Warsaw University
Faculty of Mathematics, Informatics and Mechanics

Wojciech Jaworski

Ontology-Based Knowledge Discovery from Documents in Natural Language

PhD dissertation

Supervisors

dr hab. Jerzy Tyszkiewicz
Institute of Informatics
Warsaw University

dr hab. Marek Stępień
Institute of History
Warsaw University

December 2008

Author's declaration

Aware of legal responsibility. I hereby declare that I have written this dissertation myself and all the contents of the dissertation have been obtained by legal means.

. . . .
date

. . . .
Wojciech Jaworski

Supervisors' declaration

The dissertation is ready to be reviewed.

. . . .
date

. . . .
dr hab. Jerzy Tyszkiewicz

. . . .
date

. . . .
dr hab. Marek Stępień

Abstract

A vast amount of knowledge is contained in large collections of unstructured or weakly structured text documents, which started emerging soon after discovery of writing. The utility of such document collections depends on the ability of finding the relevant information. Users seek not only for information localised in specific documents but also knowledge spread across the whole document collection. Queries may concern both facts explicitly stated in documents and general principles that govern that facts. These principles are discovered by means of inductive reasoning and, in general they are correct only with a certain probability, which should be estimated.

In this dissertation, we develop a methodology, which implements the above functionality for document corpora with restricted subject-matter. Our approach is based on the assumption that we are interested in information concerning a specific domain, for which we are able to develop an ontology. We perform profound analysis of text structure suggested by theoretical linguistic which results in translation of the documents contents into the formulae of the meaning representation language. The system is augmented with the capability of processing documents whose parts describe concepts not included in the ontology and grammar. We introduce also special methodology for dealing with massive ambiguity that arises during processing textual data.

We develop a knowledge representation model which allows us to manipulate on concepts included in data by means of logic as well as answering semantic queries. We define the semantics of the meaning representation language in a model-theoretic fashion and we investigate the reference between the formula and described reality. We also define what is an ontology in a formal way and explore its role in the reference problem.

In order to infer knowledge we use machine learning techniques. We have chosen the rough set theory as a theoretical framework of our research. However, we to make it making it compatible with our knowledge representation model. We combine the rough set theory with statistical learning theory obtaining simple, human understandable classifiers, whose quality will be guaranteed by the statistical assumptions.

We describe the practical application of our system by performing experiments in two real-world, restricted domains: the Ur III Economic Text Corpus and Biobibliographical Lexicon of Polish Writers and Literary Scholars.

Keywords: artificial intelligence, natural language processing, natural language modelling, natural language understanding, semantic parsing, computational semantics, knowledge representation, data analysis, information system, rough sets, indiscernibility relation, quality measures, rule induction, missing values, multiple valued attributes

ACM Computing Classification System: I.2.7, I.2.4, I.2.6, I.5.1, I.5.2

Streszczenie

Upowszechnienie komputerowych metod przetwarzania informacji, a zwłaszcza komputeryzacja zasobów wiedzy, doprowadziły do powstania bogatych korpusów tekstów w systemach komputerowych. Aby korzystać z takich zbiorów dokumentów, potrzebne są metody wyszukiwania potrzebnej informacji oraz wnioskowania na jej podstawie. Poszukiwane są nie tylko informacje zlokalizowane w pojedynczych dokumentach lecz również te, które są rozproszone po całym korpusie. Zapytania kierowane do tych korpusów mogą dotyczyć zarówno konkretnych faktów zawartych w tekstach, jak i ogólnych praw rządzących tymi faktami. Prawa te są formułowane w wyniku wnioskowania indukcyjnego (bazującego na generalizacji) i na ogół są prawdziwe jedynie z pewnym prawdopodobieństwem, którego wielkość trzeba oszacować.

W niniejszej pracy rozwijamy metodologię realizującą powyższą funkcjonalność dla zbiorów tekstów o ograniczonej dziedzinie tematycznej. Założenie to pozwala reprezentować za pomocą ontologii strukturę informacji zawartej w tekstach. Stosujemy techniki głębokiego przetwarzania języka umożliwiające odwzorowanie znaczenia wypowiedzeń w języku naturalnym w formuły języka reprezentacji znaczenia. Pokazujemy również w jaki sposób można analizować teksty których fragmenty zawierają pojęcia nie należące do ontologii. Wprowadziliśmy metodologię, która pozwala w zwarty sposób reprezentować wszystkie możliwe interpretacje niejednoznacznego dokumentu.

Korzystamy z pojęć teorii modeli, aby formalnie zdefiniować semantykę języka reprezentacji znaczenia, co umożliwia to operowanie na pojęciach zawartych w tekstach za pomocą logiki oraz odpowiadanie na zapytania semantyczne.

Aby wnioskować indukcyjnie i dokonywać generalizacji informacji zawartych w tekstach korzystamy z teorii zbiorów przybliżonych, którą rozszerzamy na przypadek danych zapisanych za pomocą formuł języka reprezentacji znaczenia. Teorie tę łączymy ze statystyczną teorią uczenia się, aby skonstruować klasyfikator, którego jakość będzie gwarantowana przez założenia statystyczne.

Opisaną w pracy metodologię przetestowaliśmy na dwu korpusach: korpusie sumeryjskich tekstów gospodarczych z okresu III dynastii z Ur oraz słowniku biobibliograficznym współczesnych polskich pisarzy i badaczy literatury.

Słowa kluczowe: sztuczna inteligencja, przetwarzanie języka naturalnego, natural language modeling, natural language understanding, semantic parsing, computational semantics, reprezentacja wiedzy, analiza danych, system informacyjny, zbiory przybliżone, relacja nierozróżnialności, miary jakości, indukcja reguł, brakujące wartości, wielowartościowe atrybuty

ACM Computing Classification System: I.2.7, I.2.4, I.2.6, I.5.1, I.5.2

Contents

List of Tables

List of Figures

Acknowledgements

First and foremost, I thank my supervisors, Jerzy Tyszkiewicz and Marek Stępień, for their guidance and help throughout my Ph.D. process. I am deeply grateful for their inspiration in my work, help and advice, support and comments on this thesis. I am also very indebted to them for helpful discussion and encouragement throughout the years.

I would like to thank Andrzej Skowron for comprehensive support and comments during the writing of this dissertation.

I am very indebted to Bogdan Dziobkowski for discussions concerning philosophical aspects of language.

I thank also 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 grants N N206 400234 and N N516 368334 from Ministry of Science and Higher Education of the Republic of Poland.

Chapter 1

Introduction

1.1 Motivation

A vast amount of knowledge is contained in large collections of unstructured or weakly structured text documents, which started to emerge soon after the discovery of writing. The utility of such document collections depends on the ability of finding the relevant information. Users seek not only for information localised in specific documents but also knowledge spread across the whole document collection. Queries may concern both facts explicitly stated in documents and general principles that govern that facts. These principles are discovered by means of inductive reasoning and, in general they are correct only with a certain probability, which should be estimated.

In this dissertation, we develop a methodology, which makes it possible to implement in computers the above functionality for document corpora with restricted subject-matter. This assumption allows us to represent the structure of information enclosed in documents by means of ontology.

1.2 Available solutions for structured data

The issue of the retrieval of relevant data was thoroughly investigated in context of relational databases [Codd, 1970]. The introduced solutions are based on a data model which is a formal description of the structure of data. This structure is usually expressed by means of an entity relationship diagram. Similarly, for structured data, represented by means of attribute value vectors, there exist techniques of generalisation and approximate reasoning [Cichosz, 2000, Hastie et al., 2001, Mitchell, 1997, Pawlak, 1981].

The data model has two functions: it points out where the relevant data are located in the database and it allows one to extract the information enclosed in data for further processing.

1.3 The state-of-art in natural language processing

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 modelling this structure. However, the more precise is the model, the more knowledge must be introduced. This property restricts the scope of application of sophisticated models.

For example, introduction of simple bag-of-words language model [Feldman and Sanger, 2006] (applied, eg. in Internet search engines) does not require any language or domain knowledge. This solution allows us to find relevant documents, but it does not allow to infer conclusions, nor to integrate retrieved information.

Information Extraction [Moens, 2006] realises simple task of structured information retrieval from texts. However, its methodology is based on shallow parsing and thus it is unable to express ambiguity. As a consequence it is not scalable to the case when the information has complex structure.

On the other hand, models that precisely describe document contents are based on analysis of a single semantic phenomenon (ex. representation of quantification, plural, or relative clauses) [Blackburn and Bos, 2005, Charniak and Wilks, 1976, Dowty et al., 1981, Montague, 1970b] or they are developed using small sets of documents containing less than 1000 sentences [Androutsopoulos et al., 1995, Ge and Mooney, 2005, Popescu et al., 2003, Zettlemoyer and Collins, 2005].

There were also efforts to translate document corpora into logic formulae by means of linguistic knowledge only, without use of the domain knowledge concerning relations between concepts described in the texts [Bos, 2005, Bos et al., 2004, Crouch, 2005]. The so obtained logical formulae reflect the document contents, yet they do not express the structure of information in a computationally transparent way. For example they do not allow us to identify sentences that have the same meaning but differ in form in which they express this meaning.

The above results show that it is essential to develop a structure for information enclosed in natural language documents.

1.4 Thesis contributions

In this dissertation we pursue a the goal of grasping the whole process of knowledge extraction from textual 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 an 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.

This approach, applied in artificial intelligence and data mining, has not been used in natural language processing so far.

We store the information extracted from documents as a set of meaning representation language formulae. 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 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 formally define the semantics of the meaning representation language by means of model theory. The information enclosed in documents is incomplete and imprecise. That is why we express document semantics as a class of possible worlds — models consistent with its content. In this setting, we study the problem of reference between language and reality: texts describe real objects and these objects constitute possible world universes. As a consequence properties like ontological category membership are defined in the same way in each possible world.

For converting document contents into meaning representation language formulae, we use similar methodology to the one provided by Computational Linguistics [Jurafsky and Martin, 2000]. However, we adapt it to a situation in which the ontology is available.

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 methodology presented in this dissertation 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.

A semantic value is associated with each grammar symbol during parsing process. This semantic value is a meaning representation language formula which describes the semantics of the text parsed 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 develop also query language that allows us to retrieve information according to semantic patterns.

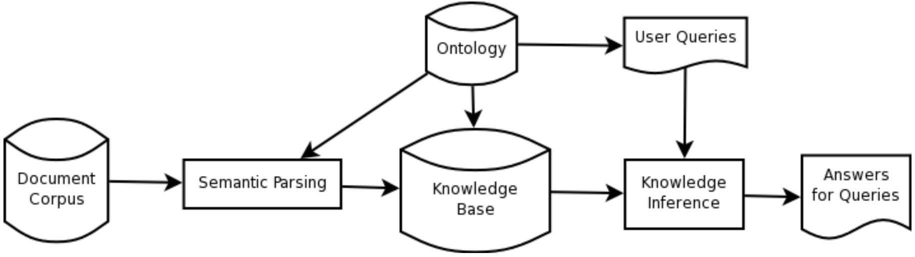
In the next step, we concentrate our efforts on the problem of inference. The ontology splits objects into categories and defines relations between these categories. Now, we look for data–induced dependencies between properties of objects belonging to certain categories. We choose the rough set theory [Pawlak, 1982, Pawlak, 1991] as a theoretical framework of our research. It provides us with tools for defining dependencies between objects in a human readable form, and in case when they are indefinable it allows us to approximate them. However, we modify the theory making it compatible with our knowledge representation model. We redefine the main concepts of the rough set theory and then we prove that we obtain generalisations of the original ideas.

The discovered dependencies represent general principles that govern objects in the world described by documents. However, available data is often incomplete. We introduce a probabilistic model of data generation process suggested by the statistical learning theory. The model allows us to explain the process of data acquisition and estimate accuracy and coverage of discovered facts.

The combination of the rough set theory and statistical learning theory provides us with tools for building simple, human understandable classifiers, whose quality will be guaranteed by the statistical assumptions. We obtain the theory that is capable of inferring knowledge from logical representations of natural language expressions, which are ambiguous and incomplete.

We demonstrate the effectiveness and universality of our system by performing experiments in two real-world, restricted domains: the Ur III Economic Text Corpus [CDLI, 2008] and Biobibliographical Lexicon of Polish Writers and Literary Scholars [Czachowska and Szałagan, 2008]. The fact that the only parts of the system that differ in the two experiments are domain dependent ontologies and language dependent syntactic analysers proves the universality of our approach. Experiments have shown, that the introduction of ontologies can

Figure 1.1: System framework



fill the gap between surface and deep processing of language. We also present results of experiments based on generalisations of the structured information extracted from documents.

Our results were a subject of two papers presented on the 52nd and 54th *Rencontre Assyriologique Internationale* [Jaworski, 2007, Jaworski, 2009]. Our findings have been positively assessed by experts.

We note that parts of the material presented in this thesis appeared in [Jaworski, 2005, Jaworski, 2006a, Jaworski, 2006b, Jaworski, 2007, Jaworski, 2008b, Jaworski, 2008a, Jaworski, 2008c].

1.5 Thesis outline

Below is a summary of the remaining chapters of this thesis:

- In Chapter 2, we describe the application domains which we utilised in our research.
- In Chapter 3, we introduce a methodology for semantic parsing. We present ontologies for our application domains and meaning representation language as well as grammars and parsing algorithms. We show also how to deal with ambiguity, damaged documents and incomplete ontology.
- In Chapter 4, we introduce the knowledge representation model, in which we define the semantics of the meaning representation language. We consider the problem of how that information is encoded into the language, and we investigate the reference between its formulae and described reality.
- In Chapter 5, we infer knowledge from logical representations of our document corpora. For that purpose, we combine the rough set theory with statistical learning theory.
- In Chapter 6, we show some preliminary results of application methodology presented in the dissertation to the experimental data.
- In Chapter 7, we conclude this thesis.

Chapter 2

Application domains

2.1 Ur III administrative text corpus

Sumerians lived from prehistoric times until late 3rd millennium BC in lower Mesopotamia (modern Iraq). Sumer was the first highly developed urban civilisation, which used cuneiform script. During the reign of the 3rd dynasty of Ur (2100 BC-2000 BC), whose power extended as far as present Iraq and western Iran, the state introduced a centrally planned economy with an extensive bureaucratic apparatus.

Civil servants reported on clay tablets agriculture and factory production, lists of worker salaries, summaries of executed work, distribution of commodities, goods, animals etc., lists of sacrificed animals, travel diets and other economical information.

Documents were written on wet clay with a sharpened stick, or stylus. Clay tablets would either be dried in the sun or fired in kilns to make the writing permanent.

Sumerian writing system was like a rebus, without punctuation signs, also with large amount of polyvalential and homophonic signs. According to the rule of polyvalence one sign could, depending on the context, be read differently. For example the `DU` sign meaning — a leg, could have also other translations like: `gin` — to go, `gub` — to stand straight, `tum2` — to bring, etc. According to the rule of homophony, one sound could be written differently, taking different meanings. The `gu` syllable had fourteen adequate graphic signs including: flax (`gu` reading), neck (`gu2`), voice (`gu3`), ox (`gu4`).

The signs were divided in groups: word-signs (ideograms), syllable signs, determinative-signs (defining category of the meaning), numeral-signs and not numerous letter-signs and phonetic complements. Determinatives were additional clues simplifying the process of reading. For example names of wooden items were preceded by prefix `gisz` (meaning tree or wood), items made of copper — `urudu`. Names of cities were preceded by `uru` or succeeded `ki` (or both of them).

Phonetical values of signs are most often single syllables. Simple words are written by single ideograms equipped with phonetically written affixes. More complicated terms can be represented by combinations of multiple signs. Due to lack of mid-word signs and agglutinative character of the Sumerian language, it is hard to identify if we read a word or a phrase, and define its borders in the text. For more information about the Sumerian language see [Labat and Malbran-Labat, 1988].

Archaeologists dug out about 100000 of tablets from that period (known also as the Neo-Sumerian period). The corpus of 44365 tablets (as of March 26, 2008) is available in electronic version in the database of Cuneiform Digital Library Initiative [CDLI, 2008], run by the University of California in Los Angeles and Max Planck Institute for the History of Science. Tablets are stored in the form of Latin transliteration, often accompanied by photographs or drawings (which are irrelevant for us).

Due to the script's polyvalence, the transliteration process includes interpretation of table's content. Tables were transliterated over decades by different translators. During those years the knowledge about meaning of each of the phrases and their interpretation has been changing. As a result, the same word, written with a few signs may be translated up to dozen or so, different translations. In addition the meaning of some phrases still remains unclear.

The fact that administrative documents are written in bureaucratic language which is full of notation abbreviations, lacks most of grammatical affixes and does not have uniform spelling for non-Sumerian words, makes the situation even more complicated.

Many symbols in the transliteration are used to describe the spatial organization of signs. It concerns for example signs written one inside of another, compound (stuck together) or written one under the other.

Texts are often physically damaged. Damages can concern one or many following signs. Sometimes many following lines are illegible. As a result comes full or partial unreadability of that fragment or uncertainty of correctness of its interpretation. In the transliterations damages are marked by special symbols or descriptions in English, German, French, etc., depending on translator's nationality.

Markings occurring in the texts are not used in any homogeneous way. For example information about the damage can be written as one of many possible comments in English: ”1-3 lines missing”, ”a few lines missing”, ”several lines broken”, ”about six lines missing”, ”1-3 lines lost”, ”some lines missing”, ”3-4 ll. missing”, ”ca. 10 lines missing” and so on.

Texts contains also lots of editors' records such as marks of sides of the tablet, place of embossing of the seal, intrusions in English, German and French or repetitions of Sumerian signs in other languages.

The other problem is that the Sumerian texts themselves are often ambiguous on both syntactic (script's polyvalence, lack of most of inflection) and semantic level. For example, the phrase `mu en {d}nanna` can be interpreted as *In the name of high priest of god Nanna* or it can be translated as *Year when high priest of god Nanna*. In the first case it represents a person who accom-

Figure 2.1: An example of a transliterated cuneiform tablet from Ur III

```
&P123831 = OIP 121, 101
tablet
obverse
1.  1(disz) sila4 ur-mes ensi2         1 lamb ur-mes governor
2.  1(disz)# sila4 da-da dumu lugal    1 lamb da-da son of king
3.  1(disz)# sila4 ga-ga-mu            1 milky lamb ga-mu or 1 lamb ga-ga-mu
reverse
1.  u4 2(u) 3(asz@t)-kam               Day 23
$ 1 line blank
3.  mu-DU                              delivery
4.  ab-ba-sa6-ga i3-dab5               ab-ba-sa6-ga received
5.  iti sze-KIN-ku5                    month sze-kin-ku5
6.  mu en {d}inanna ba-hun             Year when high priest of goddess Innana
left                                                        was elevated to office
1.  3(disz)                            3
```

plishes some role in the document. In the second it is a popular shortcut for year name `mu en {d}nanna ba-hun` and represents a date.

Mosts of texts come from the 40 year long period of time and origin in four cities. For our studies, we have selected a subcorpus of 11891 documents concerning distribution of domestic animals. This subcorpus consists of 848329 Sumerian signs.

Fig. 2.1 presents the contents of a typical Sumerian document. This documents reports the transfers of lambs from 3 people to `ab-ba-sa6-ga`, an official of the Ur III state. They took place on the 23th day of the month `sze-kin-ku5` in the year when the high priest of goddess Innana was elevated to office. The third verse of the document is ambiguous. We cannot determine whether `ga` is a part of a name or a part of an animal description.

Economic documents are an essential source of information about ancient Sumer. The corpus contains crucial information about economic, social and political history of the state, as well as its political system and administration structure. The sources of this type provide the most complete information about the daily life of those days. For more information about the economy of the Ur III kingdom see for example [Stępień, 1996, Stępień, 2006, Sharlach, 2004, Steinkeller, 1987].

2.2 A Biobibliographical Lexicon

"Współcześni polscy pisarze i badacze literatury. Słownik biobibliograficzny." [Contemporary Polish Writers and Literary Scholars. Biobibliographical Lexicon.] [Czachowska and Szałagan, 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 theatre critics, essayists, translators, historians and theorists of literature. The lexicon covers

Figure 2.2: An example of a sentence from the Biobibliographical Lexicon of Polish Writers and Literary Scholars

`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.*

writers and literary scholars who made their debuts after the year 1918 and had at least three books or dramatic works issued before 1988.

In addition to new entries, the 10th volume encloses the completion of the current biobibliographical information of previously published lexicon entries.

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, organisations' membership, place of residence, characteristics of writings and places of publication.

Biobibliographical Lexicon is written in Polish language, which is an Slavic language with rich inflection and free word order. 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. The Lexicon consists of 806721 words and 45259 sentences. Fig. 2.2 presents an example of a sentence from the biobibliographical lexicon.

Chapter 3

Semantic parsing

3.1 Background

Semantic parsing is the task of mapping natural language sentences into their computer understandable meaning representations. These meaning representations are expressed in formal languages which we call meaning representation languages.

Semantic parsing is a research area with a long history. Many early semantic parsers are natural language interfaces to databases, including LUNAR [Woods et al., 1972], CHAT-80 [Warren and Pereira, 1982], and TINA [Seneff, 1992].

Further research has focused on a few application domains which have been manually annotated and a semantic parsers have been used to transform natural language queries into formal database queries. Three most popular domains are:

- ATIS (Air Travel Information Service) [Price, 1990]; a simple domain whose semantic analysis is equivalent to filling a single semantic frame [Kuhn and de Mori, 1995, Miller et al., 1996, Popescu et al., 2004].

- GEOQUERY: The aim of this domain is to develop a natural language interface to the U.S. geography database [Zelle and Mooney, 1996, Kate et al., 2005]. Geoquery consists of 880 sentences. The average length of a sentence in this corpus is 7.48 words.

- CLANG: The RoboCup Coach Language [Chen et al., 2003]. RoboCup is an international AI research initiative using robotic soccer as its primary domain. A semantic parser is used in this domain as an interpreter for coaching instructions. CLANG was constructed by randomly selecting 300 pieces of coaching advice. On average there are 22.5 words per sentence.

The above databases are very small both in number of sentences and number of words per sentence.

Recently, there have been also efforts to convert large open domain document corpora into logical form [Bos et al., 2004, Bos, 2005, Curran et al., 2007, Crouch, 2005].

Systems for semantic parsing are manually-built [Hendrix et al., 1978, Androutsopoulos et al., 1995, Popescu et al., 2003] or use statistical learning methods [Miller et al., 1996, Ramaswamy and Kleindienst, 2000, Ge and Mooney, 2005, Zettlemoyer and Collins, 2005, Wong and Mooney, 2006]. However, statistical learning requires prior corpus annotation, which is very difficult to obtain. That is why we have decided to develop a system for lexicon and grammar acquisition instead of learning them.

Some systems use syntactic parse trees augmented with semantic features [Miller et al., 1996, Ge and Mooney, 2005, Nguyen et al., 2006, Zettlemoyer and Collins, 2005], while others remove syntactical labels from parse trees. The latter results in a semantic grammar [Allen, 1995, Kate et al., 2005], in which nonterminal symbols correspond to domain-specific concepts as opposed to syntactic categories.

Our research indicates that none of these approaches is superior to another and features should be chosen according to the corpus characteristics. We use both syntactic and semantic features while parsing a Biobibliographical Lexicon because of its rich syntactic information. On the other hand Sumerian Economic Documents are parsed using pure semantic grammar.

For a Biobilbliographical Lexicon we develop two independent grammars. One of is a semantic grammar and the other describes syntactic structure of documents. Such an approach guarantees scalability of the system and provides convenient representation for phenomena such as event nominalizations which are nouns that semantically play a role of verbs.

Grammars and grammar formalisms used by semantic parsing systems have been developed for English language. Our document corpora are written in Polish and Sumerian, respectively, which are very different from English. Thus, we choose suitable grammar formalism and we build grammar for each of our domains. For Sumerian economic documents we use grammars with the expressive power equivalent to regular languages [Hopcroft and Ullman, 1979] and for the Polish corpus we use context-free grammar whose rules are augmented with constraints [Chomsky, 1965] . Our grammars should be considered shallow because they are generated with the purpose of recognising syntactic constructions in a certain text corpora. However, they generate complete parse trees, which is a property of deep parsing.

Since our work is the first approach to Sumerian language from the computational linguistics perspective there isn't any related work. There are several papers devoted to deep processing of Polish language [Obrębski, 2002, Woliński, 2004, Przepiórkowski et al., 2002, Vetulani, 2004, Mykowiecka, 2007]. Recently also a monography devoted to shallow processing has been published [Przepiórkowski, 2008].

For Sumerian economic documents we develop an efficient parsing algorithm. For the Polish corpus we use chart parser [Earley, 1986] and also morphological analyser Morfeusz [Woliński, 2006].

Formal semantic analysis of natural languages typically uses predicate logic as the meaning representation language [Montague, 1970b, Dowty et al., 1981, Charniak and Wilks, 1976, Blackburn and Bos, 2005]. There are many different kinds of predicate logic that deal with different linguistic phenomena such as quantification, modality, underspecification, and discourse.

However, among the practical applications the following logic forms are found:

- Discourse Representation Structures [Kamp and Reyle, 1993]: a fist-order language used by [Bos, 2005] and [Curran et al., 2007]
- ATIS domain is annotated with SQL queries.
- The GEOQUERY language consists of Prolog queries augmented with several meta-predicates for dealing with quantification.
- CLANG is a variable-free, functional language that includes negation and quantifiers like "all" are included in the language.
- On the other hand, the approach taken by [Zettlemoyer and Collins, 2005] is to map natural language sentences to a lambda calculus encoding of their semantics.

In our system we use propositional logic as the meaning representation language.

One of the key characteristics of natural language is its ambiguity. Although semantic representation for ambiguity has been studied in theory [Copestake et al., 1999, Crouch, 2005, Reyle, 1993, Richter and Sailer, 1999], it hasn't been applied in practice. Systems assume that the grammar is unambiguous [Wong and Mooney, 2006], select the most probable parse trees using statistical methods [Bos, 2005] or parse only sentences which have only one semantic interpretation [Popescu et al., 2003, Popescu et al., 2004].

We develop a framework for working with ambiguous semantic representation. We distribute the meaning representation language formula across the parse tree obtaining compact representation of all the possible document interpretations.

The problem of analysing damaged documents, which is related to ambiguity, has not yet been analysed by computational linguists.

Ontologies have been thoroughly studied in contexts of artificial intelligence and machine learning [Russell and Norvig, 1995, Staab and Studer, 2004, Bazan et al., 2004, Gruber, 1993, Skowron and Stepaniuk, 1999, Nguyen et al., 2004, Sowa, 1999], however, semantic parsing systems do not encode domain knowledge in a form of an ontology. Broad coverage systems [Bos et al., 2004, Bos, 2005, Crouch, 2005] incorporate lexical semantic resources such as WordNet [Miller et al., 1990], but the obtained ontology is not sufficient to define the structure of objects.

Conversely, ontology that encodes the domain structure is one of the central ideas in our approach. The ontology determines the structure of the knowledge base, which we use for storing information extracted from documents. It also defines the components of the meaning representation language, which we use for representing data in the knowledge base. Introduction of ontology allows us to balance between flat and nested semantic representation. Ontology defines semantic constraints which reduce ambiguity of syntactic parse trees. Finally, ontology provides us with a structure which is necessary to infer knowledge out of meaning representation language formulae.

3.2 Ontology

An ontology is a formal representation of a set of concepts within a domain and the relationships between those concepts. It defines structure of the domain.

For a given domain an ontology arranges entities (or objects) that compose that domain. By the notion of concept we understand a unary relation that groups individuals possessing some common property.

Structural objects are composed out of subobjects connected by relational constraints. These constraints may be projected on the concept level metarelations that represent dependencies between concepts.

The key metarelation 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 metarelations such as *subsumption relation*: *is a subtype of*, useful in large ontologies, may be also added.

The ontology may be acquired from an expert or constructed during system development process. In our case ontologies were constructed in dialogue with experts, however, their final shape was determined by the assumption of compositionality. Experts suggested which object properties are relevant for the further applications. These properties were selected as ontological concepts, yet the dependencies between concepts were determined by the document structure

We will explore further the ontology issue in chapter 4.

3.2.1 Ontology of Neo-Sumerian economic documents

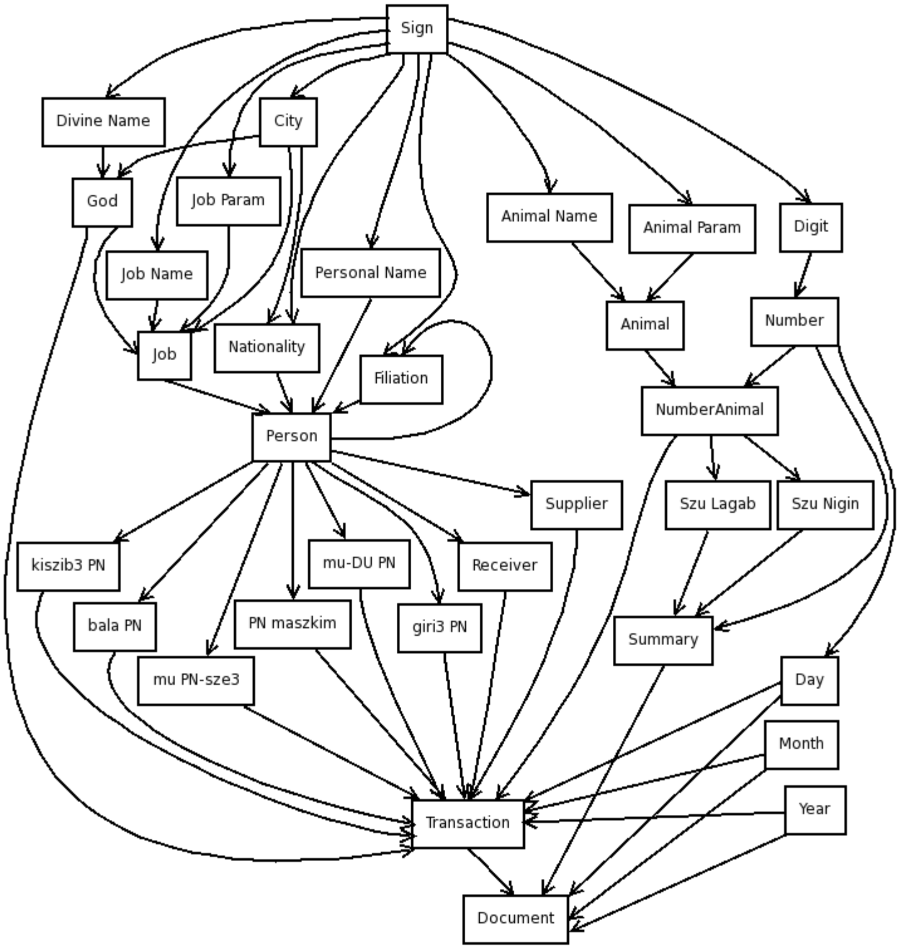
As we mentioned above, we examine the subcorpus of documents concerning distribution of breeding animals. The distribution was organised in the form of transactions. During each **Transaction** one **Person**, called **Supplier**, transfers a **Number** of animals to another Person, called **Receiver**. **Animal** description consists of information like: species, age, gender, fodder, etc. Person is described by means of his/her **Name**, **Filiation**, **Job** and/or **Nationality**. Apart from the Supplier and Receiver, other Persons might have assisted in the transaction:

giri3 PN Middleman between Supplier and and Receiver

mu-DU PN Person on whose account the transaction took place, Receiver was probably Mu Du's representative

mu PN-sze3 Person in whose name the Receiver or Supplier acted.

Figure 3.1: Ontology of Sumerian economic text corpus



kiszib3 PN Person who sealed the document

PN maszkim Overseer of the transaction

bala PN Person who provided goods as royal tax.

The roles are named after the Sumerian phrases used to introduce them (**PN** stands for **Personal Name**), their meaning is still studied by sumerologists. One of the applications of our research is to provide more facts regarding their meaning.

There are a few kinds of **Summaries** of various animal types in the documents, denoted by concepts: **Szu Lagab**, **Szu Nigin** as well as single **Number**. Dates of transactions are also provided.

Some **Documents** summarise transactions that took place in a given time interval. In such a case document's date provides additional information about the transaction dates: it assures that the document describes all the transactions that took place during that time interval.

The **Sign** category includes all Sumerian signs. It is used to describe relations between objects and signs.

The dependencies between concepts have meronymical nature. They witness the fact that one object is a part, property, attribute or description of another one. Fig. 3.1 presents the diagram of dependencies between concepts.

3.2.2 Ontology of Biobibliographical Lexicon of Polish Writers

Information in Biobibliographical Lexicon of Polish Writers and Literary Scholars is organised by means of events (see fig. 3.2 and 3.3). 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.

Figure 3.2: Ontology of Biobibliographical Lexicon of Polish Writers and Literary Scholars — events, part 1

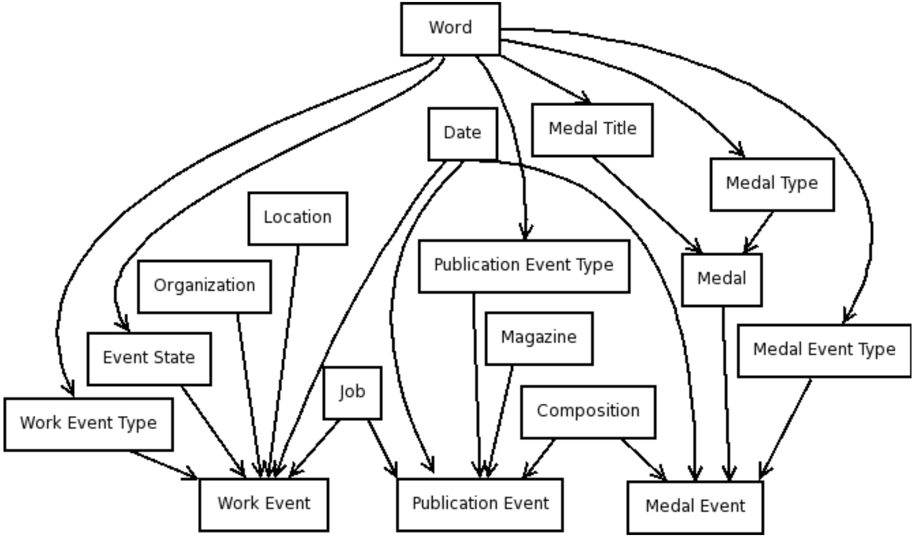


Figure 3.3: Ontology of Biobibliographical Lexicon of Polish Writers and Literary Scholars — events, part 2

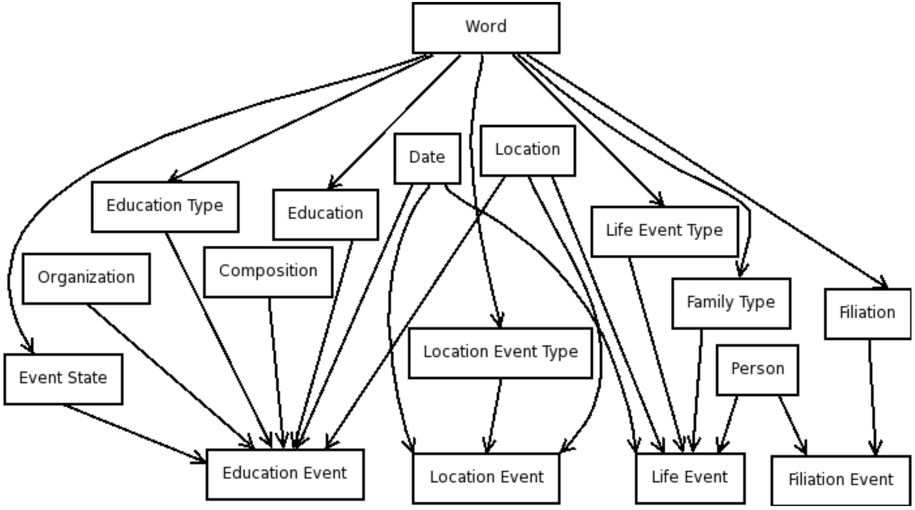
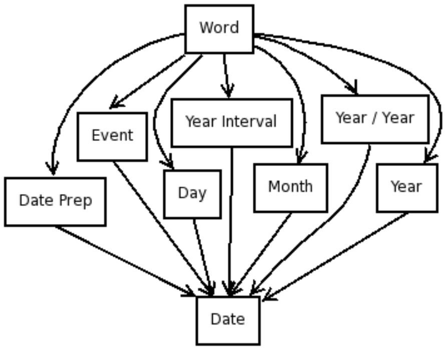


Figure 3.4: Ontology of Biobibliographical Lexicon of Polish Writers and Literary Scholars — dates



"Event types" store the word used to introduce event in a sentence. They are obligatory parameters of events and they determine its appearance. Some events are parametrised by **Event States**, which are verbs such as 'begin', 'continue', 'be'.

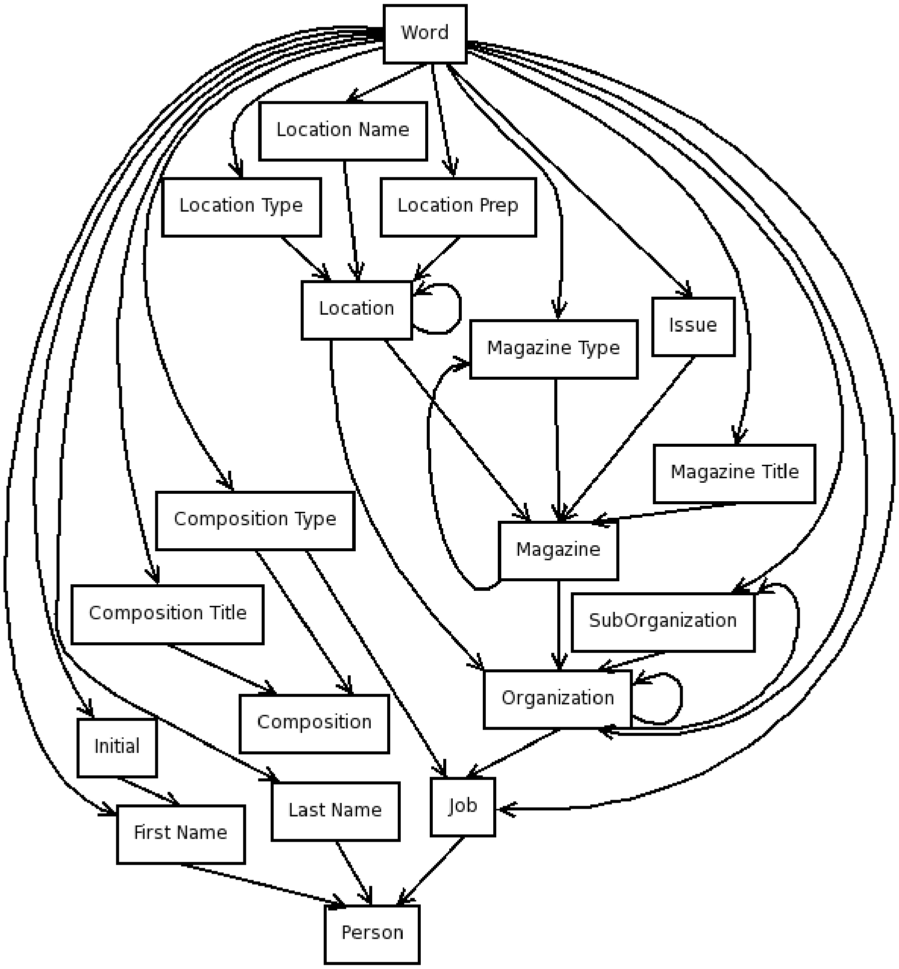
Event parameters may have internal structure (see fig. 3.4 and 3.5):

- **Date** is composed out of **Day**, **Month**, **Year** or another **Event** which determines them. It is also equipped with a **Date Prep** which describes the time interval.
- **Job** includes ordinary jobs as well as types of membership, cooperation, and writing activity.
- **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**.

The **Word** category includes all words in documents. It denotes relations between objects and words.

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.

Figure 3.5: Ontology of Biobibliographical Lexicon of Polish Writers and Literary Scholars



3.3 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. We will discuss relevance of this approach in chapter 4.

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

Signs and **Words**, which are the most primitive ontological concepts are treated in a special way: each word (or sign) is represented by a constant whose name is identical to that word (sign). If an object is described in the text by a sequence of words (signs) then this description is represented by one constant named by concatenation of these words (signs).

Fig. 3.6 provides our example Sumerian tablet written in formal language. `Number`, `Animal`, `Name`, `Person`, `Receiver`, `Year`, `Document`, etc. denote corresponding ontological categories. $q_1, q_2, a_1, m_1, del_1$, etc. represent objects described in the text.

For example:

- $\texttt{Day}(d_1, \texttt{23})$ means that d_1 is a **Day** and its number is 23.
- $\texttt{Name}(n_1, \texttt{ur-mes})$ means that n_1 is a **Name** written as `ur-mes`.
- $\texttt{Job}(j_1, \texttt{ensi2})$ means that j_1 is a **Job** called `ensi2`.
- $\texttt{Person}(p_1, n_1, j_1)$ means that p_1 is a **Person** described by n_1 and j_1.
- $\texttt{Name}(n_1, \texttt{ur-mes}) \wedge \texttt{Job}(j_1, \texttt{ensi2}) \wedge \texttt{Person}(p_1, n_1, j_1)$ means that n_1 is a **Name** written as `ur-mes` and j_1 is a **Job** called `ensi2` and p_1 is a **Person** described by n_1 and j_1.

Figure 3.6: The semantics of text. Observe the use of domain knowledge and the representation of ambiguous semantics for year.

$$\begin{array}{l}
\texttt{Number}(q_1, \texttt{1}) \wedge \texttt{Animal}(a_1, \texttt{sila4}) \wedge \texttt{Name}(n_1, \texttt{ur-mes}) \wedge \texttt{Job}(j_1, \texttt{ensi2}) \wedge \\
\qquad \wedge \texttt{Person}(p_1, n_1, j_1) \wedge \texttt{Supplier}(s_1, p_1) \wedge \\
\wedge \texttt{Number}(q_2, \texttt{1}) \wedge \texttt{Animal}(a_2, \texttt{sila4}) \wedge \texttt{Name}(n_2, \texttt{da-da}) \wedge \texttt{Job}(j_2, \texttt{lugal}) \wedge \\
\qquad \wedge \texttt{Person}(p_2, j_2) \wedge \texttt{Filiation}(f_1, \texttt{dumu}, p_2) \wedge \texttt{Person}(p_3, n_2, f_1) \wedge \texttt{Supplier}(s_2, p_3) \wedge \\
\wedge \texttt{Number}(q_3, \texttt{1}) \wedge \big(\big(\texttt{Animal}(a_3, \texttt{sila4}) \wedge \texttt{Name}(n_3, \texttt{ga-ga-mu}) \wedge \\
\qquad \wedge \texttt{Person}(p_4, n_3) \wedge \texttt{Supplier}(s_3, p_4) \wedge \texttt{Transaction}(t_3, q_3, a_3, s_3, r_1, d_1, month_1, y_1, del_1)\big) \vee \\
\qquad \vee \big(\texttt{Animal}(a_4, \texttt{sila4 ga}) \wedge \texttt{Name}(n_4, \texttt{ga-mu}) \wedge \\
\qquad \wedge \texttt{Person}(p_5, n_4) \wedge \texttt{Supplier}(s_4, p_5) \wedge \texttt{Transaction}(t_3, q_3, a_4, s_4, r_1, d_1, month_1, y_1, del_1)\big)\big) \wedge \\
\wedge \texttt{Day}(d_1, \texttt{23}) \wedge \\
\wedge \texttt{Delivery}(del_1) \wedge \\
\wedge \texttt{Name}(n_5, \texttt{ab-ba-sa6-ga}) \wedge \texttt{Person}(p_6, n_5) \wedge \texttt{Receiver}(r_1, p_6) \wedge \\
\wedge \texttt{Month}(month_1, \texttt{12}) \wedge \\
\wedge \big(\texttt{Year}(y_1, \texttt{AS5}) \vee \texttt{Year}(y_1, \texttt{IS4})\big) \wedge \\
\wedge \texttt{Transaction}(t_1, q_1, a_1, s_1, r_1, d_1, month_1, y_1, del_1) \wedge \\
\wedge \texttt{Transaction}(t_2, q_2, a_2, s_2, r_1, d_1, month_1, y_1, del_1) \wedge \\
\wedge \texttt{Number}(q_4, \texttt{3}) \wedge \texttt{Summary}(s_1, q_4) \wedge \\
\wedge \texttt{Document}(doc_1, t_1, t_2, t_3, tr_1, s_1)
\end{array}$$

Figure 3.7: The semantics of text. Observe the interpretation of relative clauses and phrases in parentheses.

$$\begin{array}{l}
\texttt{PublicationEventType}(et_1, \texttt{drukował}) \wedge \texttt{CompositionType}(ct_1, \texttt{wiersze}) \wedge \texttt{Composition}(c_1, ct_1)\big) \wedge \\
\qquad \wedge \texttt{CompositionType}(ct_2, \texttt{artykuły}) \wedge \texttt{Composition}(c_2, ct_2)\big) \wedge \\
\wedge \texttt{MagazineTitle}(mt_1, \texttt{Robotnik}) \wedge \texttt{Magazine}(m_1, mt_1) \wedge \texttt{YearInterval}(y_1, \texttt{1927-28}) \wedge \texttt{Date}(d_1, y_1) \wedge \\
\wedge \texttt{MagazineTitle}(mt_2, \texttt{Nasz Przegląd}) \wedge \texttt{Magazine}(m_2, mt_2) \wedge \texttt{YearInterval}(y_2, \texttt{1927-29}) \wedge \texttt{Date}(d_2, y_2) \wedge \\
\wedge \texttt{MagazineTitle}(mt_3, \texttt{Ster}) \wedge \texttt{Magazine}(m_3, mt_3) \wedge \texttt{Year}(y_3, \texttt{1937}) \wedge \texttt{Date}(d_3, y_3) \wedge \\
\wedge \texttt{MagazineType}(mt_4, \texttt{miesięcznik}) \wedge \texttt{MagazineTitle}(mt_5, \texttt{Wymiary}) \wedge \texttt{Magazine}(m_4, mt_4, mt_5) \wedge \\
\qquad \wedge \texttt{Year}(y_4, \texttt{1938}) \wedge \texttt{Date}(d_4, y_4) \wedge \\
\wedge \texttt{PublicationEvent}(e_1, c_1, m_1, d_1) \wedge \texttt{PublicationEvent}(e_2, c_1, m_2, d_2) \wedge \\
\qquad \wedge \texttt{PublicationEvent}(e_3, c_1, m_3, d_3) \wedge \texttt{PublicationEvent}(e_4, c_1, m_4, d_4) \wedge \\
\qquad \wedge \texttt{PublicationEvent}(e_5, c_2, m_1, d_1) \wedge \texttt{PublicationEvent}(e_6, c_2, m_2, d_2) \wedge \\
\qquad \wedge \texttt{PublicationEvent}(e_7, c_2, m_3, d_3) \wedge \texttt{PublicationEvent}(e_8, ct_2, m_4, d_4) \wedge \\
\wedge \texttt{Magazine}(m_5, mt_5) \wedge \texttt{Organisation}(o_1, m_5) \wedge \\
\wedge \texttt{EventState}(es_1, \texttt{był}) \wedge \texttt{Job}(j_1, \texttt{współredaktor}) \wedge \texttt{WorkEvent}(e_9, es_1, j_1, o_1)
\end{array}$$

- $\texttt{Year}(y_1, \texttt{AS5}) \vee \texttt{Year}(y_1, \texttt{IS4})$ means that y_1 is the 5th **Year** of reign of king Amar-Sin or the 4th **Year** of reign of king Ibbi-Sin.
- $\texttt{Transaction}(t_2, q_2, a_2, s_2, r_1, d_1, month_1, y_1, del_1)$ refers to the **Transaction** t_2, whose parameters are $q_2, a_2, s_2, r_1, d_1, month_1, y_1, del_1$.

Note that a_1 and a_2 point to different animals described by identical words. We assume that every word points to a different object. Later on, on the basis of domain knowledge, we may find equivalences between them.

For the sake of comparison, on fig. 3.7, we present our exemplary sentence from the biobibliographical lexicon. As we see the meaning representation language works in the same way as in the case of Sumerian economic documents.

3.4 Syntactic analysis of Sumerian economic text corpus

The language of Sumerian economic documents and the biobibliographical lexicon are absolutely different: the first one is positional and the other is based on inflection. As a consequence, their syntactic analysis must be carried out using different tools.

From the linguistic point of view, a Sumerian economic document is a single sentence, whose length may vary from below 50 up to more than 5000 signs. The documents from the corpus may be conveniently expressed as words of a regular language. Connections between words and phrases are determined by their positions in the text. Taking this into account we have decided to perform the syntax analysis by means of a grammar that generates a regular language.

We use semantic categories (such as divine names, personal names, job names, year names etc.) to describe dependencies between words. Apart from describing word connections, grammar plays the role of a lexicon for Sumerian and is used for determining word borders.

We define the grammar as follows:

$$G_1 = (\Sigma, N, X_I, R, +, \prec)$$

where

- Σ is a finite set of terminal symbols,
- N is a finite set of non-terminal symbols.
- $X_I \in N$ is the start-symbol of G_1.
- R is a finite set of production rules. Each production has the form $A \to \alpha$ or $A \to \beta+$, where A is a non-terminal and α is a sequence of terminals and non-terminals and $\beta \in \Sigma \cup N$. $A \to \beta+$ is a shortcut for an infinite set of rules: $A \to \beta, A \to \beta\beta, A \to \beta\beta\beta, \dots$. We call such rules *accumulation rules.*
- $\prec$ is binary relation on $\Sigma \cup N$ such that $A \prec B$ if and only if there is a rule $A \to \alpha$ in R such that B occurs in α or there is a rule $A \to B+$.
- $\prec$ is a partial order. This guarantees that G is recursion-free and generates a regular language.

We will call every subsequence parsed to a single grammar symbol a *phrase.*

Proposition 3.4.1 *Language L can be generated by a grammar of the form defined above if and only if L is regular.*

Proof Each regular language is represented by means of a regular expression composed out of terminal symbols connected with concatenation, alternation and Kleene closure. Concatenation of symbols is equivalent to their occurrence in rule. Alternation is represented in grammar by means rules that have the same symbol in their heads. $A \to \beta+$ rules correspond to Kleene closure. ▮

Names of non-terminal symbols in the grammar reflect the concept names. Example rules:

Head		**Body**
`Number`	::=	`1(disz)`
`Number`	::=	`3(disz)`
`Number`	::=	`2(u) 3(asz@t)`
`Animal`	::=	`sila4`
`Animal`	::=	`sila4 ga`
`Name`	::=	`ur mes`
`Name`	::=	`da da`
`Name`	::=	`ga ga mu`
`Name`	::=	`ga mu`
`Name`	::=	`ab ba sa6 ga`
`Job`	::=	`ensi2`
`Job`	::=	`lugal`
`NameJob`	::=	`Name Job`
`NameJob`	::=	`Name`
`NameJob`	::=	`Job`
`Filiation`	::=	`dumu NameJob`
`Person`	::=	`NameJob`
`Person`	::=	`NameJob Filiation`
`Supplier`	::=	`Person`
`Receiver`	::=	`Person i3 dab5`
`Delivery`	::=	`mu DU`
`Day`	::=	`u4 Number kam`
`Month`	::=	`iti sze KIN ku5`
`Year`	::=	`mu en d inanna ba hun`
`Summary`	::=	`left Number`
`NumberAnimal`	::=	`Number Animal`
`NumberAnimalList`	::=	`NumberAnimal +`
`NumberAnimalSupplier`	::=	`NumberAnimalList Supplier`
`NumberAnimalSupplierList`	::=	`NumberAnimalSupplier +`
`Transaction`	::=	`NumberAnimalSupplierList Day Delivery Receiver Month Year`
`Document`	::=	`Transaction Summary`

The above grammar is slightly simplified with respect to the ontology. For the sake of presentation clarity, we do not split `Animal` into `AnimalName` and `AnimalParam`, `Number` into `Digits`, etc.

3.5 Syntactic analysis of Biobibliographical Lexicon of Polish Writers and Literary Scholars

As opposed to the Sumerian economic documents, Polish texts are equipped with rich syntactic information. 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*

For the sake of efficiency, we use the grammar formalism described above for morphological analysis, lexical tagging and semantic parsing of language parts for which ideographic notation is introduced (mainly date expressions). Then we apply context-free grammar for syntactic analysis. Names of symbols in the grammar encode, divided by colons, values of features such as ontological concept names, part-of-speech and inflectional forms. Example rules:

Head		**Body**
`Digit`	::=	`0`
	...	
`Digit`	::=	`9`
`Year`	::=	`1 9 Digit Digit`
`YearInterval`	::=	`Year - Digit Digit`
`Incl:incl`	::=	`( Year )`
`Incl:incl`	::=	`( YearInterval )`
`MagazineTitle:subst:sg:loc`	::=	`Robotniku`
`MagazineTitle:subst:sg:loc`	::=	`Naszym Przeglądzie`
`MagazineTitle:subst:sg:loc`	::=	`Sterze`
`MagazineTitle:subst:pl:nom`	::=	`Wymiary`
`PublicationEventType:core:verb:acc`	::=	`Drukował`
`EventState:core:verb:inst`	::=	`był`
`CompositionType:subst:pl:acc`	::=	`wiersze`
`CompositionType:subst:pl:acc`	::=	`artykuły`
`MagazineType:subst:sg:loc`	::=	`miesięczniku`
`Job:subst:sg:inst`	::=	`współredaktorem`
`Prep:prep:loc`	::=	`w`
`Prep:prep:acc`	::=	`w`
`Conj:conj`	::=	`i`
`Conj:conj`	::=	`oraz`
`Conj:conj`	::=	`,`
`Relpron:relpron:sg:gen`	::=	`którego`

Atomic symbol names encode feature structures 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 define a grammar for parsing feature structures as follows:

$$G_2 = (\mathcal{F}_{A,V,g}, F_I, \mathcal{X}, \mathcal{C}, \mathcal{R})$$

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_2;
- $\mathcal{X} = \{X_i\}_{i=0}^{\infty}$ 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.

Below a few example syntactic rules are presented:

- identity rules that are complements for semantic rules

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

- 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

$$\begin{array}{llll} X_0 ::= X_1 X_2 & X_0.\texttt{sem} = X_1.\texttt{sem}, & & \\ & X_0.\texttt{cat} = X_1.\texttt{cat}, & X_0.\texttt{number} = X_1.\texttt{number}, & X_0.\texttt{case} = X_1.\texttt{case}, \\ & X_2.\texttt{cat} = \texttt{incl} & & \\ X_0 ::= X_1 X_2 X_3 X_4 & X_0.\texttt{sem} = X_1.\texttt{sem}, & & \\ & X_0.\texttt{cat} = X_1.\texttt{cat}, & X_0.\texttt{number} = X_1.\texttt{number}, & X_0.\texttt{case} = X_1.\texttt{case}, \\ & X_1.\texttt{cat} = \texttt{subst}, & X_2.\texttt{cat} = \texttt{,}, & X_3.\texttt{cat} = \texttt{relpron}, \\ & X_4.\texttt{cat} = \texttt{core}, & X_4.\texttt{sen} = \texttt{verb}, & X_1.\texttt{number} = X_3.\texttt{number}, \\ X_0 ::= X_1 X_2 X_3 & X_0.\texttt{sem} = X_1.\texttt{sem}, & X_1.\texttt{sem} = X_3.\texttt{sem}, & \\ & X_0.\texttt{cat} = X_1.\texttt{cat}, & X_0.\texttt{number} = X_1.\texttt{number}, & X_0.\texttt{case} = X_1.\texttt{case}, \\ & X_2.\texttt{cat} = \texttt{conj}, & X_1.\texttt{cat} = X_3.\texttt{cat}, & X_1.\texttt{case} = X_3.\texttt{case} \end{array}$$

The last three rules define sem attribute and do not require any additional semantic rule. The rest of them must be accompanied by one of the following rules:

$X_0 ::= X_1$	X_0.sem = Magazine,	X_1.sem = MagazineTitle	
$X_0 ::= X_1 X_2$	X_0.sem = Magazine,	X_1.sem = MagazineType,	X_2.sem = MagazineTitle
$X_0 ::= X_1 X_2$	X_0.sem = Magazine,	X_1.sem = Prep,	X_2.sem = Magazine
$X_0 ::= X_1$	X_0.sem = Composition,	X_1.sem = CompositionType	
$X_0 ::= X_1$	X_0.sem = Organisation,	X_1.sem = Magazine	
$X_0 ::= X_1 X_2$	X_0.sem = WorkEvent,	X_1.sem = EventState,	X_2.sem = Job
$X_0 ::= X_1 X_2$	X_0.sem = WorkEvent,	X_1.sem = Organisation,	X_2.sem = WorkEvent
$X_0 ::= X_1$	X_0.sem = PublicationEvent,	X_2.sem = PublicationEventType	
$X_0 ::= X_1 X_2$	X_0.sem = PublicationEvent,	X_1.sem = PublicationEvent,	X_2.sem = Composition
$X_0 ::= X_1 X_2$	X_0.sem = PublicationEvent,	X_1.sem = PublicationEvent,	X_2.sem = Magazine

The above rules encode dependencies between ontological concepts. Fig. 3.9 illustrates the grammar at work.

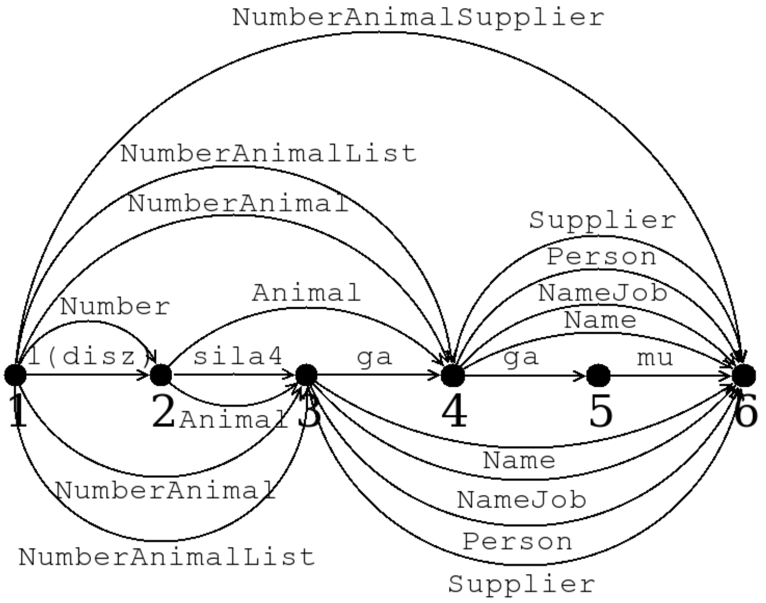


Figure 3.8: A chart for phrase `1(disz) sila4 ga-ga-mu`

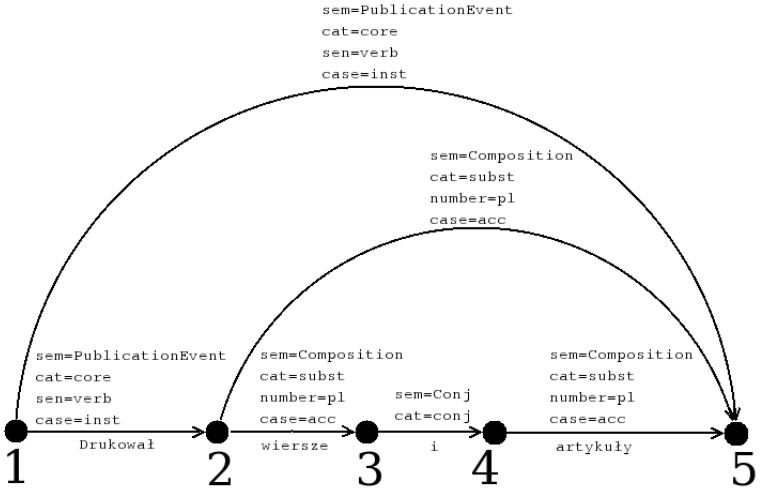


Figure 3.9: Simplified chart for phrase `Drukował wiersze i artykuły`

3.6 Text representation

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 signs (in case of Polish documents, signs are encoded feature structures). 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 positions in text. Formally, let $\{\sigma_i\}_1^n$, $\sigma_i \in \Sigma$ be a sequence. We create a graph with vertices $V = \{v_0, \dots, v_n\}$ and set of edges $E = \{v_0 \xrightarrow{\sigma_1} v_1, \dots, v_{n-1} \xrightarrow{\sigma_n} v_n\}$.

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.

In order to apply the rule $A \to \alpha_1, \dots, \alpha_k$ we find all paths

$$v_{a_0} \xrightarrow{\alpha_1} v_{a_1} \xrightarrow{\alpha_2} v_{a_2} \dots v_{a_{k-1}} \xrightarrow{\alpha_k} v_{a_k}$$

and we add for each of them the edge

$$v_{a_0} \xrightarrow{A} v_{a_k}$$

to the graph.

While applying the $A \to \beta+$ rule, we find all paths

$$v_{a_0} \xrightarrow{\beta} v_{a_1} \xrightarrow{\beta} v_{a_2} \dots v_{a_{k-1}} \xrightarrow{\beta} v_{a_k}$$

and we add for each of them the edge

$$v_{a_0} \xrightarrow{A} v_{a_k}$$

to the graph.

We will denote an edge labelled α from vertex i to vertex j by $\alpha_{i,j}$.

As a result of parsing process we obtain an edge from the first vertex to the last vertex of the graph, labelled by the start symbol of the grammar. Fig. 3.8 and 3.9 illustrate the text representation structure.

3.7 Parser algorithm

Since we use two different grammars, we need also two parsing algorithms.

The first is a robust algorithm which we developed for parsing with large G_1 grammars. Before parsing we prepare the grammar dividing the set of symbols into layers: Let $N_0 = \Sigma$ and let

$$N_{n+1} = \{A \mid \exists \text{ "}A \to \alpha_1 \dots \alpha_k\text{" } \forall i(\alpha_i \in N_n)\} \cup \{A \mid \exists \text{ "}A \to \beta + \text{" } (\beta \in N_n)\} \cup N_n.$$

Then we divide the set of rules R into layers. Let $R_{-1} = \emptyset$ and

$$R_n = \{A \to \alpha_1 \dots \alpha_k \mid \forall i\ \alpha_i \in N_n\} \cup \{A \to \beta+ \mid \beta \in N_n\} \setminus \bigcup_{i=0}^{n-1} R_i.$$

Since we do not allow recurrent symbols to occur there are finitely many layers.

For example, for the grammar created for our example Sumerian tablet we obtain:

$$
\begin{array}{ll}
N_0 = & \{\texttt{1(disz)}, \texttt{3(disz)}, \texttt{2(u)}, \texttt{3(asz@t)}, \texttt{sila4}, \texttt{ga}, \texttt{ur}, \texttt{mes}, \texttt{da}, \texttt{ga}, \texttt{mu}, \texttt{ab}, \texttt{ba}, \\
 & \texttt{sa6}, \texttt{ga}, \texttt{ensi2}, \texttt{lugal}, \texttt{dumu}, \texttt{i3}, \texttt{dab5}, \texttt{mu}, \texttt{DU}, \texttt{u4}, \texttt{kam}, \texttt{iti}, \texttt{sze}, \texttt{KIN}, \texttt{ku5}, \\
 & \texttt{mu}, \texttt{en}, \texttt{d}, \texttt{inanna}, \texttt{ba}, \texttt{hun}, \texttt{left}\} \\
N_1 \setminus N_0 = & \{\texttt{Number}, \texttt{Animal}, \texttt{Name}, \texttt{Job}, \texttt{Delivery}, \texttt{Month}, \texttt{Year}\} \\
N_2 \setminus N_1 = & \{\texttt{NameJob}, \texttt{Day}, \texttt{Summary}, \texttt{NumberAnimal}\} \\
N_3 \setminus N_2 = & \{\texttt{NumberAnimalList}, \texttt{Filiation}\} \\
N_4 \setminus N_3 = & \{\texttt{Person}\} \\
N_5 \setminus N_4 = & \{\texttt{Supplier}, \texttt{Receiver}\} \\
N_6 \setminus N_5 = & \{\texttt{NumberAnimalSupplier}\} \\
N_7 \setminus N_6 = & \{\texttt{NumberAnimalSupplierList}\} \\
N_8 \setminus N_7 = & \{\texttt{Transaction}\} \\
N_9 \setminus N_8 = & \{\texttt{Document}\}
\end{array}
$$

and

$$
R_0 = \begin{array}{l}
\texttt{Number} ::= \texttt{1(disz)} \\
\texttt{Number} ::= \texttt{3(disz)} \\
\texttt{Number} ::= \texttt{2(u) 3(asz@t)} \\
\texttt{Animal} ::= \texttt{sila4} \\
\texttt{Animal} ::= \texttt{sila4 ga} \\
\texttt{Name} ::= \texttt{ur mes} \\
\texttt{Name} ::= \texttt{da da} \\
\texttt{Name} ::= \texttt{ga ga mu} \\
\texttt{Name} ::= \texttt{ga mu} \\
\texttt{Name} ::= \texttt{ab ba sa6 ga} \\
\texttt{Job} ::= \texttt{ensi2} \\
\texttt{Job} ::= \texttt{lugal}
\end{array}
$$

$$
R_1 = \begin{array}{l}
\texttt{NameJob} ::= \texttt{Name Job} \\
\texttt{NameJob} ::= \texttt{Name} \\
\texttt{NameJob} ::= \texttt{Job} \\
\texttt{Day} ::= \texttt{u4 Number kam} \\
\texttt{Summary} ::= \texttt{left Number} \\
\texttt{NumberAnimal} ::= \texttt{Number Animal}
\end{array}
$$

$$
R_2 = \begin{array}{l}
\texttt{NumberAnimalList} ::= \texttt{NumberAnimal +} \\
\texttt{Filiation} ::= \texttt{dumu NameJob}
\end{array}
$$

etc.

Rules belonging to each layer are independent. Hence, we may go through the sequence once for each layer and apply all matching rules simultaneously.

For each path in the chart, the algorithm finds all rules that match edge labels on that path. For each matched rule the algorithm adds to the chart a new edge from the beginning to the end of the matched path and labels it with the rule head. We begin with the chart (V, E_0), where E_0 is labelled

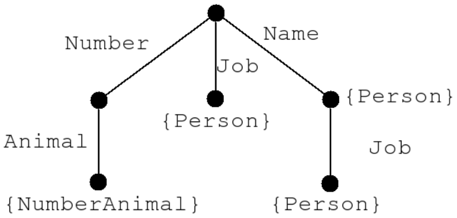


Figure 3.10: Prefix tree

with the segmented text. Applying to (V, E_n) rules from R_n, we obtain the chart (V, E_{n+1}). For each text's subsequence we find all its possible syntactic derivation trees.

In order to apply efficiently a large number of rules we organize rules in the data structure called a *prefix tree*. For each rule $A \to \alpha_1 \dots \alpha_k$ 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. Fig. 3.10 presents a prefix tree generated for layer R_1.

We mentioned in the previous section that we encode feature structures as atomic symbols. Now, we translate G_2 into the rules of context-free grammar.

G_2 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 $P = Q \cup P_1 \cup P_2 \cup \dots$, where

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

For example rules

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

are composed into the following rule

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

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

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

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

$$\begin{array}{lllll}
X_0 ::= X_1 X_2 & X_0.\texttt{cat} = \texttt{pp}, & X_1.\texttt{cat} = \texttt{prep}, & X_2.\texttt{cat} = \texttt{subst}, \\
 & X_1.\texttt{case} = \texttt{nom} & X_2.\texttt{case} = \texttt{nom} & \\
 & X_0.\texttt{sem} = \texttt{Magazine}, & X_1.\texttt{sem} = \texttt{Prep}, & X_2.\texttt{sem} = \texttt{Magazine} \\
X_0 ::= X_1 X_2 & X_0.\texttt{cat} = \texttt{pp}, & X_1.\texttt{cat} = \texttt{prep}, & X_2.\texttt{cat} = \texttt{subst}, \\
 & X_1.\texttt{case} = \texttt{gen} & X_2.\texttt{case} = \texttt{gen} & \\
 & X_0.\texttt{sem} = \texttt{Magazine}, & X_1.\texttt{sem} = \texttt{Prep}, & X_2.\texttt{sem} = \texttt{Magazine}
\end{array}$$

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. That is why we organise rules of the obtained grammar in a prefix tree. We parse this grammar using Earley Parser [Earley, 1986].

3.8 Semantic values of grammar symbols

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.

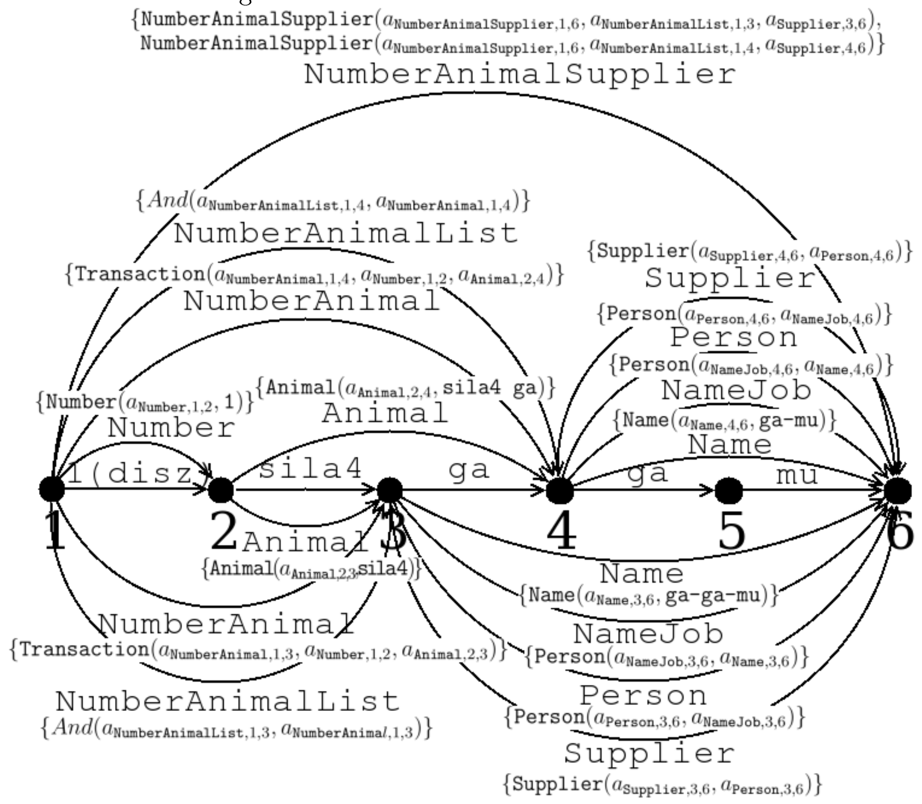
For that to be possible the meaning representation language formulae must be coherent with the syntactic decomposition. The following relation between syntax and semantics is required: we assume that each phrase describes an entity. Chart edges are created as a result of parsing phrases, so, for each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

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

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 $a_{\alpha_1^k,i_1^k,j_1^k}, \dots, a_{\alpha_{m_k}^k,i_{m_k}^k,j_{m_k}^k}$. 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.

Figure 3.11: A chart with semantic values.



{NumberAnimalSupplier($a_{\texttt{NumberAnimalSupplier},1,6}, a_{\texttt{NumberAnimalList},1,3}, a_{\texttt{Supplier},3,6}$),
NumberAnimalSupplier($a_{\texttt{NumberAnimalSupplier},1,6}, a_{\texttt{NumberAnimalList},1,4}, a_{\texttt{Supplier},4,6}$)}
NumberAnimalSupplier
{$And(a_{\texttt{NumberAnimalList},1,4}, a_{\texttt{NumberAnimal},1,4})$}
NumberAnimalList
{Transaction($a_{\texttt{NumberAnimal},1,4}, a_{\texttt{Number},1,2}, a_{\texttt{Animal},2,4}$)}
NumberAnimal
{Number($a_{\texttt{Number},1,2}$, 1)}
Number
{Animal($a_{\texttt{Animal},2,4}$, sila4 ga)}
Animal
{Supplier($a_{\texttt{Supplier},4,6}, a_{\texttt{Person},4,6}$)}
Supplier
{Person($a_{\texttt{Person},4,6}, a_{\texttt{NameJob},4,6}$)}
Person
{Person($a_{\texttt{NameJob},4,6}, a_{\texttt{Name},4,6}$)}
NameJob
{Name($a_{\texttt{Name},4,6}$, ga-mu)}
Name
1(disz)
sila4
ga
ga
mu
1
2
3
4
5
6
Animal
{Animal($a_{\texttt{Animal},2,3}$,sila4)}
NumberAnimal
{Transaction($a_{\texttt{NumberAnimal},1,3}, a_{\texttt{Number},1,2}, a_{\texttt{Animal},2,3}$)}
NumberAnimalList
{$And(a_{\texttt{NumberAnimalList},1,3}, a_{\texttt{NumberAnimal},1,3})$}
Name
{Name($a_{\texttt{Name},3,6}$, ga-ga-mu)}
NameJob
{Person($a_{\texttt{NameJob},3,6}, a_{\texttt{Name},3,6}$)}
Person
{Person($a_{\texttt{Person},3,6}, a_{\texttt{NameJob},3,6}$)}
Supplier
{Supplier($a_{\texttt{Supplier},3,6}, a_{\texttt{Person},3,6}$)}

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

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

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. 3.11 we present an example of a chart. The names of predicates associated with the grammar symbols are usually identical to their names. The first argument of each predicate is a constant that represents an entity described by the phrase. Indices of this constant identify the edge into which this phrase has been parsed.

On fig. 3.12 we show semantic values for some of the edges for chart on fig. 3.11. For example the semantic value of the edge $\texttt{NumberAnimal}_{1,4}$ (denoted as $[\![\texttt{NumberAnimal}]\!]_{1,4}$) is calculated to be the formula $\texttt{Transaction}(a_{\texttt{NumberAnimal},1,4},\ a_{\texttt{Number},1,2}, a_{\texttt{Animal},2,4})$, because the constant $a_{\texttt{Number},1,2}$ refers to the edge $\texttt{Number}_{1,2}$ and $a_{\texttt{Animal},2,4}$ to $\texttt{Animal}_{2,4}$.

Figure 3.12: Semantic values of grammar symbols

$$\llbracket\texttt{Name}\rrbracket_{4,6} = \texttt{Name}(a_{\texttt{Name},4,6}, \texttt{ga-mu})$$

$$\llbracket\texttt{NumberAnimal}\rrbracket_{1,4} = \texttt{Transaction}(a_{\texttt{NumberAnimal},1,4}, a_{\texttt{Number},1,2}, a_{\texttt{Animal},2,4})\wedge$$
$$\wedge\llbracket\texttt{Number}\rrbracket_{1,2} \wedge \llbracket\texttt{Animal}\rrbracket_{2,4} =$$
$$= \texttt{Transaction}(a_{\texttt{NumberAnimal},1,4}, a_{\texttt{Number},1,2}, a_{\texttt{Animal},2,4})\wedge$$
$$\wedge\texttt{Number}(a_{\texttt{Number},1,2}, \texttt{1}) \wedge \texttt{Animal}(a_{\texttt{Animal},2,4}, \texttt{sila4 ga})$$

$$\llbracket\texttt{NumberAnimalSupplier}\rrbracket_{1,6} =$$
$$\big(\texttt{NumberAnimalSupplier}(a_{\texttt{NumberAnimalSupplier},1,6}, a_{\texttt{NumberAnimalList},1,3}, a_{\texttt{Supplier},3,6})\wedge$$
$$\wedge\llbracket\texttt{NumberAnimalList}\rrbracket_{1,3} \wedge \llbracket\texttt{Supplier}\rrbracket_{3,6}\big) \vee$$
$$\vee \big(\texttt{NumberAnimalSupplier}(a_{\texttt{NumberAnimalSupplier},1,6}, a_{\texttt{NumberAnimalList},1,4}, a_{\texttt{Supplier},4,6})\wedge$$
$$\wedge\llbracket\texttt{NumberAnimalList}\rrbracket_{1,4} \wedge \llbracket\texttt{Supplier}\rrbracket_{4,6}\big)$$

Ambiguous phrases may be parsed to a number of predicates. Each predicate generated for a given subsequence is a possible description of an entity. That is why we point that entity by the same constant in each predicate (consider for example the semantic value of an edge $\llbracket\texttt{NumberAnimalSupplier}\rrbracket_{1,6}$).

3.9 Semantic attachments

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.

Let $A \to \alpha_1 \ldots \alpha_k$ be a syntactic rule and $f_{A\to\alpha_1\ldots\alpha_k}$ be a semantic attachment assigned to it. Assume that the rule was matched to the path $\alpha_{1,i_0,i_1}, \ldots, \alpha_{k,i_{k-1},i_k}$. As the rule was applied the symbol A_{i_0,i_k} was created. The predicate set associated with A_{i_0,i_k} is constructed as follows:

$$[A]_{i_0,i_k} := \{f_{A\to\alpha_1\ldots\alpha_k}(a_{A,i_0,i_k}, a_{\alpha_1,i_0,i_1}, \ldots, a_{\alpha_k,i_{k-1},i_k})\} \cup [A]_{i_0,i_k}.$$

$[A]_{i_0,i_k}$ on the right side of assignment is the predicate set associated with the edge A_{i_0,i_k} before the rule application. If A_{i_0,i_k} did not exist before the rule application this set is empty. The first argument of the semantic attachment is the constant a_{A,i_0,i_k} whose value is the entity described by phrase parsed to A_{i_0,i_k}.

We may add a few different semantic attachments to a syntactic rule, obtaining rules that are grammatically identical but differ on semantic level.

Upon the end of the parsing process we obtain an edge labelled by the start symbol of the grammar. Its semantic value is a formula equivalent to a disjunction of all the possible translations of the entire text into the meaning representation language.

We express semantic attachments as unnamed functions using the notation taken from the λ-calculus. For example formula $\lambda x_0 x_1.\texttt{Number}(x_0, \texttt{1})$ defines a function whose arguments are x_0 and x_1 and whose value is $\texttt{Number}(x_0, \texttt{1})$. Function value is independent from x_1.

Below, we enclose the grammar rules for Sumerian economic documents together with their semantic attachments:

Rule	**Semantic Attachment**
`Number ::=1(disz)`	$\lambda x_0 x_1.\texttt{Number}(x_0, \texttt{1})$
`Number ::= 3(disz)`	$\lambda x_0 x_1.\texttt{Number}(x_0, \texttt{3})$
`Number ::= 2(u) 3(asz@t)`	$\lambda x_0 x_1 x_2.\texttt{Number}(x_0, \texttt{23})$
`Animal ::= sila4`	$\lambda x_0 x_1.\texttt{Animal}(x_0, \texttt{sila4})$
`Animal ::= sila4 ga`	$\lambda x_0 x_1 x_2.\texttt{Animal}(x_0, \texttt{sila4 ga})$
`Name ::= ur mes`	$\lambda x_0 x_1 x_2.\texttt{Name}(x_0, \texttt{ur-mes})$
`Name ::= da da`	$\lambda x_0 x_1 x_2.\texttt{Name}(x_0, \texttt{da da})$
`Name ::= ga ga mu`	$\lambda x_0 x_1 x_2 x_3.\texttt{Name}(x_0, \texttt{ga-ga-mu})$
`Name ::= ga mu`	$\lambda x_0 x_1 x_2.\texttt{Name}(x_0, \texttt{ga-mu})$
`Name ::= ab ba sa6 ga`	$\lambda x_0 x_1 x_2 x_3 x_4.\texttt{Name}(x_0, \texttt{ab-ba-sa6-ga})$
`Job ::= ensi2`	$\lambda x_0 x_1.\texttt{Job}(x_0, \texttt{ensi2})$
`Job ::= lugal`	$\lambda x_0 x_1.\texttt{Job}(x_0, \texttt{lugal})$
`NameJob ::= Name Job`	$\lambda x_0 x_1 x_2.\texttt{Person}(x_0, x_1, x_2)$
`NameJob ::= Name`	$\lambda x_0 x_1.\texttt{Person}(x_0, x_1)$
`NameJob ::= Job`	$\lambda x_0 x_1.\texttt{Person}(x_0, x_1)$
`Filiation ::= dumu NameJob`	$\lambda x_0 x_1 x_2.\texttt{Filiation}(x_0, \texttt{dumu}, x_2)$
`Person ::= NameJob`	$\lambda x_0 x_1.\texttt{Person}(x_0, x_1)$
`Person ::= NameJob Filiation`	$\lambda x_0 x_1 x_2.\texttt{Person}(x_0, x_1, x_2)$
`Supplier ::= Person`	$\lambda x_0 x_1.\texttt{Supplier}(x_0, x_1)$
`Receiver ::= Person i3 dab5`	$\lambda x_0 x_1 x_2 x_3.\texttt{Receiver}(x_0, x_1)$
`Delivery ::= mu DU`	$\lambda x_0 x_1 x_2.\texttt{Delivery}(x_0)$
`Day ::= u4 Number kam`	$\lambda x_0 x_1 x_2 x_3.\texttt{Day}(x_0, x_2)$
`Month ::= iti sze KIN ku5`	$\lambda x_0 x_1 x_2 x_3 x_4.\texttt{Month}(x_0, \texttt{12})$
`Year ::= mu en d inanna ba hun`	$\lambda x_0 x_1 x_2 x_3 x_4 x_5 x_6.\texttt{Year}(x_0, \texttt{AS05})$
`Year ::= mu en d inanna ba hun`	$\lambda x_0 x_1 x_2 x_3 x_4 x_5 x_6.\texttt{Year}(x_0, \texttt{IS04})$
`Summary ::= left Number`	$\lambda x_0 x_1 x_2.\texttt{Summary}(x_0, x_2)$
`NumberAnimal ::= Number Animal`	$\lambda x_0 x_1 x_2.\texttt{Transaction}(x_0, x_1, x_2)$
`NumberAnimalList ::= NumberAnimal +`	$\lambda x_0 x_1 \ldots x_n.And(x_0, x_1, \ldots, x_n)$
`NumberAnimalSupplier ::= NumberAnimalList Supplier`	$\lambda x_0 x_1 x_2.\texttt{Transaction}(x_0, x_1, x_2)$
`NumberAnimalSupplierList ::= NumberAnimalSupplier +`	$\lambda x_0 x_1 \ldots x_n.And(x_0, x_1, \ldots, x_n)$
`Transaction ::= NumberAnimalSupplierList Day Delivery Receiver Month Year`	$\lambda x_0 x_1 x_2 x_3 x_4 x_5 x_6.\texttt{Transaction}(x_0, x_1, x_4, x_2, x_3, x_5, x_6, x_3)$
`Document ::= Transaction Summary`	$\lambda x_0 x_1 x_2.\texttt{Document}(x_0, x_1, x_2)$

A typical semantic attachment introduces a predicate, whose arguments are constants referring to words or other predicates. The rule for `Year` recognition is written two times with different semantic attachments, which provide the two possible readings of the year name. Two rules introduce a special predicate *And*. We use this predicate for describing the conjunction in natural language. We will describe it in detail in the next section. These two rules are also accumulation rules. They may match to a variable number of symbols. That is why their semantic attachments are defined as templates depending on the number of matched symbols. Fig. 3.13 presents the meaning representation language formula obtained as a result of parsing our example Sumerian tablet (we have simplified constant names and the representation of semantic value for

Figure 3.13: The meaning representation language formula for the Sumerian economic document.

$$
\begin{aligned}
&\texttt{Number}(q_1, \texttt{1}) \wedge \texttt{Animal}(a_1, \texttt{sila4}) \wedge \texttt{Transaction}(na_1, q_1, a_1) \wedge And(nal_1, na_1) \wedge \\
&\qquad \wedge \texttt{Name}(n_1, \texttt{ur-mes}) \wedge \texttt{Job}(j_1, \texttt{ensi2}) \wedge \texttt{Person}(nj_1, n_1, j_1) \wedge \\
&\qquad \wedge \texttt{Person}(p_1, nj_1) \wedge \texttt{Supplier}(s_1, p_1) \wedge \texttt{Transaction}(nas_1, nal_1, s_1) \wedge \\
&\wedge \texttt{Number}(q_2, \texttt{1}) \wedge \texttt{Animal}(a_2, \texttt{sila4}) \wedge \texttt{Transaction}(na_2, q_2, a_2) \wedge And(nal_2, na_2) \wedge \\
&\qquad \wedge \texttt{Name}(n_2, \texttt{da-da}) \wedge \texttt{Person}(nj_2, n_2) \wedge \texttt{Job}(j_2, \texttt{lugal}) \wedge \texttt{Person}(p_2, j_2) \wedge \texttt{Filiation}(f_1, \texttt{dumu}, p_2) \wedge \\
&\qquad \wedge \texttt{Person}(p_3, nj_2, f_1) \wedge \texttt{Supplier}(s_2, p_3) \wedge \texttt{Transaction}(nas_2, na_2, s_2) \wedge \\
&\wedge \texttt{Number}(q_3, \texttt{1}) \wedge \big(\big(\texttt{Animal}(a_3, \texttt{sila4}) \wedge \texttt{Transaction}(na_3, q_3, a_3) \wedge And(nal_3, na_3) \wedge \\
&\qquad \wedge \texttt{Name}(n_3, \texttt{ga-ga-mu}) \wedge \texttt{Person}(nj_3, n_3) \wedge \\
&\qquad \wedge \texttt{Person}(p_4, nj_3) \wedge \texttt{Supplier}(s_3, p_4) \wedge \texttt{Transaction}(nas_3, na_3, s_3)\big) \vee \\
&\qquad \vee \big(\texttt{Animal}(a_4, \texttt{sila4 ga}) \wedge \texttt{Transaction}(na_4, q_3, a_4) \wedge And(nal_4, na_4) \wedge \texttt{Name}(n_4, \texttt{ga-mu}) \wedge \\
&\qquad \wedge \texttt{Person}(nj_4, n_4) \wedge \texttt{Person}(p_5, nj_4) \wedge \texttt{Supplier}(s_4, p_5) \wedge \texttt{Transaction}(nas_3, na_4, s_4)\big)\big) \wedge \\
&\qquad \wedge And(nasl_1, nas_1, nas_2, nas_3) \\
&\wedge \texttt{Day}(d_1, \texttt{23}) \wedge \\
&\wedge \texttt{Delivery}(del_1) \wedge \\
&\wedge \texttt{Name}(n_5, \texttt{ab-ba-sa6-ga}) \wedge \texttt{Person}(nj_5, n_5) \wedge \texttt{Person}(p_6, nj_5) \wedge \texttt{Receiver}(r_1, p_6) \wedge \\
&\wedge \texttt{Month}(month_1, \texttt{12}) \wedge \\
&\wedge \big(\texttt{Year}(y_1, \texttt{AS5}) \vee \texttt{Year}(y_1, \texttt{IS4})\big) \wedge \\
&\wedge \texttt{Transaction}(tr_1, nasl_1, r_1, d_1, month_1, y_1, del_1) \wedge \\
&\wedge \texttt{Number}(q_4, \texttt{3}) \wedge \texttt{Summary}(s_1, q_4) \wedge \\
&\wedge \texttt{Document}(doc_1, tr_1, s_1)
\end{aligned}
$$

accumulation rules).

Now, we switch to the Biobibliographical Lexicon. First we define semantic attachments for its G_1 grammar:

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)

The only piece of new notation that appears in the above table is $[x_i]$, which we use to extract a specific digit value in order to compose the year number. $[x_i]$ is defined under the assumption that x_i points to a predicate which has the form

$$p(x_i, w),$$

where w is a word. If the condition is satisfied then $[x_i]$ is replaced with w.

In case of G_2 grammar only rules that define sem attribute possess semantic attachments.

$X_0 ::= X_1 X_2$	X_0.sem = X_1.sem, X_0.cat = X_1.cat, X_2.cat = incl $\lambda x_0 x_1 x_2 . Rel(x_0, x_1, x_2)$	X_0.number = X_1.number,	X_0.case = X_1.case,
$X_0 ::= X_1 X_2 X_3 X_4$	X_0.sem = X_1.sem, X_0.cat = X_1.cat, X_1.cat = subst, X_4.cat = core, $\lambda x_0 x_1 x_2 x_3 x_4 . Rel(x_0, x_1, x_4)$	X_0.number = X_1.number, X_2.cat = ,, X_4.sen = verb,	X_0.case = X_1.case, X_3.cat = relpron, X_1.number = X_3.number,
$X_0 ::= X_1 X_2 X_3$	X_0.sem = X_1.sem, X_0.cat = X_1.cat, X_2.cat = conj, $\lambda x_0 x_1 x_2 x_3 . And(x_0, x_1, x_3)$	X_1.sem = X_3.sem, X_0.number = X_1.number, X_1.cat = X_3.cat,	X_0.case = X_1.case, X_1.case = X_3.case
$X_0 ::= X_1$	X_0.sem = Magazine, $\lambda x_0 x_1$.Magazine(x_0, x_1)	X_1.sem = MagazineTitle	
$X_0 ::= X_1 X_2$	X_0.sem = Magazine, $\lambda x_0 x_1 x_2$.Magazine(x_0, x_1, x_2)	X_1.sem = MagazineType,	X_2.sem = MagazineTitle
$X_0 ::= X_1 X_2$	X_0.sem = Magazine, $\lambda x_0 x_1 x_2$.Magazine(x_0, x_2)	X_1.sem = Prep,	X_2.sem = Magazine
$X_0 ::= X_1$	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

We use the special symbol *Rel* to introduce relative clauses and insertions. We will define it in the next section. Fig. 3.14 presents the meaning representation language formula for the part of our example sentence. Fig. 3.15 presents the meaning representation language formula for the whole sentence. Due to a high ambiguity we decided to show the formula associated with a single parse tree.

Figure 3.14: The meaning representation language formula for sentence `Drukował wiersze i artykuły`.

$$\begin{aligned}
&\texttt{PublicationEventType}(et_1, \texttt{drukował}) \wedge \texttt{PublicationEvent}(e_1, et_1) \wedge \\
&\wedge \texttt{CompositionType}(ct_1, \texttt{wiersze}) \wedge \texttt{CompositionType}(ct_2, \texttt{artykuły}) \wedge \\
&\wedge \big(\big(And(ct_3, ct_1, ct_2) \wedge \texttt{Composition}(c_1, ct_3) \big) \vee \\
&\ \ \vee \big(\texttt{Composition}(c_2, ct_1) \wedge \texttt{Composition}(c_3, ct_2) \wedge And(c_1, c_2, c_3) \big) \big) \wedge \\
&\wedge \texttt{PublicationEvent}(e_2, e_1, c_1)
\end{aligned}$$

Figure 3.15: The meaning representation language formula for the example sentence.

$$\begin{aligned}
&\texttt{PublicationEventType}(et_1, \texttt{drukował}) \wedge \texttt{PublicationEvent}(e_1, et_1) \wedge \\
&\qquad \wedge \texttt{CompositionType}(ct_1, \texttt{wiersze}) \wedge \texttt{CompositionType}(ct_2, \texttt{artykuły}) \wedge \\
&\qquad \wedge And(ct_3, ct_1, ct_2) \wedge \texttt{Composition}(c_1, ct_3)) \wedge \\
&\wedge \texttt{MagazineTitle}(mt_1, \texttt{Robotnik}) \wedge \texttt{YearInterval}(y_1, \texttt{1927-28}) \wedge Rel(r_1, mt_1, y_1) \wedge \texttt{Magazine}(m_1, r_1) \wedge \\
&\wedge \texttt{MagazineTitle}(mt_2, \texttt{Nasz Przegląd}) \wedge \texttt{YearInterval}(y_2, \texttt{1927-29}) \wedge Rel(r_2, mt_2, y_2) \wedge \texttt{Magazine}(m_2, r_2) \wedge \\
&\wedge \texttt{MagazineTitle}(mt_3, \texttt{Ster}) \wedge \texttt{Year}(y_3, \texttt{1937}) \wedge Rel(r_3, mt_3, y_3) \wedge \texttt{Magazine}(m_3, r_3) \wedge \\
&\wedge \texttt{MagazineType}(mt_4, \texttt{miesięcznik}) \wedge \texttt{MagazineTitle}(mt_5, \texttt{Wymiary}) \wedge \texttt{Year}(y_4, \texttt{1938}) \wedge Rel(r_4, mt_5, y_4) \wedge \\
&\wedge \texttt{EventState}(es_1, \texttt{był}) \wedge \texttt{Job}(j_1, \texttt{współredaktor}) \wedge \texttt{WorkEvent}(e_4, es_1, j_1) \wedge Rel(r_5, r_4, e_4) \wedge \\
&\wedge \texttt{Magazine}(m_4, mt_4, r_6) \wedge And(m_5, m_1, m_2) \wedge And(m_6, m_3, m_4) \wedge And(m_7, m_5, m_6) \wedge \\
&\wedge \texttt{PublicationEvent}(e_2, e_1, c_1) \wedge \texttt{PublicationEvent}(e_3, e_2, m_7) \wedge
\end{aligned}$$

3.10 Special symbols processing

Meaning representation language formulae obtained as a result of parsing process must be refined. There are three phenomena that should be eliminated from the formula.

First, we investigate the case when the formula is in disjunctive normal form, i.e.:

$$\bigvee_{i=1}^{n} \bigwedge_{j=1}^{n_i} \varphi_{i,j},$$

where each $\varphi_{i,j}$ is a predicate. Then we switch to the general case.

We define the relation $\prec$ on the set of predicates as the smallest transitive relation such that for each pair of predicates

$$p(a_0, a_1, \ldots, a_n) \prec q(b_0, b_1, \ldots, b_m) \iff a_0 \neq b_0 \wedge \exists_{1 \leq i \leq m}\, a_0 = b_i$$

For a given formula obtained as a result of parsing process, we introduce $\prec$ on the set of predicates that compose this formula. In such case $\prec$ is an irreflexive partial order. We define partial order on We say a predicate φ is on the *root level* of the formula if φ is is a minimal element of $\prec$.

We introduced a special predicate *And* for representing the semantics of the conjunctions in natural language. We push up *And* predicate towards the root of the formula on the basis of the following observation

$$p(a_0, a_1, \ldots, a_n) \wedge And(a_i, b_1, \ldots, b_k)$$

is equivalent to

$$And(a_0, a_0^1, \ldots, a_0^k) \wedge p(a_0^1, a_1, \ldots, a_{i-1}, b_1, a_{i+1}, \ldots, a_n) \wedge \ldots$$
$$\cdots \wedge p(a_0^k, a_1, \ldots, a_{i-1}, b_k, a_{i+1}, \ldots, a_n).$$

At the root level *And* predicate is equivalent to the $\wedge$ conjunction and may be omitted. For example, consider the formula

$$And(ct_3, ct_1, ct_2) \wedge \texttt{Composition}(c_1, ct_3)$$

when we push up *And* predicate we obtain

$$\texttt{Composition}(c_1^1, ct_1) \wedge \texttt{Composition}(c_1^2, ct_2) \wedge And(c_1, c_1^1, c_1^2).$$

There are also occurrences of objects, which are properties of objects from the same ontological category:

$$p(a_0, a_1, \ldots, a_n) \wedge p(a_i, b_1, \ldots, b_k)$$

The above is equivalent to

$$p(a_0, a_1, \ldots, a_{i-1}, b_1, \ldots, b_k, a_{i+1}, \ldots, a_n).$$

For example, consider

$$\texttt{Person}(nj_2, n_2) \wedge \texttt{Person}(p_3, nj_2, f_1).$$

We simplify the formula obtaining

$$\texttt{Person}(p_3, n_2, f_1).$$

The next operation which we analyse is the elimination of *Rel* predicate, which appears as a result of parsing of relative clauses, adjectival participle phrases, event nominalisations and inclusions. 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. In case of relative clauses, adjectival participle phrases and inclusions we require b to be a word. Inclusions tend to be composed out of several independent components divided with '`;`' which should be split into distinct *Rel* predicates. Then we check whether c describes an event.

If this is the case and if c is not an event nominalisation then we parse b and c into a consistent formula. For example

$$\texttt{MagazineTitle}(mt_5, \texttt{Wymiary}) \wedge$$
$$\wedge \texttt{EventState}(es_1, \texttt{był}) \wedge \texttt{Job}(j_1, \texttt{współredaktor}) \wedge \texttt{WorkEvent}(e_4, es_1, j_1) \wedge Rel(r_1, mt_5, e_4)$$

will be transformed into

$$\texttt{MagazineTitle}(mt_5, \texttt{Wymiary}) \wedge \texttt{Magazine}(m_4, mt_5) \wedge \texttt{Organisation}(o_1, m_4) \wedge$$
$$\wedge \texttt{EventState}(es_1, \texttt{był}) \wedge \texttt{Job}(j_1, \texttt{współredaktor}) \wedge \texttt{WorkEvent}(e_4, es_1, j_1) \wedge$$
$$\wedge \texttt{WorkEvent}(e_5, o_1, e_4) \wedge Rel(r_1, mt_5, e_5)$$

Then if c describes an event we push up the predicate towards the root of the formula. When we achieve the root we replace *Rel* with *And*.

Otherwise (i.e., when c does not describe an event) we check whether c may be an argument of b. It this is the case then we replace *Rel* with the ontological category of b, else we push up the predicate one level and check again.

In case of *Rel* predicate the push up operation is performed as follows:

$$p(a_0, a_1, \ldots, a_n) \wedge And(a_i, b, c)$$

is replaced by

$$And(a_0, b, c) \wedge p(a_0, a_1, \ldots, a_{i-1}, b, a_{i+1}, \ldots, a_n)$$

In the case when disjunction appears in the formula we factor it out and execute rules described above. We avoid combinatorial explosion performing operations in a proper order: we begin we the formulae describing simple ontological concepts, and then we process formulae that describe concepts composed out of them. We factor out disjunction only when it is necessary. When we introduce a new variable we check if this variable is equivalent to any other variable in the formula. If so, we replace the new variable with the equivalent old one. For example, consider the formula, which we presented on Fig. 3.14:

$$\begin{array}{l}
\texttt{PublicationEventType}(et_1, \texttt{drukował}) \wedge \texttt{PublicationEvent}(e_1, et_1) \wedge \\
\wedge \texttt{CompositionType}(ct_1, \texttt{wiersze}) \wedge \texttt{CompositionType}(ct_2, \texttt{artykuły}) \wedge \\
\wedge \big(\big(And(ct_3, ct_1, ct_2) \wedge \texttt{Composition}(c_1, ct_3) \big) \vee \\
\;\; \vee \big(\texttt{Composition}(c_2, ct_1) \wedge \texttt{Composition}(c_3, ct_2) \wedge And(c_1, c_2, c_3) \big) \big) \wedge \\
\wedge \texttt{PublicationEvent}(e_2, e_1, c_1)
\end{array}$$

When we push up the $And(ct_3, ct_1, ct_2)$ predicate we obtain

$$\begin{array}{l}
\texttt{PublicationEventType}(et_1, \texttt{drukował}) \wedge \texttt{PublicationEvent}(e_1, et_1) \wedge \\
\wedge \texttt{CompositionType}(ct_1, \texttt{wiersze}) \wedge \texttt{CompositionType}(ct_2, \texttt{artykuły}) \wedge \\
\wedge \big(\big(\texttt{Composition}(c_1^1, ct_1) \wedge \texttt{Composition}(c_1^2, ct_2) \wedge And(c_1, c_1^1, c_1^2) \big) \vee \\
\;\; \vee \big(\texttt{Composition}(c_2, ct_1) \wedge \texttt{Composition}(c_3, ct_2) \wedge And(c_1, c_2, c_3) \big) \big) \wedge \\
\wedge \texttt{PublicationEvent}(e_2, e_1, c_1)
\end{array}$$

Now, we notice the fact that $c_1^1 = c_2$ and $c_1^2 = c_3$, so we replace occurrences of c_1^1 with c_2 and that of c_1^2 with c_3.

$$\begin{array}{l}
\texttt{PublicationEventType}(et_1, \texttt{drukował}) \wedge \texttt{PublicationEvent}(e_1, et_1) \wedge \\
\wedge \texttt{CompositionType}(ct_1, \texttt{wiersze}) \wedge \texttt{CompositionType}(ct_2, \texttt{artykuły}) \wedge \\
\wedge \big(\big(\texttt{Composition}(c_2, ct_1) \wedge \texttt{Composition}(c_3, ct_2) \wedge And(c_1, c_2, c_3) \big) \vee \\
\;\; \vee \big(\texttt{Composition}(c_2, ct_1) \wedge \texttt{Composition}(c_3, ct_2) \wedge And(c_1, c_2, c_3) \big) \big) \wedge \\
\wedge \texttt{PublicationEvent}(e_2, e_1, c_1)
\end{array}$$

Then we eliminate the disjunction of two identical formulae.

$$\begin{array}{l}
\texttt{PublicationEventType}(et_1, \texttt{drukował}) \wedge \texttt{PublicationEvent}(e_1, et_1) \wedge \\
\wedge \texttt{CompositionType}(ct_1, \texttt{wiersze}) \wedge \texttt{CompositionType}(ct_2, \texttt{artykuły}) \wedge \\
\wedge \texttt{Composition}(c_2, ct_1) \wedge \texttt{Composition}(c_3, ct_2) \wedge And(c_1, c_2, c_3) \wedge \\
\wedge \texttt{PublicationEvent}(e_2, e_1, c_1)
\end{array}$$

When we refine the formula presenting contents of our example Sumerian tablet (Fig. 3.13) we obtain the following formula:

$$\begin{aligned}
&\texttt{Number}(q_1, \texttt{1}) \wedge \texttt{Animal}(a_1, \texttt{sila4}) \wedge \texttt{Name}(n_1, \texttt{ur-mes}) \wedge \texttt{Job}(j_1, \texttt{ensi2}) \wedge \\
&\qquad \wedge \texttt{Person}(p_1, n_1, j_1) \wedge \texttt{Supplier}(s_1, p_1) \wedge \\
&\wedge \texttt{Number}(q_2, \texttt{1}) \wedge \texttt{Animal}(a_2, \texttt{sila4}) \wedge \texttt{Name}(n_2, \texttt{da-da}) \wedge \texttt{Job}(j_2, \texttt{lugal}) \wedge \\
&\qquad \wedge \texttt{Person}(p_2, j_2) \wedge \texttt{Filiation}(f_1, \texttt{dumu}, p_2) \wedge \texttt{Person}(p_3, n_2, f_1) \wedge \texttt{Supplier}(s_2, p_3) \wedge \\
&\wedge \texttt{Number}(q_3, \texttt{1}) \wedge \big(\big(\texttt{Animal}(a_3, \texttt{sila4}) \wedge \texttt{Name}(n_3, \texttt{ga-ga-mu}) \wedge \\
&\qquad \wedge \texttt{Person}(p_4, n_3) \wedge \texttt{Supplier}(s_3, p_4) \wedge \texttt{Transaction}(t_3, q_3, a_3, s_3, r_1, d_1, month_1, y_1, del_1)\big) \vee \\
&\qquad \vee \big(\texttt{Animal}(a_4, \texttt{sila4 ga}) \wedge \texttt{Name}(n_4, \texttt{ga-mu}) \wedge \\
&\qquad \wedge \texttt{Person}(p_5, n_4) \wedge \texttt{Supplier}(s_4, p_5) \wedge \texttt{Transaction}(t_3, q_3, a_4, s_4, r_1, d_1, month_1, y_1, del_1)\big)\big) \wedge \\
&\wedge \texttt{Day}(d_1, \texttt{23}) \wedge \\
&\wedge \texttt{Delivery}(del_1) \wedge \\
&\wedge \texttt{Name}(n_5, \texttt{ab-ba-sa6-ga}) \wedge \texttt{Person}(p_6, n_5) \wedge \texttt{Receiver}(r_1, p_6) \wedge \\
&\wedge \texttt{Month}(month_1, \texttt{12}) \wedge \\
&\wedge \big(\texttt{Year}(y_1, \texttt{AS5}) \vee \texttt{Year}(y_1, \texttt{IS4})\big) \wedge \\
&\wedge \texttt{Transaction}(t_1, q_1, a_1, s_1, r_1, d_1, month_1, y_1, del_1) \wedge \\
&\wedge \texttt{Transaction}(t_2, q_2, a_2, s_2, r_1, d_1, month_1, y_1, del_1) \wedge \\
&\wedge \texttt{Number}(q_4, \texttt{3}) \wedge \texttt{Summary}(s_1, q_4) \wedge \\
&\wedge \texttt{Document}(doc_1, t_1, t_2, t_3, tr_1, s_1)
\end{aligned}$$

Similarly, when we refine the formula presenting contents of our example sentence from the Biobibliographical Lexicon (fig. 3.15) we obtain the following:

$$\begin{aligned}
&\texttt{PublicationEventType}(et_1, \texttt{drukował}) \wedge \texttt{CompositionType}(ct_1, \texttt{wiersze}) \wedge \texttt{Composition}(c_1, ct_1)\big) \wedge \\
&\qquad \wedge \texttt{CompositionType}(ct_2, \texttt{artykuły}) \wedge \texttt{Composition}(c_2, ct_2)\big) \wedge \\
&\wedge \texttt{MagazineTitle}(mt_1, \texttt{Robotnik}) \wedge \texttt{Magazine}(m_1, mt_1) \wedge \texttt{YearInterval}(y_1, \texttt{1927-28}) \wedge \texttt{Date}(d_1, y_1) \wedge \\
&\wedge \texttt{MagazineTitle}(mt_2, \texttt{Nasz Przegląd}) \wedge \texttt{Magazine}(m_2, mt_2) \wedge \texttt{YearInterval}(y_2, \texttt{1927-29}) \wedge \texttt{Date}(d_2, y_2) \wedge \\
&\wedge \texttt{MagazineTitle}(mt_3, \texttt{Ster}) \wedge \texttt{Magazine}(m_3, mt_3) \wedge \texttt{Year}(y_3, \texttt{1937}) \wedge \texttt{Date}(d_3, y_3) \wedge \\
&\wedge \texttt{MagazineType}(mt_4, \texttt{miesięcznik}) \wedge \texttt{MagazineTitle}(mt_5, \texttt{Wymiary}) \wedge \texttt{Magazine}(m_4, mt_4, mt_5) \wedge \\
&\qquad \wedge \texttt{Year}(y_4, \texttt{1938}) \wedge \texttt{Date}(d_4, y_4) \wedge \\
&\wedge \texttt{PublicationEvent}(e_1, c_1, m_1, d_1) \wedge \texttt{PublicationEvent}(e_2, c_1, m_2, d_2) \wedge \\
&\qquad \wedge \texttt{PublicationEvent}(e_3, c_1, m_3, d_3) \wedge \texttt{PublicationEvent}(e_4, c_1, m_4, d_4) \wedge \\
&\qquad \wedge \texttt{PublicationEvent}(e_5, c_2, m_1, d_1) \wedge \texttt{PublicationEvent}(e_6, c_2, m_2, d_2) \wedge \\
&\qquad \wedge \texttt{PublicationEvent}(e_7, c_2, m_3, d_3) \wedge \texttt{PublicationEvent}(e_8, ct_2, m_4, d_4) \wedge \\
&\wedge \texttt{Magazine}(m_5, mt_5) \wedge \texttt{Organisation}(o_1, m_5) \wedge \\
&\wedge \texttt{EventState}(es_1, \texttt{był}) \wedge \texttt{Job}(j_1, \texttt{współredaktor}) \wedge \texttt{WorkEvent}(e_9, es_1, j_1, o_1)
\end{aligned}$$

3.11 Incomplete ontology and damaged documents

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.

In case of the Biobibliographical Lexicon 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.

On the other hand, in case of Sumerian economic documents we do not have syntactic features. The **Other Information** concept is always treated as a transaction attribute. Parser recognises phrases it does not understand using the following heuristics: Boundaries of transaction attributes are correlated with verse boundaries. If the parser does not recognise the content of a verse it decides that this verse contains **Other Information**.

Apart from being not understood the Sumerian documents are often damaged. There are several types of damages in documents.

When a single sign is illegible, it is denoted in the document as `x`. The parser considers the `x` symbol as a wildcard that may be matched with any terminal symbol.

When a part of a verse is broken, it is denoted in the document by `[...]`. We estimate the number of signs in the broken verse counting the number of signs in the other verses in the document. Then we replace the `[...]` symbol by a sequence of `x` symbols.

We do not estimate precise contents of broken phrases, we only determine their role in the document and ontological category. This allows us to avoid combinatorial explosion while parsing damaged documents.

When a whole verse or a number of verses is broken we assume that they have syntactic structure of a typical verse or contain **Other Information**.

3.12 Parsing results

For Sumerian economic documents we developed a grammar that consists of 9103 rules. Using this grammar we extracted 68619 transactions from the selected subcorpus with precision 86% and recall 90%.

The rules may be divided into the ones that constitute the lexicon (8900) and those which describe the document structure and follow ontological dependencies (203). The lexicon may be further divided into rules that recognise words and associate them with their ontological category (7510), and the ones that introduce domain knowledge (numbers, yearnames, etc) (1390).

In the case of the Biobibliographical lexicon we developed 120152 lexicon rules, 66 syntactic rules and 124 semantic rules. As we described in section 3.7 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%.

Syntactic information included in lexicon was generated using Morfeusz **??**, while semantic tagging was done manually. The so obtained results may be further improved if more words are semantically tagged.

Chapter 4

Knowledge representation

4.1 Background

In the previous chapter we have translated the documents written in the natural language into the formal meaning representation language formulae.

In this chapter we define the knowledge representation model which is a theoretical background for our meaning representation language. We consider the problem of how that information is encoded into the formulae, and we investigate the reference between the formulae and described reality.

We define the semantics of the meaning representation language in a model-theoretic fashion. However, the information provided by documents is incomplete and imprecise. It force us to introduce the concept of possible worlds. In order to handle the relationship between language and a set of possible worlds, we develop special symbols, which we denote *primitives*. They turn out to be closely connected to the ontology.

In this setting, we discuss the relevance of the introduction of constants for entity representation and disjunction for representing ambiguity in documents. In the end of the chapter, we bring up the issue of asking and answering questions. Although, our main goal is the analysis of textual data, we also show on examples, how our knowledge representation model works with non-lexical knowledge.

The idea of representing the semantics of a natural language in truth-conditional and model-theoretic way was first developed by Richard Montague [Montague, 1970a]. From that time semanticists created many formalisms such as Frame Semantics [Fillmore, 1976], Discourse Representation Theory [Heim, 1982, Kamp and Reyle, 1993, van Eijck and Kamp, 1997] and other [Lobner, 2002, Saeed, 2003]. However, it was typical for this field to examine in detail the chosen phenomenon, while abstracting away from other semantic phenomena. Only recently there is a trend to change that approach [Blackburn and Bos, 2005]. Knowledge representations were also studied in the context of artificial intelligence [Gruber, 1993, Russell and Norvig, 1995, Sowa, 1999, Staab and Studer, 2004].

The notion of primitives resembles the ideas of sorted logic [Zarba, 2006], yet it is more flexible.

The problem of relationship between the representation and the reality is known in the philosophy of language as the reference problem. Some ideas presented below corresponds to picture theory of language developed by Wittgenstein in [Wittgenstein, 1962]. Overview of various theories of reference in the computer science context is presented in [Hanseth and Monteiro, 1994].

The use of constants as entity representation is not typical in semantics, since most authors decide to use logical variables to denote entities [Blackburn and Bos, 2005], however, it has a long tradition in databases and brings our knowledge representation model closer to relational database model.

4.2 Reality perceived by sensors

We do not possess the direct insight into the nature of reality. We perceive it by means of sensors. Sensors generate structural data on the basis of reality, by extracting objects and relations among them. Sensors recognise properties of objects as well as relations between them. As examples of sensors, one can consider a thermometer, a camera as well as a document corpus or an expert, who communicates in a natural language. Generally, every analysis of the reality that results in structural data may be understood as obtained by means of sensor measurements.

We represent sensor measurements by means of relational structures developed using model theory [Mendelson, 1997]. That structures act as pictures of the world around (or at least, that part of the world we happen to be interested in for our application). In model theory structures are composed out of a set of individuals, and some relations among them. The set of all individuals is called as the universe of the structure. We also refer to the individuals as objects or entities.

Individuals and relations from a relational structure are represented by symbols. The set of such symbols is called a signature. The interpretation is a function that maps symbols from a given signature to individuals and relations. The interpretation assigns objects or relations to names. It assures that the object or relation represents the aspect of sensor activity stated by its name. For example, the symbol `red` is interpreted as an object that represents the red colour. The signature is a lexicon for every syntactic construct (e.g. language formula, information system, image) by means of which we describe sensor measurements.

Sensor measurements, as well as the documents which we process report facts concerning particular entities. They are free of general statements. This observation is crucial because it allows us to provide a distinct constant for each entity mentioned in the documents.

Let the structure $\mathcal{P}$ be a model of the reality observed by our sensors. We assume that $\mathcal{P}$ is a relational structure of signature $\Sigma_{\mathcal{P}}$ and with universe P.

$\Sigma_{\mathcal{P}}$ consists of symbols of constants $u_1, u_2, \ldots$ and relational symbols $a_1, a_2, \ldots$. In symbols, $\mathcal{P}$ can be described by:

$$\mathcal{P} = \langle P, u_1^{\mathcal{P}}, u_2^{\mathcal{P}}, \ldots, a_1^{\mathcal{P}}, a_2^{\mathcal{P}}, \ldots \rangle,$$

where P includes objects, $u_1^{\mathcal{P}}, u_2^{\mathcal{P}}, \ldots$, and $a_1^{\mathcal{P}}, a_2^{\mathcal{P}}, \ldots$ are interpretations of the symbols from $\Sigma_{\mathcal{P}}$. Constants are mapped to individuals and relational symbols to relations.

Let us consider the fragment of Sumerian economic reality described by our example document (fig. 2.1). In this world we have relations described by the ontology (fig. 3.1) and entities classified by ontological categories. Entities are represented by constants such as `ab-ba-sa6-ga`, `sila4 ga`, p, n_1, n_2, etc. For each ontological concept a number of relational symbols with different arities is provided. For example for **Person** concept, there are `Person`$(\cdot)$, `Person`$(\cdot,\cdot)$, `Person`$(\cdot,\cdot,\cdot)$, `Person`$(\cdot,\cdot,\cdot,\cdot)$ and `Person`$(\cdot,\cdot,\cdot,\cdot,\cdot)$ symbols in signature.

Let us consider a digital camera as another example. We may define photos, the data produced by this sensor, as a structure whose universe is composed out of pixels, colours and objects that represent the picture itself. We introduce binary relations `horizontal neighbours` and `vertical neighbours` that define the topology of pixels and the 3-argument relation `colour of pixel` whose arguments are a picture, pixel and colour. The latter relation provide us an access to the picture contents.

Objects perceived by sensors need not necessarily be the real things. Objects may be epiphenomena, created by sensors (for example by human perception), nevertheless we need them in order to operate on the reality. While modelling the reality we do not recognise existing objects. We define them.

4.3 Semantics of knowledge

Knowledge, or information, provides us with an insight into $\mathcal{P}$, the model of the reality observed by sensors. Information about $\mathcal{P}$ is represented by means of symbols connected by syntactic rules. Languages, information systems, images, tables, time series etc. are examples of such representations.

Besides the symbolic representation we do not possess any insight into $\mathcal{P}$ and this representation describes the results of sensor measurements in an incomplete and ambiguous way. It includes information only for a part of sensors and objects of interest for us. For a given information there exist lots of different models consistent with this information.

We introduce semantics for all symbolic representations. This semantics allows us to define the symbolic representations in the formal way and translate information between such representations.

Definition We describe semantics of symbolic representation to be the class of structures, which we name as the class *possible worlds* and denote as $\mathbb{P}$. Every $\mathcal{Q} \in \mathbb{P}$ is a model of possible reality, i.e., the world that does not contradict our knowledge. Each $\mathcal{Q} \in \mathbb{P}$ has signature $\Sigma_{\mathcal{Q}}$ and interpretation $I_{\mathcal{Q}}$.

The class of structures $\mathbb{P}$ defines knowledge independently from the syntactic medium. Every $\mathcal{Q} \in \mathbb{P}$ describes sensor measurements in a precise way ($\mathcal{Q}$ describes also the possible reality with the precision relative to the sensor precision). Imprecise description provided by the symbolic representation is interpreted as a collection of precise descriptions. Possible worlds consist of all possible extensions of the set of given sensor measurements without any estimation or inductive reasoning over the unknown measurements. We introduce multiple possible worlds to represent the incompleteness of us knowledge, the $\mathcal{P} \in \mathbb{P}$ statement defines its correctness: the knowledge is correct if the real world belongs to the class of possible ones.

For each $\mathcal{Q} \in \mathbb{P}$ its signature $\Sigma_{\mathcal{Q}}$ is a set of atomic symbols available for symbolic representation. Signatures differ across the possible worlds since some symbols may be unknown to us or we may assume existence of something that in fact does not really exist.

4.4 Primitives and ontology

The interpretation is a link between sensor measurements and symbols. Some of the symbols refer to the sensor construction. Interpretations of these symbols should be correlated among the possible worlds. In the digital camera example constants that represent pixels and colours as well as relations `horizontal neighbours` and `vertical neighbours` should be identical in all possible worlds.

In order to formalise the above observation we introduce *primitives*. Primitive constants denote the atomic components of sensor. Since we may have only a partial knowledge about sensor construction, some primitive constants might be unknown in some of possible worlds. Yet, in every possible world, if the primitive constant belongs to the signature it has the same interpretation as in real world.

On the other hand, primitive relations describe the way in which sensor measurements are constructed. For each possible world, for each subset of universe of this world, which consists only of real objects each primitive relation is identical as in real world.

Formally we define *primitives* as follows:

Definition For a given model of the reality $\mathcal{P}$ and a class possible worlds $\mathbb{P}$, we define *Primitives* to be symbols σ such that

- For each primitive symbol σ
$$\sigma \in \Sigma_{\mathcal{P}}.$$
- For each constant primitive symbol σ and for each $\mathcal{Q} \in \mathbb{P}$ if $\sigma \in \Sigma_{\mathcal{Q}}$ then
$$I_{\mathcal{P}}(\sigma) = I_{\mathcal{Q}}(\sigma).$$

- For each n-ary relational primitive symbol σ and for each $\mathcal{Q} \in \mathbb{P}$ if $\sigma \in \Sigma_{\mathcal{Q}}$ then
$$\forall_{u_1,\ldots,u_n \in P \cap Q} \; I_{\mathcal{P}}(\sigma)(u_1, \ldots, u_n) = I_{\mathcal{Q}}(\sigma)(u_1, \ldots, u_n).$$

In other words, each primitive has the same interpretation for all real objects in every possible world. The primitives for unreal objects may be defined in an arbitrary way. Universes of various possible worlds may differ, yet primitives provides a link between them.

We require that the interpretations of all possible worlds to preserve the constraints defined by primitives.

In case of textual data it is an ontology that provides the structure for data. We demand for ontological categories to be defined in the same way in all possible worlds. However, the properties that distinguish between objects belonging to the same ontological category depend on the possible world. E.g. an object p which is a **Person** in one possible world must be a **Person** in all other possible worlds in which he/she is present, but may have a name n_1 in one possible world and a name n_2 in another one.

We achieve this result declaring one argument predicates that define ontological categories and Sumerian signs as primitives. For example `Person`$(\cdot)$, `ab-ba-sa6-ga`, `sila4 ga` are primitive, while `Person`$(\cdot, \cdot)$, `Person`$(\cdot, \cdot, \cdot)$, `Person`$(\cdot, \cdot, \cdot, \cdot)$ and `Person`$(\cdot, \cdot, \cdot, \cdot, \cdot)$ and p, n_1, n_2 are not.

In this way the primitives solve the reference problem and provide a firm link between the language and the underlying reality.

4.5 Language

Signature symbols denote basic concepts given by sensors. Complex concepts are defined out of basic ones by means of language. A language is a set of syntactic rules for connecting signature symbols. Each syntactic rule generate a language formula. For each rule there is provided a method of calculation the interpretation of the formula generated by the rule .

Let the relation $\models$ be given between structures and formulae. The classical $\models$ is one of the possible choices.

Definition Let $\mathbb{A}$ be a set of language formulae without free variables. We say that $\mathbb{A}$ is *valid* in the structure $\mathcal{Q}$ iff for each formula $a \in \mathbb{A}$, truth is an interpretation of a in the structure $\mathcal{Q}$. We denote the above as
$$\mathcal{Q} \models \mathbb{A}.$$

We use language to set constraints on the set of possible worlds. We encode sensor properties, domain knowledge and sensor measurements by a set of language formulae $\mathbb{A}$, which we call axioms. Then we define the class $\mathbb{P}(\mathbb{A})$ of possible realities, as a class of structures that in which axioms $\mathbb{A}$ are valid:
$$\mathbb{P}(\mathbb{A}) = \{\mathcal{Q} : \mathcal{Q} \models \mathbb{A}\}.$$

Definition We say that a language formula φ is a *semantic consequence* of axioms $\mathbb{A}$ iff for each structure $\mathcal{Q}$, such that $\mathcal{Q} \models \mathbb{A}$,

$$\mathcal{Q} \models \varphi.$$

We denote the above as

$$\mathbb{A} \models \varphi.$$

We use first-order logic with the classical $\models$ as a language for knowledge representation. In its terms we define constraints for symbols whose interpretations depend on the nature of the sensor.

For example: the requirement that the arguments of a relation belong to proper categories is expressed by the formula that states that if a relation is true for a certain objects as its arguments, then these objects must belong to a certain categories. The category itself is determined by a primitive relation. For example

$$\forall_{x,y}\ \texttt{Person}(x,y) \Longrightarrow \texttt{Person}(x) \wedge$$
$$\wedge \big(\texttt{Name}(y) \vee \texttt{Job}(y) \vee \texttt{Filiation}(y) \vee \texttt{Nationality}(y)\big).$$

Similarly, we declare that the order of predicate arguments (apart from the first one) is irrelevant:

$$\forall_{x,y,z}\ \texttt{Person}(x,y,z) \Longleftrightarrow \texttt{Person}(x,z,y).$$

Similarly, we define relationships between predicates with different number of arguments

$$\forall_{x,y,z}\ \texttt{Person}(x,y,z) \Longrightarrow \texttt{Person}(x,y).$$

The above formulae are axioms which we require to be satisfied in every possible world. Together with underlying primitives they provide an axiomatic description of world and language which we use. Such an approach assures us that the given symbol represents corresponding relations in all possible worlds. For example, the relation denoted by symbol $\texttt{Person}(\cdot,\cdot)$ may vary in different possible worlds, yet it is desired for it to describe persons in each world.

We use prepositional logic (i.e., first order quantifier free formulae) as the meaning representation language, which defines values of sensor measurements (store the contents of the documents).

We connect different possible subsequence interpretations by means of disjunction. Domain knowledge provides constraints which may make the disjunction mutually exclusive. For example we may state that one **Name** cannot be written by two different sign sequences:

$$\forall_{x,y,z}\ \texttt{Name}(x,y) \wedge \texttt{Name}(z,y) \Longrightarrow x \neq z$$

Of course not all disjunctions should be treated as mutually exclusive, for example

$$\forall_{x}\ \texttt{Animal}(x, \texttt{sila4 ga}) \Longrightarrow \texttt{Animal}(x, \texttt{sila4})$$

Summarising, the knowledge is provided in three ways: Primitives defined in metalanguage assure the connection between symbols and elements of sensor measurements. Axioms expressed as language formulae describe the properties of sensors. Language is used also to formulate axioms that define values of sensor measurements.

4.6 Semantic query language

Having the formal representation of the text as a formula of the meaning representation language we can look for information using concepts from the documents. Syntax of the query language is defined as follows: we have the set of constants, a set of variables and a set of predicate names. Queries are composed of one or more predicates connected by conjunctions and/or disjunctions. The semantics of queries is defined as follows:

Definition Let $\mathbb{A}$ be a set of axioms that describe sensor properties and define values of sensor measurements. Query q matches to the sensor measurements when there exists a substitution θ of variables in q such that

$$\mathbb{A} \models q\theta$$

We have guarantee that we find all patterns that match the query, no matter how they are expressed.

We can construct quite complex questions. For example we may query for all documents concerning `Transactions` *where* `ab-ba-sa6-ga` *was a* `Receiver` *of goods provided by any governor* (`ensi2`):

$$\mathtt{Name}(x_1, \mathtt{ab} - \mathtt{ba} - \mathtt{sa6} - \mathtt{ga}) \wedge \mathtt{Person}(x_2, x_1) \wedge \mathtt{Receiver}(x_3, x_2) \wedge$$

$$\wedge \mathtt{Job}(x_4, \mathtt{ensi2}) \wedge \mathtt{Person}(x_5, x_4) \wedge \mathtt{Supplier}(x_6, x_5) \wedge \mathtt{Transaction}(x_7, x_3, x_6)$$

Since we do not expect from users of our system (ex. sumerologists or literary scholars) to understand our meaning representation language, we use language specific for a given document corpus as an interface for our query system. User simply writes a phrase in a natural language enriched with words that denotes ontological categories. The phrase is parsed to a formula of the meaning representation language, which is then used as a query.

Chapter 5

Inference

5.1 Background

In this chapter we pursue our ultimate goal: we develop methods for inferring knowledge from document collections translated into the meaning representation language formulae. We show how to generalise information provided by meaning representation language formulae. We seek for general truths and generalisations which not need to be 100% correct.

In order to realise this goal we introduce methods of inductive reasoning. We combine the rough set theory [Pawlak, 1982, Pawlak, 1991] and the statistical learning theory [Vapnik, 1998], which are two different methodologies for reasoning from data.

The rough set concept theory is a theoretical framework for describing and inferring knowledge. Examined knowledge is imperfect. It is imprecise due to vague concepts involved in knowledge representation and it is based on incomplete data. The central point of the theory is the idea of concept approximation by the set of objects that certainly belong to the concept and the set of those which may belong to the concept on the basis of possessed data. Then these two sets are described in terms of available attributes.

The main goal of statistical learning theory is to provide a framework for studying the problem of inference. For this purpose, there are introduced statistical assumptions about the way the data is generated. A probabilistic model of data generation process, which is the core of the theory, establishes the formalisation of relationships between past and future observations.

While rough set theory provides an intuitive description of relationships in data and approximations for dependencies that cannot be defined in an exact way, statistical learning theory measures the significance and correctness of discovered dependencies.

Both, rough set theory and statistical learning theory are based on the analysis of information systems consisting of sets of objects characterised by attribute value vectors. Since our our knowledge representation model is based on axiomatic representation of data we must extend rough set theory so that it could operate on data provided in a form of axioms.

We propose axiomatic representation of information systems and we define rough set concepts such as indiscernibility, definability and set approximations in this setting. We prove that, in the case of complete information systems, the proposed approach is equivalent to the approach used so far in rough set theory [Pawlak, 1982, Pawlak, 1991]. We compare our idea with extensions of rough set theory for information systems with missing values and multiple valued attributes presented in [Grzymała-Busse and Grzymała-Busse, 2007, Grzymala-Busse, 2006, Kryszkiewicz, 1998b, Kryszkiewicz, 1998a, Latkowski, 2003, Latkowski, 2004, Latkowski and Mikołajczyk, 2004, Latkowski, 2005, Lipski, 1981, Demri and Orłowska, 2002, Pawlak, 1981].

Then we show how to obtain axiomatic version of information system from meaning representation language formulae. In our case missing values refer to incomplete information included in textual data and multiple valued attributes are a simple example of ambiguity. The task of information generalisation is now mapped into the problem of seeking for dependencies between attribute values of objects in such system.

We define the probabilistic model of data generation process, which allow us to explain the process of data acquisition and to infer knowledge that would be applied for all existing objects, not only for the ones that are mentioned in data.

Then we use rough set theory together with probabilistic model of data generation process to infer knowledge from the document corpora. We concentrate on a specific type of inductive reasoning called classification.

We show how to extend set approximations from a sample to the set of all objects. Our attitude is similar to the idea of inductive extensions of approximation spaces presented, for example, in [Pawlak and Skowron, 2007, Skowron et al., 2005].

We introduce measures of approximation quality: accuracy and coverage. Taking advantage of the underlying probabilistic model we estimate values of the above indices on the set of all objects using a sample. We propose two estimators: one based on Hoeffding inequality [Hoeffding, 1963], and second based on the optimal probability bound presented in [Jaworski, 2005, Jaworski, 2006a].

The statistical nature of estimators leads us to the index, the measure called significance. Significance measures how often sample-based accuracy and coverage estimations are correct. The trade-off relation between these three measures allow us to balance the approximation between fitting to the sample and generalisation.

In order to show how the estimators behave in practice we developed a simple rule-based classifier. Estimated indices guarantee the quality of each rule, determine the required accuracy level for rule to be accepted and decide how many objects have to match the rule in order to make it significant.

Test results reveal that the obtained classifier generates highly relevant rules. Each rule is assigned with its accuracy and coverage estimations. Rules cover only that part of universe for which it is possible to predict decision with high

Table 5.1: A complete information system

	name	age
p_1	Alice	young
p_2	Alice	old
p_3	Bob	young
p_4	Bob	old
p_5	Bob	young

accuracy. As a consequence the classifier is able to judge whether it has enough knowledge to classify a certain object.

The properties of accuracy and coverage were thoroughly studied in [Tsumoto, 2002]. The author proposed a probabilistic definition of the indices, yet he neither defined any underlying probability model nor showed the trade-off between accuracy or coverage and significance. Quality measures were also examined from the statistical point of view in [Guillet and Hamilton, 2007], but without placing them in the rough set context.

[Gediga and Düntsch, 2000] propose an application of statistical techniques in rough set data analysis, yet they did not incorporate the assumptions on the data generating process required by these techniques into the presented model.

5.2 Complete information systems

In this section, we formally define information systems [Pawlak, 1981] and we propose their axiomatic representation, which covers both contents and structure of information systems.

Information systems are based on the assumption that examined domain is organised in terms of *objects* possessing *attributes*. Depending on the nature of domain, objects are interpreted as, e.g. cases, states, processes, patients, observations. Attributes are interpreted as features, variables, characteristics, conditions, etc.

Let U be a non-empty, finite set of known objects. Let A be a non-empty finite set of known attributes. Each attribute $a \in A$ has its domain V_a. An information system defines attribute values for given objects. Let

$$m(u, a)$$

denote the set of values of the attribute a for the object u in the information system.

Usually information systems are presented in a form of tables whose rows represent objects and columns are labelled by attributes. An example of information systems is presented in Table 5.1, we denote this system by $\mathcal{A}_1$. In Table 5.1, $U = \{p_1, p_2, p_3, p_4, p_5\}$, $A = \{\mathtt{name}, \mathtt{age}\}$, $V_{\mathtt{name}} = \{\mathtt{Alice}, \mathtt{Bob}\}$ and $V_{\mathtt{age}} = \{\mathtt{young}, \mathtt{old}\}$.

Usually each attribute has exactly one value for each object, i.e $m(u, a)$ contains one element for every u and a. In such a case information system is called *complete*.

We consider an information system as a sensor whose measurements determine a set of possible worlds. Constants that represent attribute values are primitives. We introduce the primitive relation `object` which collects objects described in the information system. For each $u \in U$ the statement $\texttt{object}(u)$ is true, yet the relation `object` is broader: it contains all objects that the sensor has perceived, is perceiving and will perceive.

The information system provides also the structural information about the domains of the attributes. We represent this information by means of axioms that set constraints on a set of possible worlds. For each attribute a we state

$$\forall_{x,y}\ a(x,y) \Longrightarrow \texttt{object}(x) \wedge y \in V_a.$$

The complete information system also states that every attribute has exactly one value for each object: for each attribute a we write the following axiom

$$\forall_x \left(\texttt{object}(x) \Longrightarrow \exists!_y\ a(x,y)\right).$$

We encode the contents of the information system as a set formulae in the following way: For each $u \in U$, for each $a \in A$ such that $v \in m(u, a)$ in the information system we add the following axiom:

$$a(u, v).$$

The above transformation treats both an object etiquette and an attribute value as constants. The attributes are considered as binary relations.

The above set of axioms, define the class of possible worlds determined by an information system.

For Table 5.1, our knowledge provided by the information system $\mathcal{A}_1$ is restricted to the following axiom set $\mathbb{A}$:

$$\forall_{x,y}\ \texttt{name}(x,y) \Longrightarrow \texttt{object}(x) \wedge y \in V_{\texttt{name}},$$

$$\forall_{x,y}\ \texttt{age}(x,y) \Longrightarrow \texttt{object}(x) \wedge y \in V_{\texttt{age}},$$

$$\forall_x \left(\texttt{object}(x) \Longrightarrow \exists!_y\ \texttt{name}(x,y)\right),$$

$$\forall_x \left(\texttt{object}(x) \Longrightarrow \exists!_y\ \texttt{age}(x,y)\right),$$

$$\texttt{name}(p_1, \texttt{Alice}), \texttt{age}(p_1, \texttt{young}), \texttt{name}(p_2, \texttt{Alice}), \texttt{age}(p_2, \texttt{old}), \texttt{name}(p_3, \texttt{Bob}),$$

$$\texttt{age}(p_3, \texttt{young}), \texttt{name}(p_4, \texttt{Bob}), \texttt{age}(p_4, \texttt{old}), \texttt{name}(p_5, \texttt{Bob}), \texttt{age}(p_5, \texttt{young}).$$

Axioms derived from Table 5.1 allow us to define many different structures. The set of possible worlds $\mathbb{P}(\mathbb{A})$ consists of all the possible extensions of I_1. $\mathbb{P}(\mathbb{A})$ will include

$$\mathcal{Q}_1 = \langle Q_1, p_1^{\mathcal{Q}_1}, p_2^{\mathcal{Q}_1}, \ldots, p_5^{\mathcal{Q}_1}, \texttt{Alice}^{\mathcal{Q}_1}, \texttt{Bob}^{\mathcal{Q}_1}, \texttt{young}^{\mathcal{Q}_1}, \texttt{old}^{\mathcal{Q}_1}, \texttt{name}^{\mathcal{Q}_1}, \texttt{age}^{\mathcal{Q}_1}\rangle$$

as well as

$$\mathcal{Q}_2 = \langle Q_2, p_1^{\mathcal{Q}_2}, p_2^{\mathcal{Q}_2}, \ldots, p_{10}^{\mathcal{Q}_2}, \mathtt{Alice}^{\mathcal{Q}_2}, \mathtt{Bob}^{\mathcal{Q}_2}, \mathtt{young}^{\mathcal{Q}_2}, \mathtt{old}^{\mathcal{Q}_2}, \mathtt{name}^{\mathcal{Q}_2}, \mathtt{age}^{\mathcal{Q}_2}\rangle$$

and

$$\mathcal{Q}_3 = \langle Q_3, p_1^{\mathcal{Q}_3}, p_2^{\mathcal{Q}_3}, \ldots, p_5^{\mathcal{Q}_3},$$
$$\mathtt{Alice}^{\mathcal{Q}_3}, \mathtt{Bob}^{\mathcal{Q}_3}, \mathtt{young}^{\mathcal{Q}_3}, \mathtt{old}^{\mathcal{Q}_3}, \mathtt{name}^{\mathcal{Q}_3}, \mathtt{age}^{\mathcal{Q}_3}, \mathtt{parent}^{\mathcal{Q}_3}\rangle.$$

Axioms do not provide any information about objects $p_6, \ldots, p_{10}$. They may have arbitrary properties. Relations `name`, `age` and `parent` can be specified in any way that satisfies $\mathbb{A}$. They do not need to be consistent with their counterpart in the world perceived by sensor.

5.3 Rough set theory

Rough set theory [Pawlak, 1982, Pawlak, 1991] is based on the idea of an indiscernibility relation. In this section, we define indiscernibility and set approximations for complete information systems. In the following sections we extend the definitions to the case of missing values and multivariate attributes.

In this and the following sections we assume that we are given an information system $\mathcal{A}$. U denote the finite set of objects described in $\mathcal{A}$, A is the finite set of attributes in $\mathcal{A}$ and $\mathbb{A}$ is a set of axioms derived from $\mathcal{A}$.

Let B be a nonempty subset of A. The indiscernibility relation $IND(B)$ is a relation on objects in a complete information system defined for $x, y \in U$ as follows

$$(x, y) \in IND(B) \text{ iff } \forall a \in B\big(m(x, a) = m(x, a)\big).$$

For example, for Table 5.1, p_3 and p_5 are indiscernible with respect to the attributes `name` and `age`.

IND is an equivalence relation. We will denote its equivalence class generated by object u as

$$[u]_{IND(B)}.$$

Definition For a given set of attributes $B \subseteq A$, formulae of the form

$$a(x, v),$$

where $a \in A$, $v \in V_a$ and x is a free variable, are called *descriptors* over B.

Definition By a *query* over the set of attributes B we denote any formula

$$\bigwedge_{i=1}^{n} \varphi_i(x),$$

where each φ_i is a descriptor over B and $n \leq |B|$. x is a free variable ranging over objects.

Consider the query:

$$\varphi(x) := \texttt{name}(x, \texttt{Bob}) \wedge \texttt{age}(x, \texttt{young}).$$

This formula is satisfied either if p_3 is the value of x or its value is p_5. p_3 and p_5 cannot be distinguished by formula $\varphi(x)$.

Definition The set of *conditional formulae* over B is defined as the least set containing all descriptors over B and closed with respect to the propositional connectives $\wedge$ (conjunction), $\vee$ (disjunction) and $\neg$ (negation).

Note that every query is a conditional formula.

Definition Let $\varphi(x)$ be a conditional formula. By $||\varphi(x)||_{U,\mathbb{A}}$ we will denote the set of all elements from U for which φ is a semantic consequence of $\mathbb{A}$, i.e.:

$$||\varphi(x)||_{U,\mathbb{A}} = \{x \in U \mid \mathbb{A} \models \varphi(x)\}.$$

We postulate the following definition of indiscernibility:

Definition Let $\varphi(x)$ be a query with free variable x. Let u_1 and u_2 be constants. We say that u_1 and u_2 are *indiscernible by the query* $\varphi(x)$ if

$$\big(\mathbb{A} \models \varphi(u_1)\big) \Longleftrightarrow \big(\mathbb{A} \models \varphi(u_2)\big).$$

Theorem 5.3.1 *Let $\mathcal{A}$ be a complete information system. Let B be a subset of A. Objects $u_1 \in U$ and $u_2 \in U$ are indiscernible with respect to attribute set B iff they are indiscernible with respect to every query over the set of attributes B.*

Proof If $(u_1, u_2) \in IND(B)$, for every $a \in B$ there exists v_a such that

$$m(u_1, a) = m(u_2, a) = \{v_a\}.$$

Then the set of formulae $\{a(u_1, v_a) : a \in B\}$ is a subset of $\mathbb{A}$ and the set of formulae $\{a(u_2, v_a) : a \in B\}$ is a subset of $\mathbb{A}$. So for all $a \in B$ and $v \in V_a$ we have

$$\mathbb{A} \models a(u_1, v) \Leftrightarrow \mathbb{A} \models a(u_2, v).$$

Thus for every query $\varphi(x)$ we obtain $\mathbb{A} \models \varphi(u_1) \Leftrightarrow \mathbb{A} \models \varphi(u_2)$.

If $(u_1, u_2) \notin IND(B)$ we have $a \in B$ and v_1, v_2 such that $v_1 \neq v_2$, $m(u_1, a) = \{v_1\}$ and $m(u_2, a) = \{v_2\}$. Thus

$$\mathbb{A} \models a(u_1, v_1) \wedge \mathbb{A} \not\models a(u_2, v_1)$$

and the query $\varphi(x) = a(x, v_1)$ distinguishes u_1 and u_2. ▮

Assume that we have two sensors. Measurements of both of them are represented by information systems and these information systems share their sets of objects, i.e. the relation `object` is identical for both sensors. We wish to describe the measurements of one sensor by means of measurements of the second one. Alternatively we say that we are describing the value of an attribute in an information system (which we call the decision attribute and decision system) by the values of the remaining of attributes. We may reduce this problem to the problem of description of the set objects in the information system for which the decision attribute has a certain fixed value. Such a set is either *definable* or *indefinable* by other attributes.

Definition Let X be a subset of U. We say that X is *definable* by $\mathbb{A}$ iff there exist queries $\varphi_1(x), \ldots, \varphi_n(x)$ such that

$$X = ||\varphi_1(x) \vee \cdots \vee \varphi_n(x)||_{U,\mathbb{A}}$$

Each definable set is a sum of objects that satisfy at least one of a given queries.

Proposition 5.3.2 *X is* definable *by $\mathbb{A}$ iff X is an union of equivalence classes of $IND(A)$.*

Proof X is definable by $\mathbb{A}$ if and only if

$$X = \bigcup_{i=1}^{n} \{u \in U : \mathbb{A} \models \varphi_i(u)\}.$$

Theorem 5.3.1 states that $u_1, u_2 \in \{u \in U : \mathbb{A} \models \varphi_i(u)\}$ iff $(u_1, u_2) \in IND(A)$. ▮

Any set $X \subset U$ may be approximated by two definable sets. The first one is called the *lower approximation* of X, denoted by $\underline{\mathbb{A}}X$, and is defined by

$$\bigcup\{Y \mid Y \subset X \wedge Y \text{ is definable by } \mathbb{A}\}.$$

The second set is called the *upper approximation* of X, denoted by $\overline{\mathbb{A}}X$, and is defined by

$$\bigcap\{Y \mid X \subset Y \wedge Y \text{ is definable by } \mathbb{A}\}.$$

$\overline{\mathbb{A}}X \subset U$ because every definable set is a subset of U.

Proposition 5.3.3 *The lower and the upper approximations of any set $X \subset U$ are definable.*

Proof For a given information system, there is a finite number of definable sets. Thus

$$\bigcup\{Y \mid Y \subset X \wedge Y \text{ is definable}\} = Y_1 \cup \cdots \cup Y_n,$$

where Y_i is defined by a formula $\varphi_i(x)$. Hence, $Y_1 \cup \cdots \cup Y_n$ is defined by $\varphi_1(x) \vee \cdots \vee \varphi_n(x)$. Similarly

$$\bigcap\{Y \mid X \subset Y \wedge Y \text{ is definable}\} = Y_1 \cap \cdots \cap Y_n,$$

where every Y_i is defined by a formula $\varphi_i(x)$. Hence, $Y_1 \cap \cdots \cap Y_n$ is defined by $\varphi_1(x) \wedge \cdots \wedge \varphi_n(x)$. The last formula may be transformed into a form of a disjunction of queries. ▮

Theorem 5.3.4

$$\underline{A}X = \underline{\mathbb{A}}X \textit{ and } \overline{A}X = \overline{\mathbb{A}}X.$$

Proof According to Prop. 5.3.2

$$\underline{\mathbb{A}}X = \bigcup\{[u_1]_{IND(A)} \cup \cdots \cup [u_n]_{IND(A)} \subset X\} =$$

$$= \bigcup\{[u]_{IND(A)} \mid [u]_{IND(A)} \subset X\} = \underline{A}X,$$

$$\overline{\mathbb{A}}X = \bigcap\{X \subset [u_1]_{IND(A)} \cup \cdots \cup [u_n]_{IND(A)}\}.$$

$IND(A)$ is an equivalence relation, so

$$\overline{\mathbb{A}}X = \bigcup\{[u]_{IND(A)} \mid [u]_{IND(A)} \cap X \neq \emptyset\} = \overline{\mathbb{A}}X.$$

▮

5.4 Incomplete data

Real-life data is frequently incomplete, i.e. values for some attributes are missing. We will assume three different interpretations of missing values:

- missing attribute values that are *lost*, i.e they are specified, yet their values are unknown
- attributes *not applicable* in a certain case, e.g. the date of death of a person who is still alive.
- *do not care values*: the attribute may have any value from its domain.

We will extend the definition of $m(u, a)$. $m(u, a) =?$ will mean that the value of attribute a for object u is lost, $m(u, a) = \star$ that it is 'do not care' and $m(u, a) = -$ that it is not applicable.

The problem of 'lost' and 'do not care' missing values was thoroughly studied (see e.g. [Grzymała-Busse and Grzymała-Busse, 2007, Grzymala-Busse, 2006, Kryszkiewicz, 1998b, Kryszkiewicz, 1998a]). The presented ideas are based on various modifications of indiscernibility relation so it could handle missing values and remain definable in terms of attributes.

The definitions of indiscernibility, definability, lower and upper approximation we have stated above, need not to be modified for information systems with missing values. They are equivalent to the definitions proposed in the cited papers.

We express the various types of missing value semantics using axioms:

- for each $u \in U$, for each $a \in A$ we state

  $$a(u, v),$$

  where $v \in m(u, a)$ in the information system.

Table 5.2: An information system with missing values

	name	date of death
p_1	Alice	-
p_2	Bob	1972
p_3	?	1987
p_4	$\star$	2100

- 'lost' values are defined as follows: for each $u \in U$, for each $a \in A$ we state

$$a(u, v_1) \vee \cdots \vee a(u, v_n),$$

where $v_1, \ldots, v_n$ are all possible values of attribute a.

- for each $u \in U$, for each $a \in A$ whose value is not applicable we state

$$\forall_x \neg a(u, x),$$

- for each $u \in U$, for each $a \in A$, for each v from the domain of a we state

$$a(u, v),$$

when the value of a is 'do not care' for object u.

We may describe contents of Table 5.2 using the following set of axioms (we assume that $V_{\texttt{name}} = \{\texttt{Alice}, \texttt{Bob}, \texttt{John}\}$):

$$\texttt{name}(p_1, \texttt{Alice}), \forall_x \neg \texttt{date of death}(p_1, x),$$

$$\texttt{name}(p_2, \texttt{Bob}), \texttt{date of death}(p_2, 1972),$$

$$\big(\texttt{name}(p_3, \texttt{Alice}) \vee \texttt{name}(p_3, \texttt{Bob}) \vee \texttt{name}(p_3, \texttt{John})\big), \texttt{date of death}(p_3, 1987),$$

$$\texttt{name}(p_4, \texttt{Alice}), \texttt{name}(p_4, \texttt{Bob}), \texttt{name}(p_4, \texttt{John}), \texttt{date of death}(p_4, 2100).$$

Since indiscernibility with respect to a set of attributes does not work for incomplete information systems authors used to extend it or replace it by other concepts. The extension proposed in [Kryszkiewicz, 1998b] for the information systems with 'do not care' missing values is the relation

$$(x, y) \in SIM(B) \text{ iff } \forall a \in B\big(m(x, a) = \star \vee m(y, a) = \star \vee m(x, a) = m(y, a)\big).$$

Theorem 5.4.1 *Let $\mathcal{A}$ be an information system with 'do not care' missing vales. Let B be a subset of A. If objects $u_1 \in U$ and $u_2 \in U$ are indiscernible with respect to every query over the set of attributes B then*

$$(u_1, u_2) \in SIM(B).$$

The reverse implication is not valid for information systems with nontrivial missing values.

Proof If $(u_1, u_2) \notin SIM(B)$ we have $a \in B$ and $v_1, v_2 \in V_a$ such that $v_1 \neq v_2$, $m(u_1, a) = \{v_1\}$ and $m(u_2, a) = \{v_2\}$. Thus

$$\mathbb{A} \models a(u_1, v_1) \text{ and } \mathbb{A} \not\models a(u_2, v_1)$$

and the query $\varphi(x) = a(x, v)$ distinguishes u_1 and u_2.

For the case of reverse implication let us consider Table 5.2. We have

$$(p_2, p_4) \in SIM(\{\texttt{name}\}),$$

yet the query

$$\texttt{name}(x, \texttt{Alice})$$

distinguish them. ▮

In [Grzymała-Busse and Grzymała-Busse, 2007] another approach to 'do not care' and 'lost' missing values is presented. The indiscernibility with respect to the set of attributes is replaced by the concept of characteristic set:

Definition For an object $u \in U$ the *characteristic set* $K_A(u)$ is defined as

$$K_A(u) = \bigcap_{a \in A} K(u, a),$$

where $K(u, a)$ is defined in the following way

- if $m(u, a) = \{v\}$ then

$$K(u, a) = \{u' \in U \mid m(u', a) = \{v\} \vee m(u', a) = \star\}.$$

- if $m(u, a) = ?$ or $m(u, a) = \star$ then $K(u, a) = U$.

Lemma 5.4.2 *Let $\mathcal{A}$ be an information system with 'lost' and 'do not care' missing vales. For every $a \in A$ and for each $u \in U$ such that $m(u, a) = \{v\}$*

$$x \in K(u, a) \Longleftrightarrow \mathbb{A} \models a(x, v).$$

Proof Let x be an element of $K(u, a)$. Then $m(u, a) = \{v\}$ or $m(u, a) = \star$. In both cases $a(x, v)$ is satisfied by $\mathbb{A}$.

If $\mathbb{A} \models a(x, v)$, then either the value of a on x is specified as v or it is 'do not care' missing value. In both cases $x \in K(u, a)$. ▮

Theorem 5.4.3 *Let $\mathcal{A}$ be an information system with 'lost' and 'do not care' missing vales. If X is a union of characteristic sets, then a set $X \subset U$ is definable.*

Proof Assume that X is a union of characteristic sets for objects $u_1, \ldots, u_n$:

$$X = \bigcup_{i=0}^{n} K_A(u_i) = \bigcup_{i=0}^{n} \bigcap_{a \in A} K(u_i, a) = \bigcup_{i=0}^{n} \bigcap_{a \in A_i} K(u_i, a),$$

where A_i is the set of all attributes specified for u_i. Let $m(u_i, a) = \{v_{a,i}\}$. According to Lemma 5.4.2

$$K(u_i, a) = \{x \in U \mid \mathbb{A} \models a(x, v_{a,i})\}.$$

Thus $x \in X$ iff

$$\mathbb{A} \models \bigvee_{i=0}^{n} \bigwedge_{a \in A_i} a(x, v_{a,i}).$$

▮

Theorem 5.4.4 *Let $\mathcal{A}$ be an information system with 'lost' and 'do not care' missing vales. For each $X \subset U$*

$$\underline{\mathbb{A}}X \supseteq \bigcup\{K_A(x) \mid K_A(x) \subset X\},$$

$$\overline{\mathbb{A}}X \subseteq \bigcup\{K_A(x) \mid x \in U, K_A(x) \cap X \neq \emptyset\}.$$

Proof Let $K_X = \bigcup\{K_A(x) \mid K_A(x) \subset X\}$. K_X is a union of characteristic sets, so according to Thm. 5.4.3 K_X is definable by $\mathbb{A}$. Since $K_X \subseteq X$ we obtain $\underline{\mathbb{A}}X \supseteq K_X$. ▮

We remark that lower and upper approximations are more precise than *subset* lower and upper approximations (defined in [Grzymała-Busse and Grzymała-Busse, 2007]).

5.5 Multiple valued attributes

Multiple valued attributes (introduced in [Pawlak, 1981] and studied in [Lipski, 1981]) may reflect our incomplete knowledge about their values, what makes them similar to 'lost' missing values. The may also represent attributes that have a few values simultaneously, in which case the are like 'do not care' missing values.

- 'lost' multiple values we define as follows: for each $u \in U$, for each $a \in A$ we state
$$a(u, v_1) \vee \cdots \vee a(u, v_n),$$
where $v_1, \ldots, v_n$ are all possible values of attribute a for object u mentioned in the information system.

- for each $u \in U$, for each $a \in A$, for each value v of attribute a for object u in information system
$$a(u, v),$$
when the value of a is 'do not care' multiple value for object u.

Table 5.3: A multiple valued information system

	filiation	name
p_5	wife	Bob, David
p_6	son	David, Alice

Table 5.3 presents object p_5 who is a wife of Bob or David and object p_6 who is a son of David and Alice. Table 5.3 will be described by the following formula:

$$\texttt{filiation}(p_5,\texttt{wife}),\ \texttt{name}(p_5,\texttt{Bob}) \vee \texttt{name}(p_5,\texttt{David}),$$

$$\texttt{filiation}(p_6,\texttt{son}),\ \texttt{name}(p_6,\texttt{David}),\ \texttt{name}(p_6,\texttt{Alice}).$$

Multiple valued attributes are a simple extension of the 'missing values' case and the whole theory derived for the information systems with missing attributes is applicable here.

5.6 Textual data represented as vectors of feature values

In this section, we transform meaning representation language formulae into the form of information system, so that we could use rough set theory for analysis of information extracted from document corpora. We take advantage of the fact that the ontology is a hierarchical structure. From basic concepts like **Words** or **Signs** compound concepts are constructed, by means of which consecutive more and more complex concepts are defined, until the root concept is reached. In our application domains these root concepts are the **Transaction** concept and seven types event concepts.

For each root concept in the ontology, we generate a separate information system, whose objects are members of the selected root concept. Then we flatten the ontology. For each path from the root concept to the **Word** (or **Sign**) concept in the ontology tree we create an attribute. The **Word** (or **Sign**) related to the path becomes an attribute value. Although, in theory due to the cyclic dependencies in the ontology there is an infinite number of paths, we may obtain a finite vector of attributes eliminating ones that are undefined for all objects.

We bind attributes with the initial ontology by means of axioms. For our example Sumerian document (fig. 3.6) we define attributes $a_1, a_2, \ldots, a_{10}$ by axioms:

$$\forall_x a_0(x) \iff \texttt{Transaction}(x)$$

$$\forall_{x,y} a_1(x,y) \iff \exists_z \texttt{Transaction}(x,z) \wedge \texttt{Number}(z,y)$$

$$\forall_{x,y} a_2(x,y) \iff \exists_z \texttt{Transaction}(x,z) \wedge \texttt{Animal}(z,y)$$

$$\forall_{x,y} a_3(x,y) \iff \exists_{z_1,z_2,z_3} \mathtt{Transaction}(x,z_1) \wedge$$
$$\wedge \mathtt{Supplier}(z_1,z_2) \wedge \mathtt{Person}(z_2,z_3) \wedge \mathtt{Name}(z_3,y)$$

etc.

Existence and uniqueness of attributes $a_1, a_2, \ldots$ is a straightforward consequence of the above axioms.

After this transformation, we obtain the following information system:

	Transaction									
	Number	Animal	Supplier				Receiver	Day	Month	Year
			Person				Person			
			Name	Job	Filiation		Name			
						Person				
						Job				
	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
t_1	1	sila4	ur-mes	ensi2	?	?	ab-ba-sa6-ga	23	12	AS5
t_1	1	sila4	ur-mes	ensi2	?	?	ab-ba-sa6-ga	23	12	IS4
t_2	1	sila4	da-da	?	dumu	lugal	ab-ba-sa6-ga	23	12	AS5
t_2	1	sila4	da-da	?	dumu	lugal	ab-ba-sa6-ga	23	12	IS4
t_3	1	sila4	ga-ga-mu	?	?	?	ab-ba-sa6-ga	23	12	AS5
t_3	1	sila4	ga-ga-mu	?	?	?	ab-ba-sa6-ga	23	12	IS4
t_3	1	sila4 ga	ga-mu	?	?	?	ab-ba-sa6-ga	23	12	AS5
t_3	1	sila4 ga	ga-mu	?	?	?	ab-ba-sa6-ga	23	12	IS4

where t_1, t_2 and t_3 are transaction labels. Rows that are not divided with line present alternative attribute values for transactions.

In case of the example from the Biobibliographical Lexicon (3.7), two separate information systems are generated, since there are two types of root ontological concepts.

	PublicationEvent					
	PublicationEventType	Composition	Magazine		Date	
		CompositionType	MagazineType	MagazineTitle	Year	YearInterval
	b_1	b_2	b_3	b_4	b_5	b_6
e_1	drukował	wiersze	?	Robotnik	-	1927-28
e_2	drukował	wiersze	?	Nasz Przegląd	-	1927-29
e_3	drukował	wiersze	?	Ster	1937	-
e_4	drukował	wiersze	miesięcznik	Wymiary	1938	-
e_5	drukował	artykuły	?	Robotnik	-	1927-28
e_6	drukował	artykuły	?	Nasz Przegląd	-	1927-29
e_7	drukował	artykuły	?	Ster	1937	-
e_8	drukował	artykuły	miesięcznik	Wymiary	1938	-

	WorkEvent		
	EventState	Organisation	Job
		MagazineTitle	
	b_1	b_2	b_3
e_9	był	Wymiary	współredaktor

5.7 Probabilistic model

Our logic-based knowledge representation model allows us to represent incomplete and imprecise information. Rough set theory provides methods for approximate concepts that cannot be exactly defined in terms of available attributes.

However these theories do not explain the process of data acquisition and as a consequence they are unable to infer knowledge that would be applied for all existing objects, not only for the ones that are mentioned in data.

The knowledge representation model provides us with the class of possible worlds which consists of models of all worlds which does not contradict our knowledge. We assumed that there is model of reality $\mathcal{P}$ among possible worlds.

In previous section we described how to organise a given domain in terms of *objects* possessing *attributes*. We decided that for our application domains transactions and events will remain objects while all other entities will become attribute values, this decision is arbitrary and we may select members of any ontological concept to be treated as objects. Let $\mathbb{U}$ be a set of objects for a given domain. $\mathbb{U}$ is a subset of $\mathcal{P}$ universe consisting of objects belonging to our root concept. We assume that $\mathbb{U}$ is finite. Let A be a non-empty finite set of attributes such that every $a \in A$ is a relation on $\mathbb{U} \times V_a$, where V_a is a subset of $\mathcal{P}$ universe consisting of objects belonging to one of ontological concepts.

We propose the following definition of the problem of induction. We introduce a probability measure $P_\#$ on $2^{\mathbb{U}}$ according to the following formula:

$$P_\#(X) := \frac{|X|}{|\mathbb{U}|},$$

where $|\cdot|$ denotes the number of elements in a set.

Statistical learning theory [Vapnik, 1998] assumes that the phenomena underlying generated data have statistical nature and the observed objects are independent, identically distributed random variables.

Formally we introduce a probability space $(\Omega, 2^\Omega, P)$. Observed objects $u_1, u_2, \ldots, u_i, \ldots$ are values of independent random variables $U_1, U_2, \ldots, U_i, \ldots$. Each U_i is a function $U_i : \Omega \to \mathbb{U}$. The distribution of U_i is identical to $P_\#$, i.e.:

$$\forall_i \forall_{X \subseteq \mathbb{U}} P_\#(X) = P(\{\omega \in \Omega \mid U_i(\omega) \in X\}) = P(U_i^{-1}(X))$$

Since we do not have insight into $\mathcal{P}$, we do not know $\mathbb{U}$ and $P_\#$.

Let $U \subseteq \mathbb{U}$ be a non-empty, finite set of observed objects called a *sample*. U, together with the values of attributes for elements of U is our knowledge about the domain. We denote elements of U by $u_1, \ldots, u_n$, where u_i is a realisation (or value) of the random variable U_i. We represent this knowledge in terms of information system or equivalently as a set of axioms $\mathbb{A}$. Information concerning observed objects may be incomplete and imprecise.

5.8 Extended approximations

Now, we use rough set theory together with probabilistic model of data generation process to infer knowledge from the document corpora. We concentrate on a specific type of inductive reasoning called classification. On the contrary to previous chapters where the documents were analysed independently, now we infer knowledge from all document collection at once.

Classification is a process of finding dependencies between values of attributes. Let $\mathbb{A}$ be a given set of axioms which define attributes A for objects from the set U. We select one of attributes from A which we denote as d —

decision attribute. Let $B = A \setminus \{d\}$. Our goal is to estimate the value of attribute d on the basis of other attribute values for a given object. For each value v of the decision attribute, there exist conditional formulae over B that define the lower and upper approximation of $||d(x,v)||_{U,\mathbb{A}}$. We denote them $\underline{\varphi}_v(x)$ and $\overline{\varphi}_v(x)$ respectively.

$$||\underline{\varphi}_v(x)||_{U,\mathbb{A}} \subseteq ||d(x,v)||_{U,\mathbb{A}} \subseteq ||\overline{\varphi}_v(x)||_{U,\mathbb{A}}$$

Set approximations for all decision values compose a classifier.

We extend set approximations to the whole universe $\mathbb{U}$. The assumption that past and future observations are both sampled independently from the same distribution provides us with tools for extending the approximations. However, the extension will be correct only with some probability.

Inductive reasoning is based on the assumption that the definition generated for the sample data is still valid in the general case. For a given set of attributes B, *extended approximations* are represented by means of conditional formulae over B interpreted in the universe $\mathbb{U}$. Let φ be a conditional formula over B and let $||\varphi||_{\mathbb{U},\mathbb{A}}$ denote the subset of elements of the universe $\mathbb{U}$ that satisfy the formula.

For every U_i we obtain from its definition[1]

$$P_{\#}(||a(x,v)||_{\mathbb{U},\mathbb{A}}) = P_{\#}(\{x \in \mathbb{U} \mid \mathbb{A} \models a(x,v)\}) =$$

$$= P(\{\omega \in \Omega | a(U_i(\omega), v)\}) = P(a(U_i, v)).$$

This correspondence may be easily extended on all conditional formulae.

Now, we define extended approximations using conditional formulae interpreted in the universe $\mathbb{U}$:

Definition Let $X \subseteq \mathbb{U}$ and B be a set of attributes and let $Y \subseteq \mathbb{U}$ be such that

$$Y = ||\varphi||_{\mathbb{U},\mathbb{A}},$$

where φ is a conditional formula over B. Let $\alpha, \kappa \in [0,1]$. The set $Y \subseteq \mathbb{U}$ is called *B-α-κ-approximation* of X when

$$P_{\#}(X \mid Y) \geq \alpha \text{ and } P_{\#}(Y \mid X) \geq \kappa.$$

We call α as the approximation *accuracy* and we denote κ as the approximation *coverage.*

As opposed to the standard approximations defined in a decision system, this definition does not construct a set Y, it only states whether a given set possesses a property of being an α-κ-approximation.

Accuracy and coverage are indices of the approximation quality. Accuracy measures the probability that an object belonging to the approximation belongs

[1] The latter equality introduces a standard probabilistic notation in which 'ω', '{' and '}' are omitted in expressions with random variables.

also to the approximated set. Coverage measures the fraction of objects in a set that are included in its approximation. When the approximation accuracy is equal to 1 and the coverage is maximised the approximation may be considered as *lower* one and when the approximation coverage is equal to 1 and the accuracy is maximised the approximation may be considered as *upper* one.

Accuracy and coverage are defined by means of the underlying probability distribution, according to which the sample is drawn. Since we are given only a sample and we do not know the probability distribution, we must estimate values of the indices using the sample and probabilistic inequalities of the form

$$P\big(P_{\#}(X \mid Y) \geq f_n(U_1, \ldots, U_n)\big) \geq \gamma_n.$$

The above inequality may be interpreted in the following way: if we draw $\{(u_1^i, u_2^i, \ldots, u_n^i)\}_{i=1}^{\infty}$, an infinite sequence of n-element samples, where u_j^i is a realisation of U_j^i, then according to the law of large numbers

$$P\big(P_{\#}(X \mid Y) \geq f_n(U_1, \ldots, U_n)\big) = P\big(P_{\#}(X \mid Y) \geq f_n(U_1^i, \ldots, U_n^i)\big) =$$

$$= \lim_{k \to \infty} \frac{1}{k} \cdot |\{i \leq k \mid P_{\#}(X \mid Y) \geq f_n(u_1^i, \ldots, u_n^i)\}|.$$

Hence γ_n describes how frequent it is true that $P_{\#}(X \mid Y) \geq f_n(u_1^i, \ldots, u_n^i)$ or, in other words how likely $P_{\#}(X \mid Y) \geq f_n(u_1^i, \ldots, u_n^i)$ is to happen in one occurrence. γ_n is a measure called *significance*.

We propose two methods of deriving estimators of the accuracy and the coverage on the basis of sample. The first bases on the Hoeffding inequality [Hoeffding, 1963]:

Theorem 5.8.1 *Let $Z_1, \ldots, Z_n$ be identically distributed independent random variables. Assume that each $Z_i : \Omega \to [0, 1]$. Then, for every $\varepsilon > 0$, the following inequality holds:*

$$P(EZ_1 \leq \frac{1}{n} \sum_{i=1}^{n} Z_i + \varepsilon) \geq 1 - e^{-2n\varepsilon^2}. \tag{5.1}$$

▮

We derive estimator from this theorem as follows: assume that Y is an α-κ-approximation for a set X. Let U be a sample and let $\{U_1, \ldots, U_n\} = U \cap Y$. For the purpose of accuracy estimation we declare that

$$Z_i = \begin{cases} 0, & \text{when } U_i \in X \\ 1, & \text{when } U_i \notin X \end{cases} .$$

Since

$$EZ_1 = P(Z_1 = 1) = P(U_1 \notin X \mid U_1 \in Y) = 1 - P_{\#}(X \mid Y),$$

we obtain the following inequality

$$P((1 - P_{\#}(X \mid Y)) \leq \frac{1}{n} \sum_{i=1}^{n} Z_i + \varepsilon) \geq 1 - e^{-2n\varepsilon^2}$$

Now, we take the advantage of the law of large numbers and the fact that we know the realisation of the sample U. We calculate a realisation for each Z_i in the following way

$$z_i = \begin{cases} 0, & \text{when } u_i \in X \\ 1, & \text{when } u_i \notin X \end{cases},$$

where u_i is i-th u_k such that $u_k \in Y$. The statement

$$(1 - P_{\#}(X \mid Y)) - \frac{1}{n}\sum_{i=1}^{n} z_i \leq \varepsilon$$

is likely to happen with significance $1 - e^{-2n\varepsilon^2}$.

n denotes the number of variables Z_i. It is equal, by definition, to the number of elements in the sample that belong to Y. On the other hand $Z_i = 1$ if and only if the corresponding U_i does not belong to X. Since U_i have to belong to U and Y we obtain

$$n = |U \cap Y| \quad \text{and} \quad \frac{1}{n}\sum_{i=1}^{n} z_i = \frac{|(U \cap Y) \setminus X|}{|U \cap Y|} = 1 - \frac{|U \cap Y \cap X|}{|U \cap Y|}.$$

If we assume that significance is equal to γ we obtain

$$\varepsilon = \sqrt{\frac{\ln(1-\gamma)}{-2|U \cap Y|}}$$

and the approximation accuracy is estimated from (5.1) with the significance γ according to the formula

$$P_{\#}(X \mid Y) \geq \frac{|U \cap Y \cap X|}{|U \cap Y|} - \sqrt{\frac{\ln(1-\gamma)}{-2|U \cap Y|}}.$$

The coverage estimator is developed in the analogous way from (5.1), and the following estimator is obtained

$$P_{\#}(Y \mid X) \geq \frac{|U \cap Y \cap X|}{|U \cap X|} - \sqrt{\frac{\ln(1-\gamma)}{-2|U \cap X|}}.$$

We illustrate the trade-off between these three numerical factors using the following example. Consider decision system presented in Table 5.4. We obtain the following lower approximation for the objects in the system:

$$\underline{\{a\}}||d(x,0)||_{U,\mathbb{A}} = ||a(x,0)||_{U,\mathbb{A}}, \quad \underline{\{a\}}||d(x,1)||_{U,\mathbb{A}} = ||a(x,1)||_{U,\mathbb{A}}.$$

Yet we cannot state that $||a(x,0)||_{\mathbb{U},\mathbb{A}}$ is an approximation of $||d(x,0)||_{\mathbb{U},\mathbb{A}}$ with a 100% accuracy, since there may exist an object u_{101} in $\mathbb{U} \setminus U$ such that $a(u_{101}) = 0$ and $d(u_{101}) = 1$. The given decision system suggests that such an event is unlikely, yet still it is possible.

Table 5.4: Exemplary decision system

	a	d
u_0	1	1
u_1	0	0
u_2	0	0
$\vdots$	$\vdots$	$\vdots$
u_{100}	0	0

We estimate the approximation accuracy with significance 95%:

$$P_{\#}(||d(x,0)||_{\mathbb{U},\mathbb{A}} \mid ||a(x,0)||_{\mathbb{U},\mathbb{A}}) \geq \frac{|\ ||d(x,0) \wedge a(x,0)||_{U,\mathbb{A}}|}{|\ ||a(x,0)||_{U,\mathbb{A}}|} - \sqrt{\frac{\ln(1-0.95)}{-2|\ ||a(x,0)||_{U,\mathbb{A}}|}} =$$

$$= \frac{100}{100} - \sqrt{\frac{\ln(0.05)}{-200}} = 0.88.$$

Hence, the accuracy of the approximation of the set $||d = 0||_{\mathbb{U},\mathbb{A}}$ by means of $||a = 0||_{\mathbb{U},\mathbb{A}}$ is greater than 88% with significance 95%. On the other hand, for the approximation $\underline{\{a\}}||d(x,1)||_{\mathbb{U},\mathbb{A}} = ||a(x,1)||_{\mathbb{U},\mathbb{A}}$, we do not obtain any significant accuracy estimation.

Hoeffding inequality provides us with a simple analytic formula for the approximation accuracy, yet the obtained estimator is not optimal. That is why we propose the second estimator based on the bound proposed in [Jaworski, 2005]. It results in an optimal estimator.

Theorem 5.8.2 *Let $Z_1, \ldots, Z_n$ be identically distributed independent random variables such that $Z_i : \Omega \rightarrow \{0,1\}$, $i = 1, \ldots, n$. Then, the following inequality holds:*

$$P\Bigg(EZ_1 > g_{n,\gamma}(\frac{1}{n}\sum_{i=1}^{n} Z_i)\Bigg) < \gamma,$$

where, for a given $k < n$, $g_{n,\gamma}$ satisfies the equation and $g_{n,\gamma}(1) = 1$. $g_{n,\gamma}$ provides the optimal bound of EZ_1.

The second estimator does not provide any analytic formula for the estimator value, yet $g_{n,\gamma}(\frac{k}{m})$ may be calculated using the algorithm proposed in [Jaworski, 2005].

According to the second estimator the accuracy of the approximation of the set $||d = 0||_{\mathbb{U},\mathbb{A}}$ by means of $||a = 0||_{\mathbb{U},\mathbb{A}}$ is greater than 97% with significance 95%.

5.9 Rule induction algorithm

Extended approximations of all decision classes compose a classifier. Unfortunately an extended approximation for a given set is not uniquely defined. Many algorithms for calculating approximations have been developed. Often approximations are represented by means of decision rules.

A *decision rule* for a given decision system is any expression of the form $\varphi(x) \rightarrow d(x,v)$, where φ is a conditional formula, d is a decision attribute, $v \in V_d$ and $||\varphi(x)||_{U,\mathbb{A}} \neq \emptyset$. A decision rule $\varphi(x) \rightarrow d(x,v)$ is *true* in the decision system if, and only if, $||\varphi||_{U,\mathbb{A}} \subseteq ||d = v||_{U,\mathbb{A}}$. A decision rule describes the dependence between a decision class and its approximation.

In order to illustrate the link of theory with practical results we propose a simple algorithm for rule induction. The algorithm generates a classifier calculating extended approximations for all decision classes. Each approximation is represented as a set of decision rules whose predecessors are conjunctions of descriptors. For each rule, the accuracy, the coverage and the significance are calculated. The algorithm is parametrised by minimal levels of significance and accuracy and it induces all the rules that satisfy these minimal levels of indices. As a consequence induced rules do not cover all objects, and the classifier has not enough knowledge to recognise some objects. On the other hand all the classified objects are certified to be classified correctly with a very high probability.

The algorithm works as follows:

- In the 0th step it checks using the estimator whether there is a decision value v such that the rule with empty predecessor and decision value v would have the desired accuracy and significance. If the answer is positive, then the rule is generated and the rule induction process ends. Otherwise the algorithm moves to the 1st step.

- In the 1st step the set P_1 of all the possible rule predecessors with one descriptor are generated. Each element of P_1 is checked using the estimator. If the answer is positive, then the rule is generated. Then we remove from P_1 all elements used to generate rules and we denote the remaining set as P'_1.

- In the k-th step, $k > 1$, the generates the set P_k on the basis of P'_{k-1} in the following way: each element $\varphi(x)$ of P'_{k-1} and for each descriptor $a(x,v)$ such that a does not appear in $\varphi(x)$ we add add $\varphi(x) \wedge a(x,v)$ to P_k. Each element of P_k is checked using the estimator. If the answer is positive, then the rule is generated. Then we remove from P_k all elements used to generate rules and we denote the remaining set as P'_k.

- The algorithm uses two heuristics that speed it up: it does not try to generate a rule that is more specific than any existing rule and it checks whether there sufficiently many objects matching the rule predecessor to make it significant.

- The algorithm ends when no more rules may be created.

The algorithm generates short and relevant rules that cover only a part of universe.

In the case when during classification several rules may be applied to a given object, we choose the rule with the greatest accuracy.

Many more effective algorithms for rule generation that the one described above were developed (for example, in RSES [Bazan and Szczuka, 2001] system). However, our objective was to illustrate the theory with a practical application and to show the link between set approximations and induced rules only.

Chapter 6

Experimental results

6.1 Cuneiform Documents Search Engine

During our research we developed a practical tool for Sumerologists [Jaworski, 2009]. The objective of Cuneiform Documents Search Engine [CDSE, 2008] is dedicated to retrieve cuneiform documents according to textual pattern. Its goal is to retrieve all the documents that might be relevant to the given query regardless transliteration variants and text arrangement on tablet. Eg. search engine looks for the sign sequences split between many lines. This behaviour allows us to find the documents in which text is arranged in an untypical way. It is also capable of ignoring editor records.

Searching algorithm matches sign sequences provided in query with their occurrences in documents. Then documents that possess at least one occurrence of each sequence are retrieved. Interface provides 10 entries, so search engine may retrieve documents according to the queries that have up to 10 sequences. Search engine translates query into the sequence of sign names. In case when the translation is ambiguous search engine analyses all possible interpretations. Apart from searching by sign sequences it is possible to search by tags.

The search result page provides the following information:

1. sign sequences translated into the sequence of sign names and marked by different colours;
2. number of documents found;
3. retrieved documents.

Parts of documents that match to sign sequences in query are marked by their colours.

6.2 Decision rules extraction

We extracted decision rules from the database of animal transfer transactions. These rules bring to light dependencies between Sumerian officials, animal types and transaction dates. The following table presents a few rules (generated with significance at least 95%):

Accuracy	**Coverage**	**Rule**
0.762606	0.895833	Receiver=`lu2-{d}gesz-bar-e3` → Kiszib=`a-kal-la sipa`
0.866110	1.000000	MuDu=`lum-ma` → Kiszib=`ensi2 u3-da`
0.888281	0.873563	MuSze=`ur-ra`, Giri=`{d}en-lil2-la2` → Kiszib=`{d}szul-gi-a-a-mu`
0.863621	0.739130	Year=`SS09`, Supplier=`du-du` → Kiszib=`u4-de3-nig2-sag10`
0.799641	0.226601	Year=`SZ41`, Animal=`udu` → Maszkim=`en-{d}nansze-ki-ag2`
0.775528	0.928571	Year=`AS07`, Kiszib=`ab-ba-sa6-ga` → Maszkim=`du-du`
0.770440	0.857143	MuSze=`ensi2 zimbir{ki}` → Maszkim=`ur-{d}gubalag nar ta2-hi-isz-a-tal`
0.827830	0.273504	Receiver=`lu2-mah` → MuDu=`{d}szara2`

The first rule states that if `lu2-{d}gesz-bar-e3` is a receiver of goods in transaction then with a probability 76% `a-kal-la sipa` seals the document. And this rule covers 89% of cases when `a-kal-la sipa` seals any document. The remaining rules are interpreted analogically.

We extracted 12841 rules. These rules help to direct Sumerological research pointing out interesting dependencies. Sumerologists may, for example, try to deduct from documents the reasons for a given dependence. Also, analysis of the whole set of rules is interesting, because it provide broad picture of Sumerian economy.

6.3 Animal flow graph

We have determined relations between Sumerian officials in terms of number of animals that were transferred between them. We have generated the graph of animal flow between the officials in Ur III kingdom.

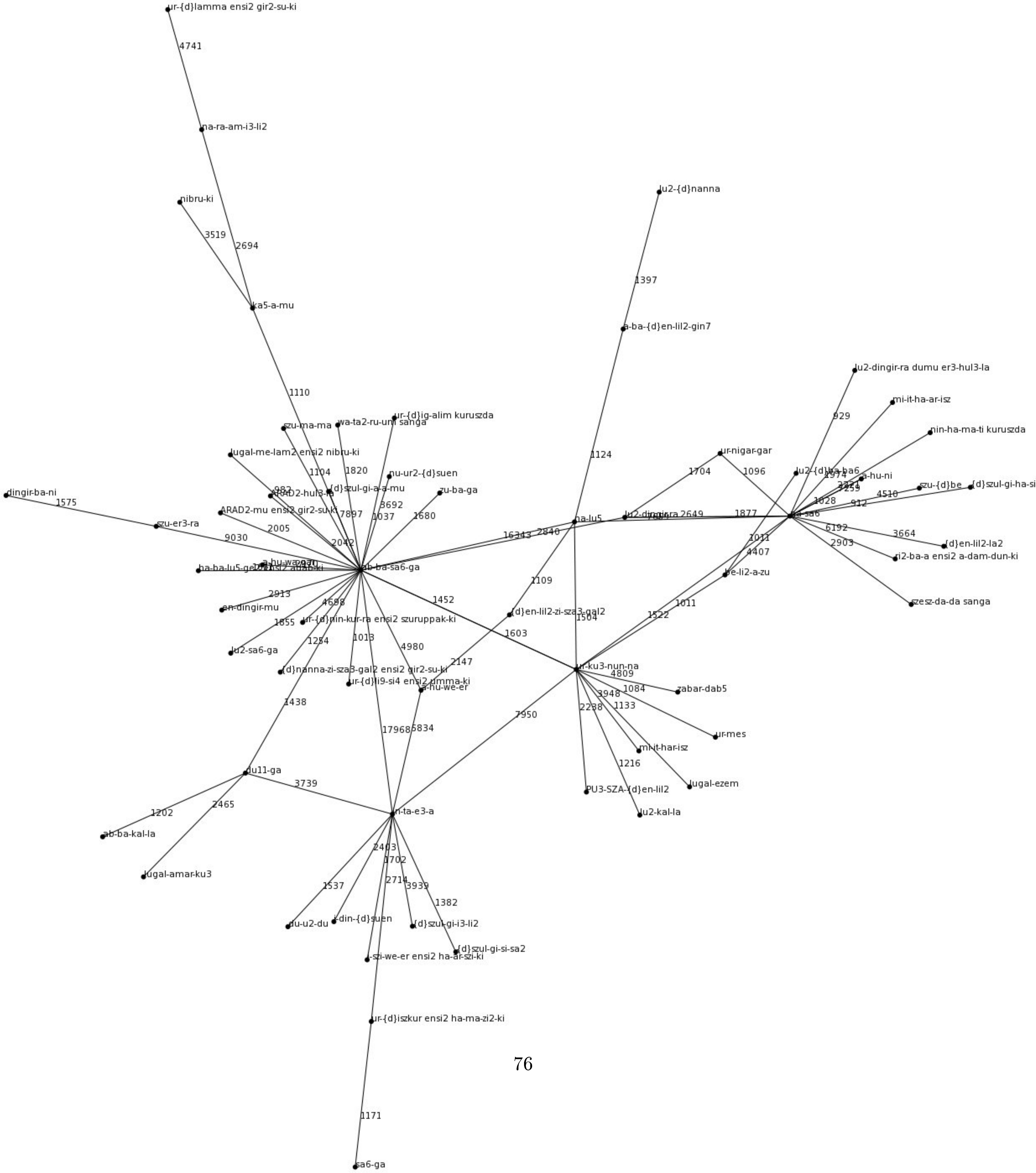
Vertices of the graph represent officials. Graph edges are labelled with quantities of animals transferred between them. On figure 6.1, we present a fragment of the animal flow graph. We selected edges labelled with animal quantities greater than 900. The complete graph that encloses all extracted transactions has 2754 vertices and 5275 edges.

The graph illustrates network of connections between officials, which has a similar structure to the present-day airline connection network. We can see that some of fulfil a function of interconnected hubs, while the rest of officials cooperate mainly with only one of those hubs.

6.4 Economical fluctuations

The following table encloses monthly summaries of quantities of animals, transactions and documents generated by an official named `ab-ba-sa6-ga` during the reign of king `{d}amar-{d}suen`.

Figure 6.1: Animal flows between the officials in Ur III kingdom.



Month	Number of Animals	Number of Transactions	Number of Documents	Animals / Transactions	Animals / Documents	Transactions / Documents
1	15327	835	216	18,36	70,96	3,87
2	8031	517	158	15,53	50,83	3,27
3	11689	549	192	21,29	60,88	2,86
4	3644	554	170	6,58	21,44	3,26
5	28038	824	182	34,03	154,05	4,53
6	12514	678	192	18,46	65,18	3,53
7	28719	816	177	35,19	162,25	4,61
8	44056	1084	271	40,64	162,57	4,00
9	38773	1220	311	31,78	124,67	3,92
10	12761	833	233	15,32	54,77	3,58
11	14164	886	204	15,99	69,43	4,34
12	7953	776	201	10,25	39,57	3,86

The last three columns consist proportion of animals to transaction, animals to documents and transactions to document. We may observe monthly fluctuations of the number of distributed animals, which are consequences of annual farming cycle. As we see 8th and 9th month are exceptional in respect of the number of distributed animals.

We also are able to observe macro-economical changes that happened during the Ur III period.

In the table below we present the activity of `in-ta-e3-a` counted in the number of animals, number of transactions and number of documents generated during consecutive years of king `{d}szu-{d}suen`'s reign:

Year	Number of Animals	Number of Transactions	Number of Documents	Animals / Transactions	Animals / Documents	Transactions / Documents
1	3039	280	42	10,85	72,36	6,67
2	13626	310	259	43,95	52,61	1,20
3	3424	327	300	10,47	11,41	1,09
4	7783	281	211	27,70	36,89	1,33
5	3396	291	233	11,67	14,58	1,25
6	2838	148	88	19,18	32,25	1,68
7	3204	173	149	18,52	21,50	1,16
8	4443	129	109	34,44	40,76	1,18
9	1053	117	108	9,00	9,750	1,08

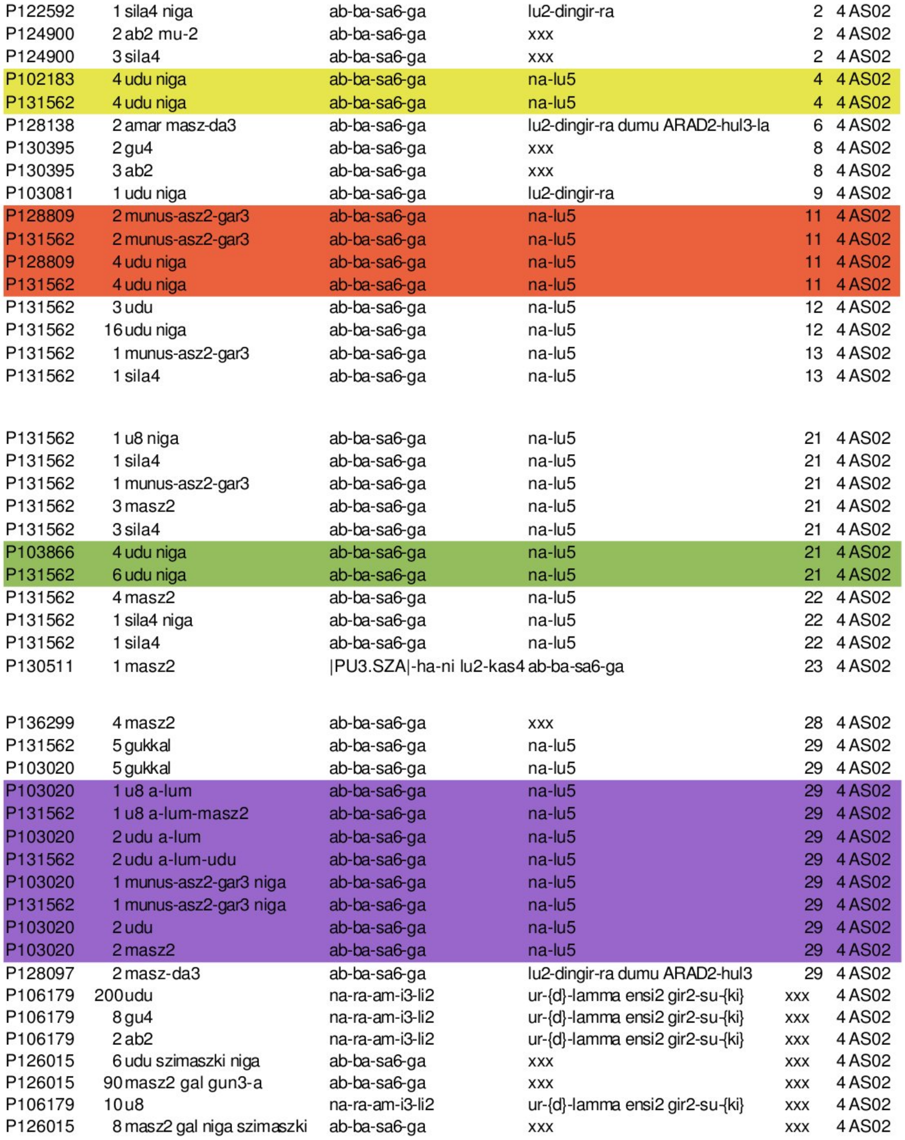
Data presented in table clearly indicates the uniqueness of the second year of `{d}szu-{d}suen`'s reign. This anomaly may be interpreted as an economic footprint of preparations for the war expedition against Simanum, which took place next year.

6.5 Detection of identical transactions

When we sort the transaction list (Fig. 6.2) we are able to find groups of documents that contain information about the same transaction (Fig. 6.3). This allows us to understand the way in which documents were generated: I.e. large documents were written on the basis of short receipts generated during a given time period.

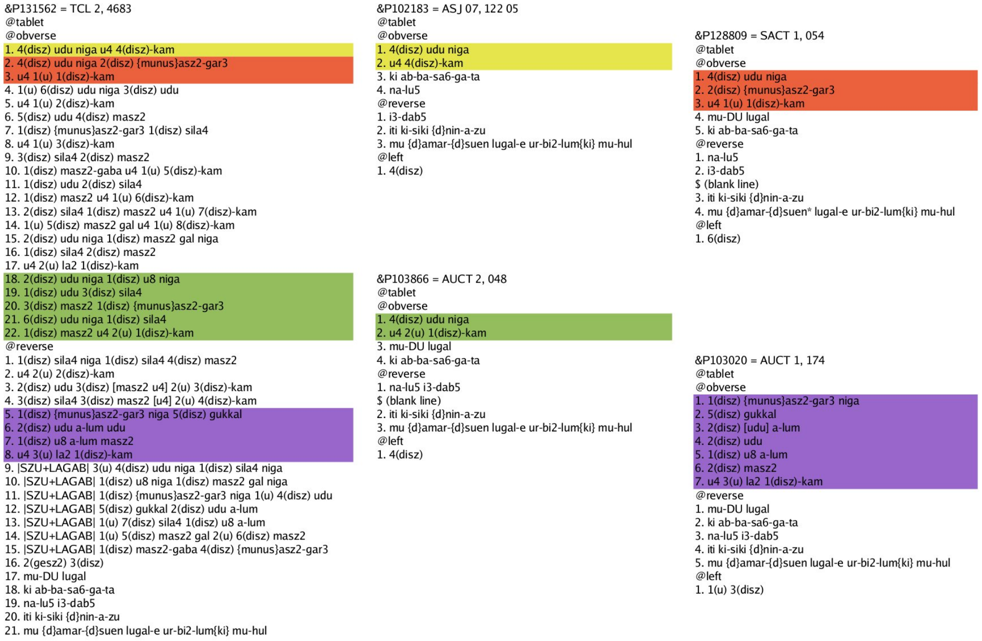
We may estimate the percentage of documents that is not included in the database. For example documents presented on fig. 6.3 suggest that we have

Figure 6.2: List of extracted transactions.



P122592	1	sila4 niga	ab-ba-sa6-ga	lu2-dingir-ra	2	4	AS02
P124900	2	ab2 mu-2	ab-ba-sa6-ga	xxx	2	4	AS02
P124900	3	sila4	ab-ba-sa6-ga	xxx	2	4	AS02
P102183	4	udu niga	ab-ba-sa6-ga	na-lu5	4	4	AS02
P131562	4	udu niga	ab-ba-sa6-ga	na-lu5	4	4	AS02
P128138	2	amar masz-da3	ab-ba-sa6-ga	lu2-dingir-ra dumu ARAD2-hul3-la	6	4	AS02
P130395	2	gu4	ab-ba-sa6-ga	xxx	8	4	AS02
P130395	3	ab2	ab-ba-sa6-ga	xxx	8	4	AS02
P103081	1	udu niga	ab-ba-sa6-ga	lu2-dingir-ra	9	4	AS02
P128809	2	munus-asz2-gar3	ab-ba-sa6-ga	na-lu5	11	4	AS02
P131562	2	munus-asz2-gar3	ab-ba-sa6-ga	na-lu5	11	4	AS02
P128809	4	udu niga	ab-ba-sa6-ga	na-lu5	11	4	AS02
P131562	4	udu niga	ab-ba-sa6-ga	na-lu5	11	4	AS02
P131562	3	udu	ab-ba-sa6-ga	na-lu5	12	4	AS02
P131562	16	udu niga	ab-ba-sa6-ga	na-lu5	12	4	AS02
P131562	1	munus-asz2-gar3	ab-ba-sa6-ga	na-lu5	13	4	AS02
P131562	1	sila4	ab-ba-sa6-ga	na-lu5	13	4	AS02
P131562	1	u8 niga	ab-ba-sa6-ga	na-lu5	21	4	AS02
P131562	1	sila4	ab-ba-sa6-ga	na-lu5	21	4	AS02
P131562	1	munus-asz2-gar3	ab-ba-sa6-ga	na-lu5	21	4	AS02
P131562	3	masz2	ab-ba-sa6-ga	na-lu5	21	4	AS02
P131562	3	sila4	ab-ba-sa6-ga	na-lu5	21	4	AS02
P103866	4	udu niga	ab-ba-sa6-ga	na-lu5	21	4	AS02
P131562	6	udu niga	ab-ba-sa6-ga	na-lu5	21	4	AS02
P131562	4	masz2	ab-ba-sa6-ga	na-lu5	22	4	AS02
P131562	1	sila4 niga	ab-ba-sa6-ga	na-lu5	22	4	AS02
P131562	1	sila4	ab-ba-sa6-ga	na-lu5	22	4	AS02
P130511	1	masz2	|PU3.SZA|-ha-ni lu2-kas4	ab-ba-sa6-ga	23	4	AS02
P136299	4	masz2	ab-ba-sa6-ga	xxx	28	4	AS02
P131562	5	gukkal	ab-ba-sa6-ga	na-lu5	29	4	AS02
P103020	5	gukkal	ab-ba-sa6-ga	na-lu5	29	4	AS02
P103020	1	u8 a-lum	ab-ba-sa6-ga	na-lu5	29	4	AS02
P131562	1	u8 a-lum-masz2	ab-ba-sa6-ga	na-lu5	29	4	AS02
P103020	2	udu a-lum	ab-ba-sa6-ga	na-lu5	29	4	AS02
P131562	2	udu a-lum-udu	ab-ba-sa6-ga	na-lu5	29	4	AS02
P103020	1	munus-asz2-gar3 niga	ab-ba-sa6-ga	na-lu5	29	4	AS02
P131562	1	munus-asz2-gar3 niga	ab-ba-sa6-ga	na-lu5	29	4	AS02
P103020	2	udu	ab-ba-sa6-ga	na-lu5	29	4	AS02
P103020	2	masz2	ab-ba-sa6-ga	na-lu5	29	4	AS02
P128097	2	masz-da3	ab-ba-sa6-ga	lu2-dingir-ra dumu ARAD2-hul3	29	4	AS02
P106179	200	udu	na-ra-am-i3-li2	ur-{d}-lamma ensi2 gir2-su-{ki}	xxx	4	AS02
P106179	8	gu4	na-ra-am-i3-li2	ur-{d}-lamma ensi2 gir2-su-{ki}	xxx	4	AS02
P106179	2	ab2	na-ra-am-i3-li2	ur-{d}-lamma ensi2 gir2-su-{ki}	xxx	4	AS02
P126015	6	udu szimaszki niga	ab-ba-sa6-ga	xxx	xxx	4	AS02
P126015	90	masz2 gal gun3-a	ab-ba-sa6-ga	xxx	xxx	4	AS02
P106179	10	u8	na-ra-am-i3-li2	ur-{d}-lamma ensi2 gir2-su-{ki}	xxx	4	AS02
P126015	8	masz2 gal niga szimaszki	ab-ba-sa6-ga	xxx	xxx	4	AS02

Figure 6.3: Corresponding documents.



```
&P131562 = TCL 2, 4683
@tablet
@obverse
1. 4(disz) udu niga u4 4(disz)-kam
2. 4(disz) udu niga 2(disz) {munus}asz2-gar3
3. u4 1(u) 1(disz)-kam
4. 1(u) 6(disz) udu niga 3(disz) udu
5. u4 1(u) 2(disz)-kam
6. 5(disz) udu 4(disz) masz2
7. 1(disz) {munus}asz2-gar3 1(disz) sila4
8. u4 1(u) 3(disz)-kam
9. 3(disz) sila4 2(disz) masz2
10. 1(disz) masz2-gaba u4 1(u) 5(disz)-kam
11. 1(disz) udu 2(disz) sila4
12. 1(disz) masz2 u4 1(u) 6(disz)-kam
13. 2(disz) sila4 1(disz) masz2 u4 1(u) 7(disz)-kam
14. 1(u) 5(disz) masz2 gal u4 1(u) 8(disz)-kam
15. 2(disz) udu niga 1(disz) masz2 gal niga
16. 1(disz) sila4 2(disz) masz2
17. u4 2(u) la2 1(disz)-kam
18. 2(disz) udu niga 1(disz) u8 niga
19. 1(disz) udu 3(disz) sila4
20. 3(disz) masz2 1(disz) {munus}asz2-gar3
21. 6(disz) udu niga 1(disz) sila4
22. 1(disz) masz2 u4 2(u) 1(disz)-kam
@reverse
1. 1(disz) sila4 niga 1(disz) sila4 4(disz) masz2
2. u4 2(u) 2(disz)-kam
3. 2(disz) udu 3(disz) [masz2 u4] 2(u) 3(disz)-kam
4. 3(disz) sila4 3(disz) masz2 [u4] 2(u) 4(disz)-kam
5. 1(disz) {munus}asz2-gar3 niga 5(disz) gukkal
6. 2(disz) udu a-lum udu
7. 1(disz) u8 a-lum masz2
8. u4 3(u) la2 1(disz)-kam
9. |SZU+LAGAB| 3(u) 4(disz) udu niga 1(disz) sila4 niga
10. |SZU+LAGAB| 1(disz) u8 niga 1(disz) masz2 gal niga
11. |SZU+LAGAB| 1(disz) {munus}asz2-gar3 niga 1(u) 4(disz) udu
12. |SZU+LAGAB| 5(disz) gukkal 2(disz) udu a-lum
13. |SZU+LAGAB| 1(u) 7(disz) sila4 1(disz) u8 a-lum
14. |SZU+LAGAB| 1(u) 5(disz) masz2 gal 2(u) 6(disz) masz2
15. |SZU+LAGAB| 1(disz) masz2-gaba 4(disz) {munus}asz2-gar3
16. 2(gesz2) 3(disz)
17. mu-DU lugal
18. ki ab-ba-sa6-ga-ta
19. na-lu5 i3-dab5
20. iti ki-siki {d}nin-a-zu
21. mu {d}amar-{d}suen lugal-e ur-bi2-lum{ki} mu-hul
```

```
&P102183 = ASJ 07, 122 05
@tablet
@obverse
1. 4(disz) udu niga
2. u4 4(disz)-kam
3. ki ab-ba-sa6-ga-ta
4. na-lu5
@reverse
1. i3-dab5
2. iti ki-siki {d}nin-a-zu
3. mu {d}amar-{d}suen lugal-e ur-bi2-lum{ki} mu-hul
@left
1. 4(disz)
```

```
&P103866 = AUCT 2, 048
@tablet
@obverse
1. 4(disz) udu niga
2. u4 2(u) 1(disz)-kam
3. mu-DU lugal
4. ki ab-ba-sa6-ga-ta
@reverse
1. na-lu5 i3-dab5
$ (blank line)
2. iti ki-siki {d}nin-a-zu
3. mu {d}amar-{d}suen lugal-e ur-bi2-lum{ki} mu-hul
@left
1. 4(disz)
```

```
&P128809 = SACT 1, 054
@tablet
@obverse
1. 4(disz) udu niga
2. 2(disz) {munus}asz2-gar3
3. u4 1(u) 1(disz)-kam
4. mu-DU lugal
5. ki ab-ba-sa6-ga-ta
@reverse
1. na-lu5
2. i3-dab5
$ (blank line)
3. iti ki-siki {d}nin-a-zu
4. mu {d}amar-{d}suen* lugal-e ur-bi2-lum{ki} mu-hul
@left
1. 6(disz)
```

```
&P103020 = AUCT 1, 174
@tablet
@obverse
1. 1(disz) {munus}asz2-gar3 niga
2. 5(disz) gukkal
3. 2(disz) [udu] a-lum
4. 2(disz) udu
5. 1(disz) u8 a-lum
6. 2(disz) masz2
7. u4 3(u) la2 1(disz)-kam
@reverse
1. mu-DU lugal
2. ki ab-ba-sa6-ga-ta
3. na-lu5 i3-dab5
4. iti ki-siki {d}nin-a-zu
5. mu {d}amar-{d}suen lugal-e ur-bi2-lum{ki} mu-hul
@left
1. 1(u) 3(disz)
```

access to about 25% of documents. Of course such conclusions are limited to certain places and periods of documents origin.

Sometimes we may also reconstruct broken documents or correct transliteration errors.

6.6 Lexicon of magazines

Our methodology is capable of translating the Biobibliografical Lexicon into a lexicon of magazines, whose entries would be all magazines mentioned in the Biobibliografical Lexicon.

We illustrate this idea on the example of ”Kamena”. Using events extracted from the Biobibliografical Lexicon we obtain general information about this periodic. It was published in Lublin and it was ”czasopismo literackie” (literary magazine): ”miesięcznik” (monthly), ”dwutygodnik” (biweekly) or ”tygodnik” (weekly) depending on the time period.

Extracted Work Events reveal editorial board of this periodic:

Date	**Name**	**WorkEventType / EventState**	**Job**	**SubOrganization**
W 1933	K. A. Jaworski	był	współzałożyciel	
do 1939	K. A. Jaworski	był	redaktor	
1935	Z. Jasiński	był	twórca-redaktor	specjalny marynistyczny numer
Od 1945	K. A. Jaworski	należał	redaktor naczelny	redakcja
	K. Bielski	należał		kolegium redakcyjne
	Z. Mikulski	pełnił funkcję	kierownik	dział literacki
w 1952-62	K. A. Jaworski	należał	członek	kolegium redakcyjne
Od 1952	M. K. Bechczyc-R.	była	członek	kolegium redakcyjne
W 1953-59	J. N. Kłosowski	należał		kolegium redakcyjne
w 1956-60	H. Platta		członek	zespół redakcyjny
Od 1957	A. Markowa	pracowała		redakcja
1958-65	A. Kamieńska	była związana		
w 1960-61	T. Kłak	redakcja		kolumna ”W Stronę Młodych”
w 1960-64	M. K. Bechczyc-R.	była	redaktor naczelny	
od 1961 do 1964	W. L. Babinicz	wchodził w skład		kolegium redakcyjne
w 1962-64	J. Pleśniarowicz		członek	kolegium redakcyjne
1963-65	Z. Mikulski	pełnił funkcję	sekretarz	redakcja
w 1963-66	Z. Jastrzębski	pracował na pół etatu		redakcja
od 1963	K. Bielski	prowadził		dział poezji
W 1965	S. Wolski	wchodził w skład		kolegium redakcyjne
w 1970-71	T. Kłak		członek	kolegium redakcyjne
1971-73	H. Makarski		współredaktor	kolumna literacka ”Zapole”
W 1971-74	B. Madej	był	członek	kolegium redakcyjne
W 1984-86	H. Pająk	pracował	dziennikarz	

Publication Events shows debutants

Date	Issue	Name	CompositionType	CompositionTitle
w 1958	nr 23/24	K Dmitruk	fragment poematu	”Cela”
w 1961	nr 20	A. Fiala	utwór	”Małe formy”
w 1968	nr 5	A. Fiala	mikroopowiadanie	”Moja książka”
w tymże roku	nr 23/24	K. Frejdlich	wiersz	”Epitaphium”
w 1972	nr 13	Z. W. Fronczek	wiersz	”Zmarła ciotka...”
w 1935	nr 1	L. Gomolicki	artykuł	”Ze współczesnej poezji rosyjskiej”
w 1967	nr 11	S. Gostkowski	wiersz	”Zmęczenie”
. . .				

as well as regular authors:

Date	Name	CompositionType
1956-1978	A. Aleksandrowicz-Ulrich	artykuły
1933/34-34/35, 1938	F. Arnsztajnowa	wiersze, przekłady
1959-68	W. Bacewicz	wiersze, nowele, artykuły, reportaże
od 1934/35	S. Bąkowski	wiersze, przekłady z francuskiego, przekłady z włoskiego
	M. K. Bechczyc-Rudnicka	artykuły, recenzje teatralne
1933-34/35, 1938	K. Bielski	wiersze, proza
1937-39	Z. Bieńkowski	wiersze, przekłady z francuskiego
	T. Bocheński	wiersze, felietony na tematy literackie, przekłady
	T. Bocheński	wiersze, przekłady
1954-67	J. Brzostowska	wiersze, przekłady
1933-37	J. Brzękowski	artykuły
od 1973	N. Chadzinikolau	utwory, przekłady z literatury greckiej
1957-84	T. Chróścielewski	wiersze, artykuły, recenzje, przekłady
1956-60	E. Cichla-Czarniawska	wiersze
. . .		

Chapter 7

Conclusions

This thesis presents a new framework for computers-aided inference of knowledge from textual data. In three chapters, we introduced methods for semantic parsing, knowledge representation and statistical inference.

When we process data into meaning representation language formulae we obtain knowledge base, which we may use for various data mining applications.

We can look for information using concepts from the documents. We describe properties of desired objects by means of queries and then we find the set of objects that satisfy the queries.

The meaning representation language formulae provide us also with features for clustering and classification of the structural objects. If we define a decision attribute, we may construct rule based classifiers that use queries as selectors in decision rules.

The direct impact of the results is aimed on sumerological and biographical research. Yet there is a lot of other potential application fields for our methodology. System may be also extended i adapted to fulfil many other tasks.

While parsing documents we simultaneously apply two sets of rules: syntactic and semantic ones. This property may be utilised for automatic extension of semantic lexicon. Syntactic rules can define the connection between unknown word and the rest of sentence, then semantic rules could determine its possible ontological category.

The coreference problem may be solved in the same fashion. Semantic rules define constraints on a semantic category of object pointed to by a pronoun considerably reducing the number of possible objects.

The fact that we use statistical learning theory for describing the data generation process suggest a question of comparison of our statistical model with classical probabilistic models used in natural language processing, which should be a subject of further study.

Finally, methodology developed in the thesis may be converted to be a high quality checker for spelling, punctuation and grammatical errors, and even semantic inconsistency of documents.

Bibliography

[Allen, 1995] Allen, J. (1995). *Natural Language Understanding.* Benjamin Cummings, Redwood City, California, 2nd edition.

[Androutsopoulos et al., 1995] Androutsopoulos, I., Ritchie, G. D., and Thanisch, P. (1995). Natural Language Interfaces to Databases–an introduction. *Journal of Language Engineering*, 1(1):29–81.

[Bazan et al., 2004] Bazan, J., Nguyen, S. H., Nguyen, H. S., and Skowron, A. (2004). Rough set methods in approximation of hierarchical concepts. In Tsumoto, S., Slowinski, R., Komorowski, J., and Grzymala-Busse, J., editors, *Proc. of the Fourth International Conference on Rough Sets and Current Trends in Computing (RSCTC 2004)*, volume 3066 of *Lecture Notes in Artificial Intelligence*, pages 346–355, Heidelberg. Springer-Verlag.

[Bazan and Szczuka, 2001] Bazan, J. and Szczuka, M. (2001). RSES and RSESlib - a collection of tools for rough set computations. In Ziarko, W. and Yao, Y., editors, *Proc. of the Second International Conference on Rough Sets and Current Trends in Computing (RSCTC 2000)*, volume 2005 of *Lecture Notes in Computer Science*, pages 106–113, Heidelberg. Springer.

[Blackburn and Bos, 2005] Blackburn, P. and Bos, J. (2005). *Representation and Inference for Natural Language: A First Course in Computational Semantics.* CSLI Publications, Stanford, California.

[Bos, 2005] Bos, J. (2005). Towards wide-coverage semantic interpretation. In *Proc. of Sixth International Workshop on Computational Semantics (IWCS-6)*, pages 42–53.

[Bos et al., 2004] Bos, J., Clark, S., Steedman, M., Curran, J. R., and Hockenmaier, J. (2004). Wide-coverage semantic representations from a ccg parser. In *Proc. of the 20th International Conference on Computational Linguistics (COLING '04)*, pages 1240–1246, Geneva, Switzerland. COLING.

[CDLI, 2008] CDLI (2000–2008). The cuneiform digital library initiative (CDLI). `http://cdli.ucla.edu`.

[CDSE, 2008] CDSE (2008). The cuneiform documents search engine (CDSE). `http://www.ur3.historia.uw.edu.pl`.

[Charniak and Wilks, 1976] Charniak, E. and Wilks, Y., editors (1976). *Computational Semantics.* North-Holland / American Elsevier, Amsterdam.

[Chen et al., 2003] Chen, M., Foroughi, E., and Heintz, F. (2003). *Users manual: RoboCup soccer server manual for soccer server version 7.07 and later.*

[Chomsky, 1965] Chomsky, N. (1965). *Aspects of the theory of syntax.* MIT Press, Cambridge, Massachusetts.

[Cichosz, 2000] Cichosz, P. (2000). *Systemy uczące się [Machine learning systems].* Wydawnictwa Naukowo–Techniczne, Warszawa.

[Codd, 1970] Codd, E. F. (1970). A relational model of data for large shared data banks. *Commun. ACM*, 13(6):377–387.

[Copestake et al., 1999] Copestake, A., Flickinger, D., and Sag, I. (1999). Minimal recursion semantics: an introduction. Technical report, CSLI, Stanford, CA.

[Crouch, 2005] Crouch, D. (2005). Packed rewriting for mapping semantics to KR. In *Proc. of Sixth International Workshop on Computational Semantics.*

[Curran et al., 2007] Curran, J. R., Clark, S., and Bos, J. (2007). Linguistically motivated large-scale NLP with C&C and boxer. In *Proc. of the ACL 2007 Demonstrations Session (ACL-07 demo)*, pages 29–32. The Association for Computer Linguistics.

[Czachowska and Szałagan, 2008] Czachowska, J. and Szałagan, A., editors (1994,1996,1997,1999,2001,2003,2004,2008). *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.

[Demri and Orłowska, 2002] Demri, S. and Orłowska, E. (2002). *Incomplete Information: Structure, Inference, Complexity.* Monographs in Theoretical Computer Science. An EATCS Series. Springer.

[Dowty et al., 1981] Dowty, D. R., Wall, R. E., and Peters, S. (1981). *Introduction to Montague Semantics.* Reidel, Dordrecht.

[Earley, 1986] Earley, J. (1986). An efficient context-free parsing algorithm. *Communications of the ACM*, 6(8):451–455.

[Feldman and Sanger, 2006] Feldman, R. and Sanger, J. (2006). *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data.* Cambridge University Press.

[Fillmore, 1976] Fillmore, C. J. (1976). Frame semantics and the nature of language. In Harnad, S., editor, *Origins and evolution of language and speech*, pages 155–202. Academy of Sciences.

[Ge and Mooney, 2005] Ge, R. and Mooney, R. J. (2005). A statistical semantic parser that integrates syntax and semantics. In *Proc. of 9th Conf. on Computational Natural Language Learning (CoNLL-2005)*, pages 9–16, Ann Arbor, MI.

[Gediga and Düntsch, 2000] Gediga, G. and Düntsch, I. (2000). *Rough Set Data Analysis — A Road to Non-Invasive Knowledge Discovery.* Methodos Publishers, UK.

[Gruber, 1993] Gruber, T. R. (1993). A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199–220.

[Grzymala-Busse, 2006] Grzymala-Busse, J. W. (2006). A rough set approach to data with missing attribute values. In Wang, G., Peters, J. F., Skowron, A., and Yao, Y., editors, *Proc. of Rough Sets and Knowledge Technology, First International Conference (RSKT 2006)*, volume 4062 of *Lecture Notes in Computer Science*, pages 58–67. Springer.

[Grzymała-Busse and Grzymała-Busse, 2007] Grzymała-Busse, J. W. and Grzymała-Busse, W. J. (2007). An experimental comparison of three rough set approaches to missing attribute values. *T. Rough Sets*, 6:31–50.

[Guillet and Hamilton, 2007] Guillet, F. and Hamilton, H. J., editors (2007). *Quality Measures in Data Mining*, volume 43 of *Studies in Computational Intelligence*. Springer.

[Hanseth and Monteiro, 1994] Hanseth, O. and Monteiro, E. (1994). Modelling and the representation of reality: some implications of philosophy on practical systems development. *Scand. J. Inf. Syst.*, 6(1):25–46.

[Hastie et al., 2001] Hastie, T., Tibshirani, R., and Friedman, J. H. (2001). *The Elements of Statistical Learning.* Springer.

[Heim, 1982] Heim, I. (1982). *The Semantics of Definite and Indefinite Noun Phrases.* PhD thesis, University of Massachusetts, Amherst.

[Hendrix et al., 1978] Hendrix, G. G., Sacerdoti, E. D., Sagalowicz, D., and Slocum, J. (1978). Developing a natural language interface to complex data. *ACM Transactions on Database Systems*, 3(2):105–147.

[Hoeffding, 1963] Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30.

[Hopcroft and Ullman, 1979] Hopcroft, J. E. and Ullman, J. D. (1979). *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley Publishing Company, Inc., Reading, MA.

[Jaworski, 2005] Jaworski, W. (2005). Model selection and assessment for classification using validation. In Ślęzak, D., Wang, G., Szczuka, M. S., Düntsch, I., and Yao, Y., editors, *Proc. of Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, 10th International Conference (RSFDGrC 2005)*, volume 3641 of *Lecture Notes in Computer Science*, pages 481–490. Springer.

[Jaworski, 2006a] Jaworski, W. (2006a). Bounds for validation. *Fundam. Inform*, 70(3):261–275.

[Jaworski, 2006b] Jaworski, W. (2006b). Learning compound decision functions for sequential data in dialog with experts. In Greco, S., Hata, Y., Hirano, S., Inuiguchi, M., Miyamoto, S., Nguyen, H. S., and Slowinski, R., editors, *Proc. of Rough Sets and Current Trends in Computing, 5th International Conference, (RSCTC 2006)*, volume 4259 of *Lecture Notes in Computer Science*, pages 627–636. Springer.

[Jaworski, 2007] Jaworski, W. (2007). Automatic tool for semantic analysis of Neo-Sumerian documents. In *Proc. of 52nd Rencontre Assyriologique Internationale.* (to appear).

[Jaworski, 2008a] Jaworski, W. (2008a). Contents modelling of Neo-Sumerian Ur III economic text corpus. In *Proc. of the 22nd International Conference on Computational Linguistics (COLING 2008)*, pages 369–376, Manchester, UK. Coling 2008 Organizing Committee.

[Jaworski, 2008b] Jaworski, W. (2008b). Generalized indiscernibility relations: Applications for missing values and analysis of structural objects. *Transactions of Rough Sets*, 8:116–145.

[Jaworski, 2008c] Jaworski, W. (2008c). Rule induction: Combining rough set and statistical approaches. In Chan, C.-C., Grzymala-Busse, J. W., and Ziarko, W., editors, *Proc. of Rough Sets and Current Trends in Computing, 6th International Conference (RSCTC 2008)*, volume 5306 of *Lecture Notes in Computer Science*, pages 170–180. Springer.

[Jaworski, 2009] Jaworski, W. (2009). Search engine for the cuneiform economic documents corpus. In *Proc. of 54th Rencontre Assyriologique Internationale.* (to appear).

[Jurafsky and Martin, 2000] Jurafsky, D. and Martin, J. H. (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* Prentice Hall, Englewood Cliffs, New Jersey.

[Kamp and Reyle, 1993] Kamp, H. and Reyle, U. (1993). *From Discourse to Logic: Introduction to Modeltheoretic Semantics in Natural Language, Formal Logic and Discourse Representation Theory, Vol. 2.* Kluwer Academic Publishers, Dordrecht.

[Kate et al., 2005] Kate, R. J., Wong, Y. W., and Mooney, R. J. (2005). Learning to transform natural to formal languages. In Veloso, M. M. and Kambhampati, S., editors, *Proc. of The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference*, pages 1062–1068. AAAI Press / The MIT Press.

[Kryszkiewicz, 1998a] Kryszkiewicz, M. (1998a). Properties of incomplete information systems in the framework of rough sets. In Polkowski, L. and Skowron, A., editors, *Rough Sets in Knowledge Discovery 1. Methodology and Applications*, Studies in Fuzziness and Soft Computing, pages 422–450. Physica-Verlag, Heidelberg.

[Kryszkiewicz, 1998b] Kryszkiewicz, M. (1998b). Rough set approach to incomplete information systems. *Inf. Sci*, 112(1-4):39–49.

[Kuhn and de Mori, 1995] Kuhn, R. and de Mori, R. (1995). The application of semantic classification trees to natural language understanding. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(5):449–460.

[Labat and Malbran-Labat, 1988] Labat, R. and Malbran-Labat, F. (1988). *Manuel d'epigraphie akkadienne. Signes, Syllabaire, ideogrammes.* Geuthner (Librarie Orientaliste Paul Geuthner S.A.), 6th edition.

[Latkowski, 2003] Latkowski, R. (2003). On decomposition for incomplete data. *Fundam. Inform*, 54(1):1–16.

[Latkowski, 2004] Latkowski, R. (2004). On indiscernibility relations for missing attribute values. In Lindemann, G., Burkhard, H. D., Ludwik Czaja, A. S., Schlingloff, H., and Suraj, Z., editors, *Proc. of the Workshop on Concurrency, Specification and Programming (CSP 2004)*, volume 170 of *Informatik-Bericht*, pages 330–335, Berlin. Humboldt Universität.

[Latkowski, 2005] Latkowski, R. (2005). Flexible indiscernibility relations for missing attribute values. *Fundam. Inform*, 67(1-3):131–147.

[Latkowski and Mikołajczyk, 2004] Latkowski, R. and Mikołajczyk, M. (2004). Data decomposition and decision rule joining for classification of data with missing values. In Tsumoto, S., Slowinski, R., Komorowski, J., and Grzymala-Busse, J., editors, *Proc. of the Fourth International Conference on Rough Sets and Current Trends in Computing (RSCTC 2004)*, volume 3066 of *Lecture Notes in Artificial Intelligence*, pages 254–263, Berlin. Springer-Verlag.

[Lipski, 1981] Lipski, W. J. (1981). On Databases with Incomplete Information. *Journal of the Association of Computing Machinery*, 28(1):41–70.

[Lobner, 2002] Lobner, S. (2002). *Understanding Semantics.* Arnold Publishers, London.

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

[Miller et al., 1990] Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. (1990). Five papers on WORDNET. Technical Report CSL 43, Cognitive Science Laboratory, Princeton University.

[Miller et al., 1996] Miller, S., Stallard, D., Bobrow, R., and Schwartz, R. (1996). A fully statistical approach to natural language interfaces. In Joshi, A. and Palmer, M., editors, *Proc. of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics*, pages 55–61, San Francisco. Morgan Kaufmann Publishers.

[Mitchell, 1997] Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, New York.

[Moens, 2006] Moens, M.-F. (2006). *Information Extraction: Algorithms and Prospects in a Retrieval Context (The Information Retrieval Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[Montague, 1970a] Montague, R. (1970a). English as a formal language. In et al., B. V., editor, *Linguaggi nella Societá e nella Tecnica*, pages 189–224. Edizioni di Communità, Milan.

[Montague, 1970b] Montague, R. (1970b). Universal grammar. *Theoria*, 36:373–398.

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

[Nguyen et al., 2006] Nguyen, M. L., Shimazu, A., and Phan, X. H. (2006). Semantic parsing with structured SVM ensemble classification models. In *Proceedings of the COLING/ACL on Main conference poster sessions*. The Association for Computer Linguistics.

[Nguyen et al., 2004] Nguyen, S. H., Bazan, J., Skowron, A., and Nguyen, H. S. (2004). Layered learning for concept synthesis. *LNCS Transactions on Rough Sets*, 3100(1):187–208.

[Obrębski, 2002] Obrębski, T. (2002). *Automatyczna analiza składniowa języka polskiego z wykorzystaniem gramatyki zależnościowej [Automatic syntactic analysis of language using dependence grammar]*. PhD thesis, IPI PAN, Poznań.

[Pawlak, 1981] Pawlak, Z. (1981). Information systems — theoretical foundations. *Information Systems*, 6(3):205–218.

[Pawlak, 1982] Pawlak, Z. (1982). Rough sets. *International Journal of Computer and Information Sciences*, 11(5):341–356.

[Pawlak, 1991] Pawlak, Z. (1991). *Rough Sets: Theoretical Aspects of Reasoning about Data.* Kluwer Academic Publishers, Dordrecht.

[Pawlak and Skowron, 2007] Pawlak, Z. and Skowron, A. (2007). Rough sets: Some extensions. *Inf. Sci*, 177(1):28–40.

[Popescu et al., 2004] Popescu, A.-M., Armanasu, A., Etzioni, O., Ko, D., and Yates, A. (2004). Modern natural language interfaces to databases: Composing statistical parsing with semantic tractability. In *Proc. of the Twentieth International Conference on Computational Linguistics (COLING–04)*, pages 30–39.

[Popescu et al., 2003] Popescu, A.-M., Etzioni, O., and Kautz, H. (2003). Towards a theory of natural language interfaces to databases. In *Proc. of the 2003 International Conference on Intelligent User Interfaces*, Full Technical Papers, pages 149–157.

[Price, 1990] Price, P. J. (1990). Evaluation of spoken language systems: The atis domain. In *Proc. of the Third DARPA Speech and Natural Language Workshop*, pages 91–95.

[Przepiórkowski, 2008] Przepiórkowski, A. (2008). *Powierzchniowe przetwarzanie języka polskiego [Shallow processing of the Polish language].* Akademicka Oficyna Wydawnicza EXIT, Warszawa.

[Przepiórkowski et al., 2002] Przepiórkowski, A., Kupść, A., Marciniak, M., and Mykowiecka, A. (2002). *Formalny opis języka polskiego. Teoria i implementacja [A formal description of the Polish language. Theory and implementation].* Akademicka Oficyna Wydawnicza Exit, Warszawa.

[Ramaswamy and Kleindienst, 2000] Ramaswamy, G. and Kleindienst, J. (2000). Hierarchical feature-based translation for scalable natural language understanding. In *Proc. of 6th International Conference on Spoken Language Processing.*

[Reyle, 1993] Reyle, U. (1993). Dealing with ambiguities by underspecification: Construction, representation, and deduction. *Journal of Semantics*, 10(2):123–179.

[Richter and Sailer, 1999] Richter, F. and Sailer, M. (1999). Underspecified semantics in HPSG. In *Proc. of the Second International Workshop on Computational Semantics*, pages 95–112. Kluwer Academic Publishers.

[Russell and Norvig, 1995] Russell, S. and Norvig, P. (1995). *Artificial Intelligence a Modern Approach.* AI. Prentice Hall.

[Saeed, 2003] Saeed, J. I. (2003). *Semantics.* Blackwell, 2nd edition.

[Seneff, 1992] Seneff, S. (1992). Tina: A natural language system for spoken language applications. *Computational Linguistics*, 18(1):61–86.

[Sharlach, 2004] Sharlach, T. (2004). *Provincial Taxation and the Ur III State*. Leiden-Boston.

[Skowron and Stepaniuk, 1999] Skowron, A. and Stepaniuk, J. (1999). Towards discovery of information granules. In Zytkow, J. M. and Rauch, J., editors, *Proc. of the 3rd European Conference on Principles of Data Mining and Knowledge Discovery (PKDD-99)*, volume 1704 of *LNAI*, pages 542–547, Berlin. Springer.

[Skowron et al., 2005] Skowron, A., Świniarski, R., and Synak, P. (2005). Approximation spaces and information granulation. *LNCS Transactions on Rough Sets*, 3400(3):175–189.

[Sowa, 1999] Sowa, J. F., editor (1999). *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks Cole Publishing Co., Pacific Grove, CA, USA.

[Staab and Studer, 2004] Staab, S. and Studer, R., editors (2004). *Handbook on Ontologies*. Springer-Verlag, Heidelberg and Berlin.

[Steinkeller, 1987] Steinkeller, P. (1987). The administrative and economic organization of the ur iii state: The core and the periphery. In Biggs, R. and Gibson, M., editors, *The Organization of Power: Aspect of Bureaucracy in the Ancient Near East*, SAOC 46, pages 19–41. Chicago.

[Stępień, 1996] Stępień, M. (1996). *Animal Husbandry in the Ancient Near East: A Prosopographic Study of Third-Millennium Umma*. Bethesda, Md. : CDL Press.

[Stępień, 2006] Stępień, M. (2006). *Ensi w czasach III dynastii z Ur: aspekty ekonomiczne i administracyjne pozycji namiestnika prowincji w świetle archiwum z Ummy [Ensi in the Third Dynasty of Ur: Economic and Administrative Aspects of the Province Governor Position on the Basis of Umma Archive]*. Wydawnictwa Uniwersytetu Warszawskiego.

[Tsumoto, 2002] Tsumoto, S. (2002). Accuracy and coverage in rough set rule induction. *Lecture Notes in Computer Science*, 2475:373–380.

[van Eijck and Kamp, 1997] van Eijck, J. and Kamp, H. (1997). Representing discourse in context. In van Benthem, J. and ter Meulen, A., editors, *Handbook of Logic and Language*, pages 179–237. Elsevier.

[Vapnik, 1998] Vapnik, V. N. (1998). *Statistical Learning Theory*. John_Wiley.

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

[Warren and Pereira, 1982] Warren, D. H. D. and Pereira, F. C. N. (1982). An efficient easily adaptable system for interpreting natural language queries. *American Journal of Computational Linguistics*, 8(3).

[Wittgenstein, 1962] Wittgenstein, W. (1962). *Tractatus Logico-Philosophicus.* Routledge and Kegan Paul, London.

[Woliński, 2006] Woliński, M. (2006). Morfeusz — a practical tool for the morphological analysis of polish. In Kłopotek, M., Wierzchoń, S., and Trojanowski, K., editors, *Proc. of Intelligent Information Processing and Web Mining (IIS:IIPWM'06)*, pages 503–512. Springer.

[Woliński, 2004] Woliński, M. (2004). *Komputerowa weryfikacja gramatyki Świdzińskiego [Computer driven verification of Świdziński's grammar].* PhD thesis, IPI PAN, Warszawa.

[Wong and Mooney, 2006] Wong, Y. and Mooney, R. J. (2006). Learning for semantic parsing with statistical machine translation. In *Proc. of Human Language Technology Conference / North American Chapter of the Association for Computational Linguistics Annual Meeting (HLT–NAACL–06)*, pages 439–446, New York City, NY.

[Woods et al., 1972] Woods, W., Kaplan, R., and Nash-Webber, B. (1972). The lunar sciences natural language information system: final report. Technical Report 2378, Bolt, Beranek and Newman, Cambridge, MA.

[Zarba, 2006] Zarba, C. G. (2006). *Many-Sorted Logic.*

[Zelle and Mooney, 1996] Zelle, J. M. and Mooney, R. J. (1996). Learning to parse database queries using inductive logic programming. In *Proc. of the 14th National Conference on Artificial Intelligence*, pages 1050–1055, Portland, OR. AAAI Press/MIT Press.

[Zettlemoyer and Collins, 2005] Zettlemoyer, L. S. and Collins, M. (2005). Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proc. of the 21th Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*, pages 658–666. AUAI Press.