Enhancing the WordNet Exploration and Visualization in Neo4J with a Tag Cloud Based Approach

Enrico G. Caldarola[*][†], Antonio Picariello[*], Antonio M. Rinaldi[*][‡], Marco Sacco[†]
[*]Department of Electrical Engineering and Information Technologies
University of Naples Federico II, Napoli, Italy
[†]Institute of Industrial Technologies and Automation
National Research Council, Milano, Italy
[‡]IKNOS-LAB Intelligent and Knowledge Systems
University of Naples Federico II, LUPT
Email: enricogiacinto.caldarola@unina.it, antonio.picariello@unina.it, antoniomaria.rinaldi@unina.it, marco.sacco@itia.cnr.it

***Abstract*—Visualization is the closer phase to the users within the data life cycles phases, so, there is no doubt that an effective, efficient and impressive representation of the analysed data may result as important as the analytic process itself. Starting from previous experiences in importing, querying and visualizing WordNet database within Neo4J and Cytoscape and extending a previous work by the authors, this work aims at improving the WordNet Graph visualization by exploiting the features and concepts behind tag clouds. The objective of this study is twofold: first, we argue that the proposed visualization style is able to put order in the messy and dense structure of nodes and edges of WordNet, showing as much as possible information from the lexical database and in a clearer way; secondly, we think that the tag cloud approach applied to the synonyms rings reinforces the human cognition in recognizing the different usages of words in a language like English. The ultimate goal of this work is, on the one hand, to facilitate the comprehension of WordNet itself and, on the other hand, to investigate techniques and approaches to get more insights from the visual representation and analytics of large graph databases.**

Keywords–WordNet; Big Data; Data and Information Visualization; Neo4J; Graph Database; NoSQL.

I. INTRODUCTION

In this paper we extend a previous work in visualizing and exploring the WordNet lexical database [1], by using one of the most spread tool in the graph databases management systems landscape, namely Neo4J. At the end of this section, the major contributions to this work with respect to the original one have been illustrated, while in the following paragraph a panorama of the main concepts related to Information Visualization will be provided.

A subtle difference exists between *data* and *information*. The first are raw, they simply exist and have no significance beyond its existence (in and of themselves) [2]. Data are just numbers, bits of information, which '*...have no way of speaking for themselves. We speak for them. We imbue them with meaning.*' [3]. On the contrary, information is data that have been given meaning by way of relational connection, by providing context for them. Even more subtle is the distinction between *Data Visualization* and *Information Visualization*. If the main goal of the first one is to communicate information clearly and efficiently to users, involving the creation and study of the visual representation of data – i.e., "information that has been abstracted in some schematic form, including attributes or variables for the units of information" [4] – the main task of the second one is the study of (interactive) visual representations of abstract data to reinforce human cognition. The abstract data may include both numerical and non-numerical data, such as text and geographic information. Beyond Information Visualization, an other outgrowth field is *Visual Analytics* that can be defined as 'the science of analytical reasoning facilitated by interactive visual interfaces.' [5]. Today, in many spheres of human activity, massive sets of data are collected and stored. As the volumes of data available to various stakeholders such as business people or scientists increase, their effective use becomes more challenging. Keeping up to date with the flood of data, using standard tools for data management and analysis, is fraught with difficulty. The field of visual analytics seeks to provide people with better and more effective ways to understand and analyse these large datasets, while also enabling them to act upon their findings immediately, in real-time [6]. Thus, the challenges that the Big Data imperative [7], [8] imposes to data management severely impact on data visualization. The "bigness" of large data sets and their complexity in term of heterogeneity contribute to complicate the representation of data, making the drawing algorithms quite complex. Just to make an example, let us consider the popular social network Facebook, in which the nodes represent people and the links represent interpersonal connections; we note that nodes may be accompanied by information such as age, gender, and identity, and links may also have different types, such as colleague relationships, classmate relationships, and family relationships. These kind of techniques can be useful in enhanced recommender systems [9]. The effective representation of all the information at the same time is really challenging. The most common solution is to use visual cues, such as color, shape, or transparency to encode different attributes. In this regard, tag clouds are a popular method for representing variables of interest (such as popularity, frequency of occurrence of a term, and so on) in the visual appearance of the keywords themselves using text properties such as font size, weight, or color [10]. The objective of the study presented in this work aims at providing a visualization style able to put order in the messy and dense structure of nodes and edges of WordNet exploiting the features of the *Labeled Property Graph Data Model*, showing as much as possible information from the lexical database and in a clearer way. At the same time, we think that the tag cloud approach applied to the synonyms rings reinforces the human cognition in recognizing the different usages of words in a language like English and that can help the comprehension of WordNet itself and, on the other hand, the research of techniques and approaches to get more insights from the visual representation and analytics of large graph databases. As mentioned at the beginning, most of the content in this paper starts from a previous work of the authors [1], this one being an extension of the first one. In particular, this paper provides a more complete state-of-the-art section, an essential description of Neo4J Graph Management System and a description of the procedure to import WordNet in Neo4j more rich in details than that in [1]. Since the study conducted in this paper consists in the visual representation of WordNet as a large graph in Neo4j [11] and Cytoscape [12], a particular attention is paid to *Graph Visualization*, referring to other well-known works in the literature for a complete review of the techniques and theories in Information Visualization [13][14][15][16]. The reminder of the paper is organized as follows. After a review of the main works existing in the literature about graph visualization in Section II, Section III describes the WordNet meta-model and clarifies the ground concepts related to WordNet landscape Section IV describes how WordNet has been imported in Neo4J and its visualization in Cytoscape. Section V goes to the hearth of this work *rationale* by illustrating the way a tags cloud approach is used to effectively draw the graph of WordNet synonyms rings in Cytoscape. Finally, Section VI draws the conclusion summarizing the major findings and outlining future investigations.

II. Related works

In the most common sense of the term, a graph is an ordered pair ***G=(V,E)*** comprising a set ***V*** of vertices or nodes together with a set ***E*** of edges or lines, which are 2-element subsets of V (i.e., an edge is related with two vertices, and the relation is represented as an unordered pair of the vertices with respect to the particular edge). Graphs are traditional and powerful tools for visually representing sets of data and the relations among them by drawing a dot or circle for every vertex, and an arc between two vertices if they are connected by an edge. If the graph is directed, the direction is indicated by drawing an arrow. A graph drawing should not be confused with the graph itself (the abstract, non-visual structure) as there are several ways to structure the graph drawing. All that matters is which vertices are connected to which others by how many edges and not the exact layout. In practice it is often difficult to decide if two drawings represent the same graph. Depending on the problem domain some layouts may be better suited and easier to understand than others. The pioneering work of W. T. Tutte [17] was very influential in the subject of graph drawing, in particular he introduced the use of linear algebraic methods to obtain graph drawings. The basic graph layout problem is very simple: given a set of nodes with a set of edges, it only needs to calculate the positions of the nodes and draw each edge as curve. Despite the simplicity of the problem, to make graphical layouts understandable and useful is very hard. Basically, there are generally accepted aesthetic rules to draw a graph [18], which include: distribute nodes and edges evenly, avoid edge crossing, display isomorphic substructures in the same manner, minimize the bends along the edges. However, since it is quite impossible to meet all rules at the same time, some of them conflict with each other or they are very computationally expensive, practical graphical layouts are usually the results of compromise among the aesthetics. Another issue about graph layout is predictability. Due to the task of graph visualization, it is important and necessary to make the results of layout algorithm predictable [19], which means two different results of running the same algorithm with the same or similar data inputs should also look the same or alike. There exists different graph visualization layouts in literature, such as: the Tree Layout, the Space Division Layout, the Matrix Layout and the Spring Layout[20], to mention a few.

Below is a brief overview of graph layouts and visualization techniques grouped by categories.

A. Node-link layouts

To this category belong the following layouts:

1) Tree Layout: it uses links between nodes to indicate the parent-child relationships. A very satisfactory solution for node-link layout comes from Reingold et al. [21]. Their classical algorithm is simple, fast, predictable, and produces aesthetically pleasing trees on the plane. However, it makes use of screen space in a very inefficient way. In order to overcome this limitation, some compact tree layout algorithms have been developed to obtain more dense tree, while keeping the classical tree looks [22]. Eades [23] proposes another node-link layout called radial layout that recursively positions children of a sub-tree into a circular wedge shape according to their depths in the tree. Generally, radial views, including its variations [24], share a common characteristic: the focus node is always placed at the center of the layout, and the other nodes radiate outward on separated circles. Balloon layout [25] is similar to radial layout and are formed where siblings of subtrees are placed in circles around their father node. This can be obtained by projecting cone tree onto the plane.

2) Tree Plus Layout: since large graphs are much more difficult to handle than trees, tree visualization is often used to help users understand graph structures. A straightforward way to visualize graphs is to directly layout spanning trees for them. Munzner [26] finds a particular set of graphs called quasi-hierarchical graphs, which are very suitable to be visualized as minimum spanning trees. However, for most graphs, all links are important. It could be very hard to choose a representative spanning tree. Arbitrary spanning trees can also possibly deliver misleading information.

3) Spring Layout: this layout, also known as *Force-Directed* layout, is another popular strategy for general graph layouts. In spring layout, graphs are modelled as physical systems of rings or springs. The attractive idea about spring layout is that the physical analogy can be very naturally extended to include additional aesthetic information by adjusting the forces between nodes. As one of the first few practical algorithms for drawing general graphs, spring layout is proposed by Eades in 1984 [27]. Since then, his method is revisited and improved in different ways [20], [28]. Mathematically, Spring layout is based on a cost (energy) function, which maps different layouts of the same graph to different non-negative numbers. Through approaching the minimum energy, the layout results reaches better and better aesthetically pleasing results. The main differences between different spring approaches are in the choice of energy functions and the methods for their minimization.

B. Space Division Layout

In this case, the parent-child relationship is indicated by attaching child node(s) to the parent node. Since the parent-child and sibling relationships are both expressed by adjacency, the layout should have a clear orientation cue to differentiate these two relationships.

1) Space Nested Layout: nested layouts, such as Treemaps [29], draw the hierarchical structure in the nested way. They place child nodes within their parent node.

C. 3D Layout

In this case, the extra dimension can give more space and it would be easier to display large structures. Moreover, due to the general human familiarity with 3D in the real world, there are some attempts to map hierarchical data to 3D objects we are familiar with.

D. Matrix Layout

Graphs can be presented by their connectivity matrixes. Each row and each column corresponds to a node. The glyph at the interaction of (i, j) encodes the edge from node i to node j. Edge attributes are encoded as visual characteristics of the glyphs, such as color, shape, and size. The major benefit of adjacency matrices is the scalability.

In this work the Spring layout will be used as it represents one of the most spread strategy in graph visualization and shows aesthetically pleasing results.

Specifically concerning the visualization of WordNet, there are not many works in the literature. In [30], the authors make an attempt to visualize the WordNet structure from the vantage point of a particular word in the database, this in order to overcome the down-side of the large coverage of WordNet, i.e., the difficulty to get a good overview of particular parts of the lexical database. An attempt to apply design paradigms to generate visualizations that maximize the usability and utility of WordNet is made in [31], whereas, in [32] a radial, space-filling layout of hyponymy (IS-A relation) is presented with interactive techniques of zoom, filter, and details-on-demand for the task of document visualization, exploiting the WordNet lexical database. The visualization approach used in this work uses the Spring layout to draw the graph-based representation of WordNet in Cytoscape and a tag cloud-based strategy to represent the synonim rings from WordNet. Moreover, as a general rule the principled representation methodology we agree on is the *Visual Information Seeking Mantra* presented by Scheiderman in [33]. It can be summarized as follows: “overview first, zoom and filter, then details-on-demand”.

III. Neo4J WordNet Case Study

The case study presented in this paper consists in the *reification* of the WordNet database inside the Neo4J GraphDB. In the following subsections a description of wordNet and Neo4J is provided.

A. Neo4J

Neo4J is a scalable, native graph database purpose-built to leverage data and its relationships. It is developed by Neo Technology, Inc and its developers describe it as an ACID-compliant transactional database with native graph storage and processing. Neo4j is one of the most popular graph database according to db-engines.com. In Neo4j, everything can be modelled by the *Labeled Property Graph Data Model*, which is based on connected entities (the nodes) which can hold any number of attributes (key-value-pairs). Nodes can be tagged with labels representing their different roles in your domain. In addition to contextualizing node and relationship properties, labels may also serve to attach metadataindex or constraint informationto certain nodes. Relationships provide directed, named semantically relevant connections between two node-entities. A relationship always has a direction, a type, a start node, and an end node. Like nodes, relationships can have any properties. In most cases, relationships have quantitative properties, such as weights, costs, distances, ratings, time intervals, or strengths. As relationships are stored efficiently, two nodes can share any number or type of relationships without sacrificing performance. Note that although they are directed, relationships can always be navigated regardless of direction. One of the most appreciated feature of Neo4J Graph Management System is Cypher, a declarative, SQL-inspired language for describing patterns in graphs visually using an iconic, acscii-art language [34]. It allows to state what we want to select, insert, update or delete from our graph data without requiring us to describe exactly how to do it.

B. WordNet

WordNet [35][36] is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. In this context, we have defined and implemented a meta-model for the WordNet reification using a conceptualization as much as possible close to the way in which the concepts are organized and expressed in human language [37]. We consider concepts and words as nodes in Neo4J, whereas semantic, linguistic and semantic-linguistic relations become Noeo4J links between nodes. For example, the hyponymy property can relate two concept nodes (nouns to nouns or verbs to verbs); on the other hand a semantic property links concept nodes to concepts and a syntactic one relates word nodes to word nodes. Concept and word nodes are considered with *DatatypeProperties*, which relate individuals with a predefined data type. Each word is related to the represented concept by the ObjectProperty *hasConcept* while a concept is related to words that represent it using the ObjectProperty *hasWord*. These are the only properties able to relate words with concepts and vice versa; all the other properties relate words to words and concepts to concepts. Concepts, words and properties are arranged in a class hierarchy, resulting from the syntactic category for concepts and words and from the semantic or lexical type for the properties. The subclasses have been derived from the related categories. There are some union classes useful to define properties domain and codomain. We define some attributes for Concept and Word respectively: Concept *hasName* that represents the concept name; *Description* that gives a short description of concept. On the other hand Word has Name as attribute that is the word name. All elements have an ID within the WordNet offset number or a user defined ID. The semantic and lexical properties are arranged in a hierarchy. In Table I some of the considered properties and their domain and range of definition are shown.

Figures 1(a) and 1(b) show that the two main classes are: Concept, in which all the objects have defined as individuals and Word which represents all the terms in the ontology.

The subclasses have been derived from the related categories. There are some union classes useful to define properties domain and co-domain. We define some attributes for Concept and Word respectively: Concept *hasName* that represents the concept name; *Description* that gives a short description of concept. On the other hand Word has Name as attribute that is the word name. All elements have an ID within the WordNet offset number or a user defined ID. The semantic and lexical properties are arranged in a hierarchy (see figure 2(a) and 2(b)). In Table I some of the considered properties and their domain and range of definition are shown.

TABLE I. PROPERTIES

Property	**Domain**	**Range**
hasWord	Concept	Word
hasConcept	Word	Concept
hypernym	NounsAnd VerbsConcept	NounsAnd VerbsConcept
holonym	NounConcept	NounConcept
entailment	VerbWord	VerbWord
similar	AdjectiveConcept	AdjectiveConcept

The use of domain and codomain reduces the property range application. For example, the hyponymy property is defined on the sets of nouns and verbs; if it is applied on the set of nouns, it has the set of nouns as range, otherwise, if it is applied to the set of verbs, it has the set of verbs as range. In Table II there are some of defined constraints and we specify on which classes they have been applied w.r.t. the considered properties; the table shows the matching range too.

TABLE II. MODEL CONSTRAINTS

Costraint	**Class**	**Property**	**Constraint range**
AllValuesFrom	NounConcept	hyponym	NounConcept
AllValuesFrom	AdjectiveConcept	attribute	NounConcept
AllValuesFrom	NounWord	synonym	NounWord
AllValuesFrom	AdverbWord	synonym	AdverbWord
AllValuesFrom	VerbWord	also_see	VerbWord

Sometimes the existence of a property between two or more individuals entails the existence of other properties. For example, being the concept dog a hyponym of animal, we can assert that animal is a hypernymy of dog. We represent this characteristics in OWL, by means of property features shown in Table III.

TABLE III. PROPERTY FEATURES

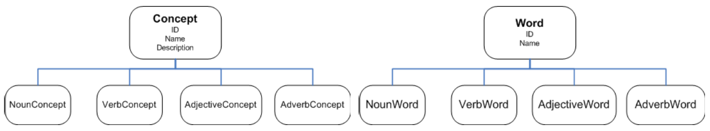
Property	**Features**
hasWord	*inverse* of hasConcept
hasConcept	*inverse* of hasWord
hyponym	*inverse* of hypernym; *transitivity*
hypernym	*inverse* of hyponym; *transitivity*
cause	*transitivity*
verbGroup	*symmetry* and *transitivity*

IV. Importing WordNet in Neo4J and Visualizing It in Cytoscape

The WordNet lexical database has been imported in Neo4J [38] and afterward visualized in Cytoscape according to a procedure similar to that described in a previous work by the authors [39]. The whole process is illustrated in Figure 3 and it involves three phases and three components: the *importing from WordNet* module, the *serializer* module and the *importing within Neo4J* module. The first phase has been implemented using a Java-based script that access the WordNet database through JWI (MIT Java Wordnet Interface) API [40][41] and passes all the information related to synsets, words, semantic relations and lexical relations to the serializer module, producing appropriate serialized data, following a proper schema that will be described in the following. This one, the main actor of the second phase, serializes the wordnet database into five csv files, suitable to be efficiently and effectively imported into Neo4J database. The last component, which is related to the third phase of the process, is responsible for importing the previously serialized information into Neo4J database. The importing from WordNet takes place via five different sub-operations which respectively retrieve: the information related to synsets, the semantic relations among synsets, the words, the lexical relations among words and finally the links between the semantic and the lexical world, i.e., how a word is related to its concepts (or its meaning) and *viceversa*.

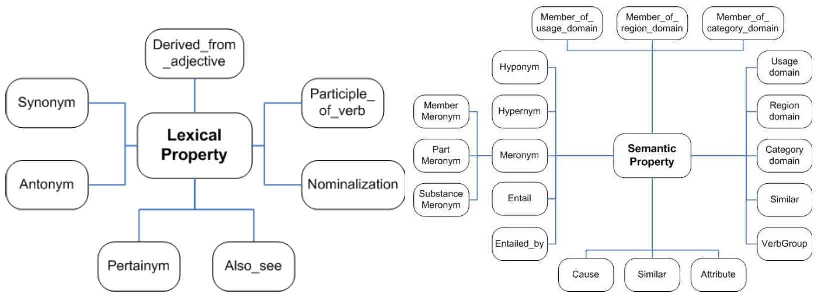
The intentional schema of each serialized data is shown as follow:

1) The synset file contains the following fields:
    a) *Id*: the univoque indentifier for the synset;
    b) *SID*: the Synset ID as reported in the WordNet database;
    c) *POS*: the synset's part of speech;
    d) *Gloss*: the synset's gloss, which express its meaning.
2) The semantic relations file contains the following fields:
    a) *Prop*: the semantic relation linking the source and the destination synsets;
    b) *Src*: the source synset;
    c) *Dest*: the destination synset;
3) The words file contains the following fields:
    a) *Id*: the univoque indentifier for the word;
    b) *WID*: the Word ID as reported in the WordNet database;
    c) *POS*: the word's part of speech;
    d) *Lemma*: lexical represenntation of the word;
    e) *SID*: the synset Id whose the word is related.
4) The lexical relations file contains the following fields:
    a) *Prop*: the lexical relation linking the source and the destination words;
    b) *Src*: the source word;
    c) *Dest*: the destination word;
5) The lexical-semantic relations file contains the following fields:
    a) *Word Id*: the word id of the word that is linked to the synset on the right via the *hasConcept* relation;



(a) Concept (b) Word

Figure 1. Concept and Word



(a) Lexical Properties (b) Semantic Properties

Figure 2. Linguistic properties

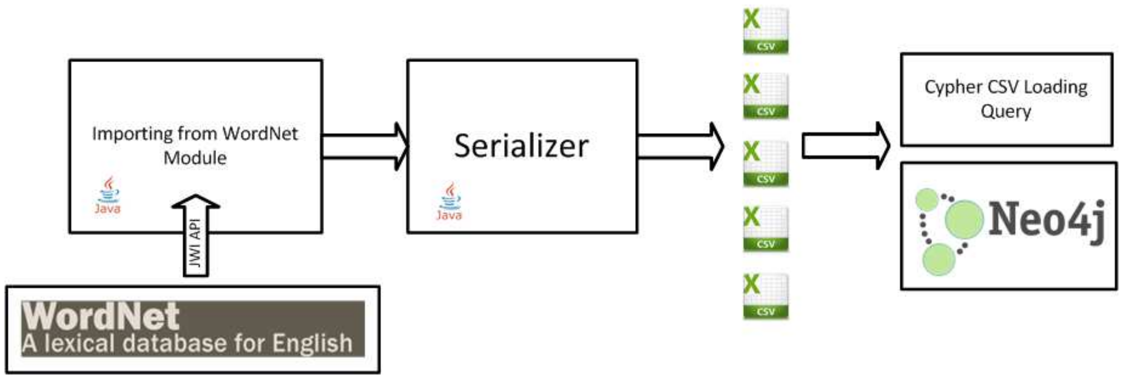


Figure 3. WordNet importing schema

b) *Synset Id*: the synset id of the synset that is linked to the word on the left via the *hasWord* relation;;

In order to import all the information contained in the serialized data and translate them into a graph data structure, the meta-model described in the previous section has been used: each synset and word has been converted into a node of the graph with label respectively: *Concept* and *Word*. Each semantic relation has become an edge between two concept nodes with the *type* property expressing the specific semantic relation holding between the concepts. Each lexical relation has been converted into an edge between two word nodes with a type property expressing the specific lexical relation between the word nodes. Finally, the word nodes have been connected to their related concept nodes through the *hasConcept* relation.

The Cypher query code used to import all the serialized information stored into csv lines is shown as follows:

```
USING PERIODIC COMMIT 1000
LOAD CSV WITH HEADERS FROM "PATH_TO_THE_FIRST_FILE"
AS csvLine

CREATE (c: Concept {
 id: toInt(csvLine.id),
 sid: csvLine.SID, POS:
 csvLine.POS,
 gloss: csvLine.gloss })

CREATE CONSTRAINT ON (c: Concept)
ASSERT c.id IS UNIQUE

USING PERIODIC COMMIT 1000
LOAD CSV WITH HEADERS FROM "PATH_TO_THE_SECOND_FILE"
AS csvLine
MATCH (src:Concept { id: toInt(csvLine.Src)}),
 (dest:Concept { id: toInt(csvLine.Dest)})

CREATE (src)-[:semantic_property
{ type: csvLine.Prop }]->(dest)

USING PERIODIC COMMIT 1000
LOAD CSV WITH HEADERS FROM "PATH_TO_THE_THIRD_FILE"
AS csvLine

CREATE (w: Word {
 id: toInt(csvLine.id),
 wid: csvLine.WID,
 POS: csvLine.POS,
 lemma: csvLine.lemma,
 sid: toInt(csvLine.SID) })

CREATE CONSTRAINT ON (w: Word)
ASSERT w.id IS UNIQUE

USING PERIODIC COMMIT 1000
LOAD CSV WITH HEADERS FROM "PATH_TO_THE_FOURTH_FILE"
AS csvLine

MATCH (src:Word { id: toInt(csvLine.Src)}),
 (dest:Word { id: toInt(csvLine.Dest)})

CREATE (src)-[:lexical_property
{ type: csvLine.Prop }]->(dest)
```

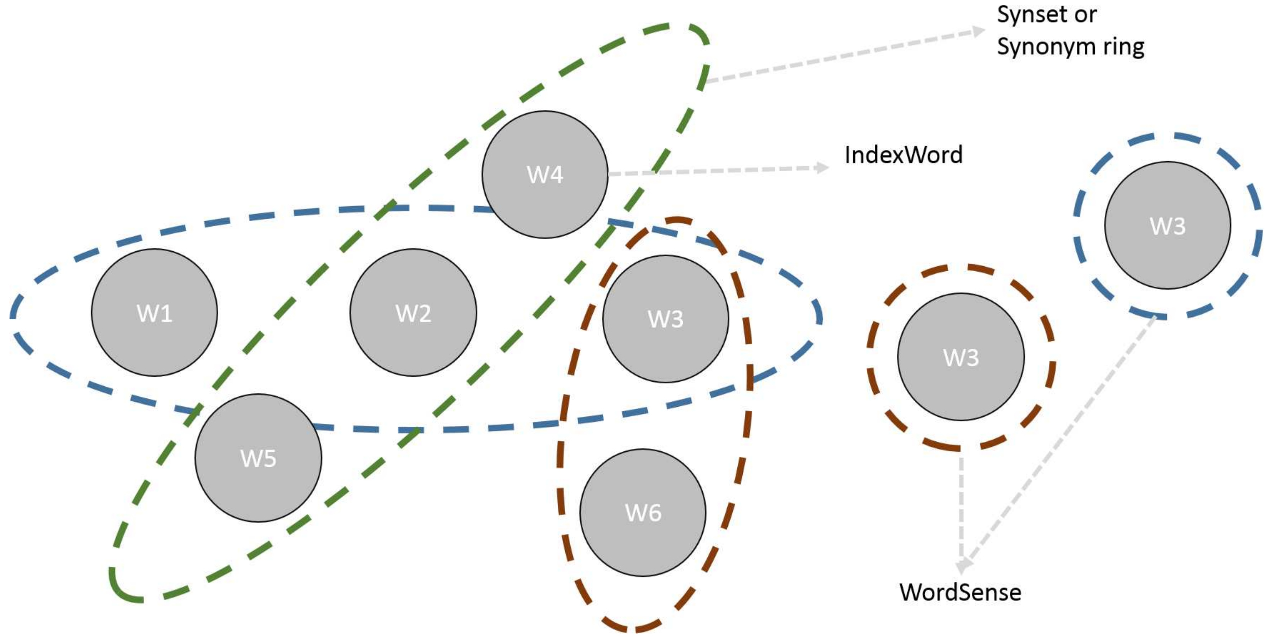


Figure 4. WordNet synsets, index words and word senses.

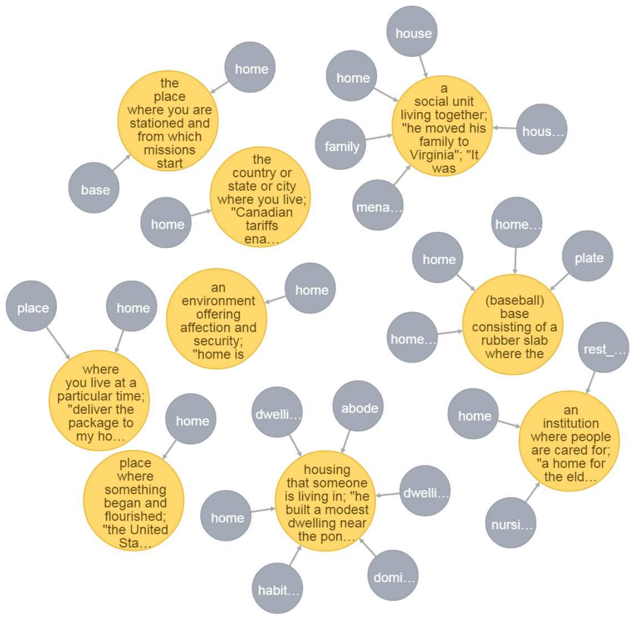
```
USING PERIODIC COMMIT 1000
LOAD CSV WITH HEADERS FROM "PATH_TO_THE_FIFTH_FILE"
AS csvLine

MATCH (src:Word { id: toInt(csvLine.Word)}),
(dest:Concept { id: toInt(csvLine.SID)})
CREATE (src)-[:hasConcept]->(dest)
```

Having applied the importing procedure described above, this work focuses on the visualization of WordNet and the most expensive part of the work has consisted in defining a Cytoscape custom style to represent the *synonyms rings* as tag clouds in an effective and clear way. This surely represents the novelty of this approach. We preferred to load WordNet objects from JWI APIs and serialize them in custom csv files, which were then imported throughout Cypher macros, instead of using already existing WordNet RDF serialization [42], because, this way, we could add some useful information in the csv lines like the *word frequency*, the *polysemy*, and so forth, for the sake of the successive representation in Cytoscape. And that is also why we prefer to create a custom tool to import the WordNet database in Neo4J instead of using already existing tools. Before diving into the procedure details, it is worth to clarify the distinction and provide some useful definitions coming from JWI APIS about *synsets*, *synsets (or synonyms) rings*, *index words* and *word senses*. Figure 4 try to put light on this. As discussed in the previous section, a synset is a concept, i.e., an entity of the real world (both physical or abstract) meaning something whose meaning can be argued by reading the *gloss* definition provided by WordNet. Its meaning can be also understood by analysing the semantic relations linking it to other synsets or by the synset (or synonyms) ring. This one is a set of words (hereafter mentioned as index words) generally used in a specific language (such as English) to refer to that concept. The term synset itself is used to refer to set of synonyms meaning a specific concept. On the contrary, an index word is just a term, i.e., a *sign* without meaning; so that, only when we link it to a specific concept we obtain a word sense, i.e., a word provided with a meaning. An index word has got different meanings according to the context in which it is used and because of a general characteristic of languages: the *polysemy*. For example, the term *home* has nine different meanings if it is used as noun, and so, it belongs to nine different synsets. In fact, the WordNet answer when we search for *home* is the following:

```
1. (430) home, place -- (where you live at a particular time; "deliver the
      package to my home"; "he doesn't have a home to go to"; "your place or
      mine?")
2. (350) dwelling, home, domicile, abode, habitation, dwelling house -- (housing
       that someone is living in; "he built a modest dwelling near the pond"; "
      they raise money to provide homes for the homeless")
3. (116) home -- (the country or state or city where you live; "Canadian tariffs
       enabled United States lumber companies to raise prices at home"; "his
      home is New Jersey")
4. (43) home -- (an environment offering affection and security; "home is where
      the heart is"; "he grew up in a good Christian home"; "there's no place
      like home")
5. (38) home, nursing home, rest home -- (an institution where people are cared
      for; "a home for the elderly")
6. (36) base, home -- (the place where you are stationed and from which missions
       start and end)
7. (7) family, household, house, home, menage -- (a social unit living together;
       "he moved his family to Virginia"; "It was a good Christian household";
      "I waited until the whole house was asleep"; "the teacher asked how many
      people made up his home")
8. (7) home plate, home base, home, plate -- ((baseball) base consisting of a
      rubber slab where the batter stands; it must be touched by a base runner
      in order to score; "he ruled that the runner failed to touch home")
9. (3) home -- (place where something began and flourished; "the United States
      is the home of basketball")
```

In addition to synsets glosses, WordNet gives us some useful statistic information about the usage of the term *home* in each synset. The position of the term in each synonyms ring tell us how usual is the use of the term to signify that concept. The position of the term in each synset is a measure of the usage frequency of the term for each concept: higher the position, higher the frequency. Moreover, by counting the number of synsets which a term belongs to, it is possible to obtain its polysemy (e.g., the number of possible meanings of *home*). JWI is able to tell us all this information about synset and word senses. In particular, for each synset we have collected the following fields in the csv files:

1) *Id*: the univoque indentifier for the synset;
2) *SID*: the Synset ID as reported in the WordNet



house
home
home
the place where you are stationed and from which missions start
a social unit living together; "he moved his family to Virginia"; "It was
hous…
family
base
the country or state or city where you live; "Canadian tariffs ena…
mena…
home
home…
home
plate
an environment offering affection and security; "home is
home
place
home
(baseball) base consisting of a rubber slab where the
home…
rest_…
where you live at a particular time; "deliver the package to my ho…
dwelli…
abode
home
an institution where people are cared for; "a home for the eld…
home
place where something began and flourished; "the United Sta…
housing that someone is living in; "he built a modest dwelling near the pon…
dwelli…
home
nursi…
habit…
domi…

Figure 5. WordNet synset rings containing the ’home’ word

database;
3) *POS*: the synset’s part of speech (POS);
4) *Gloss*: the synset’s gloss which express its meaning;
5) *Level*: the hieararchical level of synset in the whole WordNet hierarchy.

For word senses we have collected the following fields:

1) *Id*: the univoque indentifier for the word sense;
2) *POS*: the word’s part of speech (POS);
3) *polysemy*: the word polysemy;
4) *frequency*: the word frequency of the word sense as previously explicated.

A third csv file stores the semantic links existing between synset by reporting the IDs of the source and target synset and the type of semantic link existing between them, such as hypernym, hyponym, meronym, etc.

In addition to the previous files, a final file lists the links between each word sense and each synset. This file is very simple, it just contains a line for each pair (Word Sense, Synset) in WordNet. Other minor and not significant fields have been added for the sake of the visualization in Cytoscape, such as *label* (a human readable label for the nodes) and *dimension* (used to suggest a plausible diameter for the Synset node representation according to its depth in the WordNet hierarchy). The code to convert WordNet synsets into csv tables is available at https://github.com/eureko/WordNetToCSVFiles/.

In order to import all the information contained in the csv files and translate them into a graph data structure inside Neo4J [11], the meta-model described in Section II has been used. Each synset and word sense have been converted into a node of the graph with label respectively: *Synset* and *WordSense*. Each semantic relation has become an edge between two synset nodes with the *type* property expressing the specific semantic relation holding between the concepts. Finally, the word sense nodes have been connected to their related concepts nodes through a specific relation. This allows to effectively represent synonyms ring through the Neo4J web visualizer. For example, Figure 5 shows the results of the following Cypher query:

```
match (a: WordSense {POS: ’NOUN’})-[r]->(c: Synset)
        where (c)<-[]-(: WordSense {label: ’home’})
                return a,r,c
```

The figure reports nine synset rings for the term *home*. The filled circles represents the synset and contain the synset gloss definition, while the white circles around contain the word terms used to signify such synset.

V. The Tag Cloud-Based Representation of WorNet Synonyms Rings

The work described in this paper has encountered challenges that are quite close to the typical Big Data scenario. In fact, this version of WordNet graph (v. 2.1) includes 117597 synsets rings containing 207106 word senses conveyed by 155327 index words, 283837 semantic relations (cfr., Section III) linking synsets each other and 207016 semantic-lexical links between index words and synsets. With these big numbers, the manipulation, the querying and the visualization of the graph become quite challenging. The visualization of the entire structure of WordNet in terms of all synsets, words, semantic and lexical relations in a way that is elegant and intelligible at the same time, is a *chimera*, due to the performance issues of the visualization tools, in particular when sophisticated drawing algorithms are used, and to the strongly connected nature of information to be represented, which often results in a messy and dense structure of nodes and edges. Just to have an idea, Figure 6 shows a representation of an excerpt of WordNet (5000 semantic relations over 3404 synsets) obtained from *Cytoscape* v.3 graph visualization tool. The Neo4j running instance has been accessed via a specific plug-in, namely cyNeo4j, that converts the query results into Cytoscape table format and then create a view according to a custom style and a selected layout like the *Force-directed graph drawing algorithm* before mentioned. The resulting figure is more considerable for global analysis, or for its look and feel, than for actual information that you can retrieve from it. Nevertheless, thanks to the force-directed algorithm, it is possible to observe agglomerates of nodes and edges, which correspond to specific semantic categories and can help users in zooming the desired semantic area.

Thus, it is necessary to simplify the representation of the network by following some functional and esthetic criteria. In this regard, we have selected some simple representation criteria, listed as follows:

1) the efficiency of the visualization, i.e., avoid the information redundancy and the proliferation of useless signs and graphics as much as possible;
2) the effectiveness of the visualization, i.e., grant that the graphical representation of the network covers the whole informative content of the WordNet graph-based implementation;
3) the clearness of visualization, i.e., use light colors, such as gray, light blue, dark green, etc. with a proper level of brightness and with an appreciable contrast.

Furthermore, the adoption of tag cloud based representation for the synonyms rings brought us to use the statistical linguistics measures of *polisemy* and *frequency* of a term as visual cues in drawing the word signs attached to a certain synset.

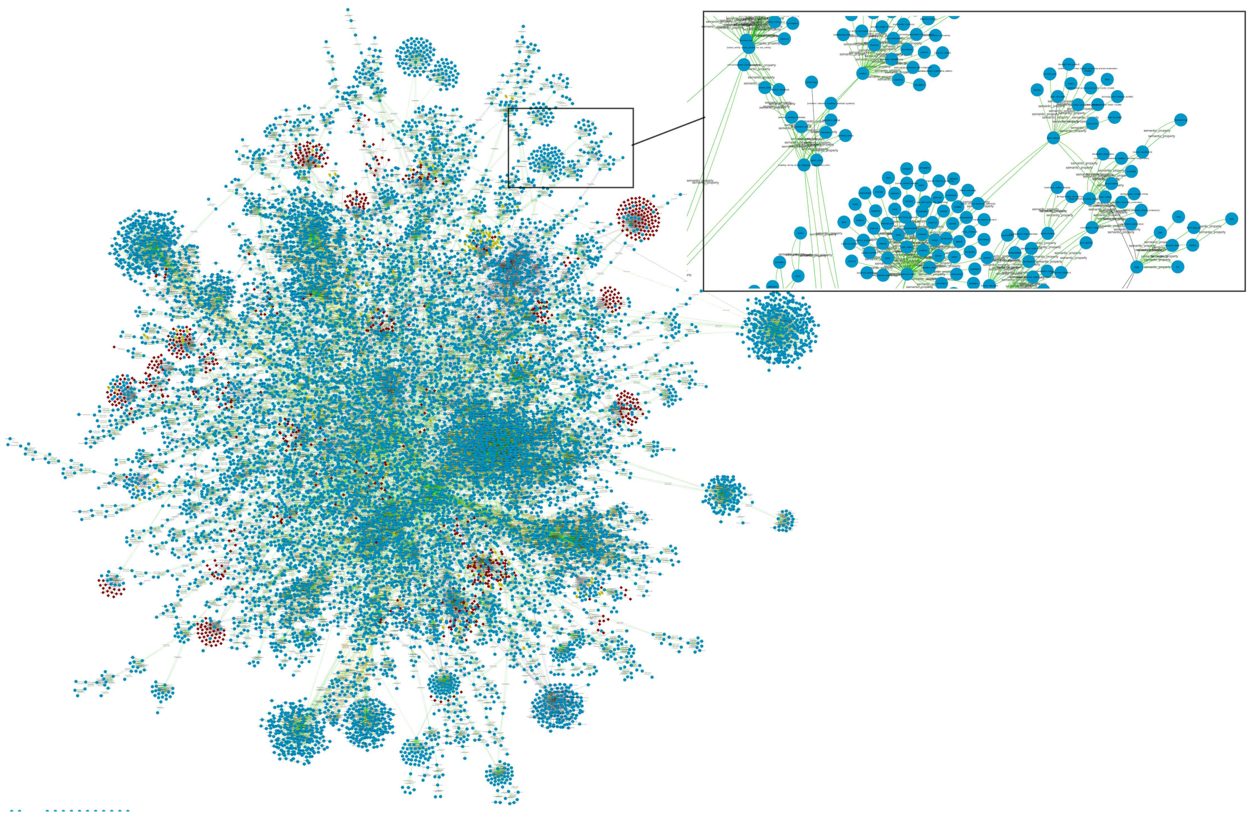


Figure 6. Large scale representation of 5000 relations and 3404 synsets in WordNet

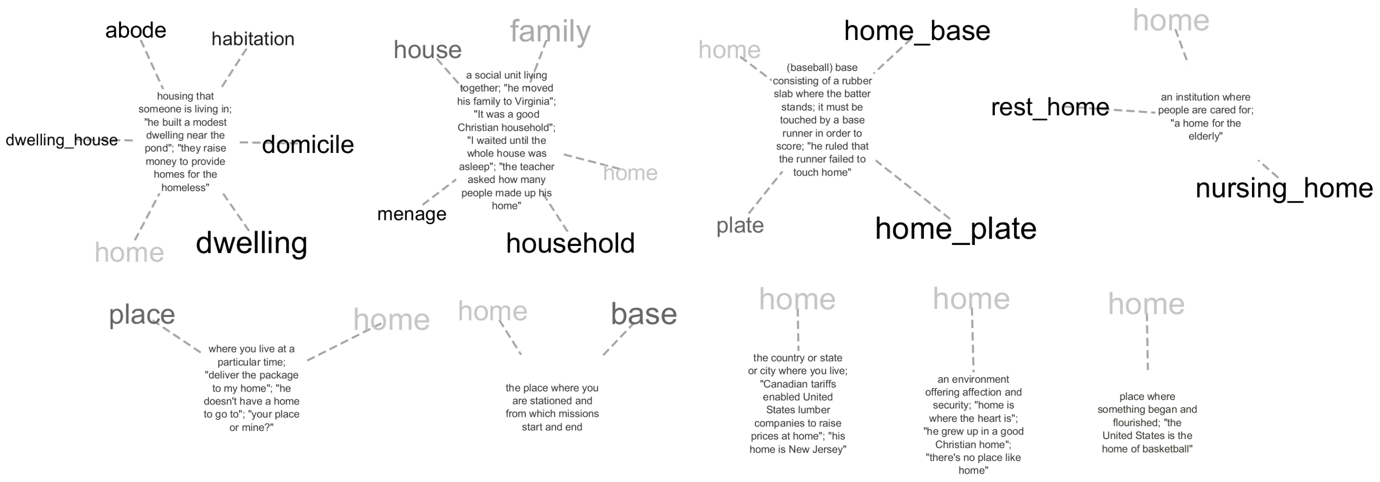
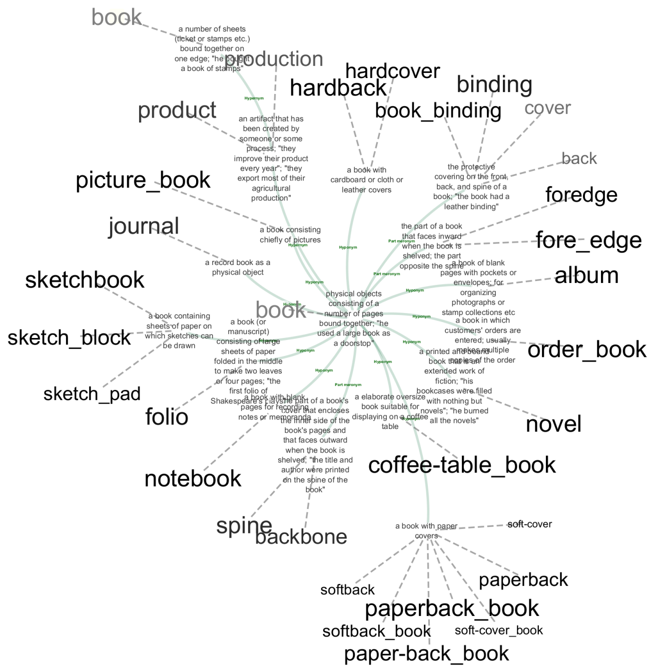
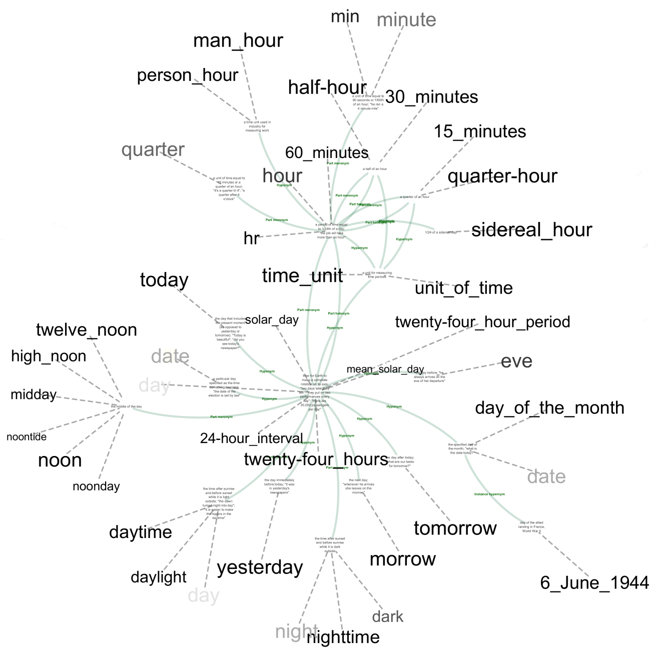

abode
habitation
dwelling_house
domicile
housing that someone is living in; "he built a modest dwelling near the pond"; "they raise money to provide homes for the homeless"
home
dwelling
house
family
a social unit living together; "he moved his family to Virginia"; "It was a good Christian household"; "I waited until the whole house was asleep"; "the teacher asked how many people made up his home"
home
menage
household
home
home_base
(baseball) base consisting of a rubber slab where the batter stands; it must be touched by a base runner in order to score; "he ruled that the runner failed to touch home"
plate
home_plate
home
rest_home
an institution where people are cared for; "a home for the elderly"
nursing_home
place
home
where you live at a particular time; "deliver the package to my home"; "he doesn't have a home to go to"; "your place or mine?"
home
base
the place where you are stationed and from which missions start and end
home
the country or state or city where you live; "Canadian tariffs enabled United States lumber companies to raise prices at home"; "his home is New Jersey"
home
an environment offering affection and security; "home is where the heart is"; "he grew up in a good Christian home"; "there's no place like home"
home
place where something began and flourished; "the United States is the home of basketball"

Figure 7. WordNet synset rings containing the 'home' word with the customized style in Cytoscape

Technically, higher the frequency of the word sense, larger is the font used to represent such word in the corresponding synonym ring, as well as, higher is the polysemy of a word in the whole WordNet, lighter is the shade of gray used to tag such word. All the word senses are connected to the corresponding synset through a blank gray line and each synset is represented through a short text containing its gloss. Semantic relations between synsets are represented through a transparent green arc showing a label that report the type of the semantic link (e.g., hypernym, hyponim, meronym, etc.). Figure 7 shows the application of the style rules described above to the same cypher query from *home* mentioned in the previous section. For each sense of the term *home*, the figure shows the tag cloud based representation. Some considerations arise from the figure above. The lighter gray used for the term 'home' is due to its high polysemy (9). This color is intentionally weak to demonstrate how vague is the term alone without a context making it meaningful. Things get worse, for



(a) Something about *book*.



(b) Something about *time*.

Figure 8. WordNet excerpts in Cytoscape with custom style.

example, with terms like *head* or *line* with a polysemy equal to 33 and 29, respectively. On the contrary, the term *home_plate* is large in size and as a strong gray shade because of its low polysemy (1) and high frequency when used in the context of baseball.

Figures 8(a) and 8(b) show more representations of WordNet excerpts to fully demonstrating the customized style resulting from this work. The figure are obtained through the following Cypher query where 'keyword' is substituted with *book* and *time*:

```
MATCH (a: WordSense {label: '<keyword>'})-[r]->
        (b: Synset)-[t: semantic_property]->
                (f: Synset)<-[s]-(c: WordSense)
        return a,r,b,t,f,s,c
```

The figures above also highlights the semantic relations existing between synsets showing a more complete representation of WordNet with the new visualization style described in this work.

VI. Conclusion and Future Work

Starting from previous experiences in importing, querying and visualizing WordNet in Neo4J and Cytoscape, a tag cloud based approach has been proposed in this paper as a new solution to make more effective and intelligible the representation of the WordNet graph. The results shown in this work are twofold: first, the new visualization style is able to put order in the messy and dense structure of nodes and edges of WordNet, showing as much as possible information from the lexical database and in a clearer way; secondly, the tag cloud approach is able to reinforce the human cognition in recognizing the different usages of words in English, w.r.t. the concepts they are related to. In fact, the proposed solution not only shows the synsets and the semantic relations holding between them, but also gives clues about the frequency of use of the synonyms for each synset. Future investigation may surely go in the direction of improving the criteria to simplify the WordNet representation with an evaluation for the visualization methods also validated by usability tests in which the user can express a consensus whether the representation is friendly or not, and the information inside WordNet is easily accessible or not. Finally, according to other studies, which aim at improving the tag cloud with semantics [43] and adding multimedia information to the knowledge representation model [44], we will investigate on the use of semantic properties and more efficient metrics to measure the relatedness among WordNet terms [45], also applying other visual features to combine these information [46] and improve the quality of WordNet visualization. Moreover, the process of ontology integration using general and domain specific ontologies [47], [48] needs the investigation of more efficient visualization tool to improve the understanding of very large knowledge base content and structure.

References

[1] E. G. Caldarola, A. Picariello, and A. M. Rinaldi, "Wordnet exploration and visualization in neo4j, a tag cloud based approach," in Proceedings of the Eighth International Conference on Information, Process, and Knowledge Management. IARIA, 2016, pp. 94–99.

[2] G. Bellinger, D. Castro, and A. Mills, "Data, information, knowledge, and wisdom," 2004.

[3] N. Silver, The signal and the noise: Why so many predictions fail-but some don't. Penguin, 2012.

[4] M. Friendly, "Milestones in the history of data visualization: A case study in statistical historiography," in Classificationthe Ubiquitous Challenge. Springer, 2005, pp. 34–52.

[5] J. J. Thomas and K. A. Cook, Illuminating the Path: The R&D Agenda for Visual Analytics National Visualization and Analytics Center. National Visualization and Analytics Center - U.S. Department of Homeland Security, 2005.

[6] D. A. Keim, J. Kohlhammer, G. Ellis, and F. Mansmann, Mastering the information age-solving problems with visual analytics. Florian Mansmann, 2010.

[7] E. G. Caldarola, A. Picariello, and D. Castelluccia, "Modern enterprises in the bubble: Why big data matters," ACM SIGSOFT Software Engineering Notes, vol. 40, no. 1, 2015, pp. 1–4.

[8] E. G. Caldarola, M. Sacco, and W. Terkaj, "Big data: The current wave front of the tsunami," ACS Applied Computer Science, vol. 10, no. 4, 2014, pp. 7–18.

[9] V. Moscato, A. Picariello, and A. M. Rinaldi, "A recommendation strategy based on user behavior in digital ecosystems," in Proceedings of the International Conference on Management of Emergent Digital EcoSystems. ACM, 2010, pp. 25–32.

[10] S. Bateman, C. Gutwin, and M. Nacenta, "Seeing things in the clouds: the effect of visual features on tag cloud selections," in Proceedings of the nineteenth ACM conference on Hypertext and hypermedia. ACM, 2008, pp. 193–202.

[11] Neo Technology Inc. Neo4j: The world's leading graph database. [Online]. Available: http://neo4j.com (2016)

[12] M. E. Smoot, K. Ono, J. Ruscheinski, P.-L. Wang, and T. Ideker, "Cytoscape 2.8: new features for data integration and network visualization," Bioinformatics, vol. 27, no. 3, 2011, pp. 431–432.

[13] R. Spence, Information visualization. Springer, 2001, vol. 1.

[14] R. Mazza, Introduction to information visualization. Springer Science & Business Media, 2009.

[15] U. M. Fayyad, A. Wierse, and G. G. Grinstein, Information visualization in data mining and knowledge discovery. Morgan Kaufmann, 2002.

[16] C. Ware, Information visualization: perception for design. Elsevier, 2012.

[17] W. T. Tutte, "How to draw a graph," Proc. London Math. Soc, vol. 13, no. 3, 1963, pp. 743–768.

[18] H. Purchase, "Which aesthetic has the greatest effect on human understanding?" in Graph Drawing. Springer, 1997, pp. 248–261.

[19] I. Herman, M. Delest, and G. Melancon, "Tree visualisation and navigation clues for information visualisation," Computer Graphics Forum, vol. 17, no. 2, 1998, pp. 153–165.

[20] T. M. Fruchterman and E. M. Reingold, "Graph drawing by force-directed placement," Softw., Pract. Exper., vol. 21, no. 11, 1991, pp. 1129–1164.

[21] E. M. Reingold and J. S. Tilford, "Tidier drawings of trees," Software Engineering, IEEE Transactions on, no. 2, 1981, pp. 223–228.

[22] L. Beaudoin, M.-A. Parent, and L. C. Vroomen, "Cheops: A compact explorer for complex hierarchies," in Visualization'96. Proceedings. IEEE, 1996, pp. 87–92.

[23] W. Huang, S.-H. Hong, and P. Eades, "Effects of sociogram drawing conventions and edge crossings in social network visualization." J. Graph Algorithms Appl., vol. 11, no. 2, 2007, pp. 397–429.

[24] G. J. Wills, "Nicheworksinteractive visualization of very large graphs," in Graph Drawing. Springer, 1997, pp. 403–414.

[25] J. Carriere and R. Kazman, "Research report. interacting with huge hierarchies: beyond cone trees," in Information Visualization, 1995. Proceedings. IEEE, 1995, pp. 74–81.

[26] T. Munzner, "H3: Laying out large directed graphs in 3d hyperbolic space," in Information Visualization, 1997. Proceedings., IEEE Symposium on. IEEE, 1997, pp. 2–10.

[27] P. Eades, "A heuristics for graph drawing," Congressus numerantium, vol. 42, 1984, pp. 146–160.

[28] E. R. Gansner and S. C. North, "Improved force-directed layouts," in Graph Drawing. Springer, 1998, pp. 364–373.

[29] B. Johnson and B. Shneiderman, "Tree-maps: A space-filling approach to the visualization of hierarchical information structures," in Visualization, 1991. Visualization'91, Proceedings., IEEE Conference on. IEEE, 1991, pp. 284–291.

[30] J. Kamps and M. Marx, "Visualizing wordnet structure," Proc. of the 1st International Conference on Global WordNet, 2002, pp. 182–186.

[31] C. Collins, "Wordnet explorer: applying visualization principles to lexical semantics," Computational Linguistics Group, Department of Computer Science, University of Toronto, Toronto, Ontario, Canada, 2006.

[32] ——, "Docuburst: Radial space-filling visualization of document content," Knowledge Media Design Institute, University of Toronto, Technical Report KMDI-TR-2007-1, 2007.

[33] B. B. Bederson and B. Shneiderman, The craft of information visualization: readings and reflections. Morgan Kaufmann, 2003.

[34] Neo Technology Inc. Neo4j: Cypher query language. [Online]. Available: https://neo4j.com/developer/cypher/ (2016)

[35] C. Fellbaum, "Wordnet," The Encyclopedia of Applied Linguistics, 1998.

[36] G. A. Miller, "Wordnet: a lexical database for english," Communications of the ACM, vol. 38, no. 11, 1995, pp. 39–41.

[37] A. M. Rinaldi, "A content-based approach for document representation and retrieval," in Proceedings of the eighth ACM symposium on Document engineering. ACM, 2008, pp. 106–109.

[38] J. Webber, "A programmatic introduction to neo4j," in Proceedings of the 3rd annual conference on Systems, Programming, and Applications: Software for Humanity. ACM, 2012, pp. 217–218.

[39] E. Caldarola, A. Picariello, and A. M. Rinaldi, "Big graph-based data visualization experiences - the wordnet case study," in Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, 2015, pp. 104–115.

[40] M. A. Finlayson, MIT Java Wordnet Interface (JWI) User's Guide, Version 2.2.x, 2013.

[41] ——, "Java libraries for accessing the princeton wordnet: Comparison and evaluation," in Proceedings of the 7th Global Wordnet Conference, Tartu, Estonia, 2014.

[42] Wordnet RDF. [Online]. Available: http://wordnet-rdf.princeton.edu/ (2016)

[43] A. M. Rinaldi, "Improving tag clouds with ontologies and semantics," in IEEE International Workshop on Text-Based Information Retrieval (TIR'12) in conjunction with DEXA12. IEEE, 2012.

[44] ——, "A multimedia ontology model based on linguistic properties and audio-visual features," Information Sciences, vol. 277, 2014, pp. 234–246.

[45] ——, "An ontology-driven approach for semantic information retrieval on the web," ACM Transactions on Internet Technology (TOIT), vol. 9, no. 3, 2009, p. 10.

[46] ——, "Using multimedia ontologies for automatic image annotation and classification," in 2014 IEEE International Congress on Big Data. IEEE, 2014, pp. 242–249.

[47] A. Cataldo and A. M. Rinaldi, "An ontological approach to represent knowledge in territorial planning science," Computers, Environment and Urban Systems, vol. 34, no. 2, 2010, pp. 117–132.

[48] E. G. Caldarola, A. Picariello, and A. M. Rinaldi, "An approach to ontology integration for ontology reuse in knowledge based digital ecosystems," in Proceedings of the 7th International Conference on Management of Computational and Collective intElligence in Digital EcoSystems, ser. MEDES '15. New York, NY, USA: ACM, 2015, pp. 1–8. [Online]. Available: http://doi.acm.org/10.1145/2857218.2857219