Proceedings of the Federated Conference on
Computer Science and Information Systems pp. 109–115

ISBN 978-83-60810-51-4

Knowledge patterns in RDF graph language for English sentences

Rostislav Miarka
University of Ostrava, Faculty of Science,
Department of Informatics and Computers,
30. dubna 22, 701 03 Ostrava, Czech Republic
Email: rostislav.miarka@osu.cz

Martin Žáček
University of Ostrava, Faculty of Science,
Department of Informatics and Computers,
30. dubna 22, 701 03 Ostrava, Czech Republic
Email: martin.zacek@osu.cz

Abstract—Each language has its word order, which determines the way of ordering words in a sentence. This paper presents knowledge patterns for sentences in the English language. Knowledge patterns are general patterns of knowledge which can be used in any knowledge base or ontology. While using them, the general symbols from the pattern are renamed to special symbols from the modeled domain. The RDF graph language is used to represent knowledge pattern. The paper contains examples of usage of knowledge patterns.

I. Introduction

SENTENCES in any language are formed by words of natural language. Each word in a sentence belongs to a certain word class; it can be a noun, pronoun, verb, adverb, etc. Words of natural language form vocabulary of a language. Each language has its grammar, which determines the way of construction of sentences. Grammar of the language does not work with word classes; it works with constituents of a sentence (members). There are two basic members of a sentence: subject and predicate. These two members must be a part of each sentence, but there are some exceptions, e.g. imperative sentences ("Stop!", "Do it!", etc.). In addition to these members, there are members which extend the sentence. There is an object, attribute and adverbial complement.

When constructing a sentence, it is important which member represents the particular word. It is not important which word class it belongs to. The term word order is closely related to the process of construction of a sentence. Word order determines order of members in a sentence. There are two basic types of word order – fixed word order and free word order. Fixed word order has stricter rules for ordering words in a sentence. This word order is used by Germanic languages, e. g. English. Free word order allows changing order of members in sentence. This is possible in languages which enable declension and inflexion. Among these languages belongs the Czech language.

As mentioned above, the essential members of each sentence are subject and predicate. Basic types of word orders are marked by three members – these two members mentioned above and the object. All three members are represented by a letter. Subject is marked by letter S, predicate is marked by V (verb) and object is marked by O. A combination of these three basic members determines the type of word order [2], [3]. There are six basic types of word order: SVO (subject verb object), SOV (subject object verb), VSO (verb subject object), VOS (verb object subject), OSV (object subject verb) and OVS (object verb subject). Languages using the SVO word order include English, Romance languages, Bulgarian and Chinese. Languages using the SOV word order include Japanese, Mongolian, Turkish and Korean. Word order marked as VSO is used by the following languages: Classical Arabic, Insular Celtic languages and Hawaiian. Word order VOS is used in the Fijian language (Fiji) and the Malagasy language (Madagascar). The OSV word order is used in languages Xavante (Brazil) and Warao (Venezuela). The OVS word order is used by the Hixkaryana language (Brazil).

II. Structure of English sentence

The English language uses fixed word order marked as SVO (subject verb object). Letter O in this shortcut marks direct object. The basic word order can be a little different because an auxiliary verb (do, have, be, will, can, etc.) or an indirect object can appear in the sentence. Sentences in each language can express different things. It can state facts – declarative sentences (positive or negative). It can be ask about things – questions. It can command – imperative sentences. Because this article is focused on knowledge patterns, which are closely related to knowledge bases or ontologies, the only type of sentence which is important for us is the declarative sentence. Questions or imperative sentences are not a part of any knowledge base or ontology.

Declarative sentences can state either positive or negative facts. The structure of both positive and negative sentence is

The research described here has been financially supported by the University of Ostrava grant SGS02/PřF/2012. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

quite similar in the English language. A sentence in its negative form contains auxiliary word “not”.

The simplest type of a positive declarative sentence is bare sentence, which contains only a subject and predicate. Examples of this type of sentence are “It rains.”, “I know.” etc. A more complex type of sentence is that which contains an object. This type of a sentence is in form SVO (subject verb object). Examples of this type of a sentence are “I see John.”, “John likes chocolate.” etc. Another type of a sentence contains an adverbial complement. It can be an adverbial complement of manner, place or time. A sentence can contain one of these complements or combination of them. A sentence containing all three types of adverbial complements has word order marked as SVOMPT (subject verb object manner place time). Examples of sentences with adverbial complements are “I play football daily.”, “I speak English very well.” etc. Another type of a sentence contains an attribute. An attribute extends the subject or the object in a sentence. Examples of attributes are “small”, “big”, “red”, “blue” etc.

Types of a negative declarative sentence are the same as for a positive declarative sentence. Negation in a sentence is related to a verb and it changes the verb to its opposite meaning.

III. Knowledge Patterns

The term “Knowledge pattern” [1] is closely related to the process of creating ontologies or knowledge bases. While building ontology or knowledge base, we describe terms and relations among them in the modeled domain. Nowadays, the most used approach for describing a modeled domain is by the help of a concept-oriented approach. The basic building block of this approach is a concept, which represents a set of objects from the modeled domain that share some features. Concepts can form hierarchy or can be related to other concepts.

During the process of creating ontology one can find out that some structures of concepts are the same. The only difference is in labels of concepts. The same structures of concepts (i.e. the same structures of knowledge) can be recorded as knowledge patterns in which general terms are labels. While using knowledge patterns, these general terms are mapped to concrete terms from the modeled domain. The process of renaming general terms to concrete terms is called morphism. The basic advantage of using knowledge patterns is reusing of knowledge. It is not necessary to create relations among concepts every time, but one can use knowledge pattern and make only the morphism. Examples of simplest knowledge patterns are specialization of concept (ISA relation) or the unit-part relation. These knowledge patterns can be found in almost every ontology or knowledge base.

For representation of knowledge patterns we propose to use an RDF (Resource Description Framework) graph language [6], [7], specifically the RDF graph language which is extended by quantifiers enabling to state negation [4]. Facts in the RDF language are called statements and they are represented by RDF triples. One RDF triple represents one statement and consists of a subject, a predicate and an object. Ontology or knowledge base in the RDF graph language is represented by a directed graph. Nodes in this graph represent subjects and objects of statements, arcs represent relations (predicates). RDF contains only binary relations, more complex relations must be decomposed to a set of binary relations. Nodes and edges in an RDF graph are identified by URI (Uniform Resource Identifier) references. A classical RDF graph uses only solid lines. To distinguish knowledge patterns from the classical RDF statements, we use dashed lines. Example of a classical RDF triple and a triple which represents knowledge pattern is shown in Figure 1.

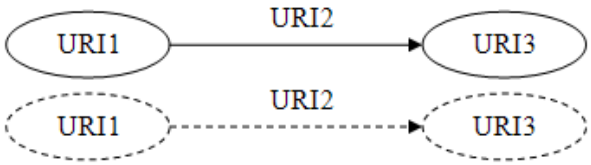

Fig. 1 Classical RDF triple (above) and RDF triple representing knowledge pattern (below)

An important part of use of knowledge patterns is morphism. By the help of morphism, the general terms from the pattern are mapped to special terms from the problem domain. Morphism represents a relation of specialization – the general term is renamed to a special one from the domain of interest. Mapping of one term will be represented by one classical RDF triple (with solid line). The subject of this triple will be the special term from the domain of interest, predicate will be “isa” (relation Is-a) and object will be the general term from the pattern. An example of morphism of one term is shown in Figure 2.

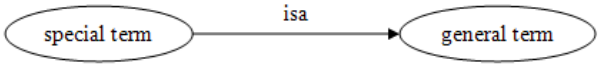

Fig. 2 Morphism

Identification of nodes and edges in an RDF graph is realized by the help of URIs, which can be quite long. An RDF graph with full URIs would be confusing to the users. It is possible to use shortcuts for full URIs. In the following RDF graphs will be used shortcuts which are shown in Table I.

IV. Knowledge Patterns for English Sentences

In this section we present knowledge patterns for English sentences. It will concern knowledge patterns for positive declarative sentence and for negative declarative sentence. For each type of these sentences, we will present

TABLE I.
SHORTCUTS AND FULL URIS USED IN RDF GRAPHS

Shortcut	URI
subject	http://en.wikipedia.org/wiki/Subject_(grammar)
predicate	http://en.wikipedia.org/wiki/Predicate_(grammar)
object	http://en.wikipedia.org/wiki/Object_(grammar)
isa	http://en.wikipedia.org/wiki/Is-a
rdf:Bag	http://www.w3.org/1999/02/22-rdf-syntax-ns#Bag
rdf:type	http://www.w3.org/1999/02/22-rdf-syntax-ns#type
rdf:statement	http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement
rdf:subject	http://www.w3.org/1999/02/22-rdf-syntax-ns#subject
rdf:predicate	http://www.w3.org/1999/02/22-rdf-syntax-ns#predicate
rdf:object	http://www.w3.org/1999/02/22-rdf-syntax-ns#object
complement	http://en.wikipedia.org/wiki/Adverbial_complement
extendedBy	http://en.wiktionary.org/wiki/extend
manner	http://en.wiktionary.org/wiki/manner
place	http://en.wiktionary.org/wiki/place
time	http://en.wiktionary.org/wiki/time
attribute	http://en.wiktionary.org/wiki/attribute
Andrew	http://en.wiktionary.org/wiki/Andrew
writes	http://en.wiktionary.org/wiki/write
fairy tale	http://en.wiktionary.org/wiki/fairy_tale
Michael	http://en.wiktionary.org/wiki/Michael
fears	http://en.wiktionary.org/wiki/fear
dogs	http://en.wiktionary.org/wiki/dog
little	http://en.wiktionary.org/wiki/little
Susan	http://en.wiktionary.org/wiki/Susan
wears	http://en.wiktionary.org/wiki/wear
clothes	http://en.wiktionary.org/wiki/clothes
red	http://en.wiktionary.org/wiki/red
spiders	http://en.wiktionary.org/wiki/spider
Lucy	http://en.wiktionary.org/wiki/Lucy
pink	http://en.wiktionary.org/wiki/pink
at home	http://en.wiktionary.org/wiki/home
sweater	http://en.wiktionary.org/wiki/sweater
daily	http://en.wiktionary.org/wiki/daily

a sentence in the basic form, a sentence with an attribute and a sentence with an adverbial complement.

A. Positive declarative sentence

A positive declarative sentence states some fact. The bare sentence contains only two members – subject and predicate. One fact represented by an RDF graph must contain three members – subject, predicate and object, so the bare sentence cannot be represented by an RDF graph.

A sentence in the basic form contains three members and represents one fact which can be represented by one RDF triple. The knowledge pattern for this type of a sentence is shown in Figure 3. This pattern will be marked as KPS1.

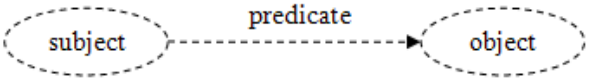

Fig. 3 KPS1 – sentence in basic form

Members of the sentence (subject, predicate and object) form the RDF triple. In the following, the word “subject” without any prefix represents subject as the member of sentence. Similarly, the words “predicate” and “object” represent the members of sentence. The subject from sentence is represented by the node “subject”, the object is represented by the node “object” and the predicate is represented by the arc “predicate”. An example of this sentence is the sentence “Andrew writes fairy tale”. The morphism for this sentence is shown in Figure 4.

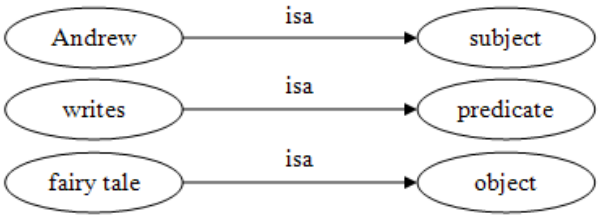

Fig. 4 KPS1 – morphism

While using knowledge pattern KPS1, the labels of nodes and arc in KPS1 will be renamed with the morphism labels on the left side. The result – one classical RDF triple is shown in Figure 5.

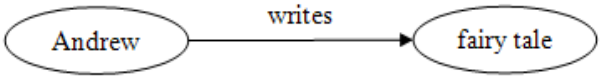

Fig. 5 KPS1 – result

Another type of a positive declarative sentence is that containing an attribute. An attribute can extend the subject or the object. An attribute is usually represented by an adjective. Examples of attributes are colors (red, blue, green etc.) or sizes (big, small etc.). Knowledge pattern for a sentence with an attribute extending the subject is shown in Figure 6 and it will be marked as KPS2.

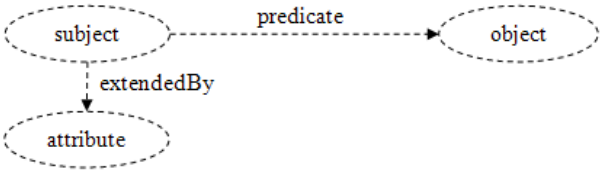

Fig. 6 KPS2 – sentence with an attribute extending the subject

KPS2 contains two RDF triples. The first triple represents the sentence in its basic form (subject – predicate – object) and the second triple represents extension of the subject by the attribute (subject – extendedBy – attribute). The predicate of the second triple could also be “is”. Both triples share a common node – the one with “subject” label. An example of this type of a sentence is “Little Michael fears dogs”. The morphism for this sentence is shown in Figure 7 and the resulting RDF graph after renaming labels of nodes and arcs is shown in Figure 8.

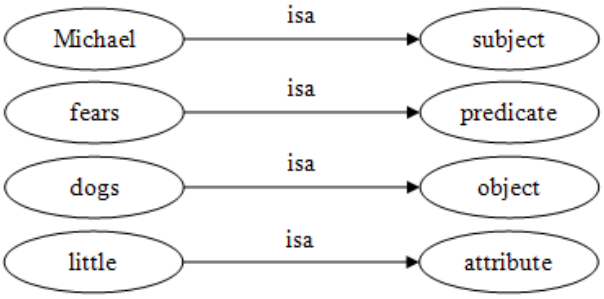

Fig. 7 KPS2 – morphism

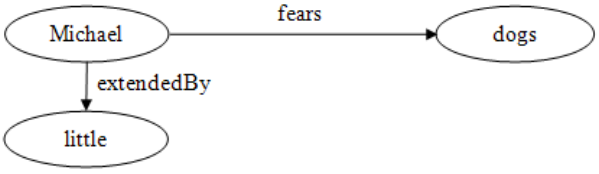

Fig. 8 KPS2 – result

The case when an attribute extends the object in a sentence is marked as KPS3 and is shown in Figure 9.

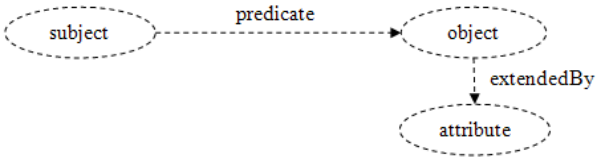

Fig. 9 KPS3 - sentence with an attribute extending the object

In this case, the graph also contains two RDF triples. The first triple is subject – predicate – object. The second triple is object – extendedBy – attribute. In the second triple, the member “object” is in the place of the subject of the RDF triple. RDF triples in this knowledge pattern share the node with “object” label and are connected via this node. Example of this type of a sentence is “Susan wears red clothes”. Figure 10 shows morphism for this sentence and figure 11 shows the resulting RDF graph.

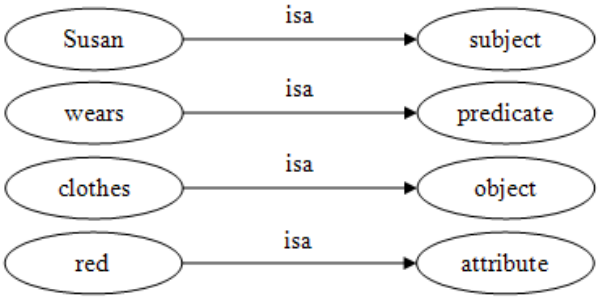

Fig. 10 KPS3 – morphism

Fig. 11 KPS3 – result

Another type of a sentence is a type containing a subject, predicate, object and adverbial complement. It can be adverbial complement of manner, place or time. The sentence can contain one, two or three types of adverbial complement. The adverbial complement is bound to the verb (predicate) in the sentence, it extends its meaning. In the RDF graph, it means that an adverbial complement extends the triple representing a sentence in the form subject – predicate – object. For representation of an RDF triple as a whole, one can use predefined resource of the RDF language called `rdf:statement`. Representation of parts of the statement is done by the help of predefined properties `rdf:subject`, `rdf:predicate` and `rdf:object`. A knowledge pattern for a sentence containing adverbial complements (all three types) is shown in Figure 12 and it will be marked as KPS4. For representation of an adverbial complement is used RDF container `rdf:Bag`.

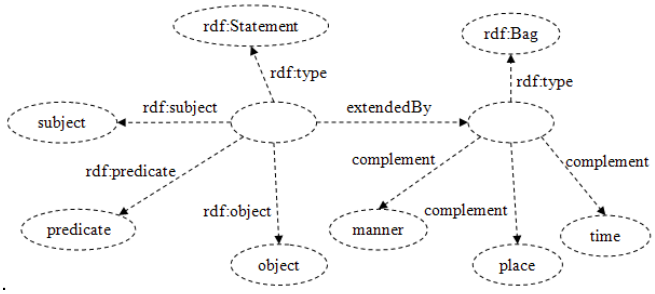

Fig. 12 KPS4 - the sentence with adverbial complements

RDF container `rdf:Bag` is used for representation of unordered collection of resources. The node with 'subject' label represents the member of sentence subject, while the arc with `rdf:subject` label represents the predefined RDF property – the part of RDF triple. If the sentence contains only an adverbial complement of place, the knowledge pattern does not contain the nodes with 'manner' and 'time' label and the arcs connecting them with the blank node representing RDF container `rdf:Bag`. The knowledge pattern for the sentence without an adverbial complements of manner and time (KPS5) is shown in Figure 13.

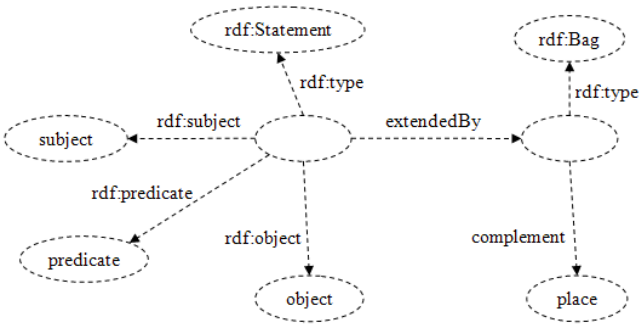

Fig. 13 KPS5 - the sentence with adverbial complement of place

As an example of this sentence we use the sentence "Andrew writes fairy tale at home ". The morphism for this sentence is shown in Figure 14 and the result after renaming labels of nodes and arcs is shown in Figure 15.

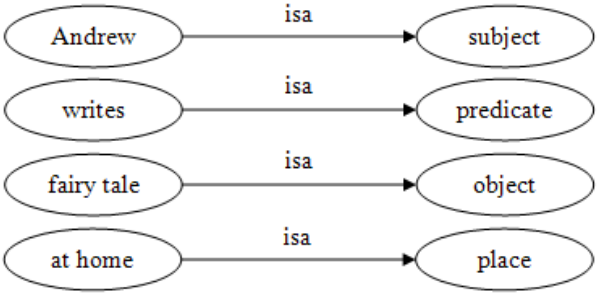

Fig. 14 KPS5 – morphism

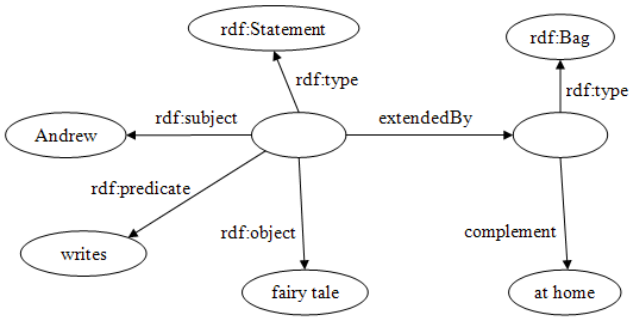

Fig. 15 KPS5 - result

B. Negative declarative sentence

For a negative declarative sentence, we will introduce the same types of sentences as for a positive declarative sentence. A negative declarative sentence states some negative fact, i.e. some statement is not true. To express negation in the RDF graph language, one can use a symbol called falsum (Figure 16), which was introduced in [4]. Falsum is always bound to the predicate of the sentence.

Fig. 16 Falsum

The first type of a negative declarative sentence is the one which contains only a subject, predicate and object. A knowledge pattern representing this type of a sentence is shown in Figure 17 and is marked as KPS6. An example of this sentence could be the sentence "Susan does not fear spiders". Figure 18 shows morphism for this sentence and Figure 19 shows resulting RDF graph (one RDF triple) after applying morphism.

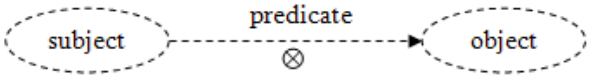

Fig. 17 KPS6 - negative sentence in basic form

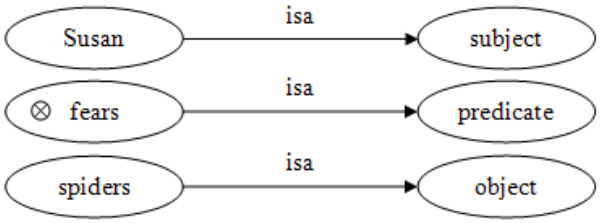

Fig. 18 KPS6 – morphism

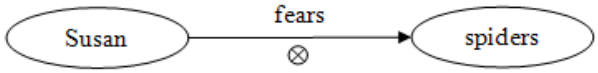

Fig. 19 KPS6 – result

The second and the third type of a negative declarative sentence is a sentence with an attribute. The former type of a sentence contains an attribute extending the subject; the latter type contains an attribute extending the object. A knowledge pattern for the sentence with an attribute by the subject (KPS7) is shown in Figure 20; a knowledge pattern for the sentence with an attribute by the object (KPS8) is shown in Figure 21.

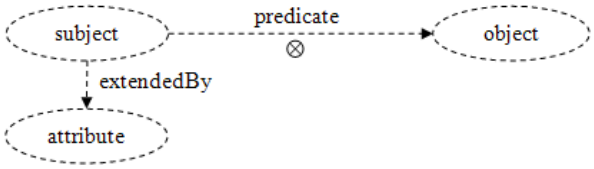

Fig. 20 KPS7 - sentence with an attribute of the subject

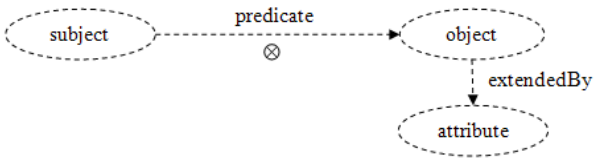

Fig. 21 KPS8 - sentence with an attribute of the object

An example of using KPS7 is sentence "Little Michael does not fear dogs". Morphism for this sentence is shown in Figure 22 and the resulting RDF graph in Figure 23.

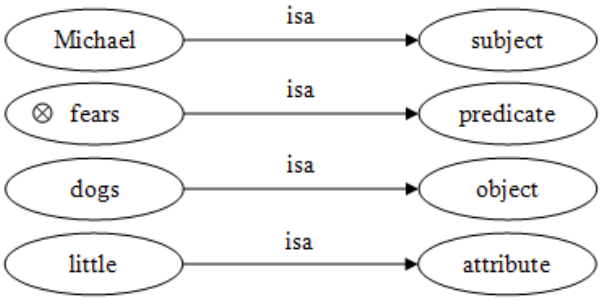

Fig. 22 KPS7 – morphism

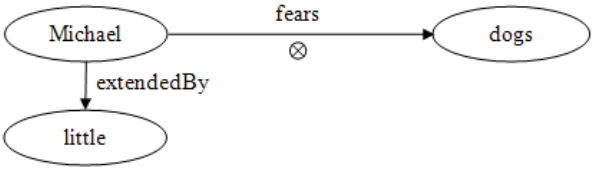

Fig. 23 KPS7 – result

An example of using KPS8 is sentence "Lucy does not wear pink clothes". Figure 24 shows morphism for this sentence, Figure 25 shows result.

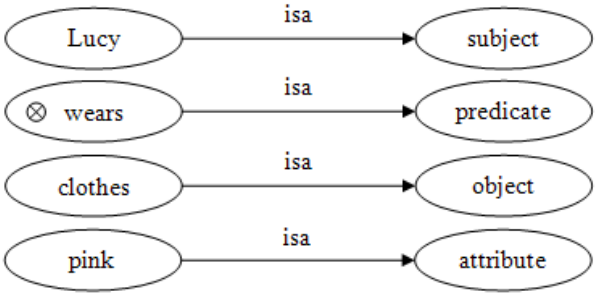

Fig. 24 KPS8 – morphism

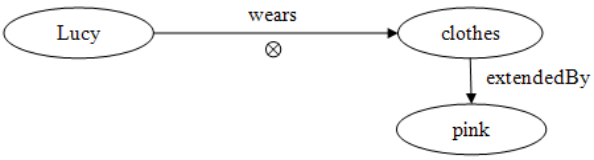

Fig. 25 KPS8 – result

The last type of a sentence described here is a negative declarative sentence containing an adverbial complements. As in a positive sentence, it can be adverbial complement of manner, place or time. A knowledge pattern for a sentence with all three types of adverbial complement is marked KPS9 and is shown in Figure 26.

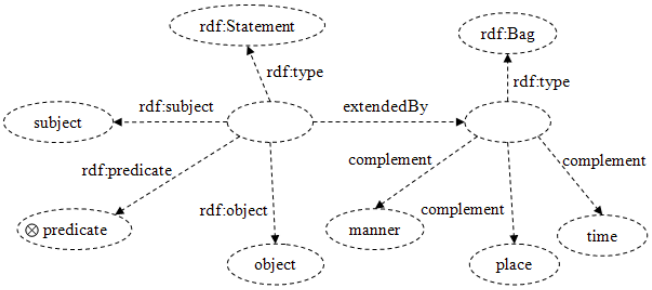

Fig. 26 KPS9 - negative declarative sentence with adverbial complements

If a sentence contains only one or two types of adverbial complement, the knowledge pattern will be modified the same way as in the case of a positive sentence. The knowledge pattern for a negative declarative sentence with an adverbial complement of time is shown in Figure 27 and is marked as KPS10.

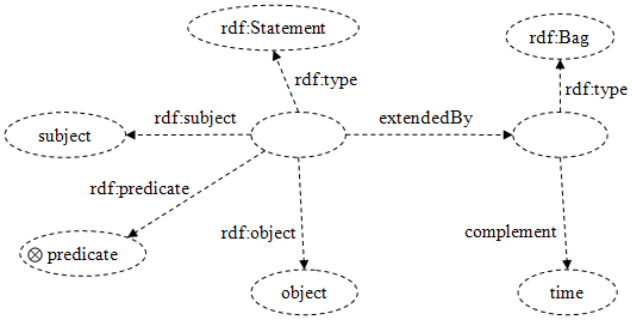

Fig. 27 KPS10 - negative declarative sentence with an adverbial complement of time

An example of this type of a sentence is “Michael does not wear sweater daily.”. Morphism for this type of a sentence is shown in Figure 28; the resulting RDF graph is shown in Figure 29.

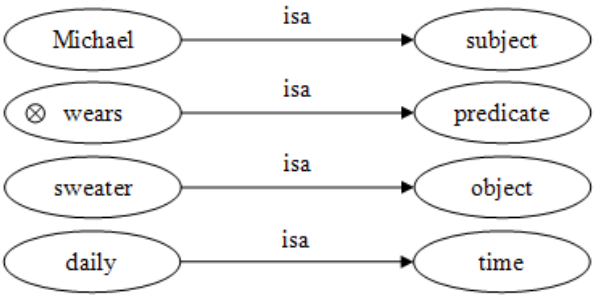

Fig. 28 KPS10 – morphism

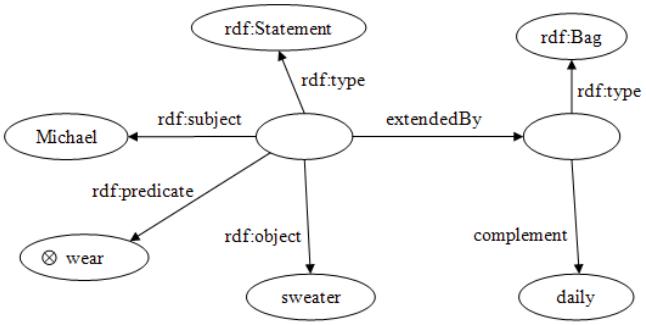

Fig. 29 KPS10 - result

V. Conclusion

Each natural language has its vocabulary and grammar. An important part of grammar is word order of the language, which determines order of particular members in a sentence. In this paper, we have introduced knowledge patterns for English sentences captured by the help of RDF graph language, including examples of knowledge patterns.

References

[1] P. Clark, J. Thompson and B. Porter, “Knowledge patterns,” in Handbook on Ontologies, S. Staab and R. Studer, Eds. Berlin: Springer-Verlag, 2004, ISBN 3-540-40834-7, pp. 191–207.

[2] D. Crystal, “The Cambridge Encyclopedia of Language (2nd edition ed.)”, Cambridge: Cambridge University Press, 1997, ISBN 0-521-55967-7.

[3] M. S. Dryer, “Order of Subject, Object, and Verb“, http://linguistics.buffalo.edu/people/faculty/dryer/dryer/DryerWalsSOVNoMap.pdf

[4] A. Lukasová, M. Vajgl and M. Žáček, “Reasoning in RDF graphic formal system with quantifier,” Proceedings of the International Multiconference on Computer Science and Information Technology, 2010, ISBN 978-83-60810-22-4, pp. 67–72.

[5] S. Staab, M. Erdmann, A. Maedche and S. Decker “Ontologies in RDF(S),” [online] http://www.ida.liu.se/ext/etai/received/semaweb/010/paper.pdf.

[6] W3C: Resource Description Framework (RDF): Concepts and Abstract Syntax, http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/

[7] W3C: RDF Primer, http://www.w3.org/TR/2004/REC-rdf-primer-20040210/