypool Publishers, San Rafael (2011). https://doi.org/10.2200/S00334ED1V01Y201102WBE001
15. Hernández, D., Hogan, A., Krötzsch, M.: Reifying RDF: what works well with wikidata? In: Liebig, T., Fokoue, A. (eds.) International Workshop on Scalable Semantic Web Knowledge Base Systems Co-located with ISWC, pp. 32–47. CEUR-WS.org (2015)
16. Waitelonis, J., Exeler, C., Sack, H.: Linked data enabled generalized vector space model to improve document retrieval. In: NLP & DBpedia Workshop in Conjunction with ISWC 2015. CEUR (2015)
17. Kertkeidkachorn, N., Ichise, R.: An automatic knowledge graph creation framework from natural language text. IEICE Trans. **101**(D(1)), 90–98 (2018). https://doi.org/10.1587/transinf.2017SWP0006
18. Landis, J.R., Koch, G.G.: The measurement of observer agreement for categorical data. Biometrics **33**, 159–174 (1977)
19. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J.R., Bethard, S., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: Annual Meeting of the Association for Computational Linguistics (ACL), pp. 55–60 (2014)

20. Martinez-Rodriguez, J.L., Hernandez, J., Lopez-Arevalo, I., Rios-Alvarado, A.B.: A strategy for the integration of named entity extraction and linking results. In: Proceedings of the 3rd International Workshop on Semantic Web 2018 Co-located with 15th International Congress on Information (INFO 2018), 7 March 2018, Havana, Cuba, pp. 13–20. CEUR-WS.org (2018)
21. Martinez-Rodriguez, J.L., Hogan, A., Lopez-Arevalo, I.: Information extraction meets the semantic web: a survey. Semant. Web J. (2018, to appear)
22. Milton, N.R.: Knowledge Acquisition in Practice: A Step-by-Step Guide. Springer, Heidelberg (2007). https://doi.org/10.1007/978-1-84628-861-6
23. Pinto, A.M., Oliveira, H.G., Alves, A.O.: Comparing the performance of different NLP toolkits in formal and social media text. In: 5th Symposium on Languages, Applications and Technologies, SLATE, pp. 3:1–3:16 (2016)
24. Randolph, J.J.: Free-marginal multirater kappa (multirater k [free]): An alternative to fleiss' fixed-marginal multirater kappa. In: Joensuu Learning and Instruction Symposium (2005)
25. Rusu, D., Fortuna, B., Mladenic, D.: Automatically annotating text with linked open data. In: Bizer, C., Heath, T., Berners-Lee, T., Hausenblas, M. (eds.) WWW2011 Workshop on Linked Data on the Web. CEUR-WS.org (2011)
26. Unger, C., Freitas, A., Cimiano, P.: An introduction to question answering over linked data. In: Koubarakis, M., et al. (eds.) Reasoning Web 2014. LNCS, vol. 8714, pp. 100–140. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10587-1_2