Intelligent Information Systems 9999
ISBN 666-666-666, pages 1–14

Ontology-Based Content Extraction from Polish Biobibliographical Lexicon

Wojciech Jaworski

Institute of Informatics, University of Warsaw
Banacha 2, 02-097 Warsaw, Poland
wjaworski@mimuw.edu.pl

Abstract

A vast amount of knowledge is contained nowadays in large collections of unstructured or weakly structured text documents. In this paper we pursue the goal of translation the textual data into a structural form, which is necessary for processing information enclosed in such data. Our methodology is based on the assumption that we are interested in information concerning a specific domain for which we are able to develop a relevant ontology. We perform profound analysis of text structure using two cooperating grammars: one is a language dependent grammar which performs syntactic decomposition of text and the other is an ontology dependent semantic grammar which defines semantic constraints between objects described by particular phrases. We introduce the special methodology for dealing with massive ambiguity that arises during processing textual data. The system is also augmented with the capability of processing documents whose parts describe concepts not included in the ontology. We we demonstrate effectiveness of our system by performing experiments in a real-world, restricted domain: the Biobibliographical Lexicon of Polish Writers and Literary Scholars.

Keywords: natural language processing, natural language understanding, semantic parsing, semantic annotation, content extraction

1 Introduction

A vast amount of knowledge is contained nowadays in large collections of unstructured or weakly structured text documents. The utility of such document collections depends on the ability of finding the relevant information. Users seek not only for information localized in specific documents but also knowledge spread across the whole document collection.

The main obstacle while processing text documents lies in the fact that the structure in textual data is computationally opaque. The structure of information must be reconstructed in order to find the relevant data in the document corpora and extract the information from them. The knowledge about the language in which the documents are written and about the world described in them is needed for modeling this structure.

In this paper we pursue the goal of translating the textual data into a structural form. Our methodology is based on the assumption that we are interested in information concerning a specific domain, for which we are able to develop a relevant ontology. This ontology represents the domain knowledge: it splits the set of objects belonging to a given domain into categories (types) and it determines relations between these categories. The ontology is a data model analogous to entity relationship diagram. The knowledge necessary to develop the ontology is provided by experts.

In comparison to other systems (such as LingInfo (Buitelaar *et al.*, 2005) and SKOS (Miles *et al.*, 2005)) our ontology formalism is very simple. We adjusted it to the task of content extraction excluding parts of formalism which are not useful, such as object oriented notation and names of relations that hold between concepts in the ontology.

We store the information extracted from documents using a meaning representation language, which is based on First Order Predicate Logic (Mendelson, 1997). Sentences in this language are composed of predicates connected by conjunctions and/or disjunctions, whose arguments are constants. Constants play the role of labels for objects described in the texts. Predicates represent ontological concepts. They express the membership of an object to a category and relationships between objects. We assume that documents describe only positive dependencies between definite objects, so it is possible to label each object with a unique constant and express document contents without any use of variables and quantifiers.

We process documents by means of a grammar which describes the way in which phrases are constructed from words and other simpler phrases. At the core of the presented methodology lies the assumption of compositionality: conformity of the structure that describes the world (i.e., the ontology) with the structure of the language that describes it. We assume, that syntactic operations constructing compound phrases on the basis of simpler ones correspond to the presentation of complex objects by means of their components. This observation leads us to the idea of introducing two cooperating grammars: one is a language dependent grammar which performs syntactic decomposition of text and the other is an ontology dependent semantic grammar which defines semantic constraints between objects described by particular phrases.

Such an approach guarantees scalability of the system and provides convenient representation for the fact that text contents is independent from syntactic features used to express it (for example phenomena such as event nominalizations which are nouns that semantically play a role of verbs). Our approach differs from the one introduced in Information Extraction systems (for ex. Mykowiecka *et al.* (2005)). Classical IE's methodology is based on shallow parsing and thus it is unable to express ambiguity and process phenomena such as coordination and inclusions. The other advantage of our methodology is that ontological concept membership is recognized on the level of lexicon. Then, compound concepts are constructed out of simpler ones by means of grammar, without the use of IE rules and additional frame filling phrase of processing.

Since our goal is to recognize dependencies between words we ignore some linguistic phenomena which are irrelevant for this task. As a consequence our grammar is significantly simpler that grammars proposed in literature (Przepiórkowski *et al.*, 2002; Vetulani, 2004; Mykowiecka, 2007).

A semantic value is associated with each grammar symbol during parsing process. This semantic value is a meaning representation language formula which assigns the semantics of the parsed text to the grammar symbol. Textual data is ambiguous, especially when the documents are damaged or contain errors, hence every sentence has an enormous number of interpretations. This property leads us to the idea of integrating the meaning representation language formulae with the syntactic decomposition of the text, so that the ambiguity is incorporated into its formulae.

The system is augmented with the capability of processing documents whose parts describe concepts not included in the ontology and grammar. This property allows us to keep the ontology small, without rare and/or irrelevant concepts.

The meaning representation language formulae obtained as the result of parsing are further processed. We suitably transform language constructions such as the relative clauses, participle phrases, nominalizations or inclusions which are not compositional in the sense of our grammar in order to obtain a representation of document contents that has an explicit structure determined by the ontology.

We also replace natural language conjunctions with logical conjunctions and reduce ambiguity. It should be mentioned that complete removing of ambiguity is not possible, and its presence is a major difference between the information stored in a form of meaning representation language formulae and a relational database.

We demonstrate the effectiveness of our system by performing experiments in a real-world, restricted domain: the Biobibliographical Lexicon of Polish Writers and Literary Scholars (Czachowska and Szałagan, 1994–2008).

2 A Biobibliographical Lexicon

"Współcześni polscy pisarze i badacze literatury. Słownik biobibliograficzny." [Contemporary Polish Writers and Literary Scholars. Biobibliographical Lexicon.] (Czachowska and Szałagan, 1994–2008) is a 10 volume lexicon edited by Jadwiga Czachowska and Alicja Szałagan in the Laboratory of Contemporary Literature Documentation of Literary Research Institute of Polish Academy of Sciences.

It contains extensive information about the life and work of about 2000 contemporary Polish poets, novelist, playwriters, literary and theater critics, essayists, translators, historians and theorists of literature. The lexicon covers writers and literary scholars who made their debuts after the year 1918 and had at least three books or dramatic works issued before 1988.

Each entry is divided into three parts: a biography of the writer, the bibliography of his compositions and the bibliography of studies concerning his/her work. In our research we restrict to the biography because it is the only part of lexicon entry that is written in plain text.

Biographies raise the following issues: dates of life, social origin, education, debut, conspiratorial activity, professional activity, organizations' membership, place of residence, characteristics of writings and places of publication.

Biographies are written using long sentences (the average length of a sentence in this corpus is 17,82 words), with extensive use of conjunctions and insertions.

`Drukował wiersze i artykuły w`	*He published poems and articles in*
`Robotniku (1927-28),`	*Robotnik (1927-28),*
`Naszym Przeglądzie (1927-29),`	*Nasz Przegląd (1927-29),*
`Sterze (1937) oraz`	*Ster (1937) and*
`miesięczniku Wymiary (1938),`	*monthly Wymiary (1938),*
`którego był współredaktorem.`	*of which he was coeditor.*

FIGURE 1: An example of a sentence from the Biobibliographical Lexicon of Polish Writers and Literary Scholars and its English translation

The Lexicon consists of 806721 words and 45259 sentences. Fig. 1 presents an example of a sentence from the biobibliographical lexicon.

3 Ontology

An ontology represents the domain knowledge: it splits the set of objects belonging to a given domain into categories (types) and it determines relations between objects belonging to these categories. Categories arranges entities (or objects) that compose a domain and therefore it defines the structure of the domain.

Apart from the category membership, there are present relations that order objects belonging to different categories. The key relation is the *meronymy relation* (written as *part-of*) that alongside with its variants such as *attribute-of*, *description-of* constitutes the core of the ontology. The other relations such as *subsumption relation*: *is a subtype of*, useful in large ontologies, may be also added.

Information in Biobibliographical Lexicon of Polish Writers and Literary Scholars is organised by means of events (see fig. 2). There are several types of events:

- **Publication Event** describes **Compositions** and **Magazines** (where they were published) alongside with publication **Date**. Here also is given information about debuts (**Publication Event Type**) and the type of writing activity is included.
- **Medal Event** documents awards.
- **Work Event** gives an account of relations between the writer and **Organisation**. **Work Event Type** defines such relations as employment, membership, cooperation, etc.
- **Education Event** reports **Organisation**, **Location**, type of educational activity and achieved degrees.
- **Location Event** gives an account of **Locations** where writer lived.
- **Filiation Event** describes family relations.
- **Life Event** documents birth, marriages and decease.

"Event types" store the word used to introduce event in a sentence. They are obligatory parameters of events and they determine its appearance.

Event parameters may have internal structure:

- **Job** includes ordinary jobs as well as types of membership, cooperation, and writing activity.

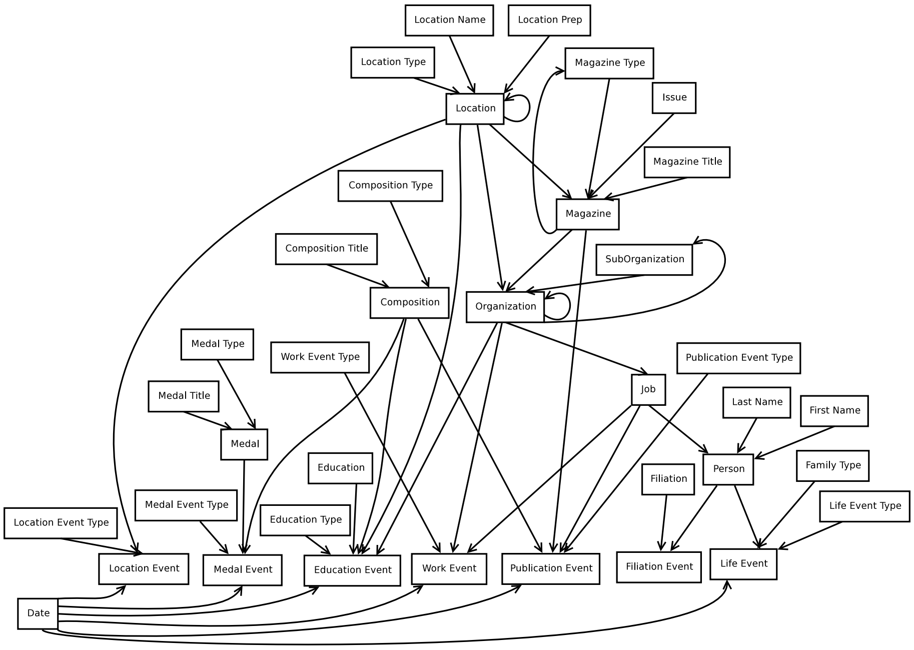

FIGURE 2: Ontology of Biobibliographical Lexicon of Polish Writers and Literary Scholars. Names of categories are written in boxes while arrows connect categories which contain related objects.

- **Organisation** is any type of corporation, school, society, etc.
- **Sub Organisation** is a part of **Organisation**.
- **Person** is denoted by its **First Name**, **Last Name** and **Job**.
- **Composition** is a piece of writer's work.
- **Magazine** is a type of **Organisation** in which **Compositions** are published, often periodically.
- **Location** is defined by **Location Name**, **Location Type**, or by proximity to another **Location**.

A person who is a subject of the biography is the default parameter of each event. Yet, since the agent is never mentioned explicitly in the text we do not include this parameter in the ontology.

4 Lexical Analysis

During the lexical analysis we identify grammatical and ontological category of words and multiword expression. We represent this information by means of *feature structures*, which we formally define as follow:

Definition Let A be a finite set of attributes and V a finite set of values. Let g be a function from A to $P(V)$, which defines attribute domains. We call $\mathcal{F}_{A,V,g}$ a set of feature structures over A, V and g if

$$\mathcal{F}_{A,V,g} = \{X \subseteq A \times V : \forall_{(a,v)\in X} v \in g(a) \wedge \forall_{(a_1,v_1)\in X} \forall_{(a_2,v_2)\in X} a_1 \neq a_2\}$$

Each attribute-value pair we will denote as *feature*.

We base on a lexicon which is represented as a set of finite state rules. For the sake of efficiency, we encode feature structures into atomic symbols. Example rules (together with their semantic attachments described in Sect. 8):

Rule	**Semantic Attachment**
`Digit` ::= `0`	$\lambda x_0 x_1$.`Digit`(x_0, `0`)
...	...
`Digit` ::= `9`	$\lambda x_0 x_1$.`Digit`(x_0, `9`)
`Year` ::= `1 9 Digit Digit`	$\lambda x_0 x_1 x_2 x_3 x_4$.`Year`($x_0$, `19`[$x_3$][$x_4$])
`YearInterval` ::= `Year - Digit Digit`	$\lambda x_0 x_1 x_2 x_3 x_4$.`Year`($x_0$, [$x_1$], [$x_3$][$x_4$])
`Incl:incl` ::= `( Year )`	$\lambda x_0 x_1 x_2 x_3$.`Date`(x_0, x_2)
`Incl:incl` ::= `( YearInterval )`	$\lambda x_0 x_1 x_2 x_3$.`Date`(x_0, x_2)
`MagazineTitle:subst:sg:loc` ::= `Robotniku`	$\lambda x_0 x_1$.`MagazineTitle`(x_0, `Robotnik`)
`MagazineTitle:subst:sg:loc` ::= `Naszym Przeglądzie`	$\lambda x_0 x_1 x_2$.`MagazineTitle`(x_0, `Nasz Przegląd`)
`MagazineTitle:subst:sg:loc` ::= `Sterze`	$\lambda x_0 x_1$.`MagazineTitle`(x_0, `Ster`)
`MagazineTitle:subst:pl:nom` ::= `Wymiary`	$\lambda x_0 x_1$.`MagazineTitle`(x_0, `Wymiary`)
`PublicationEventType:core:verb:acc` ::= `Drukował`	$\lambda x_0 x_1$.`PublicationEventType`(x_0, `drukował`)
`EventState:core:verb:inst` ::= `był`	$\lambda x_0 x_1$.`EventState`(x_0, `był`)
`CompositionType:subst:pl:acc` ::= `wiersze`	$\lambda x_0 x_1$.`CompositionType`(x_0, `wiersze`)
`CompositionType:subst:pl:acc` ::= `artykuły`	$\lambda x_0 x_1$.`CompositionType`(x_0, `artykuły`)
`MagazineType:subst:sg:loc` ::= `miesięczniku`	$\lambda x_0 x_1$.`MagazineType`(x_0, `miesięcznik`)
`Job:subst:sg:inst` ::= `współredaktorem`	$\lambda x_0 x_1$.`Job`(x_0, `współredaktor`)
`Prep:prep:loc` ::= `w`	$\lambda x_0 x_1$.`Prep`(x_0, `w`)
`Prep:prep:acc` ::= `w`	$\lambda x_0 x_1$.`Prep`(x_0, `w`)
`Conj:conj` ::= `i`	$\lambda x_0 x_1$.`Conj`(x_0, `i`)
`Conj:conj` ::= `oraz`	$\lambda x_0 x_1$.`Conj`(x_0, `oraz`)
`Conj:conj` ::= `,`	$\lambda x_0 x_1$.`Conj`(x_0, `,`)
`Relpron:relpron:sg:gen` ::= `którego`	$\lambda x_0 x_1$.`Relpron`(x_0, `którego`)

Names of symbols in the above rules encode, divided by colons, values of features such as ontological concept names, part-of-speech and inflectional forms. The encoding is done in the following way: The first feature is always an ontological category of symbol, which becomes the value of `sem` attribute. Then a part-of-speech, named `cat`, is described. The following parts of speech are distinguished: `subst` (nouns and noun phrases), `adj` (adjectives), `part` (adjectival participles), `prep` (prepositions), `pp` (preposition phrases), `core` (verbs and adverbial participles), `adv` (adverbs), `conj` (conjunctions), `ppron3` (third person pronouns), `relpron` (relative pronouns), `relpp` (relative pronouns).

Remaining features depend on the `cat` value. For `subst`, `adj`, `part`, `ppron3` and `relpron` it is `number` (with values `sg` or `pl`) and `case` (with one of the following values: `nom`, `gen`, `dat`, `acc`, `inst`, `loc`).

In case of `part` there may be one more feature, `case2`, which denotes the case of the participle object. Similarly `prep` has one feature, `case`, which denotes the case of noun phrase with whom it is bounded.

`pp`, `adv`, `conj`, `relpp` do not have any additional features.

The `core` has the feature `sen` which distinguishes between verbs (`verb` value) and adverbial participles (`part` value). It may have also one more feature, `case`, which denotes the case of its object.

We ignore some features (for example gender or tense) because our goal is not to completely describe language but to recognise dependencies between words by means om grammar. The omission of some features results in significant simplification of the grammar and little increase of ambiguity.

Syntactic information included in lexicon was generated using Morfeusz (Woliński, 2006), while semantic tagging was done manually.

5 Syntactic analysis

We define a grammar for parsing feature structures as follows:

$$G = (\mathcal{F}_{A,V,g}, F_I, \mathcal{X}, \mathcal{C}, \mathcal{R})$$

where

- $\mathcal{F}_{A,V,g}$ is a set of feature structures;
- $F_I \in \mathcal{F}_{A,V,g}$ is the start-symbol of G;
- $\mathcal{X} = \{X_i\}_{i=0}^{\infty}$ is an infinite sequence of variables;
- $\mathcal{C}$ is a set of constraints, such that

  $$\mathcal{C} = \{X_i.a = v \mid a \in A \wedge v \in g(a)\} \cup \{X_i.a_1 = X_j.a_2 \mid a_1, a_2 \in A \wedge g(a_1) = g(a_2)\};$$

- $\mathcal{R}$ is a finite set of rules. Each rule is a pair $(\{X_i\}_{i=0}^{n}, C)$, where $C \subset \mathcal{C}$ such that all variables used in constraints in C belong to $\{X_i\}_{i=0}^{n}$. $\{X_i\}_{i=0}^{n}$ is denoted by means of expression $X_0 ::= X_1 X_2 \ldots X_n$. We denote X_0 as *rule head* and $X_1 X_2 \ldots X_n$ as *rule body.*

Let us consider the rule $(X_0 ::= X_1 X_2 \ldots X_n, C)$. The body of this rule is matched with the sequence $F_1, \ldots, F_n$ of feature structures and rule head generates a new feature structure F_0. In order to apply the rule each constraint must be satisfied. There are the following conditions for constraint satisfaction (we assume that $i > 0$ and $j > 0$)

- $X_i.a = v$ — F_i must have an attribute a defined and its value must be equal to v.
- $X_0.a = v$ — attribute a with value v for feature structure F_0 is generated,
- $X_i.a_1 = X_j.a_2$ — F_i must have an attribute a_1, F_j must have an attribute a_2 and values of these attributes must be equal,
- $X_0.a_1 = X_j.a_2$ — F_j must have an attribute a_2, an attribute a_1 with value $F_j.a_2$ for feature structure F_0 is generated.

Our grammar is composed out of two sets of rules: syntactic rules assure that every parsed phrase has a proper syntactic structure, while semantic rules guarantee that every phrase describes a sound ontological concept. In order to create a new symbol a pair of rules must be successfully matched with the same symbol sequence. One of these rules must be a syntactic rule (which defines `cat` attribute) and the other must be a semantic rule (which defines `sem` attribute). This approach is a consequence of the assumption that every phrase is semantically consistent, i.e., there exists an ontological category related to the phrase content.

The grammar was created on the basis of the biobibliografical lexicon and it is attuned to the language used in the lexicon. As a consequence the grammar is capable of parsing long sentences with the extensive use of coordination, relative clauses, inclusions, participle phrases and verb nominalizations. On the other hand it does not cover phenomena that were rare, absent or irrelevant in the lexicon, such as: verb conjugation, negation, numerals and pronouns (other than relative ones). However, due to its simplicity the may be easily modified for other usecases.

Below a few example syntactic rules are presented:

- identity rules that are complements for semantic rules

$$\begin{array}{lllll} X_0 ::= X_1 & X_0.\texttt{cat} = X_1.\texttt{cat}, & X_0.\texttt{number} = X_1.\texttt{number}, & X_0.\texttt{case} = X_1.\texttt{case} \\ X_0 ::= X_1 & X_0.\texttt{cat} = X_1.\texttt{cat}, & X_0.\texttt{sen} = X_1.\texttt{sen}, & X_0.\texttt{case} = X_1.\texttt{case} \\ X_0 ::= X_1 & X_0.\texttt{cat} = X_1.\texttt{cat}, & X_0.\texttt{sen} = X_1.\texttt{sen} & \end{array}$$

- introduction of noun phrase

$$\begin{array}{llll} X_0 ::= X_1 X_2 & X_0.\texttt{cat} = X_1.\texttt{cat}, & X_0.\texttt{number} = X_1.\texttt{number}, & X_0.\texttt{case} = X_1.\texttt{case}, \\ & X_1.\texttt{cat} = \texttt{subst}, & X_2.\texttt{cat} = \texttt{subst}, & X_2.\texttt{case} = \texttt{nom} \end{array}$$

- introduction of preposition phrase

$$\begin{array}{llll} X_0 ::= X_1 X_2 & X_0.\texttt{cat} = \texttt{pp}, & & \\ & X_1.\texttt{cat} = \texttt{prep}, & X_2.\texttt{cat} = \texttt{subst}, & X_1.\texttt{case} = X_2.\texttt{case}, \end{array}$$

- rules that join predicate with its arguments

$$\begin{array}{llll} X_0 ::= X_1 X_2 & X_0.\texttt{cat} = X_1.\texttt{cat}, & X_0.\texttt{sen} = X_1.\texttt{sen}, & \\ & X_1.\texttt{cat} = \texttt{core}, & X_2.\texttt{cat} = \texttt{subst}, & X_1.\texttt{case} = X_2.\texttt{case} \\ X_0 ::= X_1 X_2 & X_0.\texttt{cat} = X_1.\texttt{cat}, & X_0.\texttt{sen} = X_1.\texttt{sen}, & \\ & X_1.\texttt{cat} = \texttt{core}, & X_2.\texttt{cat} = \texttt{pp} & \end{array}$$

- introductions of inclusion, relative clause and conjunction

$$\begin{array}{llll} X_0 ::= X_1 X_2 & X_0.\texttt{sem} = X_1.\texttt{sem}, & & \\ & X_0.\texttt{cat} = X_1.\texttt{cat}, & X_0.\texttt{number} = X_1.\texttt{number}, & X_0.\texttt{case} = X_1.\texttt{case}, \\ & X_2.\texttt{cat} = \texttt{incl} & & \\ & \lambda x_0 x_1 x_2.Rel(x_0, x_1, x_2) & & \\ X_0 ::= X_1 X_2 X_3 X_4 & X_0.\texttt{sem} = X_1.\texttt{sem}, & & \\ & X_0.\texttt{cat} = X_1.\texttt{cat}, & X_0.\texttt{number} = X_1.\texttt{number}, & X_0.\texttt{case} = X_1.\texttt{case}, \\ & X_1.\texttt{cat} = \texttt{subst}, & X_2.\texttt{cat} = \texttt{,}, & X_3.\texttt{cat} = \texttt{relpron}, \\ & X_4.\texttt{cat} = \texttt{core}, & X_4.\texttt{sen} = \texttt{verb}, & X_1.\texttt{number} = X_3.\texttt{number}, \\ & \lambda x_0 x_1 x_2 x_3 x_4.Rel(x_0, x_1, x_4) & & \\ X_0 ::= X_1 X_2 X_3 & X_0.\texttt{sem} = X_1.\texttt{sem}, & X_1.\texttt{sem} = X_3.\texttt{sem}, & \\ & X_0.\texttt{cat} = X_1.\texttt{cat}, & X_0.\texttt{number} = X_1.\texttt{number}, & X_0.\texttt{case} = X_1.\texttt{case}, \\ & X_2.\texttt{cat} = \texttt{conj}, & X_1.\texttt{cat} = X_3.\texttt{cat}, & X_1.\texttt{case} = X_3.\texttt{case} \\ & \lambda x_0 x_1 x_2 x_3.And(x_0, x_1, x_3) & & \end{array}$$

The last three rules define `sem` attribute and are assigned with semantic attachment so they do not require any additional semantic rule. The rest of them must be accompanied by one of the following rules:

$$\begin{array}{llll} X_0 ::= X_1 & X_0.\texttt{sem} = \texttt{Magazine}, & X_1.\texttt{sem} = \texttt{MagazineTitle} & \\ & \lambda x_0 x_1.\texttt{Magazine}(x_0, x_1) & & \\ X_0 ::= X_1 X_2 & X_0.\texttt{sem} = \texttt{Magazine}, & X_1.\texttt{sem} = \texttt{MagazineType}, & X_2.\texttt{sem} = \texttt{MagazineTitle} \\ & \lambda x_0 x_1 x_2.\texttt{Magazine}(x_0, x_1, x_2) & & \\ X_0 ::= X_1 X_2 & X_0.\texttt{sem} = \texttt{Magazine}, & X_1.\texttt{sem} = \texttt{Prep}, & X_2.\texttt{sem} = \texttt{Magazine} \\ & \lambda x_0 x_1 x_2.\texttt{Magazine}(x_0, x_2) & & \end{array}$$

$X_0 ::= X_1$	X_0.sem = Composition, $\lambda x_0 x_1$.Composition(x_0, x_1)	X_1.sem = CompositionType	
$X_0 ::= X_1$	X_0.sem = Organisation, $\lambda x_0 x_1$.Organisation(x_0, x_1)	X_1.sem = Magazine	
$X_0 ::= X_1 X_2$	X_0.sem = WorkEvent, $\lambda x_0 x_1 x_2$.WorkEvent(x_0, x_1, x_2)	X_1.sem = EventState,	X_2.sem = Job
$X_0 ::= X_1 X_2$	X_0.sem = WorkEvent, $\lambda x_0 x_1 x_2$.WorkEvent(x_0, x_1, x_2)	X_1.sem = Organisation,	X_2.sem = WorkEvent
$X_0 ::= X_1$	X_0.sem = PublicationEvent, $\lambda x_0 x_1$.PublicationEvent(x_0, x_1)	X_2.sem = PublicationEventType	
$X_0 ::= X_1 X_2$	X_0.sem = PublicationEvent, $\lambda x_0 x_1 x_2$.PublicationEvent(x_0, x_1, x_2)	X_1.sem = PublicationEvent,	X_2.sem = Composition
$X_0 ::= X_1 X_2$	X_0.sem = PublicationEvent, $\lambda x_0 x_1 x_2$.PublicationEvent(x_0, x_1, x_2)	X_1.sem = PublicationEvent,	X_2.sem = Magazine

The above rules encode dependencies between ontological concepts. They are assigned with semantic attachments (see Sect. 8).

6 Parser algorithm

We mentioned in section 4 that we encode feature structures as atomic symbols. Now, we translate G into the rules of context-free grammar.

G rules are split into three separable sets: a set of syntactic rules R, a set of semantic rules S and a set of rules that define both syntactic and semantic features Q. We compose syntactic rules with semantic ones obtaining the set $P = Q \cup P_1 \cup P_2 \cup \ldots$, where

$$P_n = \{(X_0 ::= X_1 X_2 \ldots X_n, C_R \cup C_S) \mid$$
$$(X_0 ::= X_1 X_2 \ldots X_n, C_R) \in R \wedge (X_0 ::= X_1 X_2 \ldots X_n, C_S) \in S\}.$$

For example rules

$X_0 ::= X_1 X_2$	X_0.cat = pp, X_1.cat = prep, X_1.case = X_2.case	X_2.cat = subst,	
$X_0 ::= X_1 X_2$	X_0.sem = Magazine,	X_1.sem = Prep,	X_2.sem = Magazine

are composed into the following rule

$X_0 ::= X_1 X_2$	X_0.cat = pp,	X_1.cat = prep,	X_2.cat = subst,
	X_1.case = X_2.case		
	X_0.sem = Magazine,	X_1.sem = Prep,	X_2.sem = Magazine

Then, we eliminate from the grammar constraints that have the form $X_i.a_1 = X_j.a_2$. Consider a rule $(X_0 ::= X_1 X_2 \ldots X_n, \{X_i.a_1 = X_j.a_2\} \cup C)$. We replace this rule by the following set of rules

$$\{(X_0 ::= X_1 X_2 \ldots X_n, \{X_i.a_1 = v, X_j.a_2 = v\} \cup C) \mid v \in g(a_1)\}.$$

The rules with contradicting constraints are eliminated from the grammar. The rule from our example is transformed into

$X_0 ::= X_1 X_2$	X_0.cat = pp,	X_1.cat = prep,	X_2.cat = subst,
	X_1.case = nom	X_2.case = nom	
	X_0.sem = Magazine,	X_1.sem = Prep,	X_2.sem = Magazine
$X_0 ::= X_1 X_2$	X_0.cat = pp,	X_1.cat = prep,	X_2.cat = subst,
	X_1.case = gen	X_2.case = gen	
	X_0.sem = Magazine,	X_1.sem = Prep,	X_2.sem = Magazine

etc.

Now, we replace constraints with feature structures that match them. We obtain a context-free grammar:

```
Magazine:pp ::= Prep:nom Magazine:subst:sg:nom
Magazine:pp ::= Prep:nom Magazine:subst:pl:nom
Magazine:pp ::= Prep:gen Magazine:subst:sg:gen
Magazine:pp ::= Prep:gen Magazine:subst:pl:gen
```

etc.

The transformation that we apply significantly increase the number of rules. In order to apply efficiently a large number of rules we organise rules in the data structure called a *prefix tree*. For each rule $A \rightarrow \alpha_1 \dots \alpha_k$ in R_n we create a path in the tree from the root labelled by symbols α_1 till α_k and we label the leaf tree node by A. For each node we merge paths that have identical labels. Prefix tree allows us to match rules with symbols in logarithmic time.

We parse this grammar using Earley Parser (Earley, 1986). We represent all the possible derivation trees for a given text and grammar by means of a directed acyclic graph whose edges are labelled by grammar symbols. We call it *a chart*.

The text is a sequence of words. We represent this sequence as an oriented graph which is a list with signs as edges. Vertices of this graph are numbers pointing to the position in text. Then for each rule we find all paths in the graph with sequences of edge labels that match the body of the rule, and add to the graph new edges from the beginnings to the ends of those paths, and label them with the rule head. We will denote an edge labelled α from vertex i to vertex j by $\alpha_{i,j}$.

7 Meaning representation language

The meaning representation language encodes concepts included in the data and dependencies between them according to the ontology.

Syntax of the language is defined as follows: We have a set of constants and a set of predicate names. Predicates are atomic formulae. Compound formulae are composed of one or more atomic formulae connected by conjunctions ($\wedge$) and/or disjunctions ($\vee$). We do not use quantifiers, functions and negation. We use disjunction for representing ambiguity in documents.

For example, $\texttt{Person}(p)$ states that p is a **Person** (belongs to the **Person** concept in the ontology). $\texttt{Person}(p, j) \wedge \texttt{Job}(j)$ means that p is a person and this person is in relation with j which is another entity. The ontological category of j is provided by predicate $\texttt{Job}(j)$.

In general, constants play the role of labels for entities described in the data. For each object mentioned in the document we introduce a unique constant; not only for entities described by a single word but also for objects denoted by phrases.

Predicates represent relations on finite sequences of entities. Each predicate consists of a name and a list of one or more arguments. Names of most predicates are identical to the names of the related ontological concepts. Number of arguments for a given predicate is not fixed. The first argument of a predicate is an object which belongs to the ontological concept denoted by predicate name. The

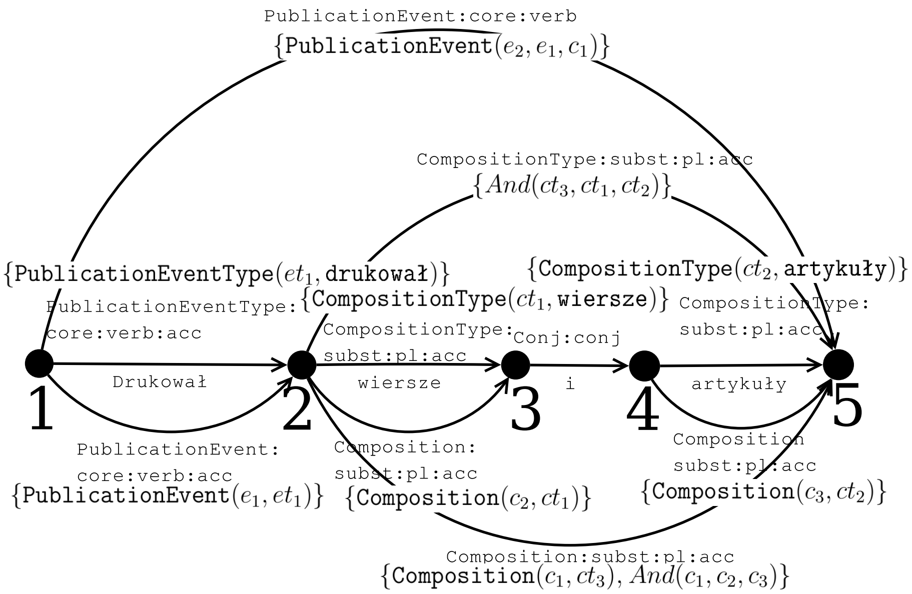

FIGURE 3: A chart with semantic values.

$$[\![\texttt{Composition:subst:pl:acc}]\!]_{2,3} = \texttt{Composition}(c_2, ct_1) \wedge [\![\texttt{CompositionType:subst:pl:acc}]\!]_{2,3} =$$
$$= \texttt{Composition}(c_2, ct_1) \wedge \texttt{CompositionType}(ct_1, \texttt{wiersze})$$
$$[\![\texttt{Composition:subst:pl:acc}]\!]_{2,5} = \big(\texttt{Composition}(c_1, ct_3) \wedge [\![\texttt{CompositionType:subst:pl:acc}]\!]_{2,5}\big) \vee$$
$$\vee \big(And(c_1, c_2, c_3) \wedge [\![\texttt{Composition:subst:pl:acc}]\!]_{2,3} \wedge [\![\texttt{Composition:subst:pl:acc}]\!]_{4,5}\big)$$

FIGURE 4: Semantic values of grammar symbols

rest of predicate arguments are objects that are parts, properties, attributes or descriptions of the first argument.

8 Semantic analysis

In case of an ambiguous grammar (as in our case), the number of possible syntax derivation trees may be exponential in the sequence length. The concept of chart is intended to be their common, compact representation. The number of possible semantic values of the sequence is equal to the number of syntax derivation trees. That is why we cannot represent them directly. Instead, we distribute the semantic values across the chart.

For that to be possible the meaning representation language formulae must be coherent with the syntactic decomposition. The following relation between syntax and semantics is required: we assume that each phrase describes an entity. Chart edges are created as a result of parsing phrases, so, for each edge, there exists an entity related to that edge.

With each edge of the chart we associate a meaning representation language formula, which we name the *semantic value of the grammar symbol.* This formula describes the properties of the entity related to the edge. Consider the edge $\alpha_{i,j}$ of chart. We denote its semantic value as $[\![\alpha]\!]_{i,j}$. We represent the entity associated with it by means of a constant $a_{\alpha,i,j}$ and the formula associated with the edge has the following structure:

$$[\![\alpha]\!]_{i,j} = \bigvee_{k=1}^{n} (p_k(a_{\alpha,i,j}, a_{\alpha_1^k,i_1^k,j_1^k}, \ldots, a_{\alpha_{m_k}^k,i_{m_k}^k,j_{m_k}^k}) \wedge \bigwedge_{l=1}^{m_k} [\![\alpha_l^k]\!]_{i_l^k,j_l^k}),$$

where p_k is a predicate that describes the ontological category of $a_{\alpha,i,j}$. Subobjects of $a_{\alpha,i,j}$ are denoted as $a_{\alpha_1^k,i_1^k,j_1^k}, \ldots, a_{\alpha_{m_k}^k,i_{m_k}^k,j_{m_k}^k}$. These subobjects are represented by phrases whose semantic values are conjuncted with predicate p_k. The disjunction at the beginning of the formula is used to connect different possible phrase interpretations.

The structure of the semantic value allows us to spread it across the chart. Each $[\![\alpha_l^k]\!]_{i_l^k,j_l^k}$ is assigned to the edge ${\alpha_l^k}_{i_l^k,j_l^k}$, so only the set of atomic formulae

$$\{p_1(a_{\alpha,i,j}, a_{\alpha_1^1,i_1^1,j_1^1}, \ldots, a_{\alpha_{m_k}^1,i_{m_k}^1,j_{m_k}^1}), \ldots, p_n(a_{\alpha,i,j}, a_{\alpha_1^n,i_1^n,j_1^n}, \ldots, a_{\alpha_{m_k}^n,i_{m_k}^n,j_{m_k}^n})\}$$

must by associated with a chart edge on the implementation level.

Semantics for a terminal symbol $\alpha_{i,j}$ is an empty set.

On fig. 3 we present an example of a chart. On fig. 4 we show semantic values for some of the edges for chart on fig. 3. For example the semantic value of the edge $\texttt{Composition:subst:pl:acc}_{2,3}$ (denoted as $[\![\texttt{Composition:subst:pl:acc}]\!]_{2,3}$) is calculated to be the formula $\texttt{Composition}(c_2, ct_1) \wedge \texttt{CompositionType}(ct_1, \texttt{wiersze})$, because the constant ct_1 refers to the edge $\texttt{CompositionType:subst:pl:acc}_{2,3}$. Ambiguous phrases may be parsed to a number of predicates: consider for example the semantic value of an edge $[\![\texttt{Composition:subst:pl:acc}]\!]_{2,5}$.

Semantic values of grammar symbols are constructed using *semantic attachments* of grammar rules. Semantic attachments are functions that compose semantics of greater objects out of the semantics of smaller ones. Predicate sets associated with edges of chart are arguments and values for these functions.

We express semantic attachments as unnamed functions using the notation taken from the λ-calculus. We present semantic attachments for our example finite-state and G grammar rules in previous sections.

A typical semantic attachment introduces a predicate, whose arguments are constants referring to words or other predicates. We use $[x_i]$ notation to extract a specific digit value in order to compose the year number. Some rules introduce a special symbols *And* and *Rel*.

We use *And* symbol for representing the conjunction in natural language, which connects events described in the sentence and shorten it drawing before the bracket words identical in all connected events. We process *And* symbol expanding the brackets. Then we replace *And* with logical conjunction '$\wedge$'.

We use the special symbol *Rel* to introduce relative clauses and insertions. It has always three arguments and we will denote it as $Rel(a, b, c)$. c, its last argument is the relative phrase, and b is the modified phrase. Then we check

$\texttt{PublicationEventType}(et_1, \texttt{drukował}) \wedge \texttt{CompositionType}(ct_1, \texttt{wiersze}) \wedge \texttt{Composition}(c_1, ct_1)) \wedge$
$\wedge \texttt{CompositionType}(ct_2, \texttt{artykuły}) \wedge \texttt{Composition}(c_2, ct_2)) \wedge$
$\wedge \texttt{MagazineTitle}(mt_1, \texttt{Robotnik}) \wedge \texttt{Magazine}(m_1, mt_1) \wedge \texttt{YearInterval}(y_1, \texttt{1927-28}) \wedge \texttt{Date}(d_1, y_1) \wedge$
$\wedge \texttt{MagazineTitle}(mt_2, \texttt{Nasz Przegląd}) \wedge \texttt{Magazine}(m_2, mt_2) \wedge \texttt{YearInterval}(y_2, \texttt{1927-29}) \wedge \texttt{Date}(d_2, y_2) \wedge$
$\wedge \texttt{MagazineTitle}(mt_3, \texttt{Ster}) \wedge \texttt{Magazine}(m_3, mt_3) \wedge \texttt{Year}(y_3, \texttt{1937}) \wedge \texttt{Date}(d_3, y_3) \wedge$
$\wedge \texttt{MagazineType}(mt_4, \texttt{miesięcznik}) \wedge \texttt{MagazineTitle}(mt_5, \texttt{Wymiary}) \wedge \texttt{Magazine}(m_4, mt_4, mt_5) \wedge$
$\wedge \texttt{Year}(y_4, \texttt{1938}) \wedge \texttt{Date}(d_4, y_4) \wedge$
$\wedge \texttt{PublicationEvent}(e_1, c_1, m_1, d_1) \wedge \texttt{PublicationEvent}(e_2, c_1, m_2, d_2) \wedge$
$\wedge \texttt{PublicationEvent}(e_3, c_1, m_3, d_3) \wedge \texttt{PublicationEvent}(e_4, c_1, m_4, d_4) \wedge$
$\wedge \texttt{PublicationEvent}(e_5, c_2, m_1, d_1) \wedge \texttt{PublicationEvent}(e_6, c_2, m_2, d_2) \wedge$
$\wedge \texttt{PublicationEvent}(e_7, c_2, m_3, d_3) \wedge \texttt{PublicationEvent}(e_8, ct_2, m_4, d_4) \wedge$
$\wedge \texttt{Magazine}(m_5, mt_5) \wedge \texttt{Organisation}(o_1, m_5) \wedge \texttt{Job}(j_1, \texttt{współredaktor}) \wedge \texttt{WorkEvent}(e_9, j_1, o_1)$

FIGURE 5: The semantics of text Fig. 1.

whether c describes an event. If this is the case then we parse b and c into a consistent formula which we connect with other events in the sentence by means of conjunction. Otherwise, we look for event which is parametrised by c and we transform c into an ordinary parameter of this event.

Fig. 5 presents the meaning representation language formula for our example sentence.

9 Incomplete ontology

In every corpus, there is a number of phrases that cannot be expressed using concepts typical for that corpus. These phrases are rare, irregular and devoid of characteristic contexts.

In order to obtain completely parsed documents we must develop a semantic representation for documents we partially do not understand. We close the ontology with the **Other Information** concept, which refers to information not included in the rest of concepts.

Parsing process is driven independently by syntactic and semantic constraints. Syntactic information concerning each word is relatively easy to obtain, so we assume that we know syntactic features of each phrase labelled as **Other Information**. Then, we treat **Other Information** as possible argument of every ontological concept and we perform parsing process as usual.

10 Experimental results

We developed 120152 finite-state rules, 66 syntactic rules and 124 semantic rules for the Biobibliographical lexicon. As we described in section 6 we composed syntactic rules with semantic ones and we obtained a grammar of 249752 rules.

Using this grammar we parsed 84% of sentences, and we extracted 88901 events with precision 100% and recall 45%.

Relatively low recall is caused by the fact that due to the high cost of manual semantic tagging only part of words were tagged. Obtained results may be further improved by means of introducing automatic tagger. However, this would probably decrease the precision of the system.

Acknowledgements I thank Jadwiga Czachowska and Alicja Szałagan for allowing me to access electronic version of "Współcześni polscy pisarze i badacze literatury. Słownik biobibliograficzny."

The research has been supported by grant N N206 400234 from Ministry of Science and Higher Education of the Republic of Poland.

References

Paul BUITELAAR, Michael SINTEK, and Malte KIESEL (2005), Feature Representation for Cross-Lingual, Cross-Media Semantic Web Applications, in *Proc. of the Workshop on Knowledge Markup and Semantic Annotation (SemAnnot2005) at the International Semantic Web Conference*, Galway, Ireland.

J. CZACHOWSKA and A. SZAŁAGAN, editors (1994–2008), *Współcześni polscy pisarze i badacze literatury. Słownik biobibliograficzny. [Contemporary Polish Writers and Literary Scholars. Biobibliographical Lexicon]*, Wyd. Szkolne i Pedagogiczne, last volumes Wyd. IBL PAN.

J. EARLEY (1986), An Efficient Context-Free Parsing Algorithm, *Communications of the ACM*, 6(8):451–455.

E. MENDELSON (1997), *Introduction to Mathematical Logic*, International Thomson Publishing, 4th edition.

Alistair MILES, Brian MATTHEWS, Michael WILSON, and Dan BRICKLEY (2005), SKOS core: simple knowledge organisation for the web, in *Proc. of the 2005 international conference on Dublin Core and metadata applications (DCMI '05)*, pp. 1–9, Dublin Core Metadata Initiative.

Agnieszka MYKOWIECKA (2007), *Inżynieria lingwistyczna. Komputerowe przetwarzanie tekstów w języku naturalnym [Linguistic Engineering. Computer processing of texts in natural language]*, Polsko-Japońska Wyższa Szkoła Technik Komputerowych.

Agnieszka MYKOWIECKA, Anna KUPŚĆ, and Małgorzata MARCINIAK (2005), Rule-Based Medical Content Extraction and Classification, in *Proc. of the Intelligent Information Processing and Web Mining Conference (IIS:IIPWM'05)*, pp. 237–246, COLING, Gdańsk, Poland.

Adam PRZEPIÓRKOWSKI, Anna KUPŚĆ, Małgorzata MARCINIAK, and Agnieszka MYKOWIECKA (2002), *Formalny opis języka polskiego. Teoria i implementacja [A formal description of the Polish language. Theory and implementation]*, Akademicka Oficyna Wydawnicza Exit, Warszawa.

Zygmunt VETULANI (2004), *Komunikacja człowieka z maszyną. Komputerowe modelowanie kompetencji językowej [Man-machine communication. Computer modeling of linguistic competence]*, Akademicka Oficyna Wydawnicza Exit, Warszawa.

Marcin WOLIŃSKI (2006), Morfeusz — a Practical Tool for the Morphological Analysis of Polish, in Mieczysław KŁOPOTEK, Sławomir WIERZCHOŃ, and Krzysztof TROJANOWSKI, editors, *Proc. of Intelligent Information Processing and Web Mining (IIS:IIPWM'06)*, pp. 503–512, Springer.