BRAIN. Broad Research in Artificial Intelligence and Neuroscience,
ISSN 2067-3957, Volume 1, July 2010,
Special Issue on Complexity in Sciences and Artificial Intelligence,
Eds. B. Iantovics, D. Rădoiu, M. Măruşteri and M. Dehmer

# MAPPING BETWEEN SEMANTIC GRAPHS AND SENTENCES IN GRAMMAR INDUCTION SYSTEM

Laszlo Kovacs

Abstract. The proposed transformation module performs mapping between two different knowledge representation forms used in grammar induction systems. The kernel knowledge representation form is a special predicate centered conceptual graph called ECG. The ECG provides a semantic-based, language independent description of the environment. The other base representation form is some kind of language. The sentences of the language should meet the corresponding grammatical rules. The pilot project demonstrates the functionality of a translator module using this transformation engine between the ECG graph and the Hungarian language.

Keywords: *statistical learning, conceptual graph, word grammar, semantic anotation*

2000 *Mathematics Subject Classification*: 68T50 Natural language processing .

## 1. INTRODUCTION

The application of statistical learning methods is a very intensively investigated area of soft computing. The key question of this problem domain is how far the human intelligence can be modelled with statistical methods. The goal of the paper is to investigate a characteristic phenomena of human beings namely the usage of a natural language, the induction of grammar using statistical methods. The term of grammar induction denotes here a system that assigns description sentences to ontology models. Most of the research activities on this field relate to natural language processing (NLP). In NLP, a large family of methods is based on statistical algorithms [16]. Most of the related grammar induction methods use either free text without any annotation [18] or free text with grammatical annotation [19]. As the experiences show [14], the syntax, the free text alone is not enough to learn the grammar at acceptable high level. On the other hand, the grammatical annotation requires a large amount of preprocessing and restricts the training pool. Our approach is based on a model where the free text is annotated with a semantic, ontology description. The main benefit of this kind of semantic annotation over grammatical annotation is a greater efficiency of mapping semantics into syntax and it resembles more the human way of learning. In the literature, there is only few research works in this field, the interest of researchers for this topic raises only in the recent years [11]. The paper describes a system model how the semantic model can be mapped to sentences. Within this frame, an important stage is to find an appropriate formalism of the semantic model. According to our analyses, the traditional semantic modelling languages are not sufficient enough for reflect the process of conceptualization. Another important point is to determine the border between semantic and grammatical elements. The proposed system uses a two phase conversion model. In the first phase, the semantic graph is mapped into a word graph. This word graph has some common characteristics with the word dependency graph and it is an intermediate step between syntax and semantic. In the second phase, the word graph is converted into a sequence of words, where the conversion rules are related to a given grammar. The main benefit of the proposed model is to provide a flexible structure for mapping of semantic graph into sentence where the modular structure can provide a clear separation of syntactical and semantic elements.

## 2. EXTENDED CONCEPTUAL GRAPH

The goal of the investigation is to develop a conceptual modelling language that can be used to describe the semantics of an agent's internal conceptual model. The model language should support a formal specification that enables the mapping of semantics into a symbolic representation of the entire conceptualization process. The analysis of existing conceptual models shows that they all have some shortcomings from the aspects of our requirements. In order to provide a model language with rich set of specific features, a specific knowledge representation model was developed. It contains beside the usual modelling elements, like the specialization relationship, additional elements to enable a more efficient and powerful description of the conceptualization process.

The proposed extended conceptual graph (ECG) model contains three primitive-types: concept, relationship and container. Based on the behaviours, the following concept subtypes are defined:

1. According to their grade of identification
    - (N) Noname concept: is a primary concept that has no context-unique identification name.
    - (R) Permanent-named concept: is a concept having a context-unique name. A permanent concept is associated with an implicit definition that enables the identification of its instances in the environment.
    - (T) Temporary-named concept: is a concept occurring in some previous snapshot(s) of the history as a noname concept
2. Categories on a logical basis
    - (P) Predicate concept: is a concept that is used to denote predicates that are usually given by verbs in sentences. Predicate concepts are the kernels of the model fragments.
    - (C) Category concept: is the term covering all non-predicate concepts. Category concepts can denote various attributes for example. Each category concept defines a subset of instances that match this category concept.

3. According to the model level
    - (F) Primary concept: is a concept at the instance level. Primary concepts correspond to instances of the agent's environment.
    - (A) Abstract (derived) concept: a higher level concept in the agent's extended knowledge model. The derivation rule is defined with a sequence of snapshots.
4. Categories on cardinality
    - (I) Single instance concept: only one object is identified by this concept
    - (M) Group concept: several instances can belong to the concept. There are different types of group defined, like AND or OR groups.

Due to the semantic integrity constraints, only the following concept types are allowed: FICN, FICT, FICR, FMCR, FMPR, AMCR and AMPR. Regarding the relationship type, it has the following categorization:

1. According to the model level:
    - (F) Primary relationship: is a relationship that can be recognized by the agent. It is detected usually in the environment.
    - (A) Abstract (derived) relationship: is a relationship that is based on primary relationships. The derivation rule is defined with a sequence of snapshots.
2. According to the logical level
    - (I) Specialization relationship: is equivalent to the usual ISA relationship. It provides inheritance. A concept may have multiple parents.
    - (R) Role relationship: arbitrary attribute of a predicate concept
3. categories on cardinality
    - (S) Single instance relationship: only one object cluster is identified by this relationship

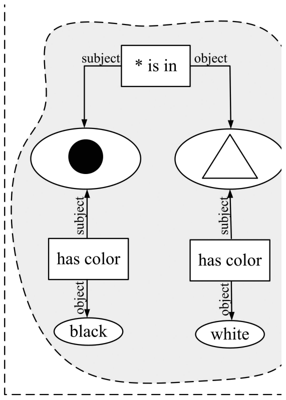


Figure 1: The ECG model

- (M) Class relationship: several instances can belong to the relationship.

The allowed relationships types are FMI, FSR, FMR, AMR. The group of container elements includes structure modules, like model fragment, model history.

Beside the mentioned semantic elements that refer to the state the environment to be observed, some other such elements can be included into the model that are related to the agent itself. These semantic elements describe internal state of the agent. The state of these internal attributes is also converted into the output sentence. The following parameters are used to describe the internal state:

- Mode (or indent): statement, open question, closed question, imperative,
- Timestamp: it denotes the time of the snapshot.

The following Figure 1 shows a sample semantic graph that describes a simple test world.

During the processing of the ECG, the base units of the graph are the ECG atoms. An ECG atom corresponds to a primitive statements related to one predicate. It has a structure of one-level deep tree, where the root of the tree is the predicate and the concepts linked to it are the leaves. The child concept of the root predicate may be not only a single concept but it can be another ECG atom. Thus, the ECG atoms can be linked into a hierarchy of ECG atoms. Our model is based on the assumption that at words at a sublevel do not contain words from the upper level. The whole sublevel is considered as an atom at the parent level. This assumption improves significantly the modularity and the efficiency of the grammar system.

## 3. STRUCTURE OF THE GRAMMAR SYSTEM

The purpose of the semantic graph is to describe the meaning of the statements about the problem domain. The grammar covers the rules how to convert the meaning into sentences. The sentence is defined as sequence of words. It is assumed here that the meaning of the statement is encoded into the following grammatical elements:

- set of the stem words;
- inflection of the words (the suffixes and prefixes alter the meaning of the stem);
- order of the words (there may be a fixed order between the words belonging to different semantic or grammatical units);
- decomposition rules of grammatical units (the chain of derivation rules assigns a sequence of words to a semantic unit).

The resulted sentence reflects the mixture of the applied grammatical encoding rules, so it is important to have a clear layering of the rules in order to have an efficient separation of the different grammatical units. One of the main difficulties of the grammar system is that very different semantic elements may affect the same syntactical unit. For example, the inflection rule of a verb in the Hungarian language may depend on the subject, on the object, on the timestamp and on the mode. Another difficulty is the fact, that a semantic unit may be mapped not only to a single word but into a word set. For example, some verb have prefix that are in some situation separated from the stem and in other situations it is merged with the stem.

The proposed grammar system should contain elements to decode the required conversion rules. The main components of the grammar system are:

- a set of stem words that are assigned to abstract concepts with a specialization graph (like: Peter, read, book, evening,..). The concepts may be either semantic concepts or grammatical concepts (car or subject).

- decomposition rules that describe the components of an abstract concepts. The components may be abstract or atomic, word level concepts. (sentence: subject, predicate, object,..).
- set of inflection rules where different grammatical units may have different transformation rules (accusative: Rule1, dative: Rule2). A rule is represented here as a string transformation function, where the basic string operations are the followings:
  - concatenation with a given suffix
  - substitution with a given infix where both arguments may be an empty string
  - concatenation with a given prefix.
  - a set of ordering rules where the nodes refer to grammatical units and the edges are directed and labelled. The label denotes the type of precedence rule.

In this investigation, the grammar rules are known and defined. The grammar is embedded in a module, so a semantic graph can be pressed with different grammar modules.

In the proposed system, the ECG model of the agent is converted into a sentence of the symbolic language. The base model of the conversion includes the following steps:

1. ECG network is converted into a rooted tree of concepts, where the root node is the primary predicate in focus. This transformation generates a hierarchy of ECG atoms.
2. From the ECG hierarchy a word-tree is generated where the words at a node corresponds to the set of words assigned to the concept at that node.
3. The word symbols of the nodes are transformed into inflected forms that reflect the syntactical and semantic roles of the nodes.
4. For the pool of generated words the ordering is determined, resulting in a sentence. The ordering of the words is generated in a top down manner.

The generated sentence reflects both the semantic and the grammar syntax. The translation module contains several sub-modules to solve the different transformation steps. In the next chapter, a special module, the word graph manager is analyzed in detail.

## 4. WORD GRAPH AND DEPENDENCY GRAMMAR

Grammars, and theories of grammar, can be classified according to whether the basic unit of sentence structure is the phrase, or the dependency between two words. A dependency-based linguistic approach to the description and analysis of natural language syntax is constituted by distinguishing a head-dependent asymmetry, and describing the relations between a head and its dependents in terms of semantically motivated dependency relations. Dependency grammar (DG) is a class of syntactic theories developed by the French linguist Lucien Tesnire [1]. His model is based on the stemma, a graphical representation of the grammatical dependencies between the words in a syntactic construction. In the sentence, the verb is seen as the highest level word, governing a set of complements, which govern their own complements themselves. Tesnire has had a major influence on linguistic theories that place more emphasis on semantics than on syntax (see [2] for a review). Klein and Simmons [3] adopted dependency grammar for a machine translation system. Valency theory [4] and Meaning-Text Theory [5], [6] are two ongoing developments of the dependency approach. Schank adopted the dependency approach in his Conceptual Dependency Graph, but he shifted the emphasis to concepts rather than words [7]. For a recent survey on dependency models see [8]. Dependency theories have also been strongly influenced by Case Grammar [9] which provides a convenient set of labels for the arcs of the graphs. Extensible Dependency Grammar [10] and Word Grammar [11] are two general frameworks for dependency grammar which aim at modelling not only the syntactic but also the semantic and phonological levels of linguistic representations.

The main reason why we turn our attention towards dependency structures is that in a phrase structure tree, discontinuous constructions can not be represented. This restriction poses problems for the analysis of word order variation, even for rigid word order languages. On the other hand, dependency grammars are not defined by a specific word order, and are thus well suited to languages with freer word order, such as Hungarian, as well as Scandinavian and Slavic languages. The modelling of other European languages where discontinuous constructions are frequent, such as German, French and Dutch can also benefit from a dependency-based approach.

Dependencies are widely accepted in theories of semantic structure. In fact, one of the main attractions of traditional DG is its close correspondence to meaning:

- syntactic dependencies are meaningful, i.e. they usually carry semantic relations; and
- syntactic dependencies are more abstract than surface order.

Nevertheless syntactic dependencies are distinct from semantic dependencies. The usual problem is how to map syntactic dependency structure to a semantic one. A distinguishing feature of our project is the aim of finding a mapping from semantic (conceptual) dependency structure to a syntactic one, where consequently the elements among which dependencies are examined are not words but concepts and syntactic units, respectively. Hence, our model got the name Conceptual Dependency Grammar (CDG)

The word graph is used to describe the words and the dependency between the words. A word $w_1$ depends on $w_2$, if $w_1$ can occur in the sentence only if $w2$ also occurs. The dependency is denoted with

$$w_2 \Rightarrow w_1.$$

Taking a sample sentence 'The cat is sitting in a chair.' the dependency graph has the following elements:

- is, sitting ⇒the, cat, in, a. chair
- cat ⇒ the
- chair ⇒ in, a

For the detection of dependency relationships, the sentence reduction method is applied. This means, that some word or words are omitted from the sentence. The meaning of the altered sentence is evaluated by a teacher. The new sentence may be judged as

- C: correct, but it has a reduced meaning compared with the original sentence, or as

- U: grammatically not correct, but understandable with a modified, reduced meaning and or as
- N: the sentence is not correct and not understandable.

Some test sentences with their evaluations are given here for demonstrations purposes:

- The cat is sitting in a : N
- The is sitting in a chair :N
- The cat is sitting : C
- Cat is sitting chair : U

The algorithm proposes a dependency relation between such words where the reduced sentences have different evaluation levels. The

$$w_2 \Rightarrow w_1$$

dependency is valid if

$$w_2 \in s, w_1 \notin s, l(s) = l_1$$

and

$$w_1 \in s, w_2 \notin s, l(s) = l_2 \Rightarrow l_1 > l_2$$

where the ordering of levels is given as

$$C > U > N.$$

## 5. STRUCTURE OF THE TRANSLATOR MODULE

The translator module described in the previous chapters can be used not only to transform the semantic graph into a sentence but it can work in the reverse way too. Thus a sentence can be mapped to semantic graph that denotes the meaning of the sentence. The main steps of the mapping function can be summarized as follows:

- Parsing the input sentence into words
- Performing a morpheme analysis on the words

- Determining the stems and the inflection classes for each words
- Matching the annotated words to the nodes of the word graph
- Checking the ordering constraints given in the graph
- Calculating the similarity value between the sentence and the word graph
- Selection of the winner graph
- Returning the corresponding conceptual graph as meaning descriptor of the sentence

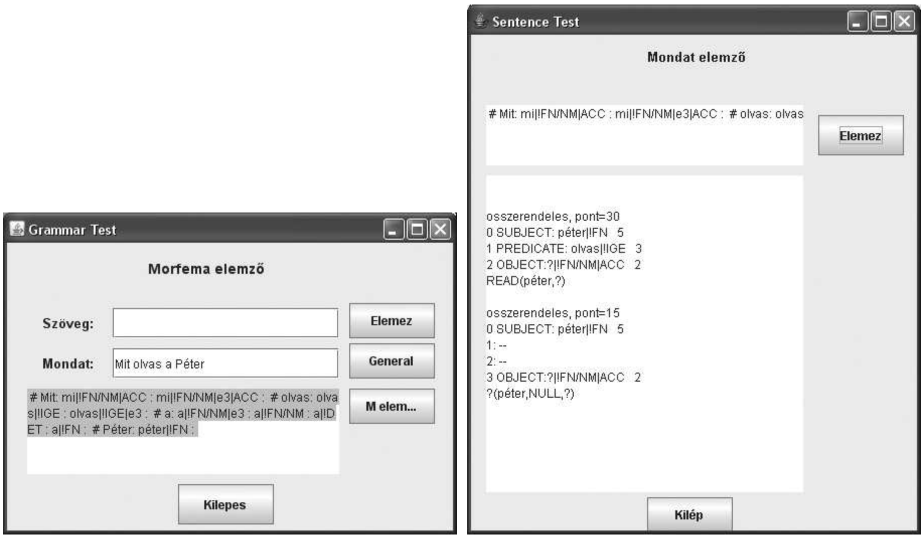
In this processing a new key element is the morpheme analyzer that contains a grammar specific engine to determine the stems and inflection parts. In the frame of the project a Morpheme Analyzer Module for the Hungarian Language was developed. For a given sentence, the analyzer returns the list of possible morpheme spectrums of the words. With coupling of two directions a translator module can be developed. The grammars of the input and output channels may be different. Thus, a sentence in the first grammar is translated into a sentence of the second grammar.

In the project, the goal was to develop a natural language interface for a service provider. The server module has an application programming interface, a set of functions that should be invoked to start the required service. The front end of the pilot system uses the Hungarian language and the output contains the predicates symbolizing the function calls.

The engine selects the best fitting conceptual graph for the incoming sentence and returns the predicate representation of the selected conceptual graph. The Fig 1 shows the input form with the input sentence (Mit olvas Peter?, What is reading Peter?) and the output form with the generated formula (READ(Peter,?))

## 6. Conclusions

The goal of the project was to develop a translation system between two knowledge representation forms. One of the representation forms is a new kind of abstract predicate oriented conceptual graphs called ECG. The second representation form is a set of sentences of a given language. While the ECG is a general, language independent tool, the sentences belong to a language and a specific grammar.



Szöveg:

Mondat: Mit olvas a Péter

```
# Mit: mi|!FN/NM|ACC : mi|!FN/NM|e3|ACC : # olvas: olva
s|!IGE : olvas|!IGE|e3 : # a: a|!FN/NM|e3 : a|!FN/NM : a|!D
ET : a|!FN : # Péter: péter|!FN :
```

```
# Mit: mi|!FN/NM|ACC : mi|!FN/NM|e3|ACC : # olvas: olvas
```

```
osszerendeles, pont=30
0 SUBJECT: péter|!FN   5
1 PREDICATE: olvas|!IGE   3
2 OBJECT:?|!FN/NM|ACC   2
READ(péter,?)

osszerendeles, pont=15
0 SUBJECT: péter|!FN   5
1: --
2: --
3 OBJECT:?|!FN/NM|ACC   2
?(péter,NULL,?)
```

Figure 2: The intput and output forms of the transformation engine

## References

[1] L. Tesnire, lments de syntaxe structurale, Paris: Klincksieck, 1959.

[2] J. F. Sowa, 'Semantic networks,' in Encyclopedia of Artificial Intelligence, S. C. Shapiro Ed., 2nd ed., Wiley, 1992.

[3] S. Klein and R. F. Simmons, 'Syntactic dependence and the computer generation of coherent discourse,' Mechanical Translation 7, 1963.

[4] D. J. Allerton, Valency and the English Verb, New York: Academic Press, 1982.

[5] I. A. Mel'cuk, 'Towards a linguistic "Meaning Text' model,' in Trends in Soviet Theoretical Linguistics, F. Kiefer Ed., Dordrecht: Reidel, pp. 35-57, 1973.

[6] J. Steele Ed., Meaning-Text Theory, Ottawa: University of Ottawa Press, 1990.

[7] R. Schank Ed., Conceptual Information Processing, Amsterdam: North-Holland Publishing Co., 1975.

[8] R. Hudson, 'Recent developments in dependency theory,' in Syntax. Ein internationales Handbuch zeitgenssischer Forschung, J. Jacobs, A. v. Stechow, W. Sternefeld and T. Vennemann Eds., pp. 329-338, Berlin: Walter de Gruyter, 1993.

[9] C. J. Fillmore, 'The case for case,' in Universals in Linguistic Theory, E. Bach and R. T. Harms Eds., New York: Holt, Rinehart and Winston, pp. 1-88, 1968.

[10] R. Debusmann, 'Extensible Dependency Grammar: A modular grammar formalism based on multigraph description,' PhD thesis, 2006.

[11] R. Hudson, Language Networks: The new Word Grammar, Oxford University Press, 2007.

[12] D. Jurafsky and J. H. Martin, Speech and Language Processing: An Introduction to NLP, Computational Linguistics, and Speech Recognition, Prentice Hall, 2nd ed., 2007.

[13] M. Ross Quillian, 'Word concepts: a theory and simulation of some basic semantic capabilities", in Behavioral Science, vol. 12, pp. 410-430, 1967.

[14] G. B. Varile, A. Zampolli, R. A. Cole, J. Mariani, H. Uszkoreit, A. Zaenen and V. Zue, Survey of the State of the Art in Human Language Technology, Cambridge University Press, 1997.

[15] E. Charniak, Statistical Language Learning, MIT Press, Cambridge, MA, 1996.

[16] C.D. Manning and H. Schtze, Foundations of Statistical Natural Language Processing, MIT Press, Cambridge, MA, 1999.

[17] A. Clark, Unsupervised Language Acquisition: Theory and Practice, PhD Dissertation, COGS, University of Sussex, 2001.

[18] A. Roberts and E. Atwell, Unsupervised Grammar Inference Systems for Natural Language, Research Report 2002.20, University of Leeds, School of Computing, 2002.

[19] A. McEnery, R. Xiao and Y. Tono, Corpus-Based Language Studies: An Advanced Resource Book, in ser. Routledge Applied Linguistics, Routledge, 2005.

Laszlo Kovacs
Department of Information Technology
University of Miskolc, Hungary
H-3515 Miskolc-Egyetemvaros
e-mail:*kovacs@iit.uni-miskolc.hu*