Semi-automatic Tool for Ontology Learning Tasks

Ondřej Šebek[1](✉), Václav Jirkovský[1], Nestor Rychtyckyj[2], and Petr Kadera[1]

[1] Czech Institute of Robotics, Informatics, and Cybernetics, Czech Technical University in Prague, Prague, Czech Republic
{ondrej.sebek,vaclav.jirkovsky,petr.kadera}@cvut.cz
[2] Ford Motor Company, Dearborn, MI, USA
nrychtyc@ford.com

Abstract. The (semi-)automated integration of new information into a data model is a functionality which is required in cases when input documents are extensive and therefore a manual integration difficult or even impossible. We proposed an ontology learning procedure combining information acquisition from structured resources, such as WordNet or DBpedia, and unstructured resources using text mining techniques based on an evaluation of lexico-syntactic patterns. This approach offers a robust way, how to integrate even previously unknown information disregarding target application or domain. The proposed solution was implemented in the form of semi-automatic ontology learning tool used for integration of Excel document containing spare part records and Ford Supply Chain Ontology.

Keywords: Ontology · Ontology learning · Web mining · Text mining · Automotive

1 Introduction

This paper is motivated by a requirement for facilitating interoperability in supply chain management together with a formation of a new suitable data model (or improvement of existing one) represented in an ontology. This problem may be defined as deriving an ontology (or its part) from given data, also known as an ontology learning task [9].

The integration of new information into an existing data model is a problem which may be observed in many systems. The integration is necessary for example when a company replaces a supplier (information about spare parts has to be imported in a company system) or the company changes a production plan and needs to adapt the data model of the information system adequately. In all cases, we have to face a heterogeneity, which is caused by different designers of given concepts or by different target applications. Thus, the heterogeneity has to be resolved for ensuring proper information integration.

V. Mařík et al. (Eds.): HoloMAS 2019, LNAI 11710, pp. 119–129, 2019.
https://doi.org/10.1007/978-3-030-27878-6_10

We proposed the solution which includes information acquisition from the Web (web mining) in the ontology learning process. This approach offers a robust way, how to integrate even previously unknown information disregarding target application or domain. The solution deals with facilitating identification of input data among existing concepts or with the definition of a new concept. The proposed solution was experimentally verified on the integration of Excel document containing spare parts and Ford Supply Chain Ontology.

The paper is organized as follows: Next section introduces the problem of ontology learning and some of the developed ontology learning methods. Section 3 describes our proposed approach to ontology learning. Application of the proposed method on the case of integration of spare part records into the Ford ontology is presented in Sect. 4. The paper concludes with a summary in Sect. 5.

2 State of the Art

The (semi-)automatic methods for ontology construction are typically referred to as the ontology learning altogether [8]. In other words, ontology learning deals with the construction of a domain model from available data.

Ontology learning can be often considered as a reverse engineering process. Implicit domain models that were created by a single or multiple authors represent given input data. The ontology learning procedure reconstructs the universal world model from these existing implicit models. The task is complex especially because only a small part of the authors' domain knowledge is involved in the data model creation process and furthermore a conceptualization (which is used by the author) is rarely mentioned explicitly.

The ontology learning process can be divided into several separated tasks:

1. Acquisition of appropriate vocabulary.
2. Identification of synonym terms and linguistic variants (possibly across languages).
3. Formation of concepts.
4. Hierarchical organization of the concepts.
5. Learning relations, properties or attributes, together with the appropriate domain and range.
6. Hierarchical organization of the relations.
7. Instantiation of axiom schemata.
8. Definition of arbitrary axioms.

These tasks with their exactly fixed order are typically referred to as the ontology learning layer cake. Various methods have been proposed to deal with particular steps of the ontology learning problem. However, the applicability of a particular method strongly depends on the target domain and the ontology learning is thus a very challenging task.

One of the approaches to the problem of semi-automated ontology learning is to employ structured datasets such as WordNet [10] or DBpedia[1]. WordNet is

[1] http://wiki.dbpedia.org.

a large lexical database of English, where nouns, verbs, and adverbs are grouped into sets of cognitive synonyms (named synsets). Every synset corresponds to a distinct concept. These synsets are interlinked by means of conceptual-semantic relations. DBpedia is a crowd-sourced community effort to extract structured content from the information created in various Wikimedia projects, i.e., DBpedia contains not only definitions of concepts but also relations between these concepts represented in some standard form (e.g., by using Dublin Core[2]). The existence of the relations between individual concepts in such datasets can be easily employed for the fourth and fifth layer of the ontology learning process outlined above.

Zhou et al. [13] proposed ontology learning method where the initial human-designed core ontology is expanded using semantic relations in WordNet. This process may be repeated until the resulting ontology is sufficiently rich. Luong et al. [7] developed a method for enriching concepts using WordNet. In their approach, correct sense of given WordNet term is evaluated on the basis of comparing hypernymy trees inferred from WordNet. Booshehri and Luksch [1,2] proposed the use of Linked Data as an additional tool (besides standard ontology generating from a text) to find new relations and thus obtain more expressive ontology.

Another approach to derive relations between concepts is based on the search of lexico-syntactic patterns in unstructured text. Such an approach was pioneered by Hearst in 1992 [4] for hyponymy/hypernymy relation. The idea behind this approach is to identify patterns which usually indicate specific semantic relation. Throughout the years this approach was also extended to other semantic relations (e.g., [3,11]). Example of ontology learning based on lexico-syntactic patterns may be found in [6].

3 Proposed Solution

Each of the ontology learning methods has its advantages and suitable domain of application. For example, the structured datasets (such as WordNet or DBpedia) already contain much valuable information, e.g., relations to concepts with broader or narrower sense, relations to meronyms and holonyms. On the other hand, these resources contain rather general concepts, and more specialized concepts may not be covered in these resources. Unstructured text resources provide much larger space for search but extraction of the required information is complicated, and the accuracy of obtained results is uncertain.

For these reasons, we proposed a hybrid ontology learning procedure which combines the acquisition of information from structured datasets as well as from unstructured text using text mining methods [5,12]. In the first step, the required information is extracted from the structured dataset. This requires to identify the concept in the dataset. However, querying the whole dataset only on the basis of the concept name may produce ambiguous results. For example, even when limiting to the words of type *artifact*, a search of word *Seal* in WordNet

[2] http://dublincore.org

produces four different results. Some additional method would have to be then employed to identify the correct concept related to the domain in interest as discussed in the previous section. In our solution we propose to use an inverse approach where the amount of concepts in the structured dataset is constrained to the domain of interest and the required term is searched only on this subset of the dataset. In this approach, the search space is limited even more, but it is ensured that if an equivalent concept is found in the dataset, it is really related to the domain in interest.

When the concept is not found in the structured dataset, text/web mining may be used for information extraction as the next step. In our approach, the hierarchy of concepts derived from the structured dataset forms base of the resulting ontology and text/web mining techniques are used to find relations between these base concepts and the more specialized concepts not covered in the structured dataset. The relations are formed by searching lexico-syntactic patterns between relevant concepts in text/web documents. The overview of the proposed solution's workflow is shown in Fig. 1.

The procedure outlined above represents the basic set of steps to follow in our proposed solution. However, the application of particular resources (WordNet, DBpedia, Web mining) may be combined arbitrarily within each step to obtain better outcomes. Naturally, this must be tailored for specific application to ensure the required results. For instance, some form of web mining may be required in a preprocessing phase in order to define suitable concept labels for the subsequent ontology learning process. Furthermore, even if some semantically equivalent concept was found in the structured dataset, it may be suitable to apply the text/web mining methods to extend the knowledge about the concept, to find paths to other concepts in the ontology, etc. Furthermore, it is good to notice the proposed solution is not aimed to be fully automatic and a certain user effort is still required during the integration.

4 Semi-automatic Tool for Ontology Learning Tasks

In this paper we implemented the proposed solution on the case of integration of spare part records into the Ford Supply Chain ontology. The input data for this case is extensive and detailed Excel document where records for various spare parts are saved. Every record of spare parts has a unique (or in other words is identified by) part-number and is characterized by a brief part label. The part-number is considered as an attribute which is difficult to utilize for subsequent semantic processing (mining) and is only used to identify whether given spare part already exists in the ontology. The part label is taken into consideration for subsequent semantic processing. An example of the spare part number is *AT4E_6701_AA* and example of the part label is *crankshaft rear oil seal*.

We evaluated the suitability of various resources and methods for this case of application in an automotive domain. We considered WordNet and DBpedia as potential structured datasets to derive the hierarchy of general concepts. As described in the previous section, we don't attempt to query the whole dataset

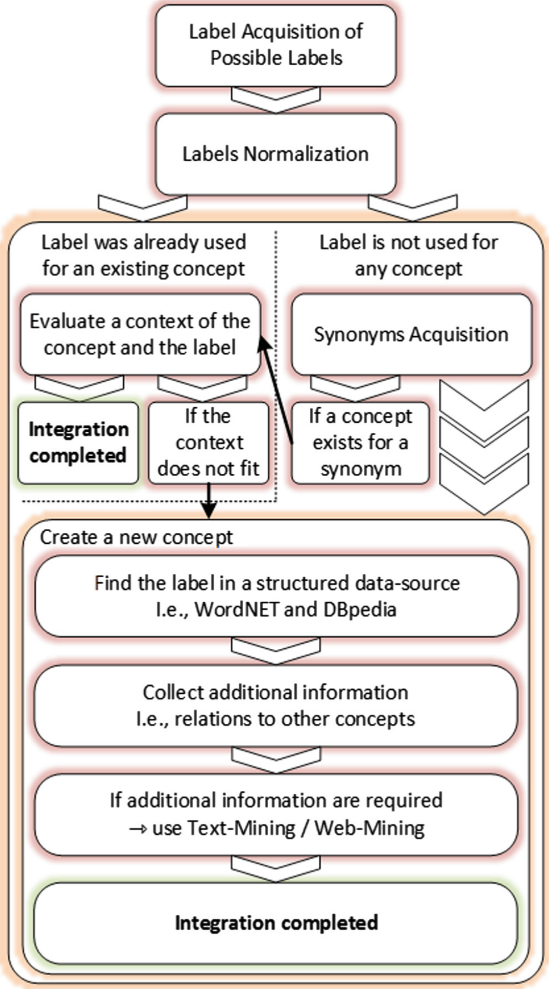
Label Acquisition of Possible Labels
Labels Normalization
Label was already used for an existing concept
Label is not used for any concept
Evaluate a context of the concept and the label
Synonyms Acquisition
Integration completed
If the context does not fit
If a concept exists for a synonym
Create a new concept
Find the label in a structured data-source I.e., WordNET and DBpedia
Collect additional information I.e., relations to other concepts
If additional information are required → use Text-Mining / Web-Mining
Integration completed

Fig. 1. Workflow of proposed solution

but only a predefined subset corresponding semantically to the target domain. To constrain the WordNet database, we consider hierarchy of concepts that are related to super-concepts *Automotive vehicle* or *Automobile* through various semantic relations (hypernymy, meronymy). Similarly, we constrain the query in DBpedia by condition that the concepts must be related to super-concept *Vehicle technology* through relations "skos:broader"[3] and "dc:subject"[4]. WordNet database was evaluated to be more suitable for this application mainly due to the existence of the meronymy relation.

[3] https://www.w3.org/TR/skos-reference/#broader.
[4] http://dublincore.org/documents/dces/.

We considered mining of semantic information on the web. However, since the spare part labels are quite specific, there are not many useful results found on the web. These terms often appear mainly in catalogues of producers without any detailed description to extract the required information from. For this reason we decided to employ text mining from technical manuals.

Overall process of integration of given spare part record into the ontology is divided into following steps:

1. Identification of spare part in the target ontology:
   (a) Find equivalent spare part in the ontology.
   (b) Find similarly labeled concepts in the ontology as possible candidates for matching.
2. Identification of spare part label in WordNet.
3. Definition of specialized concept:
   (a) Identify corresponding concepts in WordNet as possible canditates for matching.
   (b) Use text mining to find relations to general WordNet concepts.

These steps are discussed in greater detail in following sections.

The ontology learning process described above was implemented in a form of semi-automatic tool which reports results of particular steps to the user and requires verification before inclusion into the target ontology. Verification is implemented in the form of a graphical user interface based on Java Universal Network/Graph Framework[5]. Hierarchy of concepts derived within particular step of the process is displayed to the user. The graphical representation helps the user to evaluate accuracy of proposed matchings and eliminate the incorrect ones. Moreover the tool presents some additional information to the user about the concepts which can also help to evaluate correctness of obtained results. In this particular case the tool can display a description of WordNet concepts.

Structure of the ontology created by this tool is very simple. WordNet concepts are grouped as subclasses of the class *WordNetConcept* and the concepts not found in the WordNet are grouped as subclasses of the class *SpecializedConcept*. There are several object properties relating these concept classes, these properties are namely *hasPart* and its inverse *isPartOf*, *hasBroader* and its inverse *hasNarrower*, and general property between classes *isRelatedTo*. The property *hasBroader* is equivalent to the property rdfs:subClassOf. All spare parts are included as instances of class *SparePart* and given spare part is also added as an instance of corresponding concept class. Spare parts are labeled in the ontology by their part number.

In following paragraphs we describe particular steps of the above outlined procedure in greater detail and we present examples of integration for some specific cases.

[5] http://jung.sourceforge.net/.

4.1 Step 1: Identification in Target Ontology

In the first step the tool explores existing target ontology for matchings between processed spare part and spare parts and concepts already included in the ontology. First, labels of all spare part instances (i.e., part numbers) in the ontology are extracted and compared with the part number of processed spare part. If the tool finds exact agreement, this spare part was already included in the ontology and the procedure ends.

Next, labels of all concepts in the ontology are extracted and compared against part label of the processed spare part. This comparison is performed also for permutations and subsets of the spare part label. If some matching is found, it is reported to the user. The user then decides whether some of the matchings is correct and can specify a type of relation between the processed spare part and found corresponding concepts.

For example, let us assume the processed spare part has label *crankshaft rear oil seal* and the ontology already contains concept with label *rear crankshaft oil seal.* The user can decide that this concept represents the same type of spare part and order the tool to include new relation in the ontology: *exampleSparePart – isInstanceOf – rear crankshaft oil seal.* Similarly, if the ontology contained concept with label *crankshaft oil seal*, the user could order the tool to include relation *exampleSparePart – isInstanceOf – crankshaft rear oil seal – hasBroader – crankshaft oil seal.* Finally, if the ontology contained only the concept *crankshaft*, the user could order the relation *exampleSparePart – isInstanceOf – crankshaft rear oil seal – isPartOf – crankshaft.*

4.2 Step 2: Identification in WordNet

In our application each WordNet concept represents single WordNet synset. The synset is a set of synonyms, i.e., terms representing approximately the same thing. For example terms *cylinder* and *piston chamber* form one synset in WordNet and both are represented by single WordNet concept in our tool. Within this step the tool compares full part label with the set of WordNet concept labels. Multiple matches can be found in the WordNet in case that more WordNet synsets contain the same term. Within the constrained WordNet hierarchy used in our application this holds for example for the part label *cap* which matches two WordNet synsets, one with description "a top (as for a bottle)" and the other with description "something serving as a cover or protection".

When a match is found, full path from the mapped concept to the super concept is displayed. Figure 2 shows the concept hierarchy produced for search of the part label *cap.* The square represents the spare part instance and the circles represent WordNet concepts (three dashes in the concept labels split different synonyms of given WordNet synset). Dotted lines represent the relation *isInstanceOf*, dashed lines represent the relation *hasBroader*, and solid lines represent the relation *isPartOf.* The figure indicates that there are three different paths to the super concept *motor vehicle.* By clicking particular nodes in the graph, the user can select which concepts (and relations) should be added into the ontology. Selected concepts are marked by green color in this example.

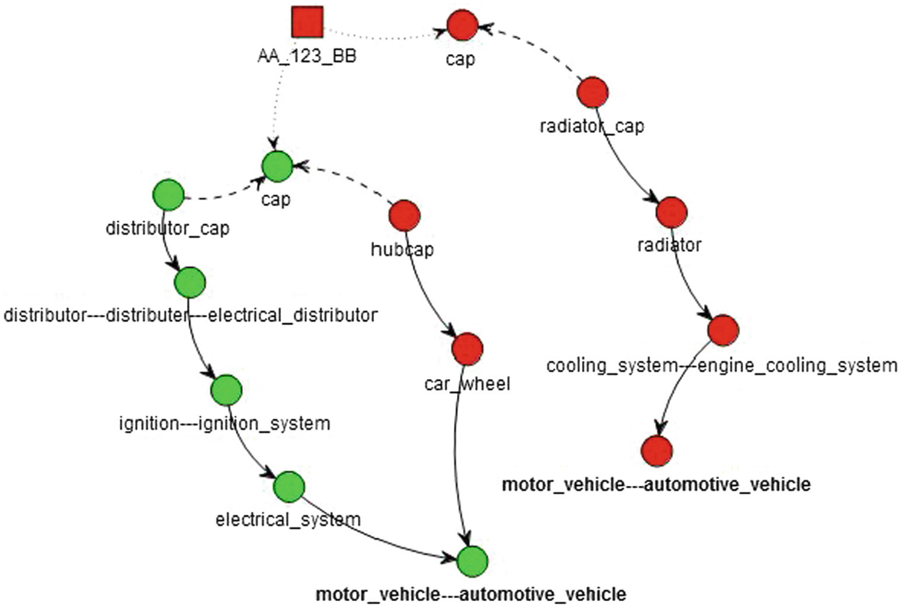
AA_123_BB
cap
radiator_cap
cap
distributor_cap
hubcap
radiator
distributor---distributer---electrical_distributor
car_wheel
cooling_system---engine_cooling_system
ignition---ignition_system
motor_vehicle---automotive_vehicle
electrical_system
motor_vehicle---automotive_vehicle

Fig. 2. Displayed concept hierarchy for search of spare part label cap. (Color figure online)

4.3 Step 3: Definition of Specialized Concept

Since the WordNet database contains rather general concepts, exact match of the full spare part label and some WordNet concepts is found only for limited number of spare parts. For the others the procedure continues by defining specialized concept and looking for semantic relations with other concepts. This task is divided in two substeps. First, possible candidates for matchings are searched in the WordNet database. This step is basically identical to the previous step but in this case permutations and subsets of the spare part label are taken into consideration. When a match is found, graphical representation of the concept hierarchy is displayed again. In this case relations between the specialized concept and corresponding WordNet concepts are shown and user can define type of relation between them by selecting one (or more). This is demonstrated in Fig. 3 on the example search of part label *piston pin retainer*. There are five relations between the specialized concept and corresponding WordNet concepts *piston—plunger* and *pivot—pin*, relation *isPartOf* in both directions (solid), *hasBroader* in both directions (dashed) and general relation *isRelatedTo* (dotted).

When all attempts to integrate the spare part performed in the previous steps of the procedure fail, the tool looks for semantic relations in unstructured text using text mining methods. Extraction of semantic relations is based on searching lexico-syntactic patterns. These pattern have form *first phrase – key phrase – second phrase*, where one of the phrases contains label of given

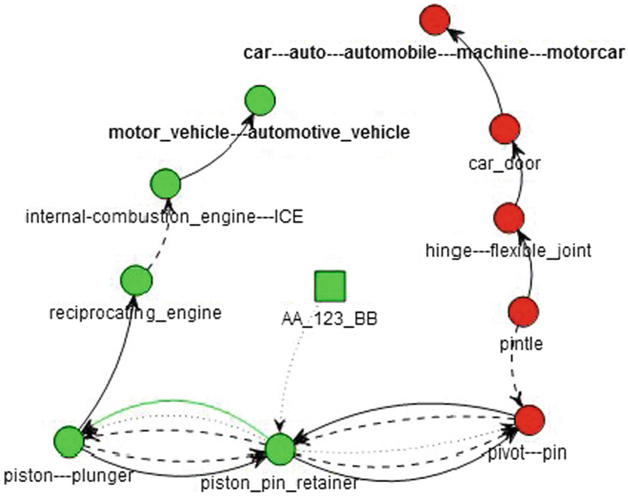

Fig. 3. Displayed concept hierarchy for search of spare part label piston pin retainer.

spare part (or permutation or subset of the label), the other contains name of some concept from the derived WordNet hierarchy, and the key phrase specifies semantic relation. These key phrases may be divided into four categories according to semantic relationship—hypernymy (e.g., “is a”, “is defined as a”, etc.), hyponymy (e.g., “called”, “like”, etc.), meronymy (e.g., “is part of”, “in”), holonymy (e.g., “consists of”, “having”, “made of”, etc.).

Found lexico-syntactic patterns may be processed in various ways. If the searched concepts are well covered by available resources, occurrence of particular patterns could be processed statistically to find the most frequent patterns as candidates for semantic relations. However, as discussed above, this domain is quite specific so that such an approach was not considered to be suitable. One also needs to decide whether to search the pattern explicitly or allow some unspecified words within the pattern. In the case of explicit search the procedure fails on inserted words, e.g., search of pattern *crankshaft–is part of–engine* fails on sentence “crankshaft is part of reciprocating engine”. On the other hand, when the pattern is not strictly specified, the search may catch incorrect dependencies, e.g., pattern *crankshaft–in–automobile* is found in sentence “crankshaft is part of reciprocating engine in automobile”, so that false relation pairs are found. In order to increase precision of obtained relation pairs, natural language processing methods are involved within analysis of found patterns, such as part-of-speech tagging, phrase tagging, and derivation of grammatical dependencies. For these tasks we use libraries developed at the Stanford Natural Language Processing Group[6].

The procedure of extracting semantic relation used in this application can be described as follows:

[6] https://nlp.stanford.edu/.

1. First, sentences corresponding to searched pattern are identified in the text.
2. The complexity of a given sentence is reduced in order to facilitate subsequent processing. This involves removing information in brackets (these are usually references to figures, etc.) or removing some types of words from the sentence (e.g., adverbs).
3. Grammatical dependencies are evaluated to find words related by given key phrase.
4. The sentence is decomposed into phrases using phrase tagging. Lowest level noun phrases which contain the two related words found in the previous step are identified.
5. It is evaluated whether these noun phrases contain the searched labels.

Let us demonstrate the procedure on a simple example of search of pattern *gasket–of–water pump*. In the first step, sentence “Install a new gasket (13) on the flange of the water pump body” is identified as possible match. The bracket in the sentence represents reference to part of figure and it is removed in the second step. Evaluation of grammatical dependencies shows that words *flange* and *body* are related by the phrase *of*. The lowest level noun phrases corresponding to these two words are *flange* and *the water pump body*. In this case the search fails because the searched term *gasket* is not contained in the first noun phrase. The procedure would succeed in this case for the search of pattern *flange–of–water pump*. In this case the tool reports the result to the user together with suggested semantic relation (meronymy). The user can then decide to add the relation *flange – isPartOf – water pump* to the ontology.

5 Summary

In this paper, we deal with a problem of integration of new information into existing knowledge model, i.e., ontology learning problem. Automated extension of existing ontology is a very demanding task and is strongly dependent on the given application as well as on available resources of additional information.

Our proposed solution to the ontology learning problem combines extraction of required knowledge in structured resources, e.g., DBpedia or WordNet, and acquisition of additional knowledge by means of text/web mining techniques. This approach aims to overcome problems when a given domain or given resources are very specific so that the existing structured data-sources are not convenient.

We presented the application of the proposed approach for integration of spare part records into Ford ontology. Proposed approach was implemented in a semi-automatic ontology learning tool which reports results of the ontology learning process to the user and allows him to verify the information to be added to the ontology. The proposed solution for seems to be promising for the facilitation ontology learning process based on the presented application in automotive domain.

Acknowledgment. This work is supported through the Ford Motor Company University Research Proposal (URP) program and by institutional resources for research by the Czech Technical University in Prague, Czech Republic.

References

1. Booshehri, M., Luksch, P.: Towards adding linked data to ontology learning layers. In: Proceedings of the 16th International Conference on Information Integration and Web-Based Applications, pp. 401–409 (2014)
2. Booshehri, M., Luksch, P.: An ontology enrichment approach by using DBpedia. In: Proceedings of the 5th International Conference on Web Intelligence, Mining and Semantics (2015)
3. van Hage, W.R., Kolb, H., Schreiber, G.: A method for learning part-whole relations. In: Cruz, I., et al. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 723–735. Springer, Heidelberg (2006). https://doi.org/10.1007/11926078_52
4. Hearst, M.: Automatic acquisition of hyponyms from large text corpora. In: Proceedings of the Fourteenth International Conference on Computational Linguistics, pp. 539–545 (1992)
5. Jirkovský, V., Šebek, O., Kadera, P., Rychtyckyj, N.: Heterogeneity reduction for data refining within ontology learning process. In: IECON 2018–44th Annual Conference of the IEEE Industrial Electronics Society, pp. 3108–3113 (2018)
6. Klaussner, C., Zhekova, D.: Lexico-syntactic patterns for automatic ontology building. In: Proceedings of the Second Student Research Workshop Associated with RANLP 2011, pp. 109–114. Association for Computational Linguistics, Hissar, September 2011. https://www.aclweb.org/anthology/R11-2017
7. Luong, H.P., Gauch, S., Speretta, M.: Enriching concept descriptions in an amphibian ontology with vocabulary extracted from WordNet. In: 2009 22nd IEEE International Symposium on Computer-Based Medical Systems, pp. 1–6 (2009)
8. Maedche, A., Staab, S.: Ontology learning for the semantic web. IEEE Intell. Syst. **16**(2), 72–79 (2001)
9. Maedche, A., Staab, S.: Ontology learning. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies. INFOSYS, pp. 173–190. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24750-0_9
10. Miller, G.A.: WordNet: a lexical database for English. Commun. ACM **38**(11), 39–41 (1995)
11. Tesfaye, D., Zock, M., Teferra, S.: Combining syntactic patterns and Wikipedia's hierarchy of hyperlinks to extract meronym relations. In: Proceedings of the NAACL Student Research Workshop, pp. 29–36. Association for Computational Linguistics, San Diego, June 2016. https://doi.org/10.18653/v1/N16-2005. https://www.aclweb.org/anthology/N16-2005
12. Šebek, O., Jirkovský, V., Rychtyckyj, N.: Concepts and relations acquisition within ontology learning process for automotive. In: Data a znalosti & WIKT, pp. 115–119 (2018)
13. Zhou, W., et al.: A semi-automatic ontology learning based on WordNet and event-based natural language processing. In: Information and Automation (2006)