Proceedings of the Federated Conference on
Computer Science and Information Systems pp. 63–68

ISBN 978-83-60810-22-4

Knowledge patterns for conversion of sentences in natural language into RDF graph language

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*— This paper describes the knowledge patterns for the conversion of sentences in natural language into RDF graph language. While creating knowledge base in RDF graph language from sentences expressed in natural language, one must convert words from sentences to nodes and arcs in RDF graphs. For this conversion, it is important to know which members of a sentence represent particular words. In this paper, knowledge patterns are proposed as a tool for conversion of sentences. In order to capture knowledge patters one can use extended RDF graph language. For the representation of knowledge patterns, further extension of this language is proposed. The paper contains four examples of knowledge patterns and their use.**

I. Word order in natural language

IN NATURAL language, people use sentences to express various statements, questions, orders etc. Each natural language is defined by its vocabulary and its grammar. Individual words are marked as vocabulary; they are words one can use in a given language. Typically, words are divided into word classes (nouns, pronouns, verbs, adverbs etc.), which determine their meaning. The grammar of a language determines the construction of sentences, which means the way of ordering particular words in sentences. Constituents of sentences (members) are basic building blocks of constructions of sentences. The basic members are subject and predicate. Other members, which extend information included in a sentence, are object, attribute and adverbial complement. The word classes of particular words are not important. It is important which member these words represent. A word-order in language, which determines the order of members in sentence, relates to the creation of sentences. Word order abides by some rules and one can identify two basic types of a word order – fixed word order and free word order. Fixed word order defines relatively strict rules of order of members in sentence. This word order is typical for Germanic languages such as English. On the other hand, free word order is greatly flexible, rules of ordering members are not so strict and they can be modified to the context of the sentence. Free word order is typical for languages which enable declension and inflexion. Among these languages are Slavonic languages such as the Czech language [6], [8].

The essential members are subject and predicate, which form bare sentences. For the marking of these members the letter S is used for subject and the letter V for predicate (verb). Except of these essential members, other, elaborative members can be used: object, attribute and adverbial complement. To mark an object the letter O is used. According to the combination of subject, verb and object we can differentiate six basic word orders in sentence: 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). English language uses word order SVO [6], [8].

II. Concepts and their relations

The typical approach for the creation of formal ontologies is presently the concept-oriented approach. A concept is a set of objects sharing some particular features. One of the methods for searching concepts is formal concept analysis [7], which has a formal context as its input and a conceptual graph as its output. Problems of the sentence creation can be depicted as a formal context R, which is represented by a table called the data matrix, where columns are particular properties and rows are objects (types of sentences – declarative sentence, question etc.). If the table cell contains "X", an object on the appropriate row has the property in the appropriate columns. If the cell contains "O", objects can have the appropriate property. The data matrix for English grammar is displayed in table I; next tables describe objects and properties.

Figure 1 shows a conceptual graph based on the data matrix in table I. Each node represents one concept and it has a label consisting of two rows. Objects in this concept are in the first row, properties in this concept are in the second row. The top concept of this graph contains all objects, which share the property f (verb) – each sentence must contain a verb. The bottom concept of this graph contains all properties and no object at all, because there are no types of sentences that contain all the properties from the data matrix. Each concept, except the top concept, has its superconcepts; they are concepts connected with particular concept and are positioned above it. Each concept, except the bottom concept, has its subconcepts; they are concepts

˜ The research described here has been financially supported by University of Ostrava grant SGS23/PřF/2011. 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.

TABLE I.
FORMAL CONTEXT R

R	a	b	c	d	e	f	g	h	i	j
1		X				X	X	X	X	X
2		X				X	X	O	O	O
3		X		X		X	X	O	O	O
4	X	X		X		X	X	X	X	X
5	X	X		X		X	X	O	O	O
6		X		X		X	X	O	O	O
7					X	X	X	O	O	O
8				X		X		O	O	O
9				X		X	X	O	O	O
10		X				X	X	X	X	X
11		X	X			X	X	X	X	X
12		X	X	X		X	X	X	O	O

TABLE II.
OBJECTS AND ATTRIBUTES IN FORMAL CONTEXT R

	objects
1	Basic construction of sentence
2	Affirmative declarative sentence
3	Negative declarative sentence
4	General interrogative sentence
5	Factual interrogative sentence
6	Declaratory interrogative sentence
7	Interrogative sentence for subject
8	Imperative – 2nd person
9	Imperative 1st and 3rd person
10	Adverbial complements
11	Adverb of frequency
12	Adverb of frequency with complex predicate

	attributes
a	Wh – Interrogative pronoun or interrogative verb
b	S – Subject
c	AF – Adverb of frequency
d	AV – Auxiliary verb
e	S (Wh) – Interrogative pronoun
f	V – Verb
g	O – Object
h	M – Adverbial complement of Manner
i	P – Adverbial complement of Place
j	T – Adverbialcomplement of Time

connected with particular concept and are positioned below it. For example, the concept containing objects 1, 4, 10, 11, 12 and properties b, f, g, h has two superconcepts (first with objects 1, 2, 3, 4, 5, 6, 10, 11, 12 and with properties b, f; second with objects 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12 and with properties f, g) and two subconcepts (first with objects 1, 4, 10, 11 and with properties b, f, g, h, i, j; second with objects 11, 12 and properties b, c, f, g, h). Further, concepts containing optional properties (h, i, j) are connected with its superconcepts and subconcepts by a dashed line. In the conceptual graph shown on figure 1, we can see how the concepts are connected.

With the help of knowledge patterns the representation of natural language sentences in extended RDF graph language [2] will be further described. This representation is useful while building an ontology or knowledge base. The representation of some types of sentences does not make sense, e.g. an imperative or interrogative sentence. In the rest of this paper the representation of affirmative declarative sentences and negative declarative sentences will be described. A basic construction of a sentence is the special case of affirmative declarative sentences.

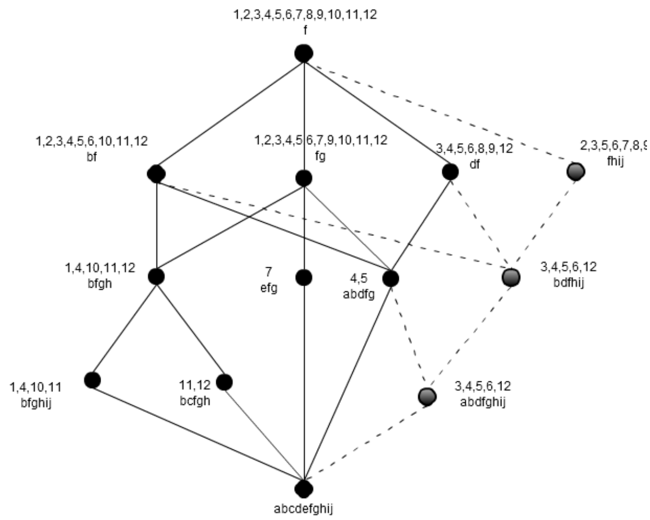

Fig. 1. Conceptual graph of formal context R

III. Knowledge patterns

The term 'knowledge pattern' was first used in [1]. While building ontologies or knowledge bases, one can see that some structures of modelled knowledge are the same. These same structures of knowledge can be captured as knowledge patterns. Knowledge patterns are general structures (patterns) of knowledge, which are not a part of the target knowledge base. They can be included into a target knowledge base by renaming their non-logical symbols. This renaming is called morphism. The morphism is an important part of using knowledge patterns.

Presently, there is no direction for capturing knowledge patterns. We propose to model knowledge patterns in RDF graph models [4], [5]. This model is simple to understand, even for amateur users. The RDF graph model is a set of RDF triples. The RDF triple consists of subject, predicate and object. Subject and object are nodes of the graph; predicate is a directed-arc from subject to object. Each more complex statement must be decomposed into individual RDF triples. In this paper, the idea of knowledge patterns for conversion of sentences in English language into RDF graph language will be introduced. For modelling of knowledge patterns extended RDF graph model introduced in [2] will be used.

RDF(S), enriched with possibility of quantification and reasoning (analogous to associative networks [9]), is an accessible tool on conceptual level even for users that do not know OWL or other formal languages based on logic which are suitable for formalization of knowledge patterns. A transcription from RDF to OWL is then quite simple.

A. Representation of knowledge patterns in RDF graph language

Solid lines are used to display nodes and arcs in classical RDF graphs. To distinguish the knowledge patterns from the classical statements in RDF graph we will use dashed lines. Figure 2 shows a classical RDF triple (above) and RDF triple representing knowledge pattern (below).

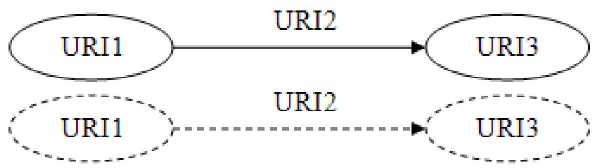

Fig. 2. RDF triples in classical RDF graph and as knowledge pattern

Apart from knowledge patterns, morphisms will be introduced for some sentences. Morphisms for mentioned knowledge patterns will be displayed as classical RDF triples. Subject of triples for morphisms will represent the term from a modelled domain, the predicate will be “isa” (Is-

Table III.
Shortcuts for 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
David	http://en.wiktionary.org/wiki/David
likes	http://en.wiktionary.org/wiki/like
chocolate	http://en.wiktionary.org/wiki/chocolate
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
play	http://en.wiktionary.org/wiki/play
the piano	http://en.wiktionary.org/wiki/piano
loudly	http://en.wiktionary.org/wiki/loudly
at home	http://en.wiktionary.org/wiki/home
daily	http://en.wiktionary.org/wiki/daily

A relation – relation of specialization) and the object will symbolize the general term from the knowledge pattern.

URI references are used to identify nodes and arcs in RDF graph. RDF graphs with full URI references would be too confusing to the users, therefore shortcuts usage for each node and arc are essential. These shortcuts are shown in table III.

1) Affirmative declarative sentence

The first described type of sentence is an affirmative declarative sentence. This sentence can have two forms. The first form is a sentence without an adverbial complement, second is one which includes it. In the first case, the sentence is in the form subject – predicate – object. This sentence forms RDF triple. The knowledge pattern for this type of sentence is shown in figure 3. This knowledge pattern will be marked as KPS1.

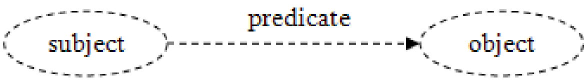

Fig. 3. KPS1

Particular members play the same role, in the RDF triple as in the sentence. Subject plays the role of subject etc. For instance, we convert the sentence “David likes chocolate.” into RDF graph language. The morphism for particular members is shown in the next figure.

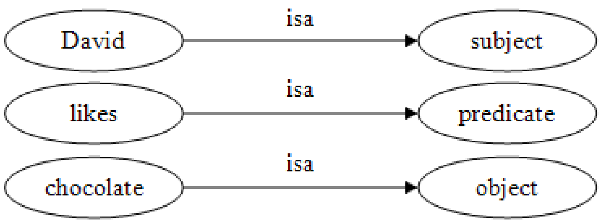

Fig. 4. KPS1 – morphism

While using the knowledge pattern KPS1 in a target knowledge base, the labels of nodes and arcs are renamed using morphism in figure 4. The resulting RDF graph (in this case, it is one RDF triple) is shown in figure 5.

Fig. 5. KPS1 – result

The second form of affirmative declarative sentence contains an adverbial complement, which extends the meaning of the verb in this sentence. It can be an adverbial complement of manner, place or time. The sentence can contain one, two or all three types of adverbial complement. In terms of RDF triples, it means that adverbial complements extend the whole RDF triple containing the appropriate verb. In order to represent the whole RDF triple the RDF data model offers the predefined resource `rdf:statement` and to represent parts of this statement it offers the predefined properties `rdf:subject`, `rdf:predicate` and `rdf:object` (full URIs for predefined resource and properties are in table III). When capturing an affirmative declarative sentence with adverbial complements as the knowledge pattern, the RDF container `rdf:Bag` is used (full URI is in table III). The knowledge pattern of an affirmative declarative sentence with adverbial complements is shown in figure 6 and will be marked as KPS2.

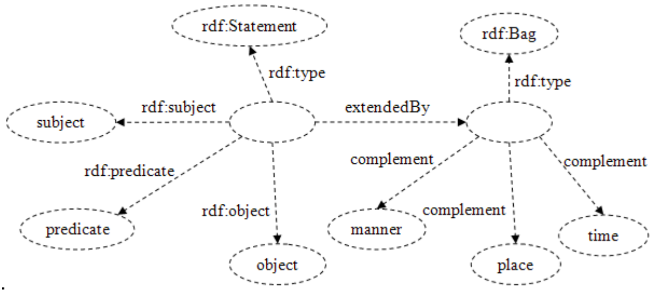

Fig. 6. KPS2

Furthermore, it is important to note that 'subject' and `rdf:subject` in KPS2 are not the same thing. The node with the 'subject' label represents the member of the sentence subject, while the arrow with label `rdf:subject` represents the predefined property from the RDF data model. Similarly, it stands both for the pair predicate – `rdf:predicate` and for the pair object – `rdf:object`.

While using pattern KPS2, the morphism contains only renaming the following symbols: subject, predicate, object, manner, place and time. The remaining symbols do not change. Let us consider the sentence “David plays the piano loudly at home daily.”. The morphism for this sentence is shown on figure 7.

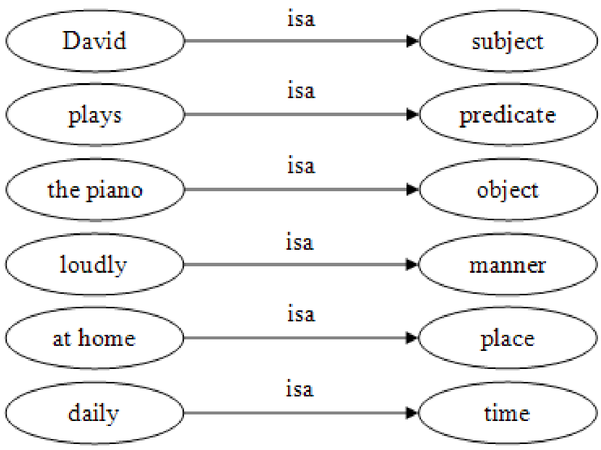

Fig. 7. KPS2 – morphism

After applying this morphism to the knowledge pattern, the symbols are renamed and the target knowledge base will contain the resulting RDF graph (figure 8).

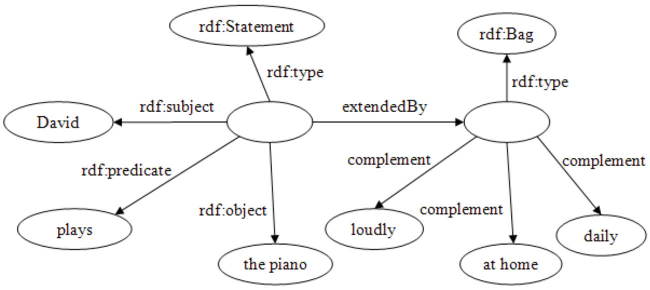

Fig. 8. KPS2 – result

2) Negative declarative sentence

The second described type of sentence is a negative declarative sentence. As well as an affirmative declarative sentence, this type of sentence can have two forms; the first form contains only subject, predicate and object, the second form contains an adverbial complement as well.

Extended RDF graph language [2] allows user to express negation of a statement. Negation is expressed by the help of special symbol called falsum (notation ⊗), which is false in all interpretations. Falsum is always bound to predicate.

The knowledge pattern for a negative declarative sentence without adverbial complements (KPS3) is shown in figure 9. The pattern is very similar to pattern KPS1, the only difference is that the arrow representing predicate is marked by falsum.

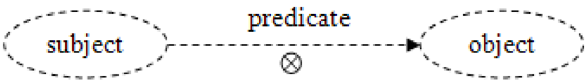

Fig. 9. KPS3

For simplicity, let us consider the knowledge base contains negation of a sentence, stated as an example in KPS1, i. e. the sentence "David does not like chocolate.". The morphism for this sentence (figure 10) is very similar to the morphism for affirmative sentences. To denote that the predicate is in its negative form, the falsum is inside the node representing the predicate.

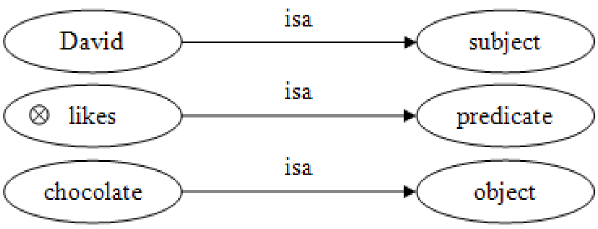

Fig. 10. KPS3 – morphism

After renaming the symbols, the resulting RDF graph contains one RDF triple (figure 11).

Fig. 11. KPS3 – result

The second form of negative declarative sentences contains adverbial complements. It can be an adverbial compelement of manner, place, time or more adverbial complements together. The knowledge pattern for this type of sentence will be marked as KPS4 and is very similar to KPS2. The only difference is, again, in using falsum for marking the predicate in the negative form. Falsum is inside the node representing the predicate. KPS4 is shown on figure 12.

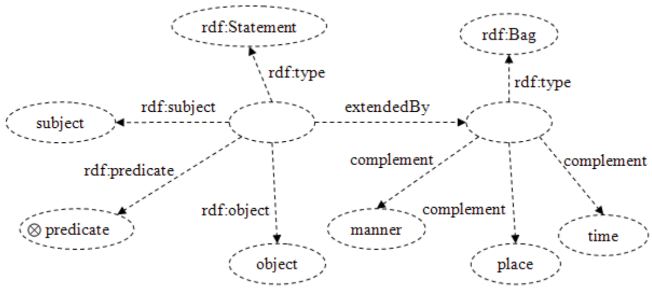

Fig. 12. KPS4

While using this knowledge pattern in a concrete knowledge base, let us consider, again for simplicity, the negation of sentence shown while using of KPS2, i. e. sentence "David does not play the piano loudly at home daily.". The morphism for this sentence is shown in figure 13.

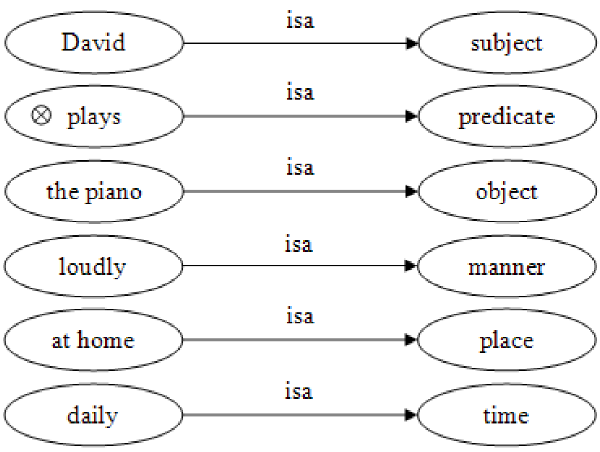

Fig. 13. KPS4 – morphism

The resulting RDF graph (figure 14) was created by renaming the symbols listed in the morphism.

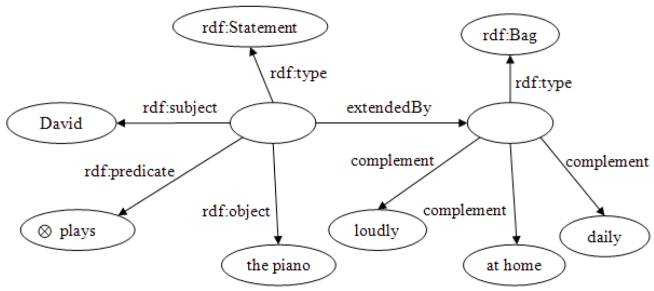

Fig. 14. KPS4 – result

IV. Conclusion and future work

While creating a knowledge base from sentences expressed in natural language, it is important to determine members of particular words in a sentence. The member of a sentence determines the position of a word in the RDF graph. In this paper, it was proposed that the help of knowledge patterns would be a way of converting natural language sentences into RDF graph language. Knowledge patterns are general structures of knowledge, which are not part of the target knowledge base or ontology. An essential part of the use of the knowledge patterns is the specification of renaming non-logical symbols. This part is called morphism. This paper introduced a way of capturing knowledge patterns in extended RDF graph language. A dashed line was used for the representation of knowledge patterns in RDF graphs (in contrast to solid line in classical RDF graphs). Several examples of knowledge patterns and their use are part of this paper.

Future work will be focused on discovering other knowledge patterns and the representation of these knowledge patterns in the RDF/XML languages.

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] 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.

[3] 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.

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

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

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

[7] B. Ganter and R. Wille, "Formal Concept Analysis: Mathematical Foundations," Berlin: Springer-Verlag, 2004, ISBN 3-540-62771-5.

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

[9] A. Lukasová, "Reprezentace znalostí v asociativních sítích," *Znalosti 2001*, ISBN 80-245-0190-2, pp. 245-250. (in Czech)