DOI 10.2298/CSIS100901038S

Representation of Texts in Structured Form

Mladen Stanojević[1] and Sanja Vraneš[1]

[1]University of Belgrade, Institute Mihajlo Pupin,
Volgina 15, 11060 Belgrade, Serbia
{mladen.stanojevic, sanja.vranes}@pupin.rs

Abstract. Although the existing knowledge representation techniques, ranging from the relational databases to the most recent Semantic web languages, are successfully applied in numerous practical applications, they are still unable to represent the information contained in text documents and web pages in structured form, suitable for productive text processing. Text files can represent text documents with no loss of information, however, this information is represented in an unstructured form. Various knowledge formalisms used in different phases of Natural Language Understanding, such as lexical, syntactic, semantic, pragmatic and discourse analysis, are still unable to represent texts in structured form with no loss of information. In this paper, we define the crucial requirements for structured text representation and then, we give a brief introduction to a representation technique that fulfills all these requirements, including the basic data types and learning techniques used to create, maintain and interpret the resulting representation formalism.

Keywords: structured representation, learning, text processing, natural language understanding, regular languages.

1. Introduction

Computers can process information efficiently, only if it is represented in a structured form. Natural language documents and web pages are the examples of the unstructured knowledge representation, so the problem is how to translate them automatically and represent the information contained in a structured form with no loss of information.

The importance of the structured data representation is obvious on an example of a relational database. Data in a relational database are represented in the structured form suitable for the automatic processing in various applications. We can dump this database into a text file that will contain the same information as the database itself and the created text file may be used to backup or transfer the database to another computer, but it is definitely not suitable for the automatic data processing. The databases are convenient means for structured data representation, and the focus of this paper is on a representation technique that will play a role of “a database for texts", where the information found in texts will be represented in the structured form.

Different knowledge representation techniques, like conventional knowledge representation techniques (e.g. relational [1] and object-oriented [2] databases), Artificial Intelligence techniques [3], [4] (e.g. logic formalism, semantic nets, conceptual dependencies, frames, scripts, rules, etc.), or Semantic Web [5] ontology and schema languages (e.g. XOL [6], SHOE [7], OML [8], RDFS [9], DAML+OIL [10], OWL [11]) with enough expressive power to represent any kind of knowledge in structured form suitable for further computer processing, have been successfully applied to support knowledge processing in many different application domains.

However, it has been widely recognized in academic circles that neither of these techniques can be successfully applied in the representation of information found in natural language documents and web pages. They cannot be used to automatically translate various texts (natural language documents, web pages, etc.) into the structured form with no loss of information.

Although these knowledge representation techniques may look rather different, they actually all share the basic principle, which limits their representational ability. This basic representational principle is related to the way we perceive the world around us. We use named symbols to distinguish different phenomena and to capture their semantics.

We observe the world as it is composed of separate and distinct phenomena, objects, entities, which are mutually connected by a set of specific relations. All these phenomena, objects and entities may be more closely described using some features. Hereby, we commonly use names to describe the meaning of observed phenomena, objects, entities, features and relations.

Almost all existing knowledge representation techniques use naming to describe the meaning of represented knowledge. Instead of just representing the knowledge, all these techniques provide also the means for interpreting its meaning using names. Naming actually creates the limitations of the existing knowledge representation techniques. These knowledge representation techniques can be called "symbolic", because they represent the real world domains using simple and complex symbols and the corresponding relations between them. However, there is a knowledge that is not symbolic in its essence (e.g. information found in paragraphs, sections, documents or web pages in natural languages), which cannot be represented using symbolic knowledge representation techniques.

The proposed representation technique is completely equivalent to text files regarding the information represented by these two techniques. However, while text files represent this information in an unstructured form, the proposed technique represents the same information in a structured form suitable for automated text processing.

The first attempt to implement a novel technique able to translate texts into the structured form without loss of information resulted in the proposal of Hierarchical Semantic Form (HSF) and SOUL algorithm [12], while a more advanced approach is applied in Natural language Implicit Meaning Formalization and Abstraction (NIMFA) [13].

The organization of the paper is the following: Section 2 presents some related work, which outlined the ideas beyond symbolic knowledge representation knowledge representation formalisms used to represent different kind of knowledge in Natural Language Understanding, Section 3 describes the main characteristics of natural languages and the requirements for structured representation of texts, Section 4 gives the basic insights into the implementation of the requirements for the representation of texts in structured form, while Section 5 provides some conclusions.

2. Related Work

There is an acute lack of references related to the structured representation of natural language documents. However, there are some papers that outline a new way of thinking, which is beyond symbolic knowledge representation.

The so-called “radical connectionism” [14] claims that the natural language is not used as representational, but rather as communicational medium. The modification of the “localist” approach of the “connectionist” model [15] could be used as a starting point for the implementations of ideas of radical connectionism. One implementation of a knowledge representation not based on symbols is represented by Hierarchical Temporal Memory (HTM) [16], used for image processing, while another solution [17], used in text processing, relies on the Hopfield-like neural networks.

However, these approaches failed in defining clear requirements for structured representation of natural language documents, which will not be based on symbols. In structured representation of natural language documents, the tasks of representing knowledge is clearly separated from the task of interpreting the meaning of the represented knowledge, which provides the basis for overcoming the limitations of symbolic knowledge representation.

The representation technique proposed in this paper should facilitate text processing and more precisely Natural Language Understanding (NLU), which is actually not in the focus of this paper, hence, we will only pass briefly through the knowledge representation formalisms used to represent different forms of knowledge relevant for NLU [18] (e.g. morphological, syntactic, semantic, pragmatic and discourse knowledge).

The morphological knowledge used in lexical analysis is usually represented by Finite State Automata (FSA) and Finite State Transducers (FSTs) to implement the electronic dictionaries for the given natural languages. The morphological knowledge used in lexical analysis corresponds to the structure of words.

In syntactical analysis various classical, statistical and connectionist approaches are used, whereby classical and statistical approaches are based on the corresponding grammars. Among the most popular grammars are the dependency grammars, which, as a result of parsing a statement, produce the corresponding dependency tree to reflect the syntactic structure of the given statement. The grammars used in syntactic analysis are language dependent and subjective, because even for the same language different linguists may propose different grammars.

The semantic knowledge, as a result of semantic analysis of a sentence, can be represented in logical form (e.g. first order predicate calculus - FOPC). However, the statements in FOPC can represent the meaning of natural language sentences only, whereby some information considered semantically irrelevant may be lost. Moreover, FOPC statements are not structured.

The discourse knowledge is used in different discourse structuring theories like the theory of segmentation, attention shift and hierarchical inclusion of topic-related discourse segments [18] and Rhetorical Structure Theory [19]. However, the structures used to represent the discourse knowledge are only temporarily used to interpret correctly the meaning of sentences in the given context. There are also different discourse representation models [20] usually implemented using the specific ontologies, but all these models are manually built.

As we can see neither of the above described knowledge representation formalism used in NLU can be used to automatically translate texts into the structured form with no loss of information. They are either too specific, enabling the representation of words or phrases or sentences, or are temporal by their nature and not designed for persistent storage of texts in the structured form.

Another way to represent information found in a text in the structured form is to use some information extraction technique, which combines an NLU technique with a suitable knowledge representation technique [21], [22]. However, information extraction techniques also have some drawbacks: 1) they cannot be used to represent the complete information found in a text, but only a fraction of it; 2) information extraction systems are language dependent (they use language dependent dictionaries and grammars); 3) for each new subset of data that should be extracted from a text, a new information extraction system must be developed.

3. Requirements for Representation of Texts in Structured Form

Various natural language documents and web pages are represented in the plain text form and it seems that there is no apparent structure behind it. However, there is an intrinsic structure behind any natural language. Unlike the syntactic structures, which are subjective and language dependent, this structure is objective and the same for all languages.

This structure is composed of letters, syllables, words, phrases, sentences, paragraphs, etc. Hereby, syllables are composed of letters, words are composed of syllables, phrases are composed of words and so on. Another important characteristic of natural languages is their sequentiality. Each structural element at the higher level is composed of a sequence of structural elements at the lower level.

Having in mind these simple observations, we can formulate two basic requirements for structured representation of text documents:

1. Provide support for context representation, i.e. provide means for determining the exact position of each structure element regarding the hierarchical level and place in the corresponding sequence(s).
2. Provide unique representation for each structure element in different contexts.

Strictly speaking, the context of a structure element at the given hierarchical level is defined by its predecessor and successor structure element, where each structure element must be uniquely represented.

Why these two requirements are so important? Because they enable an efficient search of natural language documents. Each word, each phrase, each sentence, etc. would be uniquely represented in a hierarchical network that could represent thousands of natural language documents or web pages. We would be able to efficiently identify all contexts in which the given word or phrase appear in, i.e. all sentences, paragraphs, sections and documents. The search through the hierarchically organized structure must be much faster compared to the sequential search of natural language documents represented in an unstructured form.

The above-described requirements are also in line with the fact well known in linguistics that any limited language is at the same time a regular language. Since all natural languages are limited, all text documents can be represented using a deterministic finite state machine. A Finite State Automaton (FSA) [18] in lexical analysis is usually used to locate morpho-syntactic patterns in corpora.

A common way to represent text documents is to use files, which can be observed as Finite-State Automata, where each text character is followed by the next character until the end-of-file mark is encountered leading to the terminating state. However, for information to be processed by computers effectively, it must be represented in a structured form. Since files represent text documents in plain form, they are not particularly useful for automatic processing.

Recursive Transition Networks (RTNs) [18] are used in syntactic analysis to construct various grammars. However, RTN can be also used to introduce structure to FSA, where a part of an existing FSA can be named to represent a simple graph. The only problem with RTN is that it is manually constructed where all constituent graphs must be named.

Eventually, both requirements for structured text representation are satisfied by RTN, but in its present form it is useless, because it must be constructed manually.

4. Technique for Representing Texts in Structured Form

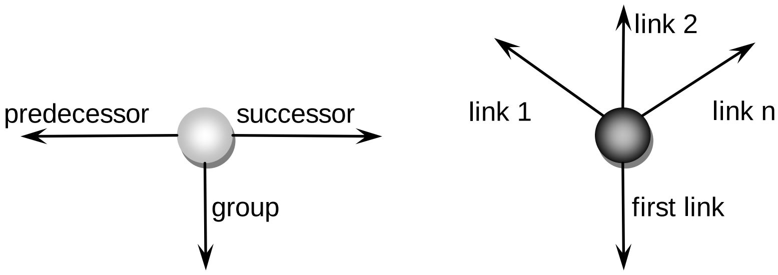
In this section, we will describe a technique for representing texts in the structured form with its basic data types and “learning by repetition” algorithm that facilitates the automatic creation and maintenance of the corresponding hierarchical network. This hierarchical network is composed of two types of nodes where **groups** are used to uniquely represent the natural language structures (e.g. words, phrases, sentences, groups of sentences, paragraphs, groups of paragraphs, documents), while **links** are used to represent these natural language structures within different contexts (e.g. words in phrases, phrases in sentences, sentences in paragraphs, etc.). We will, then, give a simple example of the hierarchical network created using the described representation technique, provide an algorithm for translating texts into structured form and finally, we will just briefly describe how the hierarchical network can be interpreted to find its meaning.

4.1. Basic Data Types

Any technique suitable for the representation of texts in the structured form must satisfy the above described requirements related to context representation and unique representation of structure elements. The network, similar to RTN, which will represent texts in structured form can be built using two basic data types corresponding to two defined requirements:

Link data type (Fig. 1.a) enables the context representation and implements the ternary, sequential relation between the previous, current and successive structure element. Since each word and any other natural language structure may appear in different contexts (e.g. the same word may appear in different phrases or sentences), links are used to represent this natural language structure in all these contexts. It contains pointers to its predecessor and successor, but also to the structure element it represents in the given context. The successor of the last link in the sequence points to the group which represents this sequence. The link data type corresponds to states in RTN.

Group data type (Fig. 1.b) supports unique representation of all structure elements (characters, syllables, words, phrases, sentences, etc.). This means that any natural language structure is represented only once in a hierarchical structure that may represent many natural language documents. Instead of repeating the same structure element in different contexts, we use links to represent the corresponding group in these contexts. Each group contains a pointer to the first link of the sequence it represents, but also pointers to all the links that represent the corresponding structure element in different contexts. The group data type corresponds to transitions in RTN.



a) link

b) group

Fig. 1. Basic data types

Any natural language document has inherent, hierarchical structure, which may not be readily visible when we read it, composed of characters at the lowest levels, then words, phrases, sentences, paragraphs, etc. Since the main characteristic of all natural languages is their sequentiality, we may say that any natural language document can be represented as a hierarchical structure comprised of sequences at different levels of abstraction. The hierarchical structure used to represent texts in structured form contains groups representing characters at the lowest level, followed by syllables which represent sequences of letters, words as sequences of syllables and letters, phrases as sequences of other phrases and words, sentences as sequences of phrases and words and so on. Hereby, sequences are represented using links.

Formally, the information in structured text representation can be represented by the space *S* defined by the triple of groups *G*, links *L* and sequences *Q* (composed of links from *L*):

$$S = \{G, L, Q\},\ g_i \in G, i = 1,r,\ l_j \in L, j = 1,s,\ q_k \in Q, k = 1,t \tag{1}$$

Initially, *G* contains only groups corresponding to letters, *L* contains only links corresponding to these atomic groups and *Q* is an empty set of sequences.

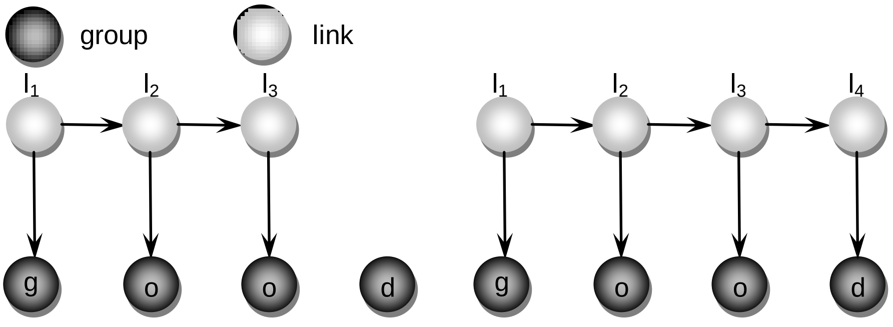
Structured text representation follows the two basic principles in knowledge representation, the principle of locality (context representation) and the principle of unique representation.

Principle of locality defines the transition *T* from the link l_t, which is the last link in the subsequence q_i, to the link l_u, when group g_c belonging to the same hierarchical level appears at the end of subsequence q_i:

$$l_u = T(l_t, g_c),\ q_p = q_i l_u,\ l_u \to g_c,\ q_i, q_p \in Q,\ g_c \in G,\ l_t, l_u \in L \tag{2}$$

The link l_u represents the group g_c in the subsequence q_p, which extends the subsequence q_i. If g_c is a new group, or link l_u does not exist, then the new link l_u must be created. The principle of locality enables learning of new sequences.

For instance, suppose that the system learns the word “good” and that it already represented the sequence “goo” (Fig. 2.a) and that it has to add now “d” to complete the sequence (Fig. 2.b). In our example, l_u = l_4, l_t = l_3, g_c corresponds to a group representing the letter “d”, q_i. corresponds to a sequence of letters “goo” represented by links l_1, l_2 and l_3, while q_p. corresponds to a sequence of letters “good” represented by links l_1, l_2, l_3 and l_4.



a) unfinished sequence

b) finished sequence

Fig. 2. Principle of locality

This principle is related to the representation of sequences at different levels of hierarchy (words, phrases, sentences, paragraphs, etc.). It basically states that paragraphs are composed of sentences and not of letters or words.

Principle of unique representation states that each subsequence (q_x) that repeats in two different sequences (contexts, q_i, q_j) must be uniquely represented by the corresponding group (g_u):

$$\begin{aligned}
&g_s \rightarrow q_i, q_i = q_a q_x q_b\ ,\ g_s \in G\ ,\ q_a, q_b, q_i, q_s, q_x \in Q \qquad (3)\\
&g_t \rightarrow q_j, q_j = q_c q_x q_d\ ,\ g_t \in G\ ,\ q_c, q_d, q_j, q_t, q_x \in Q\\
&g_u \rightarrow q_x\ ,\ l_p, l_q \rightarrow g_u\ ,\ l_p, l_q \in L\ ,\ g_u \in G\ ,\ q_x \in Q\\
&g_s \rightarrow q_i, q_i = q_a l_p q_b\\
&g_t \rightarrow q_j, q_j = q_c l_q q_d
\end{aligned}$$

Hereby, subsequences q_a or q_b, q_c or q_d may be empty, i.e. they may contain no links. When a subsequence q_x repeats in two sequences (q_i, q_j), a new group g_u will be created corresponding to this subsequence, as well as two new links (l_p, l_q) representing this subsequence in two different contexts (q_i, q_j). This is an example of self-organization of the space S, which allows an automatic identification of semantic concepts, structures and relations.

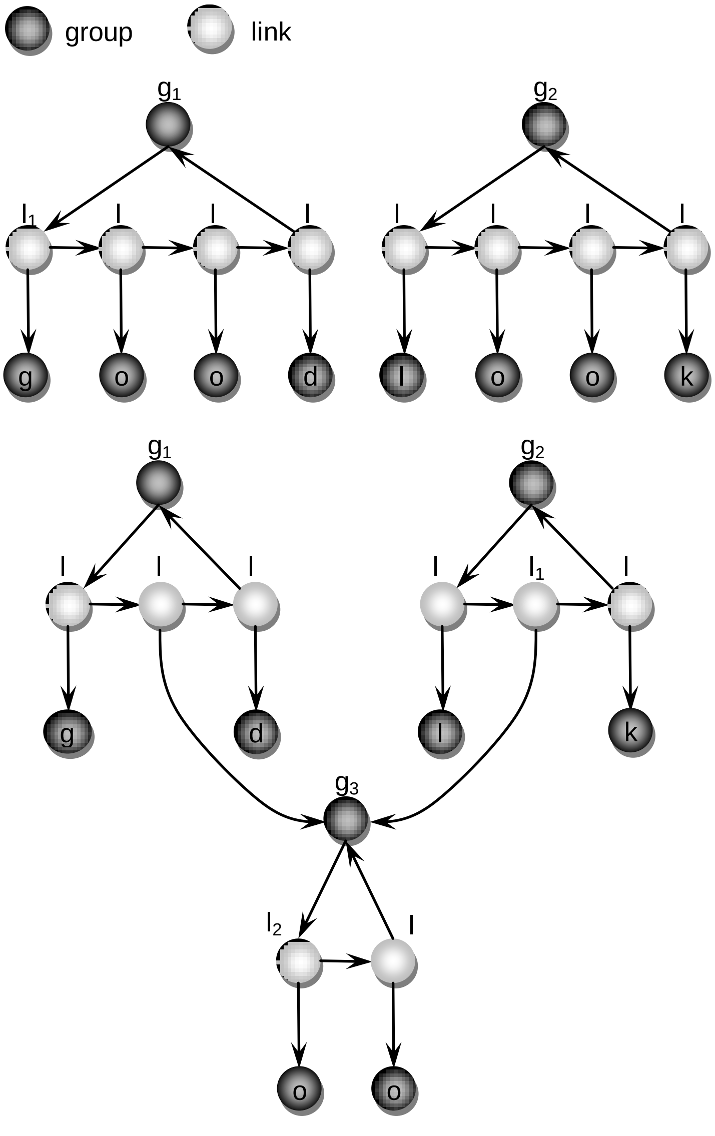


Fig. 3. Principle of unique representation

As an example of principle of unique representation we will use two words “good” and “look”. We can immediately notice that these words share the same subsequence of letters “oo”, which should be uniquely represented (Fig. 3).

In our notation group g_s = g_1 uniquely represents the word “good” and the corresponding sequence of letters q_i, group g_t = g_2 uniquely represents the word “look” and the corresponding sequence of letters q_j where q_a = l_1 corresponds to letter “g”, q_x = l_2l_3 to sequence of letters “oo” in word “good” and q_x = l_6l_7 to the same sequence of letters in word “look”, q_b = l_4 to letter “d”, q_c = l_5 to letter “l” and q_d = l_8 to letter “k”. Since the same sequence of letter “oo” appears in both words, we should represent this sequence uniquely. Therefore, we create a group g_u = g_3 to uniquely represent this sequence and then a link l_p = l_9 to represent the letters “oo” in the word “good” and a link l_q = l_{10} to represent the same sequence of letters in the word “look”.

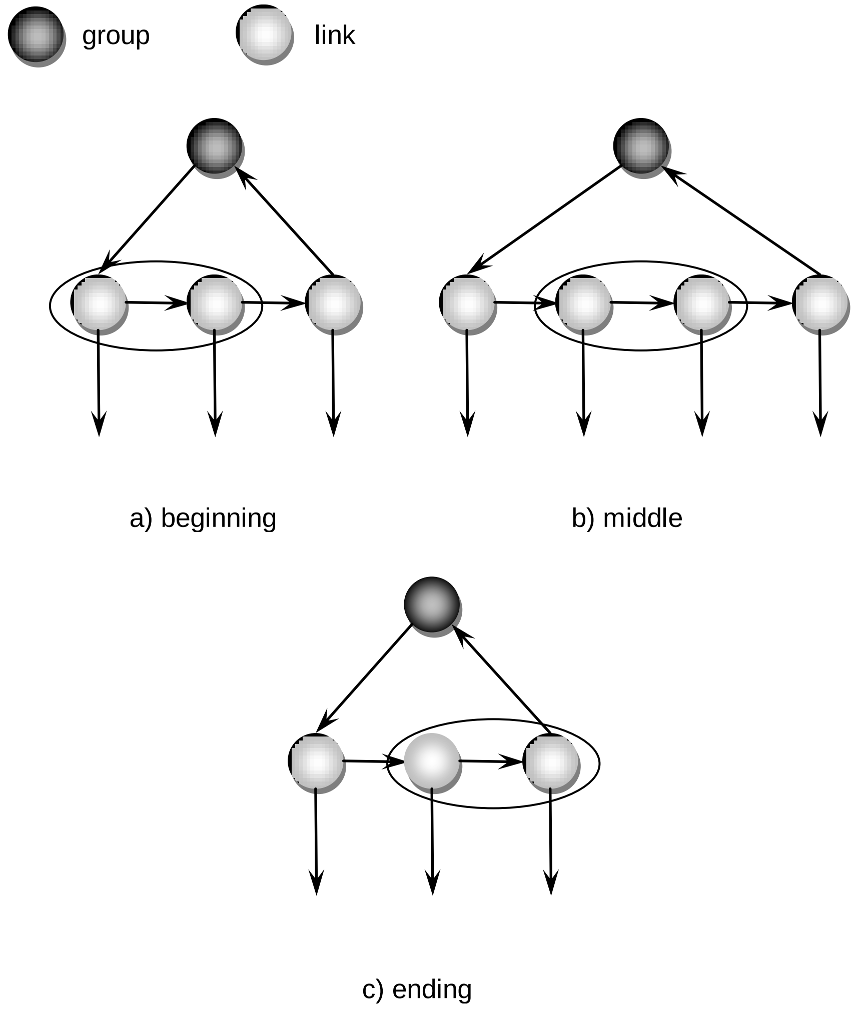
4.2. Learning by Repetition

“Learning by repetition” is one of the basic forms of learning inherent to human beings. We cannot observe different phenomena unless they are repeated. In the same way, we learn to speak languages. “Learning by repetition” enables us to learn the structure, i.e. words, phrases, sentences, etc. of any natural language. Notice that “learning by repetition” doesn’t allow us to understand the meaning of language structures, but only to distinguish them.

However, when linguists speak about the structure, they usually refer to syntactic structure of a natural language statement. A dependency tree representing the structure of a natural language statement can be created by parsing this statement using the corresponding dependency grammar. “Learning by repetition” cannot be used to create a dependency tree, because it is not related to any dependency grammar and syntactic structures. The syntactic structures defined by dependency grammars are language dependent and subjective in their essence, whereas the structures (words, phrases, etc.) created by “learning by repetition” are language independent.

“Learning by repetition” facilitates unique structure representation and can be defined in the following way: if two contexts share the same subsequence, this subsequence then represents a new structure element which is shared by two contexts. For this new structure element, a new group will be created and two new links will represent this structure element in the given contexts. Strictly speaking, this definition is slightly different from the definition used in psychology where the same structure should be repeated many times to be memorized.

In practice, one of these two contexts will be a referential context, which has been already created, while the second one is a new one and has not yet been represented. Basically, we have three possible cases when “learning by repetition” can take place: the repeated subsequence appears at the beginning of the referential context (Fig 4.a), it occurs in the middle of the referential context (Fig. 4.b), or it takes place at the end of the referential context (Fig. 5.c).



a) beginning

b) middle

c) ending

Fig. 4. Position of repeated subsequence

As an example of “learning by repetition”, we will take two sentences:

1. “Bill was in the school”
2. “Bill is in the school”.

Suppose that we have already entered all words and that we have presented the first sentence. We will have a hierarchical network as presented in Fig. 5. Notice that groups represented at the bottom of the hierarchical representation are actually not named and that these labels are used only to simplify the diagram and hide the part of structure representing the corresponding words.

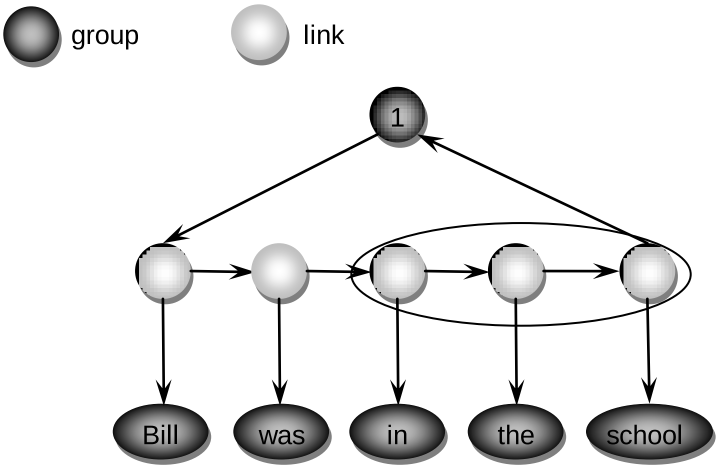


Fig. 5. Hierarchical representation of the first sentence

When the second sentence is processed by the structured text representation technique, it will discover that the ending subsequence “in the school” is repeated (Fig. 5) and reorganize the hierarchical network to represent correctly the second statement (Fig. 6).

The representation technique presented in this paper enables the creation of the self-organizing hierarchical network, which changes as new texts are fed to it. It reuses the part of information that has been already represented and adds the new one. Unlike symbolic knowledge representation technique, this knowledge representation technique is able to automatically translate any natural language text into the corresponding hierarchical structure and vice versa with no loss of information.

4.3. Example

As an example of structured text representation, we will use the same two simple sentences to represent the context correctly:

1. “Cows eat grass.”
2. “Cats eat mice.”

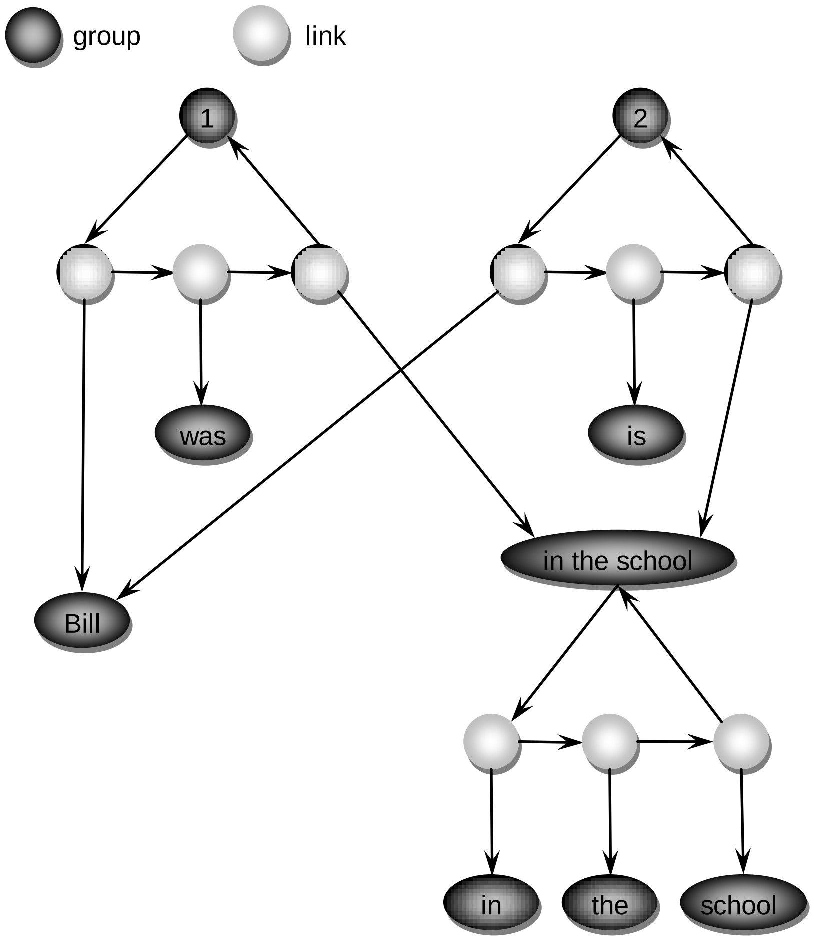


Fig. 6. Learning by repetition

At the beginning, the empty hierarchical structure is comprised only of groups representing single characters. There are no links, because no text is fed to it yet.

The learning process begins with feeding the single words. This is actually quite similar to the way babies are learning to talk. After we have fed all the words from our simple statements, we will get hierarchical network as presented in Fig. 7.

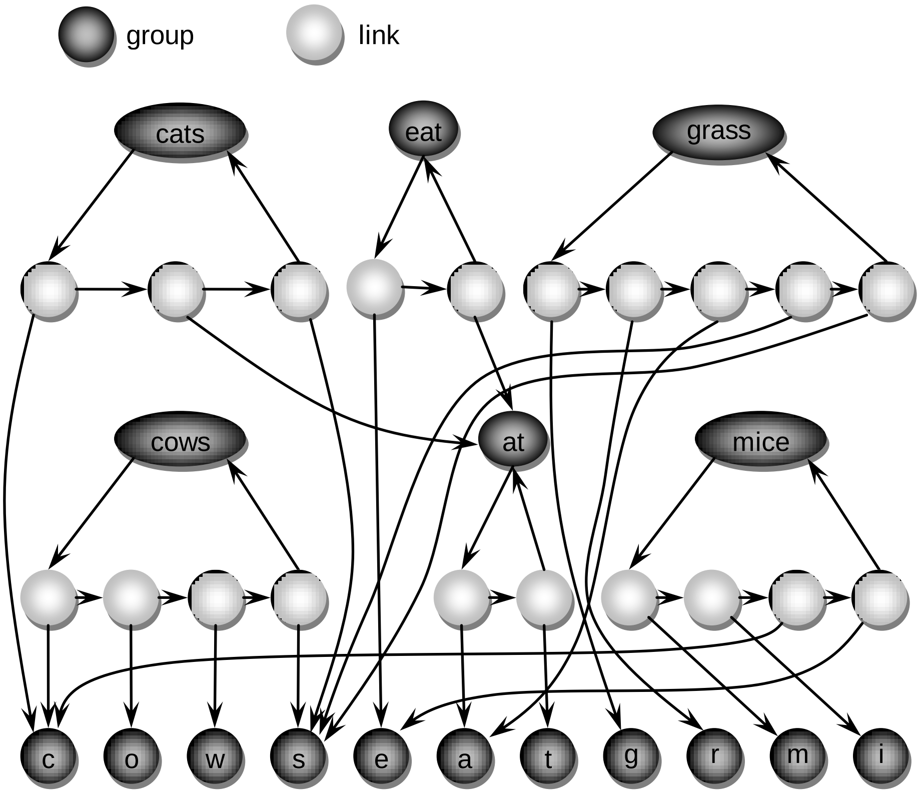


Fig. 7. Representation of single words

After both sentences are processed the hierarchical structure will have the form as presented in Fig. 8.

The same representation technique can be used to represent sentences, paragraphs, sections, etc. represented as the corresponding groups in Fig. 9.

4.4. Translation of Texts into Structured Form

Any natural language text can be translated automatically into the structured form composed of groups representing uniquely syllables, words, phrases, sentences, groups of sentences, paragraphs and group of paragraphs using the algorithm given below.

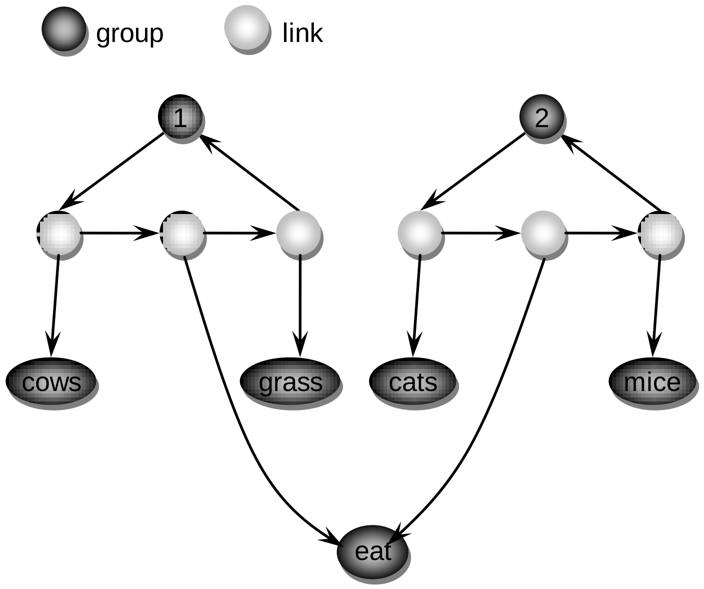


Fig. 8. Correct context representation

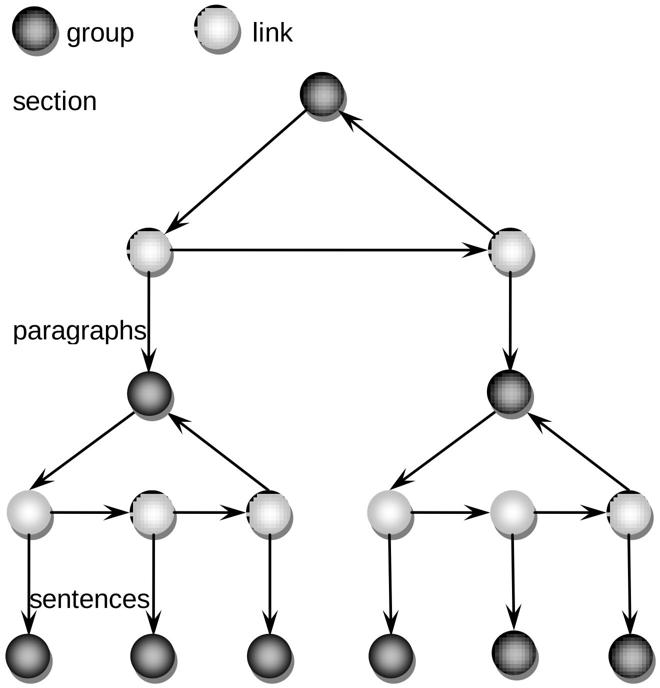


Fig. 9. Representation of sentences, paragraphs and section

```
createStructuredTextForm(fileName, currDocumentGroup)
{
    paragraphGroups = new DoubleLinkedList();
    paragraphsGroups = new DoubleLinkedList();
    textFile = new File();
    textFile.open(fileName);
    currPos = 0;
    currChar = textFile.readAt(currPos);
    // while not end of file
    while (currChar != eof)
    {
        sentenceGroups = new DoubleLinkedList();
        sentencesGroups = new DoubleLinkedList();
        // while not end of paragraph
        while (currChar != eop)
        {
            charGroups = new DoubleLinkedList();
            constructGroups = new DoubleLinkedList();
            // while not end of sentence
            while (currChar != eos)
            {
                convertCharToGroup(currChar, charGroup);
                charGroups.addTail(charGroup);
                currPos++;
                currChar = textFile.readAt(currPos);
            }
            pos = charGroups.getHeadPosition();
            while (pos != null)
            {
                // Identify existing syllables, words or
                // phrases
                identifyLongestSequence(charGroups, pos,
                    nextPos, currGroup);
                constructGroups.addTail(currGroup);
                pos = nextPos;
            }
            if (constructGroups.Count() > 1)
                // Create a new sentence
                currSentenceGroup =
                    createGroup(counstructGroups);
            else
                // The sentence is already defined
                currSentenceGroup = currGroup;
            sentenceGroups.addTail(currSentenceGroup);
            delete charGroups;
            delete constructGroups;
        }
```

```
            pos = sentenceGroups.getHeadPosition();
            while (pos != null)
            {
                  // Identify existing groups of sentences
                  identifyLongestSequence(sentenceGroups, pos,
                        nextPos, currGroup);
                  sentencesGroups.addTail(currGroup);
                  pos = nextPos;
            }
            if (sentencesGroups.Count() > 1)
                  // Create new paragraph
                  currParagraphGroup =
                        createGroup(sentencesGroups);
            else
                  // The paragraph is already defined
                  currParagraphGroup = currGroup;
            paragraphGroups.addTail(currParagrapghGroup);
            delete sentenceGroups;
            delete sentencesGroups;
      }
      pos = paragraphGroups.getHeadPosition();
      while (pos != null)
      {
            // Identify existing groups of paragraphs
            identifyLongestSequence(paragraphGroups, pos,
                  nextPos, currGroup);
            paragraphsGroups.addTail(currGroup);
            pos = nextPos;
      }
      if (paragraphsGroups.Count() > 1)
            // Create new document
            currDocumentGroup = createGroup(paragraphsGroups);
      else
            // The same document is already entered
            currDocumentGroup = currGroup;
      delete paragraphGroups;
      delete paragraphsGroups;
      textFile.close();
      delete textFile;
}
```

The algorithm is described using the `createStructuredTextForm` procedure given in a pseudo-code form. This procedure has two arguments: `filename` is an input argument representing the name of the text file; `currDocumentGroup` is an output argument representing the text file in a structured form. This hierarchically organized, structured form is composed of groups of paragraphs, paragraphs, groups of sentences, sentences, phrases, words, syllables and characters at the bottom of this hierarchy. Hereby, each of these natural language structures is uniquely represented for all text files that may be translated using the given procedure.

The procedure divides the text documents into sentences and paragraphs. It then translates one by one character found in a sentence into character groups (`convertCharToGroup`) and adds these character groups to the `charGroups` list.

The `identifyLongestSequence` procedure is used to implement the principle of unique representation based on learning by repetition. It has four arguments: the first one is a double-linked list of groups, the `pos` argument is the position in this list from where the procedure starts to identify the existing sequence, the `nextPos` argument is the position in the list where we begin the next search and the `currGroup` argument is a group representing the longest found sequence. Actually, this procedure implements three possible cases:

The group found at `pos` in linked list is not followed by the next group in the same list in any previously created sequence. The procedure returns `nextPos` as the position of the next group in the list and `currGroup` is the group found at `pos`.

All groups found in a double-linked list are already defined as a sequence represented by a single super group. The procedure returns `null` as the value of `nextPos` and the super group for `currGroup`.

Learning by repetition is applied and for the recognized subsequence of groups a new group is created. The `nextPos` points to the first group after the recognized subsequence in the linked list and the new group is returned as `currGroup`.

The function `createGroup` implements the principle of locality. The double-linked list containing groups is the only input argument and the function returns the super group representing uniquely the sequence of groups contained in the given list. For the contained groups the function first creates the corresponding links, then, it connects these links to represent a new sequence and finally creates a super group that uniquely represents the newly created sequence.

The hierarchical organization representing natural language texts serves as “a database for texts”, because it enables a structured representation of texts with “indexes” for each word, phrase, sentence, group of sentences, paragraph and group of paragraphs. However, unlike ordinary databases which require human experts to design them and create indexes manually, this “database for texts” together with the corresponding “indexes” is automatically created, with no help from human experts.

Knowing that the algorithm easily determines the repeated sentences, groups of sentences, paragraphs and groups of paragraphs, one obvious application would be the determining of originality of text documents. Not only can the algorithm identify the repeated parts of texts, but it can also easily determine the provenance of copied parts. The other approaches sequentially compare the given text with the other texts, which can be very time-consuming, especially when there are many texts to check.

Another very useful application of the algorithm would be in the keyword search. The present day search engines provide many unproductive hits, because they search for keywords in the whole document, whereas they should try to locate the sentences or paragraphs containing these keywords. The algorithm explicitly stores the information about the context (sentences, paragraphs) and since all words and phrases are indexed in all encompassing natural language structures (sentences, paragraphs, documents) it is easy to find the given keywords in any desired context in all represented documents.

4.5. Interpretation of Meaning

Symbolic knowledge representation techniques define the meaning of represented knowledge using names, so they do not have to interpret the meaning.

On the other hand, structured text representation technique does not use the names to describe the meaning; hence, the hierarchical structures used to represent plain texts are not comprehensible to humans. These hierarchical structures have to be interpreted somehow to extract the meaning from them.

However, from the practical point of view, it seems that the emphasis should not be on the meaning but on the closely related issue such as relevance. In semantic knowledge representation, the role of a domain expert is very important, because he/she decides what objects, relations, features, etc. are relevant for the given domain of application. This relevance is actually defined having in mind the application domain.

It seems that human beings determine the relevance of things based on the appearance of differences in similar contexts. For instance, in distinguishing the twin brothers, we do not rely on similarities in their appearance, but on the small differences we can notice.

In case on natural languages, our attention is also attracted by differences. If we observe the phrases:

1. “Good morning”
2. “Good afternoon”
3. “Good evening”

we will immediately notice that the first word “Good” is repeated in all of them, but also that the second word is always different. Since it spots the difference in the similar contexts, our brain considers this difference as important and generalizes the three different words appearing at the end of these three phrases. This generalization actually gives the importance to things and represents the basis of semantics. These three words may be taken as instances of a newly discovered, simple semantic category.

Since structured text representation requires the representation of context, the corresponding techniques should be able to automatically discover semantic categories in the given contexts. To do this, these techniques should support two types of learning: “learning by generalization” and “learning by specialization”.

“Learning by generalization” may be defined as follows: when two similar contexts differ from each other only by two different structures found in the same place in these contexts, then these two structures can be generalized. A new semantic category is discovered and two structures represent the instances of this newly discovered category.

“Learning by specialization” may be similarly defined: when a context contains a structure which appears at the same place as the structures that have been already generalized in similar contexts, this new structure can be considered as a new instance of the same semantic category.

In our example with three phrases, “learning by generalization” may be applied on the first two phrases, where words “morning” and “afternoon” appear at the same place in similar contexts and therefore, may be generalized to create a new semantic category. When the third phrase is considered, “learning by specialization” may be applied, because the word “evening” appears at the same place as the generalized words “morning” and “afternoon”, so, the word “evening” may be considered as a new instance of the same semantic category.

We can force “learning by generalization and by specialization” by using the explicit definitions and this is what we usually think when we speak about semantics:

1. “John is a boy”
2. “Bill is a boy”
3. “Tom is a boy”

The phrase “is a boy” is repeated in these three phrases, while “John”, “Bill” and “Tom” are instances of a semantic category. However, in the proposed knowledge representation techniques, this semantic category is actually not named and is not even defined as a separate structure. Only the structures corresponding to words “John”, “Bill” and “Tom” are marked as instances of a semantic category and when we ask a question:

“What is John”

we will be able to find the answer:

“John is a boy”

“Learning by generalization and by specialization” can be used to identify simple semantic categories and their instances. Instances of simple semantic categories are words and phrases. The process of discovering simple semantic categories is actually very similar to the process of identifying symbols in symbolic knowledge representation.

However, the processing power of the proposed representation techniques should go beyond symbolic processing. These techniques should also be able to process sentences, paragraphs, sections, documents, web pages, etc.

The instances of simple semantic categories take part in more complex structures like complex phrases, sentences, paragraphs, etc. thus creating complex semantic categories or patterns. These complex semantic categories could be used to recognize natural language commands or to find some information or documents. However, the process of interpreting the meaning of represented texts is out of the scope of this paper. More information about the possible implementation of this process can be found in [13].

5. Conclusions

Although it seems that the existing knowledge representation techniques do not have much in common, almost all of them can be described as symbolic techniques. Actually, they are all designed to represent symbols, i.e. clearly separated entities (objects, phenomena) with defined features and relations that are relevant in the given domain of application. They all use names to describe the meaning of represented knowledge. So, these techniques besides providing means for knowledge representation also provide means for the interpretation of meaning. They facilitate symbolic knowledge representation and symbolic processing.

However, the symbolic knowledge representation techniques simply do not have the necessary representational power to represent texts in structured form. They do not provide the means to represent natural language structures, which are hierarchical and sequential by their nature, nor the means to process texts effectively.

Different knowledge representation formalisms used in text processing to represent various kinds of knowledge like morphological, syntactic, semantic, pragmatic or discourse are also not suitable for structured text representation. They are actually designed to represent only a specific kind of knowledge related to word structure, syntactic structure, meaning of words or discourse context.

However, it is well known in linguistics that any limited language is at the same time a regular language. So, a Finite State Automaton could be used to represent any text document, and this is exactly how texts are represented in files. The problem with text files as representation formalism is that they are not structured and thus not convenient for automatic text processing. Recursion Transition Networks (RTN) can be used to structure a graph represented by a Finite State Automaton, but the corresponding sub-graphs must be named and that's why standard RTNs are not suitable for structured text representation.

In this paper, we have defined a novel technique for structured text representation which resembles the RPNs, but can be used to automatically translate texts into the structured form. It can be viewed as "a database for texts", where unlike the relational databases, which must be designed, this "database" is automatically created from the texts.

We have defined first two requirements that must be fulfilled by a technique to be able to represent texts in the structured form. The first requirement is related to the accurate context representation, i.e. the sequential order of natural language structures that comprise a more complex structure, while the second requirement is related to the unique representation of natural language structures in different contexts.

We have then defined two data types corresponding to two requirements for structured text representation: link data type (corresponding to RPN states) is used to satisfy context representation requirement, while group data type (corresponding to RPN transitions) satisfies the requirement for the unique structure representation.

The hierarchical structure representing text documents is created using groups and links and a special form of learning, which we call "learning by repetition". "Learning by repetition" enables learning the structure of natural languages, by identifying the repeated subsequences of structure elements. It facilitates an automatic translation of any natural language document into the structured form and vice versa with no loss of information. The created structure is self-organizing and changing as new knowledge is fed to it, whereby the old knowledge is reused and the new one is added.

Two other types of learning: "learning by generalization" and "learning by specialization" enable the interpretation of the represented knowledge. "Learning by generalization" supports the discovery of new semantic categories, while "learning by specialization" enables the definition of new instances of the existing semantic categories. These two types of learning facilitate the definition of simple semantic categories and the corresponding instances represented by words and phrases.

Complex semantic categories or patterns are represented by complex natural language structures composed of instances of simple and complex semantic categories. Complex semantic categories enable text processing needed for the understanding of natural language commands or finding the necessary complex information based on natural language queries.

Text files represent basically the same information as the proposed representation technique, but in an unstructured form. In the proposed technique information is structured in the way that each natural language structure (word, phrase, sentence, paragraph, etc.) is uniquely represented in all contexts in which it appears. The created structure facilitates an easy identification of all contexts in which some natural language structure appears giving rise to an efficient text processing and many practical applications.

There are many possible practical applications of structured text representation in areas like continuous speech recognition, question answering systems based on natural language queries, information retrieval, user interfaces based on natural language commands, machine translation, etc.

Continuous speech recognition systems are usually based on phonetic and phonological knowledge and the representation of contexts can enhance the precision of such systems by providing all candidate words that can succeed the last recognized word. The use of word context could significantly reduce the set of candidate words and considerably increase the probability that the correct word will be recognized.

As we all know, standard search engines based on keywords are not very useful when we use natural language queries. We usually get many unproductive hits. There are two reasons for such a performance: 1) search engines do not take care about the context; 2) they do not use semantic categories to abstract the complexities of natural language. All search engines implicitly take a document or web page as a context in which keywords are searched (reason for bad precision rate). However, sometimes we expect that these keywords must be found in the same sentence or the same paragraph. Standard search engines have poor recall rate when using natural language queries because the same thing can be said in many different ways. Semantic categories are coping efficiently with the richness of a natural language, but they are not supported in standard search engines. The structured text representation facilitate context representation and semantic categories, hence, it could be used to implement an efficient question answering system based on natural language queries.

Information retrieval is usually also based on keyword search, therefore, the same limitations hold as for search engines. Information retrieval based on structured text representation can improve the recall and precision rate again by using the context representation and semantic categories. For instance, instead of using combinations of keywords, patent attorneys could easily find all semantically correlated patents using the patent they would like to check.

Natural language-based user interfaces could be easily built using the structured text representation and semantic categories to cover many possible ways how humans can express natural language commands.

Machine translation could also benefit from the use of structured text representation technique, because parallel text corpora in different languages could be fed to it and the same group could then be used to represent the same word, phrase, sentence, paragraph, etc. in different languages. Thus, it would be possible to easily identify the highest level of translation (be it a paragraph, sentence, phrase or word) of a text based on the parallel text corpora represented in structured form.

The structured text representation technique presented in this text has been already successfully applied in question answering systems based on natural language queries. It was implemented first in a prototype system [12] that provides information about flight timetable for the largest European airlines, while the second implementation was in a prototype web portal [13] providing information about flights, football matches and weather forecasts.

Acknowledgement. This work has been partially supported by the EU Seventh Framework Programme project LOD2 (Pr. No.: 257943) and partly by the Ministry of Education and Science of the Republic of Serbia (Pr. No.: TR-32010).

References

1. Date, C.J.: Database in Depth: Relational Theory for Practitioners. Sebastopol, CA: O'Reilly Media, Inc. (2005).
2. Russell, C. et al: The Object Data Standard: ODMG 3.0. San Francisco, CA: Morgan Kaufmann. (2000).
3. Sowa, J.: Knowledge Representation: Logical, Philosophical, and Computational Foundations. Pacific Grove, CA: Brooks/Cole Publishing Co. (2000).
4. Vraneš, S., Stanojević, M.: Prolog/Rex - A Way to Extend Prolog for Better Knowledge Representation. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 6, No. 1, 22-37. (1994).
5. Fensel, D., Hendler, J.A., Lieberman, H., Wahlster, W. (Eds.): Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential. Cambridge, MA: MIT Press. (2003).
6. Karp, R. et al: XOL: An XML-Based Ontology Exchange Language (version 0.4). (1999). Available: http://www.ai.sri.com/pkarp/xol/ (current February 2011).
7. Heflin, J. et al: SHOE: A Knowledge Representation Language for Internet Applications. Technical Report, CS-TR-4078 (UMIACS TR-99-71), Dept. of Computer Science, University of Maryland. (1999).
8. Kent, R. Ontology Markup LanguageVersion 0.3. (1999). Available: http://www.ontologos.org/OML/OML%200.3.htm (current February 2011).
9. Brickley, D., Guha, R.V., (Eds.) RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation. (2004). Available: http://www.w3.org/TR/rdf-schema/ (current February 2011).
10. McGuinness, D., Fikes, R., Handler, J., Stein, L.: DAML+OIL: An Ontology Language for the Semantic Web. *IEEE Intelligent Systems*, Vol. 17, No. 5, 72-80. (2002).
11. McGuinness, D., van Harmelen, F. (Eds.): OWL Web Ontology Language – Overview. W3C Recommendation. (2004). Available: http://www.w3.org/TR/owl-features/ (current February 2011).
12. Stanojević, M., Vraneš, S.: Knowledge representation with SOUL. *Expert Systems with Applications*, Vol. 3, No. 1, 122-134. (2007).
13. Stanojević, M., Vraneš, S.: NIMFA - Natural language Implicit Meaning Formalization and Abstraction, *Expert Systems With Applications*, Vol. 37, No. 12, 8172-8187. (2010).
14. O'Brien, G., Opie, J.: Radical connectionism: thinking with (not in) language. Language & Communication, No. 22, 313–329. (2002).
15. Hinton, G.E.: Mapping Part-Whole Hierarchies into Conntectionist Networks. *Artificial Intelligence*, Vol. 46, No. 1-2, 47-75. (1990).
16. Hawkins, J.: Learning Like A Human. IEEE Spectrum, 17-22. (April 2007).

17. Kharlamov, A., Raevsky. V.: Networks Constructed of Neuroid Elements Capable of Temporal Summation of Signals. In Rajapakse, J., Wang L. (Eds.): Neural Information Processing: Research and Development. Springer, 56-76. (2004).
18. Allen, J.: Natural Language Understanding, Second Edition. Redwood City, CA: The Benjamin/Cummings Publishing Company. (1994).
19. Man, W.C., Taboada M.: Intro to RST (Rhetorical Structure Theory). http://www.sfu.ca/rst/ (current February 2011).
20. Groza, T., Handschuh, S., Clark, T., Buckingham Shum, S., de Waard, A.: A short survey of discourse representation models. Proceedings 8th International Semantic Web Conference, Workshop on Semantic Web Applications in Scientific Discourse. Lecture Notes in Computer Science. Berlin: Springer Verlag. (2009).
21. Feldman, R., Aumann, Y., Finkelstein-Landau, M., Hurvitz, E., Regev, Y., Yaroshevich, A.: A Comparative Study of Information Extraction Strategies. *Lecture Notes in Computer Science*, Vol. 2276, 349-359. (2002).
22. Cunningham, H.: Information Extraction, Automatic. In K. Brown, (Ed.): Encyclopedia of Language and Linguistics, 2nd Edition. Elsevier. (2005).

Dr. Mladen Stanojević is a Senior Researcher at the Mihailo Pupin Institute. He received his MSc degree in the field of belief revision/truth maintenance and PhD in the area of natural language processing, both from the University of Belgrade. He has been working on applying techniques from artificial intelligence and natural language processing in real-world decision support systems, NLP based services and user interfaces, intelligent web applications, etc. His research interests include knowledge representation techniques, natural language processing techniques, semantic web, web services, service oriented architectures, etc. In these areas he published over 100 scientific papers. He is a member of IEEE.

Prof. Dr. SanjaVraneš is jointly appointed as a Scientific Director of the Mihajlo Pupin Institute and as a Professor of Computer Science at the University of Belgrade. From 1999 she has been engaged as a United Nations Expert for information technologies, and from 2005 as the expert evaluator and reviewer of EC Framework Programme Projects. Her research interests include semantic web, knowledge management, decision support systems, multicriteria analysis algorithms, complex event processing, etc. In these areas she published over 170 scientific papers. She serves as a reviewer of respectable international journals, like IEEE Transaction on Computer, IEEE Intelligent Systems magazine. She was a postdoctoral research fellow at the University of Bristol, England in 1993 and 1994. In the period 1999-2004 she was a Scientific Consultant at the ICS-UNIDO, International Center for Science and High Technology in Trieste, Italy. She has also served as a project leader and/or principal architect of more than 20 complex software projects. She is a member of IEEE, ACM and AAAI. She is also a member of Serbian Academy of Engineering Sciences and of National Scientific Council.

Received: September 1, 2010; Accepted: July 5, 2011.