A Semantic Schema - based Approach for Natural Language Translation

MIHAELA COLHON[(1)] , NICOLAE ȚĂNDĂREANU[(2)]
Department of Computer Science
University of Craiova
street A. I. Cuza, no. 13, Craiova, 200585
ROMANIA
[(1)]mcolhon@inf.ucv.ro http://inf.ucv.ro/~ghindeanu
[(2)]ntand@rdslink.ro http://inf.ucv.ro/~ntand

Abstract: - Processing natural language statements to obtain equivalent translations in a different language has long been an area of research in Artificial Intelligence. In machine translation systems, an intermediate representation of inputs is necessary to express the result of the phrase analysis. These representations treat each phrase as a character string and construct the corresponding syntactic and/or semantic representation structure.
In the proposed approach, representation is made by means of a semantic network type structure, named semantic schema, with focus on the dependency relations existing between the sentence words. The resulted schema components are further evaluated (using an interpretation system) with the corresponding constructions from the language into which translation occurs.

Key-Words: - Semantic Schema, Natural Language, Machine Translation, Morpho-syntactic data, Dependency Relationships

1 Introduction

Since 1960, much of research on Natural Language Processing (NLP) was motivated by its potential use in communication with software products. Natural language systems have been developed to extract information from databases, to control robots, to interact with graphic systems, in generally, to interact in a human friendly way with systems specialized in some task or problem area.

Natural language provides a human-machine interaction method which presents several advantages, some of them reling on the immediate vocabulary available with a minimum training or on shielding the user from the formal acces language of the underlying system. So far, the proposed technologies can be grouped into four categories ([9]): machine-readable dictionary-based approaches, knowledge-based mechanisms, semantic network-based stuctures and corpus-based statistical approaches. The last two are the main steams today, and most semantic network-based methods adopt Princeton University's WordNet semantic dictionary.

In this paper, *translation* is considered as the task of transforming an existing text written in a *source language* (SL), into an equivalent text in a different language named the *target language* (TL) ([3]).

Translators vary greatly with respect to how they produce translations. The main problem is to find correctly the corresponding translation words. Obviously, a bilingual lexicon acts as a bridge between the words of SL and the equivalent ones from the considered TL. Another approach to this issue is given by lexicons with WordNet structures.

WordNet ([19]) is a lexical database that acts like an ontology and has a semantic network representation. English nouns, verbs, adjectives and adverbs are organized into synonym sets called *synsets* that are inter-connected with different relation links.

In the last years, the development of WordNets for languages other than English has been encouraged. EuroWordNet ([6]) is a multilingual database with wordnets for several West European languages. Part of the second module of EuroWordNet, the BalkaNet project extends the wordnet approach to the less studied East European languages. All the resulted wordnets are aligned to the Princeton Wordnet, according to the principles established by EuroWordNet ([17]).

The main reason for developing these ontologies is the desire and the necessity to create an *uniform ontological infrastructure across languages* that will simplify translations. By storing the wordnets in

a central lexical database system, a large multilingual database was created, where the English synsets from WordNet 1.5 function as an Inter-Lingual Index (ILI).

More precisely, ILI is a set of pivot nodes that allows the linkage between concepts belonging to different wordnets. The advantages of an interlingua mechanism as ILI are well known in Machine Translation (MT). In this approach it is possible to go from one synset in a wordnet to a synset in another wordnet that is linked by the same WordNet 1.5 concept. In principle, multilinguality is achieved by adding an equivalence relation for each synset in a language to the closest synset in WordNet 1.5.

An ideal translation system has a completely modular architecture with no influence of one monolingual component – modularity ensures that the system can be easily extended to new languages and language pairs ([18]).

Typically, fine-grained morphological analysis is a prerequisite for good translation results. Data driven approaches to syntactic analysis (parsing) are a very active area of the current research. The traditional problem of morphological analysis for a given word form, is to predict the set of all its possible morphological analyses. A morphological analysis has to determine the part-of-speech tag (POS), possibly other morphological features, and the lemma (basic form) corresponding to this tag and features combination.

Thus, translators usually use an intermediate mapping between the source and target language structures defined by means of the syntactic properties of the text being translated. By means of such structures, the results of the text analysis can be explored in order to identify the connections between the text's words, and furthermore to obtain the relations between the syntactic phrases of each sentence ([8]).

1.1 The proposed model

The majority of the existing syntactic representations for natural language constructions is based on *context-free grammars* rules that are usually mapped on tree type-structures. The developed symbolic approaches describes the meaning of represented constructions and the procedural knowledge used to process the underlying knowledge ([14]).

The translation mechanism we propose makes use of the syntactic information represented in semantic network structure, named *semantic schema* and then evaluates the resulted representations with the equivalent words from the target language. Evaluation is defined in order to exploit the Inter-Lingual Index mechanism of wordnet lexicons. The contextual approach is ensured by the fact that evaluation is constructed for each syntactical phrase based on its constituent components.

Resuming, the involved translation processes can be grouped into four categories:

1. Morphological analysis of the natural language input.
2. Syntactic representations by means of a semantic schema's components.
3. Identification of the dependency relations among the units of the natural language input.
4. Defining the interpretation system for the resulted semantic schema relations with equivalent constructions from TL.

The resulted semantic schema-based representation mechanism differs from other syntactic representations in that it abstracts away from language-particular properties of the sentence structure and represents the basic syntactic constituencies, annotated with the specific morpho-syntactic features. For this reason, our representation mechanism is not specific to any particular language, i.e. it is a *language-neutral representation*. In order to become a real one, we have to endow it with a specialized lemmatizer for the input language and with lexicons for both input and output languages.

In order to exemplify the translation mechanism, in this paper we will consider English as the source language and Romanian as the target language.

2 Representation by means of Semantic Schema

A semantic schema is an abstract structure that extend the concept of the semantic network and is formalized by means of a tuple of symbolic entities. In order to obtain a real representation, the entities of the tuple must be interpreted. An interpretation for a semantic schema defines the domain of its components as it happens in mathematical logic, where an interpretation establishes a logic value for some formula. If S is a semantic schema and I an interpretation for S then the pair (S, I) defines an environment for the reasoning process.

A semantic schema comprises two aspects ([16]):

- **formal aspect** by which some computations in a Peano σ-algebra are obtained; this aspect deals with the syntactic representations of the

semantic schema. The computations are based on the concept of *derivation* and the set of results is denoted by $F_{comp}(S)$. A *sort* is an element of A where A is the set of the relation symbols corresponding to the semantic schema. Based on this concept, the set $F_{comp}(S)$ can be divided into equivalence classes. An equivalence class includes all the elements with the same *sort*.

- an **evaluation aspect** described in the context of an interpretation by means of which the abstract entities defined in the previous step get values in a semantic space named *output space*. For every sort $u \in A$, each entity of an equivalence class $[u]_F$ is transformed to obtain its semantics. Using such a transformation, the set Y_u is constructed, each object of Y_u having the class u. The output space, noted with Y, becomes the union of the resulted *classes of objects*, $Y = \bigcup_{u \in A} Y_u$.

We propose a translation mechanism that uses original internal representations by means of which the structure of the source sentence is mapped on the semantic schema components. The resulted representations are then evaluated with equivalent constructions from the target language and in this way the translation is performed.

This translation method presents several advantages, such as:

- it is not necessary to specify many-to-many equivalence relations between the source and the target languages pair; the constructions of each source or target language only consider the equivalence relations to the semantic schema representations
- it is possible to develop the semantic schema representations as the central resources for a MT by means of which the matching between the source language constructions with the target language equivalents can be done more efficient and precise.
- by defining different interpretations for the same semantic schema structure, constructions in different target languages can be generated

2.1 Syntactic aspects of Semantic Schemas

Consider θ a symbol of arity 2 and a non-empty set A_0.

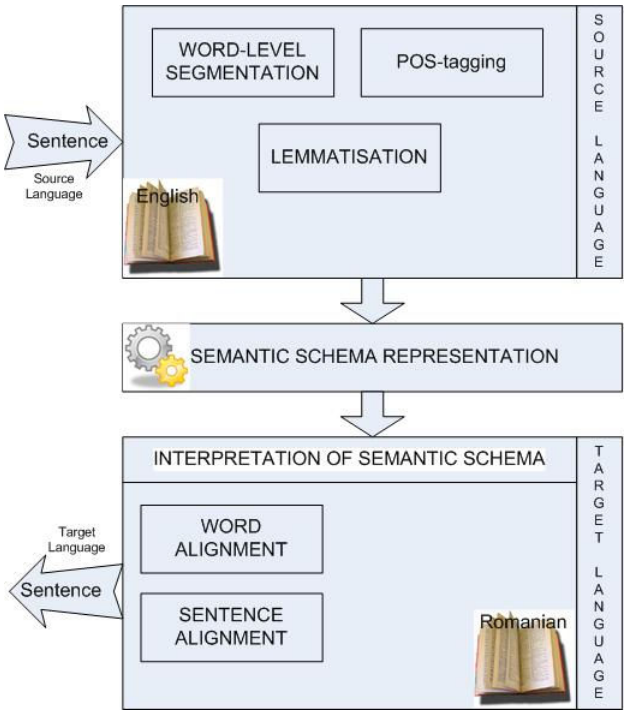

Figure 1 Translation System based on Semantic Schema

Starting from A_0 we construct the set $\overline{A_0}$ as the Peano θ-algebra over A_0. We have $\overline{A_0} = \bigcup_{n \geq 0} A_n$, where A_n are defined recursively as follows:

$$A_{n+1} = A_n \cup \{\theta(u,v) \mid u, v \in A_n\}$$

A semantic θ-schema is a system $S = (X, A_0, A, R)$ where ([16]):

- X is a non-empty set of *objects symbols*
- A_0 is a finite non-empty set of elements called *label symbols*
- $A_0 \subseteq A \subseteq \overline{A_0}$, where $\overline{A_0}$ is the Peano θ-algebra generated by A_0
- $R \subseteq X \times A \times X$ is a non-empty set of relations such that the tuples satisfy the following conditions:

(R1):
$(x, \theta(u,v), y) \in R \Rightarrow \exists z \in X : (x,u,z) \in R, (z,v,y) \in R$

(R2):
$\theta(u,v) \in A, (x,u,z) \in R, (z,v,y) \in R \Rightarrow (x, \theta(u,v), y) \in R$

(R3):
$pr_2 R = A$

The elements of $A \setminus A_0$ denote the compound relations constructed in the semantic schema by fulfilling the conditions (R1)÷(R3). Accordingly to the semantic schema definition, every relation $r \in R \setminus R_0$ can be broken in two relations $r_1, r_2 \in R$ that fulfill the *composition condition* of binary

relation: the final node of the first relation r_1 is the initial node of r_2 (results from the condition R1). Conversely, if two relations $r_1, r_2 \in R$, for $r_1 = (x, u, z)$ and $r_2 = (z, v, y)$ fulfill the composition condition, then there must be $\theta(u, v) \in A$ in order to have $(x, \theta(u, v), y) \in R$ (condition R2). The last condition R3 ensures that all the symbolic names of A are used to label the relations of R.

2.1 Evaluation aspects of Semantic Schemas

As we have already specified, by means of an appropriate interpretation, the abstract entities of a semantic schema receive values in a semantic space, named *output space*. We define the interpretation of a semantic schema as a system endowed with a set of algorithms, which organizes the output space as a set of layers hierarchically organized.

In essence, by means of an interpretation, objects of the output space are attached to the nodes of a semantic schema and based on these objects some classes of *complex objects* can be computed. The classes of the output space are defined recursively as follows:

- The object $o = A\lg_a(ob(x), ob(y))$ is a *complex object of class a* and we note this property by $cls(o) = a$ for $a \in A_0$ and $(x, a, y) \in R$.
- If $cls(o_1) = u, cls(o_2) = v$ and $\theta(u, v) \in A$ then $o = A\lg_{\theta(u,v)}(o_1, o_2)$ is a complex object of class $\theta(u, v)$ and $cls(o) = \theta(u, v)$.

We have that the complex objects are defined only by means of the set of algorithms $\{A\lg_u\}, u \in A$.

Using these notations, the interpretation of a semantic schema $S = (X, A_0, A, R)$ is a system

$$I = (Ob, ob, Y, \{A\lg_u\}_{u \in A})$$

where ([15]):

- Ob is a finite set of *objects* used to interpret the nodes of the schema
- $ob : X \to Ob$ is the function that maps the abstract symbols of the semantic schema nodes to the objects of the output space
- Y is a nonempty set of elements which are called the output elements

$$Y = \bigcup_{u \in A} Y_u$$

where

$$Y_a = \{A\lg_a(ob(x), ob(y)) \mid (x, a, y) \in R\}, a \in A_0$$

$$Y_{\theta(u,v)} = \{A\lg_{\theta(u,v)}(o_1, o_2) \mid o_1 \in Y_u, o_2 \in Y_v\}, \theta(u, v) \in A \setminus A_0$$

The output elements of the set Y divide the output space of a semantic schema into layers.

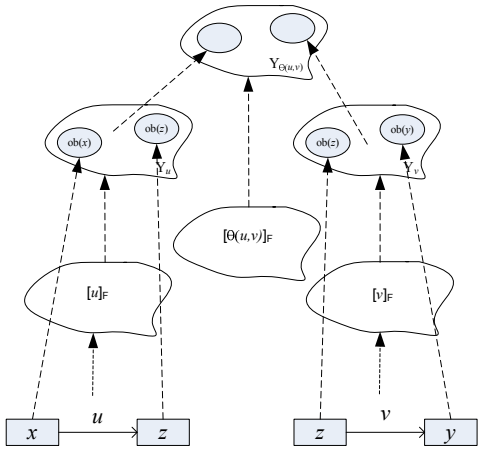

Figure 2 Formal computations in Semantic Schema

A layer is a set Y_u for some $u \in A$. We observe that each element of Y_u has the class u.

Resuming, the interpretation algorithms organize the output space as a set of layers, each layer containing objects of the same class.

3 Morpho-syntactic data

The fundamental question in Machine Translation (MT) system design is the form in which information about the source text is passed to generation. Such information must include anything relevant for translation, expressed in a form that can guarantee correct translation.

The lexical alignment of the input sentence can be done on the results provided by the segmentation, Part Of Speech tagging (shortly, POS tagging) and lemmatisation steps. Also, additional phase of meta-category annotation, more precisely, the morpho-syntactic annotations must be used in order to allow the inter-category alignment.

POS Tagging, also called grammatical tagging, is the process of marking up the words of a text as corresponding to a particular part of speech, based on both its definition, as well as its context – i.e. relationship with adjacent and related words in a given phrase or paragraph ([5]).There are two types of taggers: the first one attaches syntactic roles to each word (e.g. *subject*, *object*) and the second one attaches only functional roles (such as *noun* or *verb*).

Base forms, also known as lemmas, base forms or ground form of the words do not contain any morphological derivation of the word (such as gender, number, tense, and so on) but are crucial in order to get the corresponding target word from the dictionary entries.

Remark. *In a specific language, a word form is uniquely identified by its lemma and the corresponding morpho-syntactic information*.

Tabel 1. Relevant POS Morphological Attributes

Part-Of-Speech (POS)	**POS morpho-attributes**
Verb	*mood, time, person, number, gender*
Noun	*number, gender, type*
	type: common, proper
Adjective	*number, gender, degree*
	degree: positive, comparative, superlative
Pronoun	*type, gender, number, case*
	type: personal, possesive, interrogative demonstrative, indefinite, relative,
Determiner	*number, gender, type*
	type: article, possesive, demonstrative, interrogative, numeral quantifier
Preposition	*type*
	type: place, time, mode
Adverb	*type*
	type: place, time, manner
Numeral	*type, number*
	type: cardinal, ordinal
Conjunction	*type*
	type: coordinating, subordinating

At the heart of every sentence structure are the *relations* among words, no matter if by these relations we understand the possible grammatical functions or the links which bind words into larger syntactical units such as, *noun phrases* or *verb phrases*. Usually, these relations are formalised by means of *dependency grammar* rules where each word is considered to be *depended* on another word which links it to the rest of the sentence ([7]).

A variety of dependency relations may exist among the words of a sentence. Following this assumption, we intend to use semantic schema representations in order to store syntactic (dependency) relations identified between the source sentence's words. More precisely, in the proposed representation, the words are identified by their lemmas while the relations between words correspond to the POS tagging information which can specify the role played by a word in connection to the word that follows it in the sentence.

In 1970, Robinson formulated four axioms to govern the well-formedness of dependency representation structures, depicted below:

1. one and only one element is independent
2. all others depend directly on some element
3. single headedness and uniqueness: no element depends directly on more than one other
4. projection requirement: if *A* depends directly on *B*, and some *C* element intervenes between them (in the linear order of the sentence string), then *C* depends directly on *A* or *B* or some other intervening element.

Starting from the *dependency approach to syntactic analysis* given in [7], we propose the following representation method. Let us consider that a sentence is a sequence of words noted as:

$$Sen = (w_1, ..., w_n)$$

We shall denote by w_{n+1} a special word EOS which indicates the end of the sentence.

For each word w_i from the sentence *Sen*, an unique number, named *word id*, will be use to indentify it. This *word id* is actually, the position of the word within the sentence. Thus, the first word is identified by id_1, the second word by id_2, etc. In order to preserve the notation the EOS word is identified with a *word id* equal to n+1, where by n we note the number of the sentence's words.

In order to have an unique word identification in the resulted semantic schema representations, each *word id* will be annotated with the following attributes:

- *lemma*: the corresponding word's lemma (canonical or dictionary form)
- *ana*: the morpho-syntactic information (a string containing information about inflectional class, derivation, gender, number, gradation)
- *POS tagging*: Part-Of-Speech Marker. Accordingly to Multext-East language specific features these POS markers are: N(Noun), V(Verb), A(Adjective), P(Pronoun), D(Determiner), T(Article), R(Adverb), S(Adposition), C(Conjunction), M(Numeral), I(Interjection), X(Residual), Y(Abbreviation), Q(Particle).

In what follows we will define the structure of a natural language sentence by means of the *dependency relations* ([7]) existing between the sentence's words.

Definition. *The **dependency relation** corresponding to each word w of a sentence is described as a tuple of the form*:

$$(w, POS, w')$$

such that w is called the dependent word in the relation noted by POS, POS is the POS tagging

information corresponding to w and w' represents for w the word it depends on.

At this point we can introduce the syntactical representation mechanism of a natural language sentence by means of semantic schema.
Let us consider the sentence *Sen* =(w_1, ..., w_n). The syntactical structure of *Sen* by means of a semantic θ - schema is given by the system $S = (X, A_0, A, R)$ such that:
- $X = \{id_1,...,id_n, id_{n+1}\}$ where id_i represents the *word id* of w_i, for $i = \overline{1,n}$ and id_{n+1} correspond to the EOS word.
- A_0 is a set of POS tags identifiers that represent the parts of speech (*noun*, *verb*, *adjective*, etc.) to which the words identified by the elements of X belong
- $A_0 \subseteq A \subseteq \overline{A_0}$, where the labels of $A \setminus A_0$ are compound tags designating syntactical phrases components of the sentence
- $R \subseteq X \times A \times X$ is a set of tuples, each tuple representing a dependency relation govern by the POS tag identifier as follows: if $(id_i, u, id_j) \in R$ means that the word w_i identified by id_i belongs to the part-of-speech class denoted by u and thus w_i is in a u-relation with the word identified by id_j.

A *grammar* of a language is a set of rules which says how these parts of speech can be put together to make grammatical or 'well-formed' sentences ([2]). The compound relations, and according to them, the compound labels from $A \setminus A_0$ are determined by means of a grammar's rules in which the represented sentence is valid.

In order to exemplify all these representations, let us take the sentence:

Sen_1 = (*the*, *beautiful*, *gril*, *sings*, *few*, *songs*)

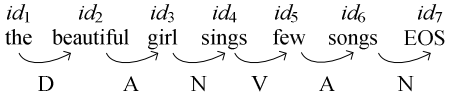

Figure 3 The dependencies between sentence words

We consider EBNF grammar specifications for describing the grammar rules in which the considered sentence structure is valid.

```
<sentence> ::= <noun_phrase><verb_phrase>|
   <noun_phrase><verb_phrase><noun_phrase>
<noun_phrase> ::= <det><adj><noun>|
   <adj><noun>
<verb_phrase> ::= <verb><noun_phrase>
<det> ::= a| the
<adj> ::= beautiful| few
```

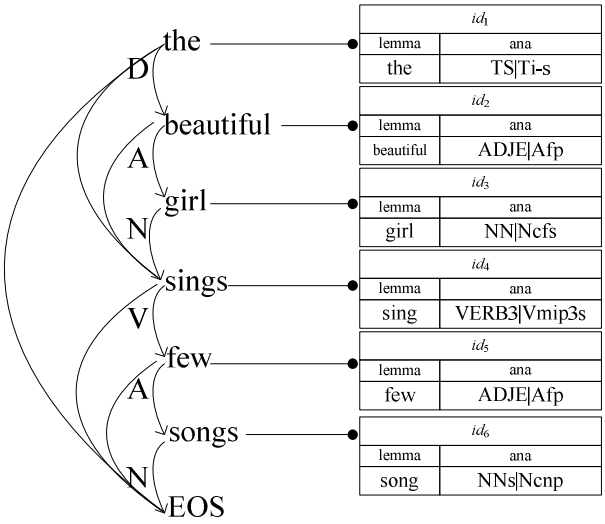

Figure 4 The morphosyntactic information of the sentence words

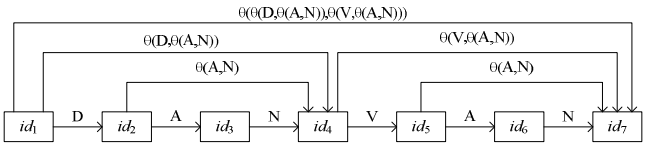

Figure 5 The semantic schema dependency structure

```
<noun> ::= girl| songs
<verb> ::= sings
```

The morphosyntactic information for the words of the sentence Sen_1 are illustrated in Fig. 2 using of Multext-East language specifications ([12]).

The dependency relations of the considered sentence are illustrated in Fig. 3 and represented in the semantic schema $S_1 = (X, A_0, A, R)$ where:
- $X = \{id_1, \ldots, id_6, id_7\}$
- $A_0 = \{D, A, N, V\}$

such that D denotes the *determiner* relation, A stands for the *adjective* relation, N for the *noun* relation and V labels a *verb* relation.
- $A = A_0 \cup \{\theta(A,N), \theta(D,\theta(A,N)), \theta(V,\theta(A,N)),$
$\theta(\theta(D,\theta(A,N)),\theta(V,\theta(A,N)))\}$

The label $\theta(D,\theta(A,N))$ designates the noun phrase corresponding to the rule:

```
<noun_phrase> ::= <det><adj><noun>
```

while $\theta(V,\theta(A,N))$ designates the verb phrase:

```
<verb_phrase> ::= <verb><noun_phrase>
<noun_phrase> ::= <adj><noun>
```

$R = \{(id_1, D, id_2), (id_2, A, id_3), (id_3, N, id_4), (id_4, V, id_5),$
$(id_5, A, id_6), (id_6, N, id_7), (id_2, \theta(A,N), id_4),$
$(id_5, \theta(A,N), id_7), (id_1, \theta(D,\theta(A,N)), id_4),$

$(id_4, \theta(V, \theta(A, N)), id_7),$
$(id_1, \theta(\theta(D, \theta(A, N)), \theta(V, \theta(A, N))), id_7)\}$

Definition. *Let us take the semantic schema* $S = (X, A_0, A, R)$ *as the dependency structure relative to the sentence Sen*=(w_1, ..., w_n). *We consider S as a correct* ***dependency structure*** *if it fulfils the following restrictions*:

1. restrictions about unique relationship between each pair of consecutive words (w_i, w_{i+1})
   a. if $(id_i, u, id_j) \in R$ then we can not have the inverse relation $(id_j, u, id_i) \in R$
   b. if $(id_i, u, id_j) \in R$ and $(id'_i, u', id'_j) \in R$ then if $id_i = id'_i$ and $id_j = id'_j$ then $u = u'$
2. Robinson's axioms restrictions
   a. (acyclicity) $\forall id_1, \ldots, id_{k-1}, id_k \in X$: (id_1, u_1, id_2), ..., $(id_{k-1}, u_{k-1}, id_k) \in R$ then $id_1 \neq id_k$
   b. (rootedness) $\exists! id_i \in X : \forall id_j \in X$, $i \neq j$ then there is no $u \in A$ such that $(id_i, u, id_j) \in R$
   c. (single headedness) $\forall id_i, id_j, id_k \in X$: (id_k, u_i, id_i), $(id_k, u_j, id_j) \in R$ then $id_i = id_j$ and $u_i = u_j$ (results also from 1.b.)

The acyclicity restriction results from the asymmetrical property of the dependency relations. Asymmetry guarantees also that the dependency relations are not reflexive (we can not have $(id_k, u, id_k) \in R, \forall u \in A$).

The Robinson's projection requirement is not reflected by any of these two restrictions for the semantic schema structure because, as we will show in the next section, each word depends directly on the following word in the sentence so there is no "intervening word" case in the schema representations.

3.1 Morpho-syntactic data mapped on Semantic Schema.

Morphology is the study of the internal structure of words, the way words are built up from smaller meaning units. Such analysis is the basis for many NLP applications, including syntax parsing and also machine translation.

We assign to each node an *agreement tuple*. Each such tuple contains the lemma of the represented word together with syntactic features such as POS, gender, number, person and case.

agr = (*lemma*, *POS*, *gender*, *number*, *person*, *case*)

The possible values for these lexical entries are:

- **POS**: *noun*, *verb*, *adjective*, *pronoun*, *determiner*, *adverb*, *conjunction*, *numeral*
- **gender**: *masculine*, *feminine*, *neuter*
- **number**: *singular* and *plural*
- **person**: *first*, *second* and *third*
- **case**: *nominative*, *genitive*, *dative*, *accusative*

The goal of a stemmer or a lemmatizer is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form:

are, *am* , *is* >> *be*
working, works, worked >> *work*

However, the form base constructed by each of these applications may differ.

Stemming usually refers to a crude heuristic process that chops off the inflectional ending of words and often includes the removal of derivational affixes. The most famous stemmer is the Porter Stemmer for English ([13]). This stemmer removes around 60 different suffixes, using rewriting rules in two steps.

Opposite to a stemmer, a lemmatizer usually performs things properly with the use of a vocabulary and full morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, known as lemma.

For this reason, in our application development, we used a lemmatizer, more precisely, **Archeus lemmatizer**, freely available on the internet ([1]) under LGPL License. This product is an integrated tool for English and Romanian morphology, part-of-speech tagging and lemmatization.

The lexicon for this lemmatizer is very compact and stores for each word, the base form together with its morphological information. The lemmatizer includes several formats for the two lexicons (for English and Romanian Language): binary, text and XML format.

For each word form of the analysed sentence, the lemmatizer determines its root, part of speech, and – if appropriate - its gender, case, number, person, tense and comparative degree. But since this analysis treats each word separately, syntactic ambiguities are not resolved.

3.2 Implementation

We implement the morpho-syntactic annotation of an English natural language phrase by means of a semantic schema structure in a Java application that uses the Archeus Lemmatizer library:

`LematizatorArcheus.jar`

and also the morphologik-stremming library, a Polish morphological analyzer ([11]), for grammar correction in LanguageTool.

```
morfologik-stemming-nodict-1.4.0.jar
    morfologik-stemming-1.4.0.jar
```

In the first step, the morphology module delivers all possible lemma for each word form. Secondly, the tagger determines the grammatical categories of the word forms.

The lemma form for some words is ambiguous. For ambiguous word forms, all possible word form and their morphological descriptions are available by means of a contextual menu.

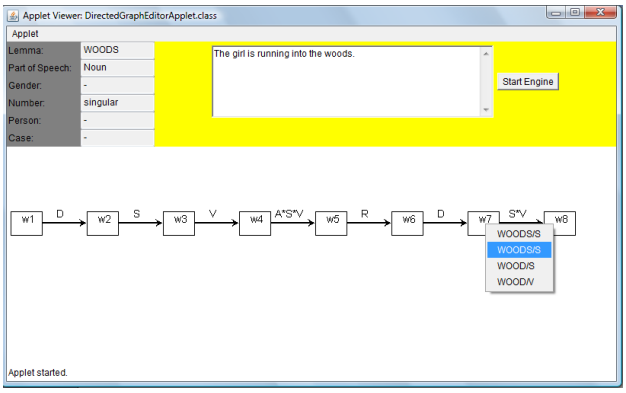

Figure 6 Morphologic data corresponding to each word's lemma

Each *node id* identifies an unit of the received input, more precisely, a word of the sentence. We define the class `Word` that encapsulates all the features of a natural language word in a member list called `senses`:

```
public class Word {
  List<Sense> senses;

  public Word()
   {senses = new ArrayList<Sense>();}

  public void addSense(String lemma,
            String morphoPOS)
   {senses.add(new Sense(lemma,
                    morphoPOS)}

 public String getSenses(int no)
   {...}

 public String getPOS()
   {...}
}
```

Each word sense is characterized by the word's lemma, the morphosyntactic information and the POS tagging data:

```
public class Sense {
    String lemma;
    char PV;
    char Gen;
    char Nr;
    char Pers;
    char Case;

      public Sense(final String lemma,
                final String morphoPOS)
        {...}

      public String getLemma()
        {return lemma;}

      public char getPV(){return PV;}

      public char getGen()
        {return Gen;}

      public char getNr(){return Nr;}

      public char getPers()
        {return Pers;}

      public char getCase()
        {return Case;}
}
```

Furthermore, this representation can be enriched with the compound relations build up against a grammar rules.

Resuming, by means of semantic schema representations, the sentence abstract structure is encoded by the set of dependency relations that exist among the words of the sentence, these relations being labelled with the corresponding part of speech of the dependent word.

In the next section we present the mechanism by means of which the translation can be made starting from the semantic schema representations.

4 Contextual Translation by means of Semantic Schema Interpretations

For an input sentence, each word (actually its lemma) must be paired with the equivalent word or words of the same part-of-speech in the target language. This process is called *word alignment*, being a hard NLP problem which can be stated as follows ([17]):

> given $<T_{SL}, T_{TL}>$ a pair of reciprocal translation texts in the Source Language, respectively Target Language, the word w_{SL} occurring in T_{SL} is said to be aligned to the word w_{TL} occurring in T_{TL} if the two words, in their context, represent reciprocal translations.

In what follows we will note by *SL words*, the words from of the Source Language, particularly from the input sentences, and by *TL words*, the

words from the Target Language. The Ob set of the interpretation will consist of all *TL words* that correspond to the meaning(s) of the *SL words*.

By means of a bilingual lexicon or wordnet lexicon which can exploit the Inter-Lingual Index ILI, the *SL words* are aligned with one or more words of the same part of speech in TL. Several cases in this matching process can occur:

M1. *In the case of non-ambiguity, there is an unique equivalent translation, which means that to each SL word corresponds exactly one TL word.*

In this case, the ob function is a bijective mapping that puts in correspondence the *SL words* with their synonyms from TL by means of the *SL words* identifiers.

M2. *Some SL words can be assigned to more than one TL word.*

In this case, the ob function has to use some evaluating criteria in order to choose the right translation from the resulted *TL words*, taking care to the whole meaning of the input sentence.

M3. *Some SL words can remain unaligned, which means there is no translation equivalent.*

In order to manage this case, the simplest solution consists of removing these *SL words* from the sentence being translated. But, of course, in this manner the translation process can be affected so the source lexicon used by the system must be carefully verified such that most of *SL words* to have at least one synonym in the system's Target Language.

If the sentence representation is given by $S = (X, A_0, A, R)$ then interpretation system of S is:

$$I = (Ob, ob, \{A\lg_u\}_{u \in A})$$

such that:

- Ob is a set of *TL words*
- $ob : X \to Ob$ the mapping that assigns one or more *TL words* to each *SL word*
- The algorithms of the system I can be grouped in two categories: algorithms that correspond to labels of A_0 and return a single *TL word* and algorithms that correspond to compound labels of $A \setminus A_0$ and return TL constructions made of more than one word.

The translation mechanism obtained by means of this interpretation system ensures that the system produces equivalent *TL words* for the *SL words* of the source sentence (ob function) and that the target sentence is as grammatical and fluent as possible (algorithms).

For this purpose, the algorithms defined for the interpretation system underline some language specific rules concerning Target Language constructions versus constructions particularities in Source Language. Indeed, in some languages like English or German the adjective must preceed its noun while in other (Spanish or Romanian), the noun must preceed the adjective ([4]).

Once the adequate *TL words* has been chosen by means of the ob function, the algorithms corresponding to the labels of A_0 include the morphological rules needed to generate the inflected forms of the target words with respect to the morpho-syntactic information of the *SL words* being translated.

4.1 Algorithms of the Interpretation system

As we have already specify from the beginning of this paper, out mechanism is intended to be a neutral-language representation. In order to resolve the substantial cross-linguistic variation of the input language with respect to the output language, we have to define corresponding interpretation algorithms.

Indeed, the algorithms that correspond to the $A \setminus A_0$ labels, that is, to syntactic groups of the input phrase, must resolve linguistic issues of the output language, such as *word order* (which carries various kinds of grammatical information) or the necessity of adding or deleting function words like determiners or prepositions ([10]).

In order to exemplify the manner in which contextual translations can be constructed using semantic schema interpretations we will reconsider the sentence Sen_1 from the previous section.

The representation of Sen_1 is given by $S_1 = (X, A_0, A, R)$ and thus, the translation is constructed by means of the interpretation system of S_1:

$$I = (Ob, ob, \{A\lg_u\}_{u \in A})$$

where:

- Ob is a set of dictionary word forms from Romanian language
- $ob : X \to Ob$ maps the *words ids* from the S_1 schema to the corresponding Romanian words of Ob
- the set of algorithms $\{A\lg_u\}, u \in A$:

• for every $u \in A_0$:

$A\lg_u(ob(id_i), ob(id_j))$

1. generate the inflected form of $ob(id_i)$ according to the $id_i.ana$
2. implement the concordances determined by the dependency *u*-relation between id_i and id_j

- $A\lg_{\theta(A,N)}(A\lg_A(o_1,o), A\lg_N(o,o_2))$

  1. verify the need of inserting or deleting function words between the adjective given by $A\lg_A(o_1,o)$ and the noun of $A\lg_N(o,o_2)$
  2. if word order in TL is *Noun-Adj* then reverse the order between parameters

- $A\lg_{\theta(D,\theta(A,N))}(A\lg_D(o_1,o), A\lg_{\theta(A,N)}(o,o_2))$

  upon the $A\lg_D(o_1,o)$ determiner type, articulate the noun phrase given by $A\lg_{\theta(A,N)}(o,o_2)$

- $A\lg_{\theta(V,\theta(A,N))}(A\lg_V(o_1,o), A\lg_{\theta(A,N)}(o,o_2))$

  verify the need of inserting or deleting function words between the verb given by $A\lg_V(o_1,o)$ and the noun phrase constructed by $A\lg_{\theta(A,N)}(o,o_2)$

- $A\lg_{\theta(\theta(D,\theta(A,N)),\theta(V,\theta(A,N)))}(A\lg_{\theta(D,\theta(A,N))}(o_1,o), A\lg_{\theta(V,\theta(A,N))}(o,o_2))$

  refine the structure of the sentence composed from the noun phrase given by $A\lg_{\theta(D,\theta(A,N))}(o_1,o)$ and the verb phrase of $A\lg_{\theta(V,\theta(A,N))}(o,o_2)$ based on the grammatical rules of TL

5 Conclusions

The presented translation method has a price to be paid and this is represented by the morpho-lexical properties transfer between source and target translation equivalents, followed by a generation of the inflected form in the target language.

On the other hand, centering the translation mechanism on lemmas and not on word forms ensure that the translation processes, that is the matching and the recombination mechanisms, can be further improved in order to ensure good translations.

Also, with the scope of providing more accurate translations, the representation of the dependencies relations by means of semantic schemas can be further refined according to the grammatical category of the dependent word.

6 Acknowledgment

This work was supported by the strategic grant POSDRU/89/1.5/S/61968, Project *ID 61986* (2009), co-financed by the *European Social Fund* within the Sectorial Operational Program Human Resources Development 2007-2013

References:

[1] Archeus Lemmatizer, www.archeus.ro.

[2] D. Arnold, L. Balkan, S. Meijer, R. Lee Humphreys, L. Sadler, *Machine Translation: an Introductory Guide*, NCC Blackwell, London, 1994, ISBN: 1855542-17x.

[3] N. Cancedda, M. Dymetman, G. Foster, C. Goutte, A Statistical Machine Translation Primer, *Learning Machine Translation*, The MIT Press Cambridge, Massachusetts, London, England, 2009.

[4] M. Colhon, D. Dănciulescu, Semantic Schemas for Natural Language Generation in Multilingual Systems, *Journal of Knowledge. Communications and Computing Technologies*, vol. II, no. 1, pp. 10—18, ISSN: 2067-0958, 2010.

[5] C. Dang, X. Luo, WordNet-based Document Summarization, *Proceedings of 7th WSEAS Int. Conf. on APPLIED COMPUTER & APPLIED COMPUTATIONAL SCIENCE* (ACACOS '08), Hangzhou, China, pp. 383—387, ISBN: 978-960-6766-49-7, ISSN: 1790-5117, 2008.

[6] EuroWordNet, http://www.illc.uva.nl/EuroWordNet/.

[7] F. Hristea, M. Popescu (eds), *Building Awareness in Language Technology*, Bucureşti University Publishing House, 2003.

[8] F. Hristea, M. F. Balcan, *Knowledge Searching and Representation in Artificial Intelligence. Theory and Applications*, Bucureşti University Publishing House, 2005 (in Romanian)

[9] G. LU, P. Huang, L. He, C. Cu, X. Li, A New Semantic Similarity Measuring Method Based on Web Search Engines, *WSEAS TRANSACTIONS on COMPUTERS Journal*, Issue 1, Volume 9, ISSN: 1109-2750, 2010.

[10] A. Menezes, C. Quirk, Syntactic Models for Structural Word Insertion and Deletion, *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pp. 735–744, Honolulu, 2008.

[11] Morfologik: Polish morphological analyzer, http://mac.softpedia.com/get/Word-Processing/Morfologik.shtml.

[12] Multext-East language specifications, http://nl.ijs.si/ME/V3/msd

[13] M. Porter. An Algorithm for Suffix Stripping. *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, 1980.

[14] M. Stanojević, S. Vraneš, Semantic Approach to Knowledge Processing, *WSEAS TRANSACTIONS on INFORMATION*

SCIENCE & APPLICATIONS Journal, vol. 5, issue 1, ISSN: 1790-0832, 2008

[15] N. Ţăndăreanu, Cooperating Systems Based on Maximal Graphs in Semantic Schemas, *Proceedings of the 11th WSEAS International Multiconference CSCC 2007* (Circuits, Systems, Communications, Computers), vol. 4, pp.517--522, Crete Island, Greece, ISSN: 1790-5117, ISBN: 978-960-8457-92-8, 2007

[16] N. Ţăndăreanu, Semantic Schemas and Applications in Logical representation of Knowledge, *Proceedings of the 10th International Conference on Cybernetics and Information Technologies, Systems and Applications* (*CITSA2004*), Orlando, Florida, USA, vol.III, pp. 82--87, 2004.

[17] D. Tufiş, Word sense disambiguation: a case study on the granularity of sense distinctions, *Proceedings of the 4th WSEAS International Conference on Signal Processing, Robotics and Automation*, Salzburg, Austria, ISBN:960-8457-09-2, 2005.

[18] P. Whitelock, Shake and Bake Translation, *Proceedings of the 14th conference on Computational Linguistics,* vol. 2, pp. 602 – 609, 1992.

[19] WordNet, http://wordnet.princeton.edu/