
**OPAALS PROJECT**

Contract n° IST-034824

# WP6: Socio-Constructivism and Language

## Del6.4 – First results from a showcase model of an applied ontology and taxonomic hierarchy


Project funded by the European Community under the "Information Society Technology" Programme

**Contract Number**: IST-034824

**Project Acronym**: OPAALS

**Deliverable N°**: 6.4

**Due date**: July 2008

**Delivery Date**: September 2008

**Short Description**:

The deliverable presents different linguistic and statistical methods that might assist in the process of ontology building from texts as well as in the analysis of ontological and taxonomic representations. Furthermore, it provides first results of applying different methods to the exemplary samples of companies' descriptions (SMEs) taken from the web. Finally, the leitmotif of this deliverable is the role and impact of the linguistic analysis in the ontology building process.

**Author**: Oxana Lapteva, Hagen Peukert

**Partners contributed**: UniKassel, LSE, IITK

**Made available to**: Consortium

**Versioning**

| Version | Date | Name, organization |
|---|---|---|
| v1 | 05.06.2008 | Oxana Lapteva |
| v2 | 30.06.2008 | Oxana Lapteva, HagenPeukert |
| v3 | 25.07.2008 | Oxana Lapteva, Hagen Peukert |
| v4 | 25.09.2008 | Oxana Lapteva, Hagen Peukert |

**Quality check**

**Internal Reviewers**: Ossi Nykänen (TUT), Alexandros Marinos (UNIS)

## Dependences:

| | |
|---|---|
| **Work Packages** | WP6, WP2, WP3, WP10, WP11 |
| **Partners** | TUT, IITK, SUAS, UNIS, LSE |
| **Domains** | Linguistics: computational linguistics, corpus linguistics, natural language processing |
| **Targets** | All partners |

# Table of Contents

# Introduction

Within digital ecosystems information and knowledge is shared among people. By this means, one has to think of the underlying architectures in terms of its representation and processing algorithms. The communicative purpose of digital ecosystems is clearly affected by the way knowledge is available and in which form information is shared, that is, communication within the system is carried out.

Humans feel most comfortable using natural language as their means of communication. It is especially true when communicating more complex ideas. Naturally, this should also be true for digital ecosystems that involve human interaction. On the other hand, large databases of natural languages are hard to maintain and to administer. While we have some approaches that try to solve the problem, they are far from perfect. So it seems to be worth investigating some more research to this open question. Again, it appears as if it boils down to finding just another algorithm that transfers natural language into a format free of ambiguity. In other words how to map natural language to a logic that machines are capable of processing. The following text explores some alternative possibilities that could lead to a more prosperous outcome.

Ontology languages are formal languages that construct ontologies. Basically they specify the data structure that is used to classify the concepts of a certain domain and how these concepts are processed and related to each other. Now we define a language ontology to be an ontology in which its concepts and their relationships among each other is natural language itself, that is, the specific but fairly simple syntax of an ontology language is replaced by the complex syntactic structures of natural language. Of course, only these parts of natural syntax are considered that are well understood. The approach, to describe an ontology by the very structure of language itself, is not a matter of course. Usually, ontologies are linked with databases that model the semantic content by simple logic statements and/or link them to each other paying no attention to the dynamics of world knowledge. The approach here will make a first step towards paving the way for a broader understanding of ontology design in which it could also be possible to take world knowledge into account because language encodes some substantial part of it. Since natural language also encodes other concepts of very specific domains, the work at hand is an attempt to give some orientation to devise a very abstract and general architecture. Within this framework, we suggest that linguistic approaches to ontology building can be exploited. The advantage is that language ontologies do not only serve the purpose of knowledge representation, which they also do as a prerequisite for general and specific domains, but also reveal important insides on the language system. Such a design

should make the processing of information in the format of natural language as the preferred form of communication among humans easier since it is already represented in a semi-formal format.

Looking at the typical process of ontology development, one can identify different stages that enter at some point into an iterative process (Figure 1). Having defined the scope of the ontology (see chapter on corpus design) and neglecting the issue of reuse, the developmental work on the ontology is restricted to find appropriate terminology, define classes, properties and constraints as well as create instances of the defined classes. Since the ontology encodes the terms of a certain kind of language, that is business language, we expect to find the right terms in the text itself. Using the information about the structure of the language, one can at least in part define the properties (attributes) of the instances of the classes that are the relations expressed by verbs in a text. Classes and instances are whole-part mappings that are also present in language or can be added by existing domain specific ontologies or heuristics.

Even though whole-part relations are a complicated matter for language engineers, one can use language expressions to a certain degree. As an illustration consider:

a) The colour of the house is red

Stripping down the genitive specifications and other specifiers, one gets: colour is red, which is a structure that reveals subset relations (see chapter 3.2).

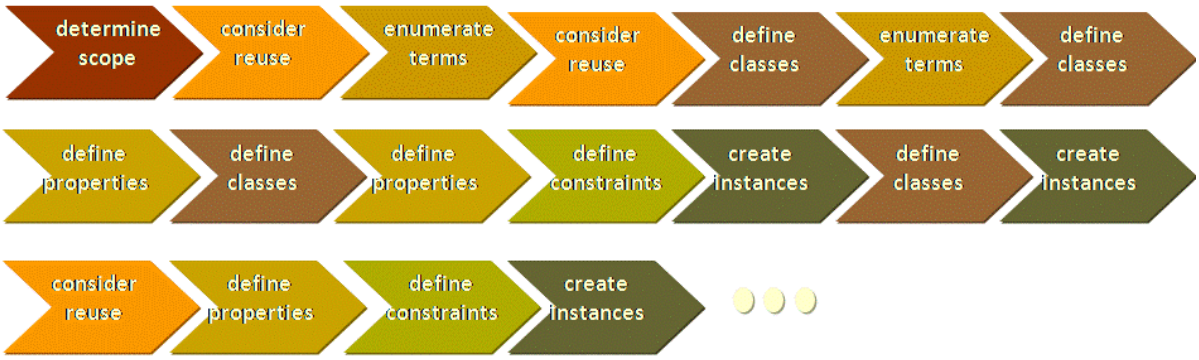


*Figure 1: Ontology Development as an iterative process. Source: http://www.co-ode.org/resources/tutorials/*

As a result, we can facilitate the work of ontology development by semi-automatic processes. Instead of defining each new term and its properties, we can use language dependent knowledge to automate this process. Thus, a text corpus can be taken as an input out of which a first ontological structure emerges. Of course, some classes and its instances we expect to give more fine tuning or complete new definitions. Still the amount of work saved by using more information that is encoded in the language input will pay off for these corrections.

To accomplish the ideal goal of building an ontology out of raw text, we have to survey possible strategies of text analysis. Thus, this deliverable presents different linguistic and statistical methods. In addition, it shows first results of their application to the exemplary samples of companies' descriptions (SMEs) taken from the web. Furthermore, we are going to show the role and impact of the linguistic analysis in the ontology building process. Some of the methods have been already implemented by UniKassel and serve different purposes:

- linguistic analysis of unstructured texts
- linguistic preprocessing and analysis of metaphors (please refer to the deliverable D6.3 for more detail)
- ontology building from text through linguistic analysis

# 1 Taxonomy, Ontology and Natural Language

A taxonomy is defined as "a partially ordered set of taxoms (classes) in which one taxon is greater than another one only if what it denotes includes what is denoted by the others" (Euzenat & Shvaiko, 2007, p.31). We refer to taxonomy as a classification of entities (or units) into different categories based on their characteristics. We can find a taxonomic representation in different areas of content structure: thesaurus, directory structure, site maps. Figure 2 illustrates an example of taxonomic structure taken from the web site of the SME called Domain Solutions (Appendix 6.1).

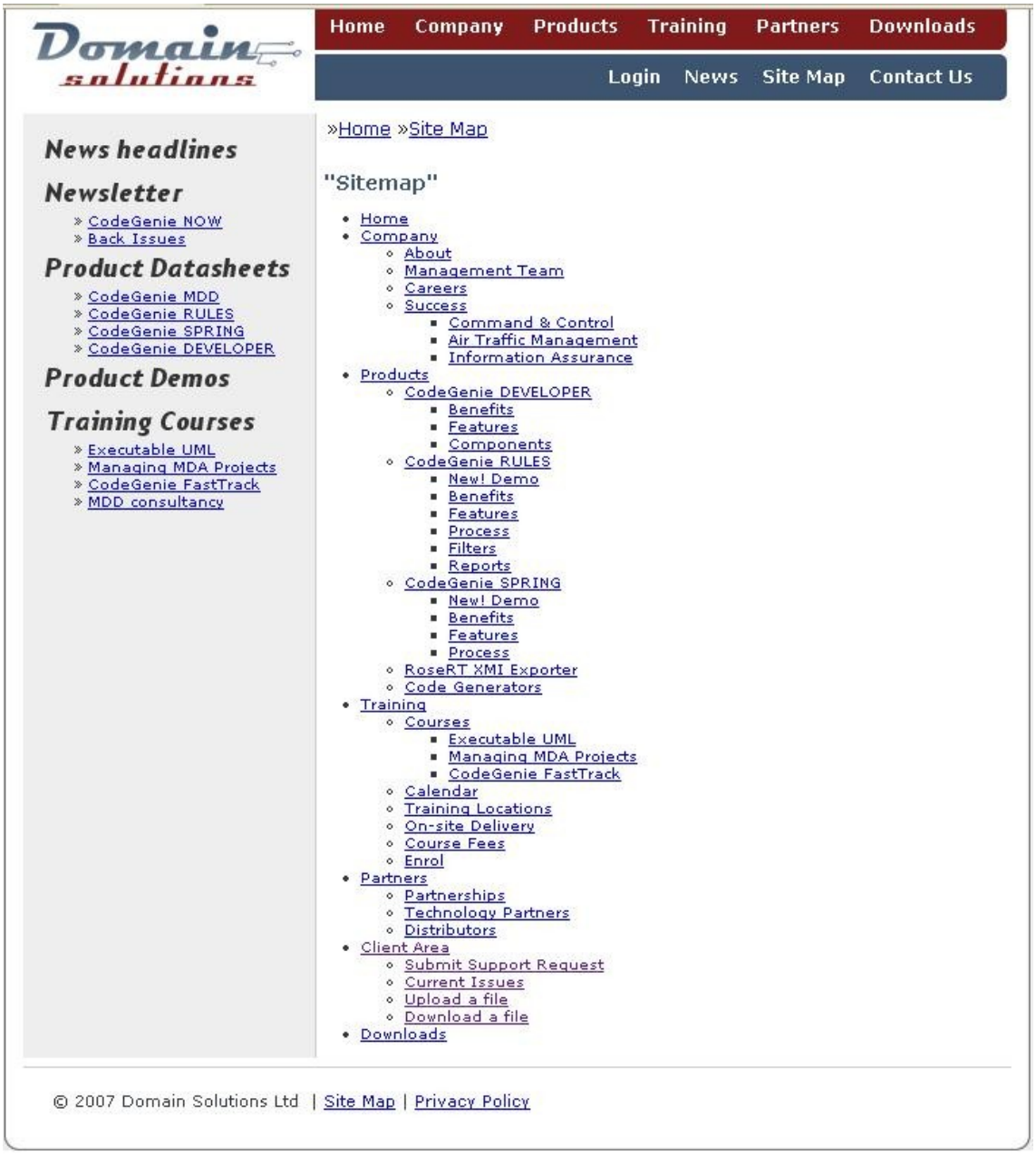


*Figure 2: Taxonomic representation. Source: http://www.domsols.com/*

A taxonomic representation involves a hierarchical structure of items according to parent-child relations between them. Taxonomies are the core structure of the thesauri that use hierarchical plus associative relationships between terms.

A good example of hierarchical and associative relationships between concepts is the AGROVOC Thesaurus developed by Food and Agriculture Organization of the United Nations (FAO). Table 1 illustrates exemplary types of relationships assigned to the term **fish** in those database:

| ***Relationships*** | ***Related term*** |
|---|---|
| Broader Term (BT) | Fishery products |
| Related Term (RT) | Seafoods |
| Scope Note: Scope Note Reference (SNR) and Is Referenced in Scope Note (SNX) | Fishes |
| Used For (UF) | Fresh fish |

*Table 1: Relationships between terms in AGROVOC Thesaurus*

The important factor in both, ontological and taxonomic representations is the relation between terms and concepts. In summary, several kinds of relationships are recognized in terminology work:

- *hierarchical relation*: relation between two concepts which may be either a generic relation or a partitive relation
  - ➔ *generic relation*: genus-species relation between two concepts where the intension of one of the concepts includes that of the other concept and at least one additional delimiting characteristic
  - ➔ *partitive relation*: part-whole relation between two concepts where one of the concepts constitutes the whole and the other concept a part of that whole
- *associative relation*: pragmatic relation between two concepts having a non-hierarchical thematic connection by virtue of experience

An ontology specifies and constructs a knowledge surface by defining specific concepts and relations between them. The most common definition of an ontology was defined by Gruber (1993): “An ontology is an explicit specification of a conceptualization”(p.199). Borst (1997) has extended this definition to: “An ontology is a formal specification of a shared conceptualization” (p. 12).

The notion of concept is the core unit of an ontology. What exactly do we mean by this term?

Figure 3 illustrates the linguistic view: a concept is the mediator that relates the symbol to its object[1] (Sowa, 2000).

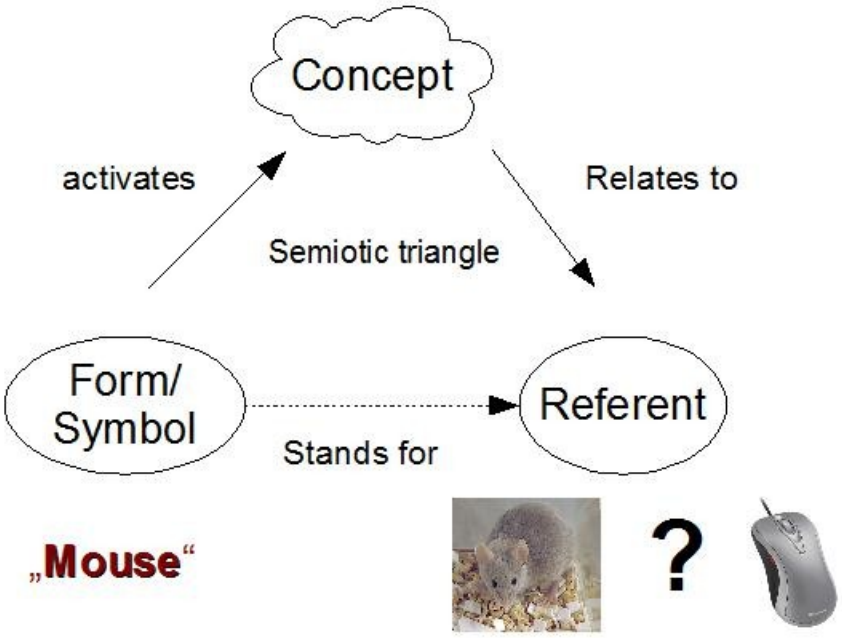


*Figure 3: Concept definition, linguistic view. Source: Ogden, C.K. & I.A. Richards (1923)*

Consider the word mouse. The sequence of characters *m-o-u-s-e* that can be easily recognised by a computer is just a string, nothing more. However, this sequence of characters refers to a real object (or several objects). By adding semantics (= meaning) to the string we link the string, its lexical reference and the concept it represents.

In our example, *mouse* can refer to an animal as well as to a computer device. In ontological representation, this concept is explicitly defined. This aspect illustrates one of the major differences between ontology and natural language. The distinction lies in the ambiguity as well as "in the constructed and overtly defined nature of ontological concepts and labels on which no human background knowledge can operate unintentionally to introduce any ambiguity, as opposed to pervasive uncontrolled ambiguity in natural language" (Nirenburg & Raskin, pp. 152-157). An important property of an ontology is its "context-independent notion of what it means to be an instance or a subclass of a given concept" (Hepp, 2008, p.7).

Furthermore, this example (mouse as an animal and as a computer device) illustrates the metaphorical mapping between different objects existing in real environments. We claim that

1 Within semiotics, the concept is what Peirce describes as the interpretant mediating between representamen and object.

business language is rich of metaphors (see Deliverable 6.3 for more detail). Therefore, the integration of results from those research to our investigations of ontologies seems to be of significance.

Generally, an ontology consists of:

1. Concepts

To people who own or operate a small or medium enterprise (SME), an applied ontology is a written representation of the concepts they use to describe:

- ➔ their product and service offerings,
- ➔ their organization,
- ➔ the organization of their customers, suppliers, partners, and regulators to the extent the SME understands these organizations,
- ➔ their policies and procedures,
- ➔ their physical, business, and political environment

When written in the SME's natural language, such an ontology is the vocabulary of the SME's business language. By publishing such an ontology in a digital ecosystem, the SME is making a commitment, an ontological commitment, that the things represented by the concepts in the ontology exist or may exist and are conceptualized in the way they are defined in the ontology. By making such a commitment, the SME is saying that others can take them at their word when interpreting descriptions such as those listed above that are made using the vocabulary. Such commitments, especially when they are available in machine readable and interpretable form, make it possible to construct software systems for deployment in a digital ecosystem to support the business of the SME.

2. Properties of concepts and relationships between them

Properties may include the taxonomic structure, that provides hierarchical representation of concepts based on the relationships (for example, is-a, part-of, etc.). Additionally, non-taxonomic relationships, roles, and attributes are involved in the ontological structure (Figure 4).

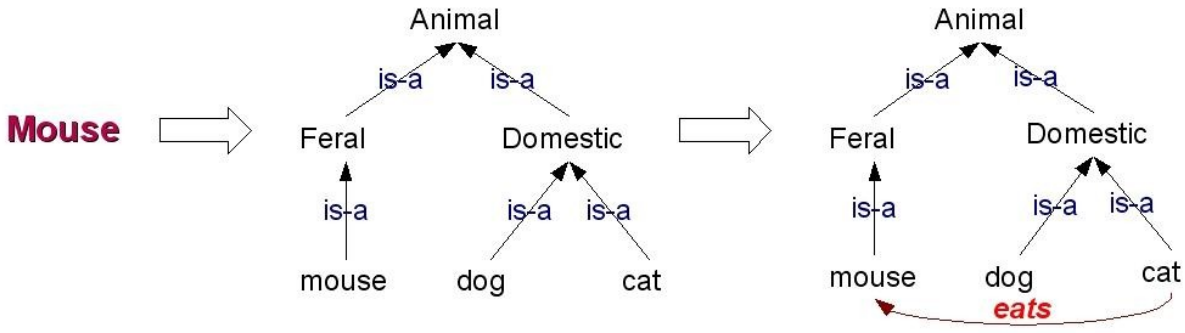


*Figure 4: Relationships in taxonomy and ontology*

An ontology includes essential relationships among its concepts. One might ask, "What makes a relationship essential?" A relationship or characteristic of a concept is essential if it is indispensible to understanding the concept. This, in turn, depends on the intent and purpose of the SME whose vocabulary it is. A person will ascribe different meanings to a word in different usage contexts. When writing for use in a digital ecosystem for advertising and performing business transactions, what is essential is what is indispensible for these purposes. It is the essence of the language for that purpose that must be included in an ontology for use in a digital ecosystem. In the end, what to include is a judgment call of the SME. In general, there are no implied constraints in a business ontology. Other concepts exist that are not included in the vocabulary. Included concepts may have other relationships and characteristics that are not included. Anything is permitted that is not expressly prohibited (unless, of course, it is impossible). Anything that is not asserted is assumed to be unknown; its negation is not assumed. There are no "default" values. These assumptions are known as the "open world" assumption. A SME may explicitly close particular concepts and relations in his or her vocabulary, but closure needs to be explicit for each concept that is closed. For example, a SME may state that their "customer" concept is closed, meaning that if you are not on their list of customers, you are not a customer. Another SME may have only anonymous customers, or know only some of their customers; "customer" is not closed in their vocabularies. Queries of their customer database would necessarily have a different interpretation than if "customer" were closed. "Customer" rules would be different.

The core structure of an ontology includes concepts and relationships between them. The extraction process of these elements from the complex textual data may involve different combinations of linguistic, statistical, and other methods and algorithms. In the context of ontology building from text, we split down the problem of finding concepts in text to identifying keywords, terms, and

synonyms that can form the conceptual surface. We argue that terms are the basic and most important units in conceptual structure of a specific knowledge domain. Furthermore, the concepts are interconnected through different kinds of relations. The identification and extraction of such relations from text requires deep linguistic analysis.

In the following chapters we introduce linguistic and statistical methods that may be involved in the extraction process of concepts and relations. We are currently developing and testing these approaches. The next sections will give an overview of the methods and some exemplary results of the analysis. Text analysis has been done based on the collection of SMEs' description taken from the web (see Appendix 6.1).

# 2 Linguistic and statistical methods in ontology extraction

## *2.1 Linguistic methods*

### 2.1.1 Corpus linguistics

To build an ontology encoding the semantic domains of natural language and some of its structural relations, we need a model case of natural language. A representative sample of natural language is not easy to draw because its population has no exact definition for such purposes. However, we do need a representative sample to have some founding for the architecture of the ontology as well as its content. The branch of science engaging in this research domain is corpus linguistics. So we have to survey this scientific field to find the most appropriate solution.

Problems of representativeness and corpus design are addressed by a number of researchers (Church/Mercer, 1993; McEnery/Wilson; 1996; Kennedy, 1999). Taken the number of citations of researchers in this field, the paper by Douglas Biber (1993) gives the best directions to approaching the problem. Biber lays down two dimensions which specify the representativeness of corpora. First, the range of text types in a language, and second, the range of linguistic distributions in a language. According to that, the representativeness of a language sample depends on how many text types (e.g. news, novels, sports) are admitted, but also how different the linguistic units occur within each type. The first dimensions hardly needs any further explanation. It is plausible even to the layman that language differs between poetry and scientific journals. This classification emanates from a content analysis, and it is therefore an internal measurement. They are usually referred to as registers or genres. The second, however, is more subtle and depends on the first. In analogy to the first dimension, it is called an external measure. The content remains unconsidered. Instead the researchers looks at co-occurrences within each text type. Such co-occurrences show characteristic patterns of word type relations, e.g. noun-adjective co-occurrences. The objective is to have text types that are minimally different in their linguistic distributions within one type though maximally different between the types.

How do we put these text types together, how many and which? For a complete language, the target population could be all of the texts ever spoken or written. Since this is impossible, we have to constrain the samples of the corpus reasonably. With regard to the purpose of the corpus study, we do not need historical texts and may neglect time as an impacting factor. Second, we are most

interested in the language of businesses. Therefore, we could even restrict samples to that area and exclude genres as literature, philosophy and the like. Even this remainder is challenging enough since within business domains, we have important differences in written media (e-mails, newsgroups, newspapers, Executive Summary Report vs. operational instructions, etc) and spoken communication (conferences, meeting, telephone calls). A pilot study should specify, first, which of these registers have to be taken into account and, second, an estimation of the emphasis thereof.

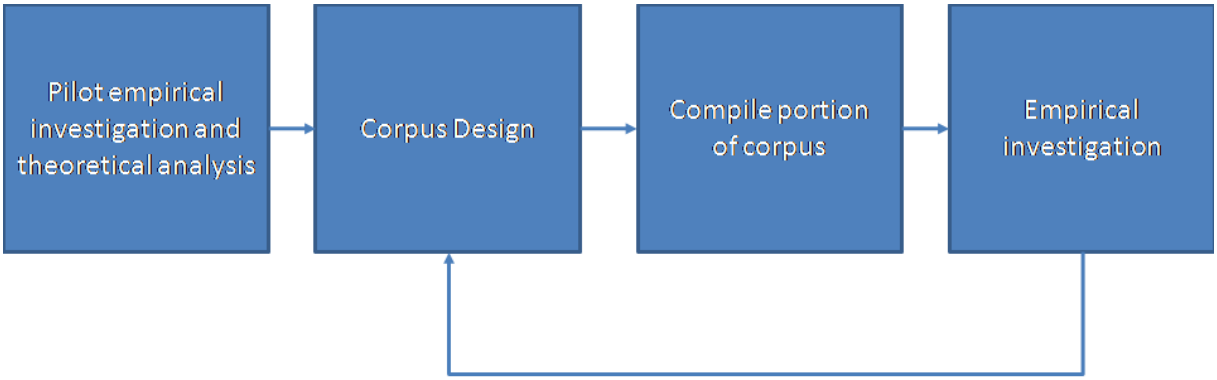


*Figure 5: Developing a representative corpus. Source: Biber (1993, p. 256)*

Following Biber's proposition of developing a representative corpus (Figure 5), a theoretical analysis and some explorative investigations are necessary provided that the target population is well defined. After that, the researcher thinks of putting the text types together. Once a certain composition of texts is put together, he/she will have to check whether this collection is correct. Accordingly a sample from this combination of text types is drawn. This sample will now be evaluated according to certain criteria that are surveyed empirically. Should the sample and empirical results deviate substantially, the corpus design has to be adjusted, that is, a different combination of text types must be chosen. Fine tuning the corpus design is therefore a cyclic process. With each cycle the representativeness of the corpus will increase until it reaches a very high level of quality.

The empirical investigation comprises the matching of corpus data with reality, i.e. does the corpus contain all properties that are relevant for the investigation. For example, a corpus representing written English should not only contain all the genres and its typical use of words and constructions of all written English, but also their correct distribution or number in relation to the corpus size. The empirical investigation serves foremost to minimize the discrepancy between target population (written English) and sample (representative corpus) according to the dimensions mentioned above.

To illustrate, we need to imagine an ideal person that resembles the average behavior of all people

involved in managing businesses. Then we can look at this person over a fixed period of time. Let's say a year and check for each day by which sources of language that person is affected. She might read a newspaper for an hour, have a meeting for two hours, write a report for another hour, write some ten e-mails for three hours, phone calls for another hour and last gives orders to employees for ten minutes. Averaging all actions taken over a year by the hours spent, will give us a normalized scale as a weight for the text types or registers.

Really, the ideal person does not exist and so it has to be founded on a sample of some thousand representatives that have to be questioned on their daily actions. The average alone might not be the best measure since actions taken differ between business branches and their hierarchical level notwithstanding that managerial tasks are more similar the closer they get to the organizational peak.

As a first step, we like to present *first result* of a showcase model. Since the production of an entire corpus of business language is a task of intense work for a decade, we have to stick to the resources available to us right now. Once all resources are accessible, they can easily be applied to the model proposed below. So our procedure can be outlined as follows. We will restrict ourselves to one text type typically involved in business corpora for this deliverable. Such a register is a company description of SMEs available on the web. The list of SMEs whose descriptions we used in our research can be found in Appendix 6.1. In subsequent steps, one could then add other registers and see whether they meet representativeness if compared to empirical data and whether they show the same results as our sample here.

## 2.1.2 Linguistic preprocessing and analysis

### *2.1.2.1 Tokenization*

The first step in almost every application involving linguistic analysis (information retrieval, text mining applications, computational linguistics, etc. ) is the identification of core units in text. Depending on the particular application, these units can be single words, sentences, or even paragraphs. In order to extract them, one needs to break up the sequence of characters in the text by locating word/sentence/paragraphs boundaries. This process is called tokenization. At first glance, it seems to be a very straightforward and easy approach. However, there are different factors affecting the difficulty of tokenizing natural language texts[2].

2 In this section we are leaving out the discussions about tokens identification in different languages and focusing

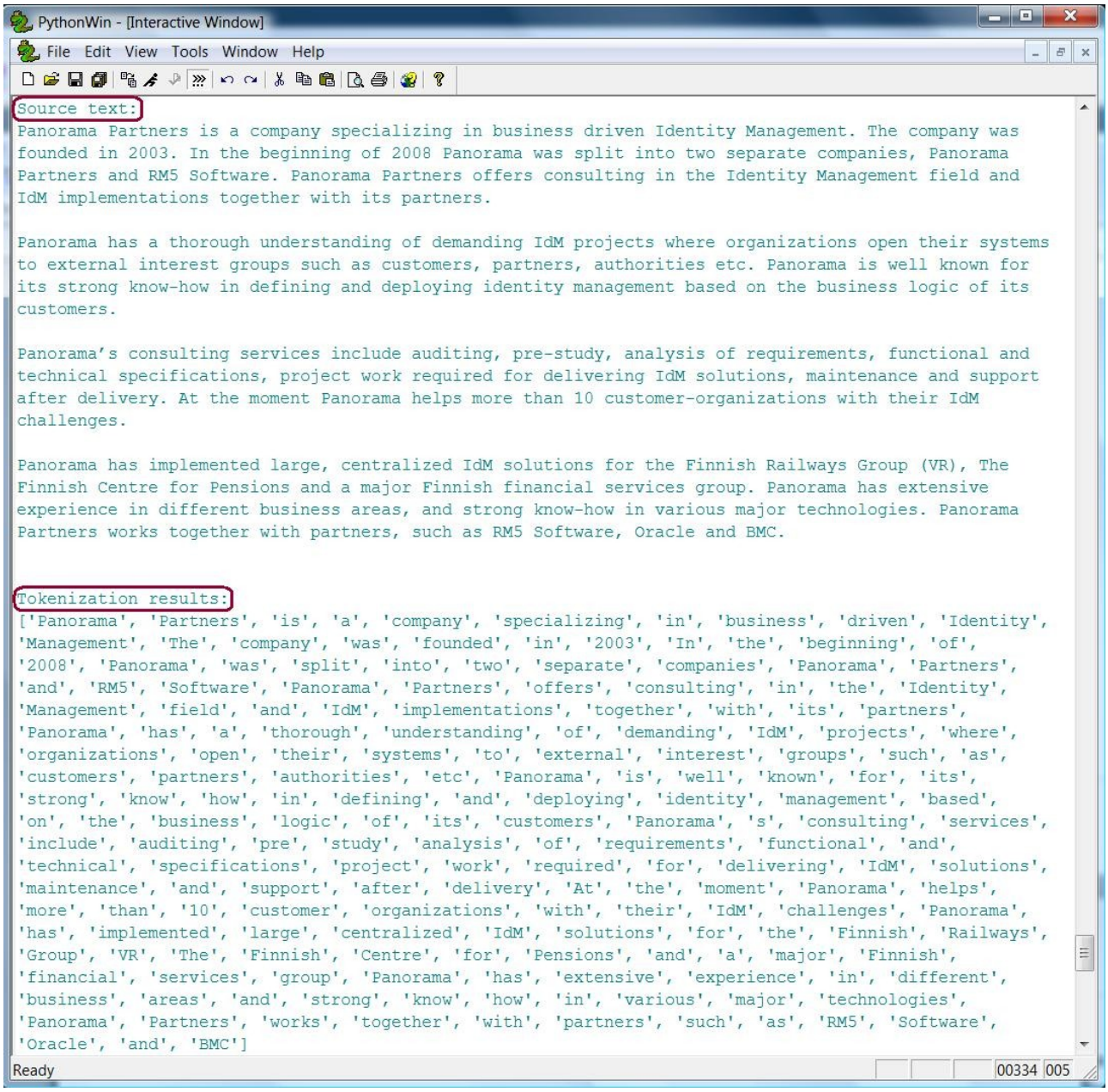


*Figure 6: Text before and after applying the tokenizer*

In our approach of ontology building and learning from text, we use words as tokens and units of analysis. For the purposes of linguistic analysis of written texts and text corpora, we developed our own tokenizer. This approach is usually used as a pre-processing step in more complex applications related to the text analysis. Taking the full control over this application from the very beginning ensures the quality of our work and provides the flexibility by implementing different modifications according to the specific needs. Figure 6 shows an exemplary output of text referring to the description of one of SMEs.

only on texts written in English.

In English texts, where words are usually separated by whitespace, we are facing the problem of ambiguity caused by use of punctuation marks, such as commas, quotation marks, apostrophes, periods (full stops), and hyphens. Not just the use of them, but also the fact that the same punctuation mark can serve different functions, causes ambiguity and as a result difficulty in identifying tokens. Palmer (2000) provides a prime example demonstrating the ambiguous character of use of punctuation marks in a single sentence:

> *Clairson International Corp. said it expects to report a net loss for its second quarter ended March 26 and doesn't expect to meet analysts' profit estimates of $3.9 to $4 million, or 76 cents a share to 79 cents a share, for its year ending Sept. 24.*

Even in such single sentence, the author identify two important aspects of ambiguity (ibid, p. 18):

- The periods are used in three different ways: within numbers as a decimal point (*$3.9*), to mark abbreviations (*Corp. and Sept.*), and to mark the end of the sentence. In this case, the tokenizer must be able to recognise that the period following the number *24* is not a decimal point.
- The sentence uses apostrophes in two ways: to mark the genitive case (*analysts'*) and to show contractions (places where letters have been left out of words, such in this case in *doesn't)*

The difficult task is to determine whether "a punctuation mark is part of another token and when it is a separate token" (Palmer, 2000, p. 18). In the context of Digital Ecosystems, language used in communication and information representation of SMEs is rich of such cases.

Even if we focus on determining tokens by a white space and taking into account the drawbacks of recognising a punctuation mark as a separate token, we may still face some problems. Consider the following examples:

*Los Angeles*

*New York*

*white space (= whitespace)*

*lower-case = lower case (= lowercase)*

There is a variety of words separated by white space that have to be considered as a single token. This problem occurs with names, compounds having several writing cases, and foreign phrases.
In the context of our research, we analysed several web pages of SMEs. Table 2 presents different cases including ambiguity in punctuation marks usage and whitespace-based tokenization:

| SME | Whitespace | Punctuation marks |
|---|---|---|
| Network Ltd | Network Ltd;<br>Network Menus™. | Web-based;<br>know-how |
| Redenet | line up | |
| Joinex Oy | high quality information systems;<br>built up | he challenge of today's business needs;<br>know-how |
| Panorama Partners | RM5 Software | Pre-study, customer-organizations;<br>know-how;<br>Panorama's consulting services; |
| Openscape | web based;<br>long term relationships; | Business-to-business;<br>real-time;<br>e-commerce;<br>backed-up;<br>with an industrial-strengh;<br>the clients' needs;<br>your company's information |
| Excellis Consultants Network | Activity based costing (ABC) & earned value analysis (EVA) | Multi-disciplinary;<br>change-plan formulation |

*Table 2: Tokenization problems in texts of SMEs*

As demonstrated in Table 2, one of the common cases is hyphenation. In the texts from SMEs we observed two types of hyphenation:

- splitting up vowels in words (for example, *e-commerce, pre-study*)
- joining words (for example, *real-time, business-to-business*)

Handling hyphens during the tokenization is very difficult. Even the short description of SMEs contains difficult cases for the process of tokenization. Currently we are working on the investigation of such problematic cases and defining major modification issues of the tokenizer.

### *2.1.2.2 Filtering*

Filtering is the process of removing the information not relevant for any further investigations in text analysis. Our filtering block includes two processes: removing of function words (so-called stop-words removal) and removing of punctuation marks. The process of filtering serve to reduce the complexity of analysed text and improve the quality of further analysis and results.

### *2.1.2.3 Stemming*

In most cases, morphological variants of words have similar semantic interpretations and can be considered as equivalent in a variety of applications. In the context of extraction and analysing ontological units in unstructured texts, stemming might be essential. Words with the same meaning appear in various morphological forms. To capture their similarity they are normalised into a common root-form, the stem. For example, the words *laughing*, *laugh*, *laughs* and *laughed* are all stemmed to ***laugh*** (Jurafsky & Martin, 2000). A number of so-called stemming algorithms, or stemmers, have been developed that attempt to reduce a word to its stem or root form. Most widely used is the Porter stemmer (Porter, 1980) that is based on a series of simple cascaded rewrite rules, such as:

| | |
|---|---|
| *sses –> ss* | *caresses –> caress* |
| *ational –> ate* | *relational –> relate* |
| *tional –> tion* | *conditional –> condition* |
| *ies –> i* | *ponies –> poni* |

The following example shows the results after applying Porter stemmer:

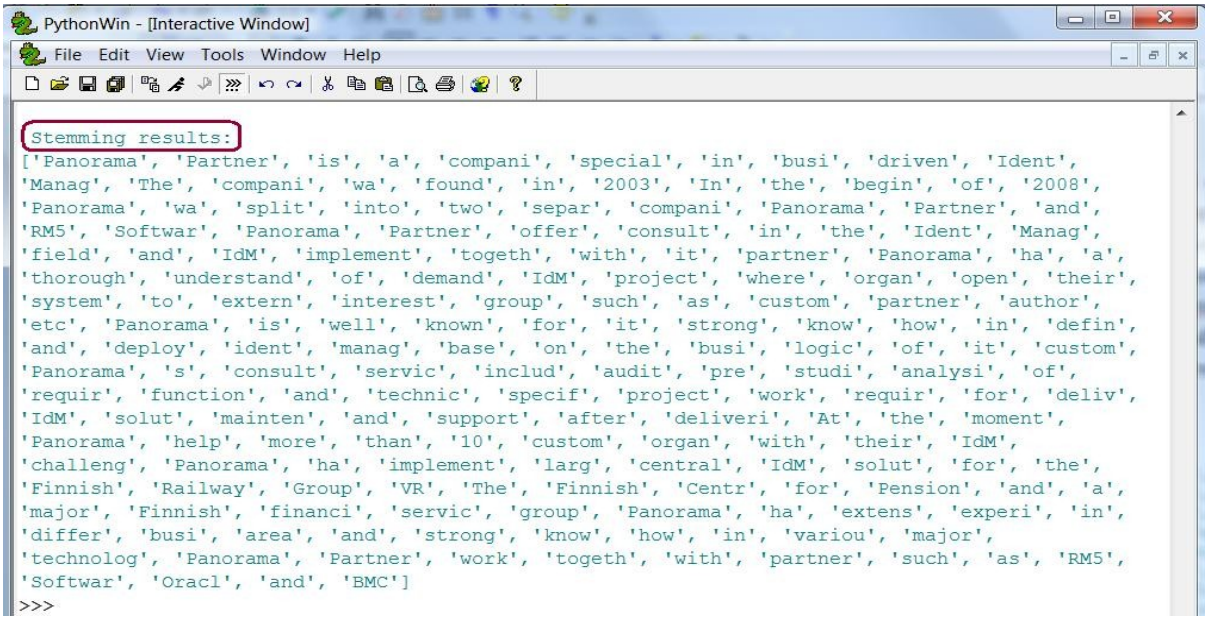


*Figure 7: Results of stemming applied to the text of Panorama Partners (Appendix 6.1)*

Stemming can reduce the complexity and dimensionality of information to be analysed. However, there are several disadvantages of applying stemming approach:

- Stemming does not recognise all morphological variants and derivations.
- Some words can be stemmed to an unusual form, for example the stem of *organization* is *organ* (Figure 7). This "false" assignment can affect the overall process of analysis.
- The reduced forms of words are difficult to interpret.

Therefore, some improvements are necessary. One possibility is to transform words into their canonical dictionary form[3], for example *specialised* into *specialise* (and not special as we can see in Figure 7).

### *2.1.2.4 Tagging*

In combination with some of the approaches described in previous sections (tokenizing, filtering and stemming), tagging can facilitate the process of ontology and taxonomy building significantly. Instead of reading and manually analysing mass of documentation in order to identify ontological concepts, relationships between concepts, building hierarchical representation, and so on, one may use the advantages of automatic identification of word's part-of-speech category.

3 This process is called lemmatisation

"Tagging is the task of labeling (or tagging) each word in a sentence with its appropriate part of speech. Many words have more than one syntactic category. In tagging, we try to determine which of these syntactic categories is the most likely for a particular use of a word in a sentence" (Manning & Schütze, 2000, p. 341). Tagging is closely related to the parsing which is described in the next section.

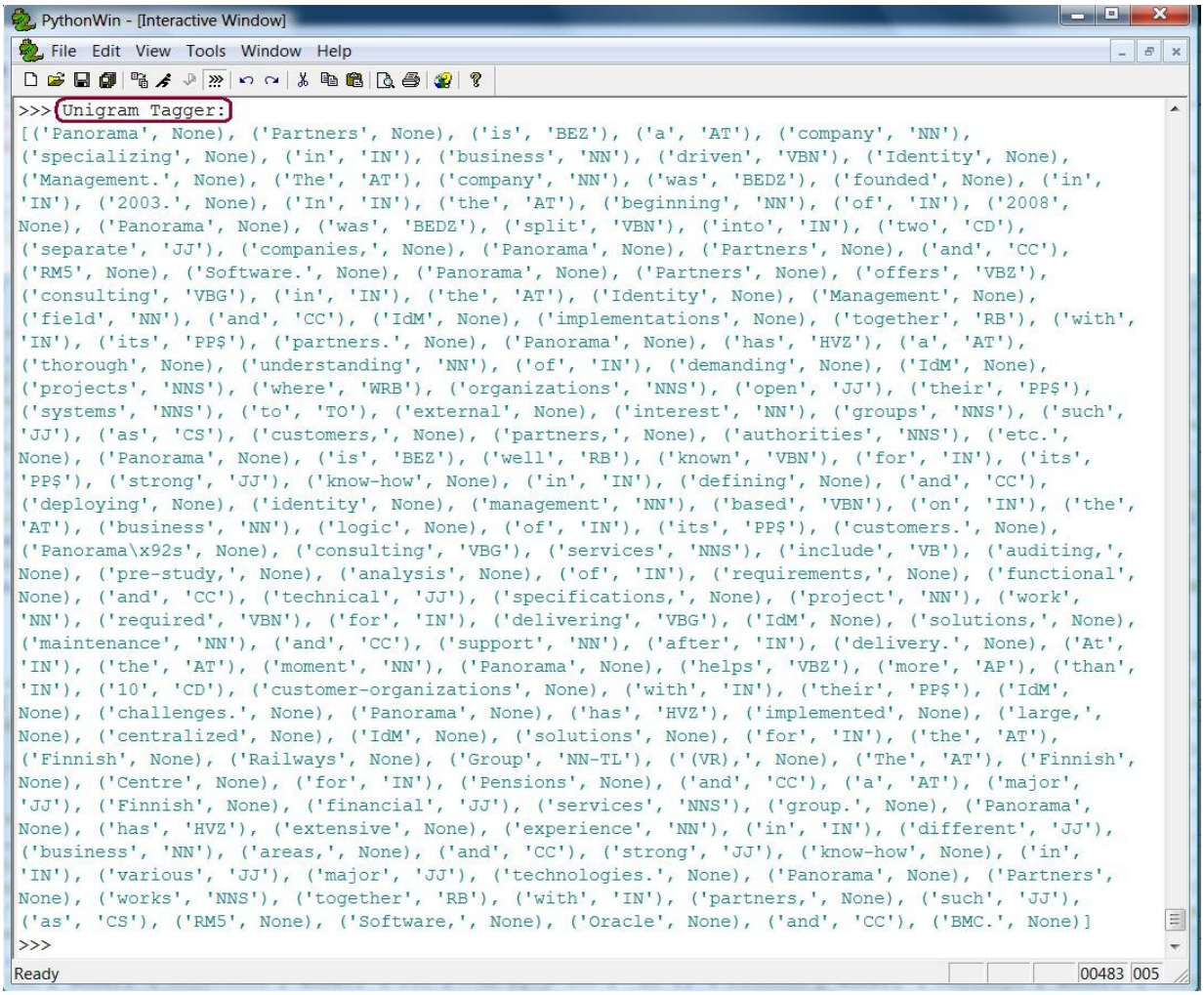


*Figure 8: Results of unigram tagger*

Figure 8 shows results of applying unigram tagger to the text. The basic idea behind the unigram tagger is to assign tags to each word *w* after analysing the training corpus (for this task we used the Brown corpus) and discovering the most frequent tag for the word *w*. At the current stage of our research, we are implementing and testing different approaches related to the tagging procedure (Unigram tagger with backoff tagger, Bigram Tagger, N-gram tagger, Regular expressions tagger) as well as use of different training corpora (Brown corpus and Penn Treebank)[4].

4 See Appendix 6.2

Furthermore, in the context of ontology extraction and learning from text, we applied the Stanford Tagger to the texts from SMEs. Here, Penn Treebank annotation has been used. An exemplary output is shown in Figure 9:

> Redenet/NNP are/VBP an/DT independent/JJ software/NN development/NN ,/, consultancy/NN and/CC technology/NN company/NN ./. By/IN working/VBG closely/RB with/IN our/PRP$ clients/NNS we/PRP strive/VBP to/TO deliver/VB software/NN solutions/NNS that/WDT enhance/VBP productivity/NN and/CC tackle/VB complex/JJ business/NN paradigms/NNS ./.

*Figure 9: Results of applying Stanford tagger*

Usually, tagging is used as a pre-step to parsing. "One difference between parsing and tagging is that parsing normally aims to analyse the overall syntactic structure of the text, whereas tagging has the goal of identifying only the grammatical categories of the individual words" (Somers, 2000, p. 379).

#### *2.1.2.5 Parsing*

In previous chapters we focused on words: how to identify them, how to analyse their morphology, and how to assign them to different classes via part-of-speech tags. Nevertheless, there seem to be linguistic regularities that cannot be analysed by using aforementioned approaches. In this section we are moving towards syntactic representation of sentences and its necessity in analysis of ontological components in texts. In particular, this approach can be successfully applied for the purposes of identifying concepts, relationships between them even some properties of them. We refer to the syntactic analysis of sentences according to a certain grammar theory as a process of parsing. Generally, parsing can be done based on two approaches (Feldman & Sanger, 2007):

1. Constituency grammar
2. Dependency grammar

##### 2.1.2.5.1 Constituency grammar

The fundamental idea of constituency is that groups of words form a single unit or phrase, called constituent (Jurafsky & Martin, 2000). Formally, the constituency grammar is defined as following (Nugues, 2006, pp. 244-245):

1. A set of designated start symbols ($\Sigma$), covering the sentences to parse.

2. A set of non-terminal symbols enabling the representation of the syntactic categories (sentence and parse categories)
3. A set of terminal symbols representing the vocabulary: words of the lexicon, possibly morphemes
4. A set of rules, *F*, where the left-hand-side symbol of the rule is rewritten in the sequence of symbols of the right-hand side.

Parsing systems based on constituency formalism are usually referred to the context-free grammars (CFG). Considering the sentence

*Our services provide a global solution for the SME* (Izanet),

following transformation rules are applied to the extraction of the syntactic structure:

*S –> NP VP*

*NP –> PRP$ NN*

*VP –> V NP*

*NP –> NP PP*

*NP –> DT JJ NN*

*NP –> DT NN*

*PP –> IN NP*

*NN –> services | solution | SME*

*V –> provide*

*DT –> a | the*

*PRP$ –> our*

The annotation schemes of constituency grammar usually reflect four main parts of speech, namely noun, verb, adverbs, and adjectives. Hence the categories of phrasal constituents are:

**Noun Phrase** (NP) consisting of a noun and its possible modifiers:

*our services*

**Verb Phrase** (VP) consisting of a verb together with its objects. Such an object can be an another noun phrase

*is a company*

or the verb can be followed by a preposition phrase (PP):

*specializing in business*

**Adjective Phrase** (AdjP) is headed by an adjective with possible modifiers.
In the sentence

*Ensure all your company's information is secure and regularly backed-up* (Openscape),

the adjective phrases would be

*secure and regularly backed-up*

*secure*

*regularly backed-up*.

**Adverbial phrase** (AdvP) is a phrase headed by an adverb. Considering the previous example, we might identify an adverbial phrase inside the adjective phrase:

*[ADJP [ADVP regularly] [V backed-up]]*

A further commonly used category is the **prepositional phrase** (PP), which is actually a noun phrase beginning with a preposition. The common transformation rule is *PP –> IN NP*
In the sentence

*Panorama Partners is a company specializing in business driven Identity Management.* (Panorama Partners),

the prepositional phrase would be:

*[PP [IN in] [NP [business driven Identity Management]] ]*

The more complete picture of a sentence structure (parse tree) based on constituency formalism is shown in Figure 10. The parse tree reflects the constituents and their hierarchical representation in the sentence

> *Redenet are an independent software development, consultancy and technology company. By working closely with our clients we strive to deliver software solutions that enhance productivity and tackle complex business paradigms.* (Redenet)

```
(ROOT
  (S
    (NP (NNP Redenet))
    (VP (VBP are)
      (NP
        (NP
          (NP (DT an) (JJ independent) (NN software))
          (NP (NN development))
          (, ,)
          (NP (NN consultancy)
            (CC and)
            (NN technology) (NN company))
          (. .))
        (SBAR
          (S
            (PP (IN By)
              (S
                (VP (VBG working)
                  (ADVP (RB closely))
                  (PP (IN with)
                    (NP (PRP$ our) (NNS clients))))))
            (NP (PRP we))
            (VP (VBP strive)
              (S
                (VP (TO to)
                  (VP
                    (VP (VB deliver)
                      (NP
                        (NP (NN software) (NNS solutions))
                        (SBAR
                          (WHNP (WDT that))
                          (S
                            (VP (VBP enhance)
                              (NP (NN productivity)))))))
                    (CC and)
                    (VP (VB tackle)
                      (NP (JJ complex) (NN business) (NNS paradigms)))))))))))
    (. .)))
```

*Figure 10: Parse tree*

### 2.1.2.5.2 Dependency grammar

As its name implies, Tesnière's (1959) *Dependency Grammar* uses interrelated dependencies within a sentence. The information encoded in the structure of a sentence can be manifold. Theories within the framework of *Dependency Grammars* concentrate on but one such source of information. As opposed to aforementioned *Phrase-Structure-Grammar*-models, *Dependency Grammars* disregard the linear order of words, but look at words in terms of how they govern other components of the same sentence. Furthermore, the function of subjects and predicates are lost sight of. Instead, the verb itself becomes central. Tesnière liked to find a hierarchical representation of language structure that avoids the dissection of the constituent structure as it is the case in *Phrase-Structure-*

*Grammars* that split up into nominal and verbal phrases (Stötzel, 1970). His approach is based on the properties (valency) of words. In particular, *Valency Theory* assumes that the verb binds other obligatory and optional components to it. Tesnière's analogy is the atomic model in chemistry, in which, for example, formic acid is composed of a carbon molecule binding one hydrogen molecule and two oxygen molecules, which either show a double connection to it or which bind another hydrogen molecule. Applied to *Dependency Grammar*, the order of words of a sentence from left to right matters as much as the order of molecules. One may simply turn and twist a formic acid molecule, and it is still the same molecule. The verb takes over the role of the carbon molecule that governs the oxygen, possibly a noun that again dominates an adjective (alias one of the hydrogen molecules). The remaining hydrogen molecule may be the subject of the sentence. So a sentence like:

*He puts the milk into the white fridge.*
would fit the pattern of a formic acid.

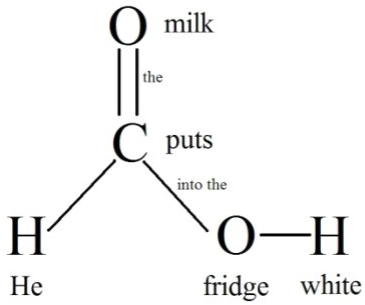


*Figure 11: Dependency structure*

From the given example, we can see that the verb dominates the entire sentence. The *s* in *puts* tells us that some third person has to be used as an agent. Probably more striking is the quality of the verb to bind obligatorily two more objects encoding theme and location or goal respectively. The meaning of the sentence is thus verb-centered. However, one of the nouns are specified by an adjective. Here, this part of the sentence can be described as noun-centered meaning.

Since English has a very poor morphology, one may argue that word order is important. Indeed, this might be an issue looking at the outer form of the language. In Figure 11 this could be marked by the double binding of the oxygen. Still, this does not matter for the concept of *Dependency Grammar*. The meaning is still unambiguous. The verb dominates the *who* (he), the *what* (milk), the *where* (fridge) and itself the action carried out. None of these concepts need a linear order that is

necessary for understanding the meaning of the utterance.

*Dependency Grammars* certainly show a hierarchy and they are definitely taxonomic. The main verb is the dominating pivot on the highest level of the hierarchy. The taxonomy goes along the lines of the verb's components. In the example above, *he*, *milk*, and *fridge* do have the same status and can be classified together. Moreover, they are on the same level of importance within the hierarchical structure since none of them is optional or can be left out. The latter is the case with the adjective *white* which could be classified as a specifier of the noun. As such it would occur at a lower level of the hierarchy because it could be left out and the meaning is still clear. The information that the specifier contributes is additional.

Considering the sentence

*She opened the door with the key.*

reveals a different hierarchy. The verb holds its pivotal position and so do agent (*she*) and theme (*door*). However, the instrument (*key*) is optional and should be put into a lower level of the hierarchy.

Regarding the initial objective – which is an appropriate form of representation that also allows efficient processing – we have to think of *Dependency Grammar* in terms of its applicability to a digital ecosystem. Although the approach became most popular for its applications in formal logic, one does not have to follow these strains necessarily. Frege's Principle of Compositionality has its limitations when becoming specific (Partee, 1984) and Montague's (1970) homomorphism is only true for some cases of natural language, but not for the language system as such.

Analyzing sentences according to this model would make it substantially easier to process an utterance into an ontological format. Ontologies, by definition, are used to classify concepts and the relations among them. This is exactly what *Dependency Grammars* aim at. The verbs define the relation among the concepts. The concepts themselves can be classified into their role fulfilling in the sentence and according to their semantic content.

#### 2.1.2.5.3 Constituency versus Dependency

Constituency grammar describe the syntactical structure of the sentences in terms of phrasal hierarchies. In contrary, dependency grammars focus on the direct relations between words in a

particular sentence. Figure 12 illustrates the difference between parsings based on constituency and dependency grammar[5].

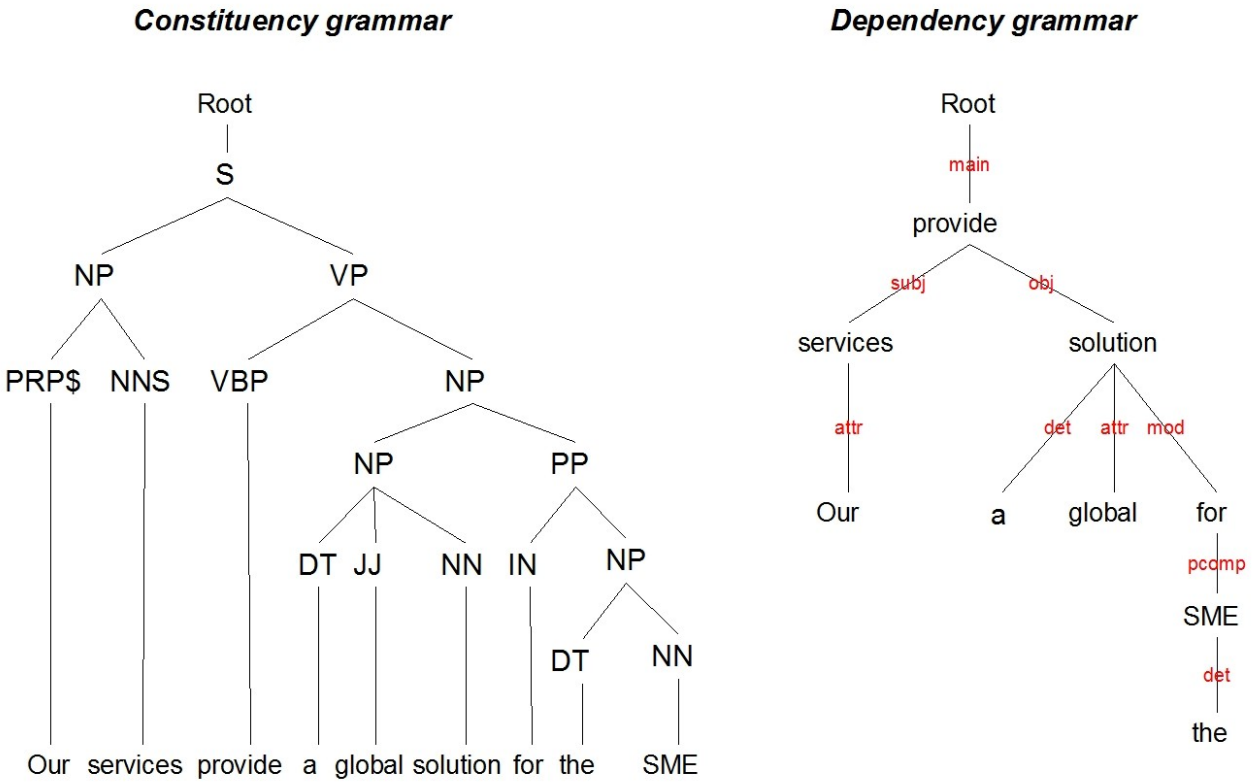


*Figure 12: Parsing trees based on constituency and dependency grammar*

In respect to ontologies, we see the relevance of applying both methods. The phrase-based approach identifies phrases and structural categories in a given sentence. Analysing the sentence structure through the lens of a constituency grammar, we might be able to extract the relevant information of the phrase-boundaries helping in identification of concepts. Dependency grammar, on the other hand, seems to be significant in identifying the relationships[6] between concepts and attributes of a particular concept. The reason is its ability to discover head-based relations (e.g. verb as a head), functional categories (e.g. subject, direct object, complement of a preposition, and others).

5 The exemplary sentence has been taken from the web site of Izanet (see Appendix 6.1)

6 See Appendix 6.2.3 for more detail

## *2.2 Statistical methods*

### 2.2.1 Collocation

Collocations are groups of words which frequently appear in the same context. Two words are collocated if the occurrence of one is closely related to the occurrence of another. Collocations, defined as arbitrary and recurrent word combinations (Benson, 1990) or institutionalised phrases (Sag et al., 2002), represent a subclass of multi-word expressions that are prevalent in language and constitute a key problem for natural language processing (NLP) and text mining.

Collocations can be divided in two classes (McKeown & Radev, 2000):

- Grammatical collocations can be in form different combinations of syntactic categories, such as verb-preposition (e.g. *focus on, specialize in*), noun-preposition (e.g. *piece of*) , and others.
- Semantic collocations refer to the "lexically restricted word pairs" (p.511) and represent a set of words that are related to each other based on a particular criteria. For example, following sets can represent semantic collocations: *{boy, girl}, {dollars, cents}, {car, wheels}*.

### 2.2.2 C-value

The C-value method is a domain-independent method used to automatically extract multi-word keywords and terms from a given document corpus. "It aims to get more accurate terms than those obtained by the pure frequency of occurrence method, especially terms that may appear as nested within longer terms" (Milios et al., 2003, p. 276). It uses statistical characteristics of the candidate string such as frequency of occurrence in the corpus, frequency as part of other longer candidate terms, the number of these longer candidate terms, the length of the candidates.

Furthermore, this method enhances the common statistical measure (frequency of occurrence) and makes the measure sensitive to a particular type of multi-word terms (the nested terms) (Frantzi et al., 2000). It applies more weight to nested terms. C-value is formally represented as follows:

$$C-value(a)=\left\{\log_2(|a|\cdot f(a))\right\} \text{ if } a \text{ is not nested}$$

$$C-value(a)=\left\{\log_2|a|(f(a)-\frac{1}{(P(Ta))}\sum_{b\in Ta} f(b))\right\} \text{ , otherwise.} \quad (1)$$

$a$ = candidate term

$b$ = longer candidate terms

$|a|$ = length of the candidate term (number of words)

$f(a)$ = frequency of occurrence of a in the corpus

$Ta$ = set of extracted candidate terms that contain $a$

$P(Ta)$ = number of candidate terms in $Ta$

$f(b)$ = frequency of occurrence of longer candidate term $b$ in the corpus.

C-value is a domain-independent method for automatic term recognition (ATR) which combines linguistic and statistical analyses; emphasis being placed on the statistical part. The linguistic analysis enumerates all candidate terms in a given text by applying part-of-speech tagging, extracting word sequences of adjectives and nouns, and stop-list. The statistical analysis assigns a "termhood" to a candidate term by using the following four characteristics:

- the occurrence frequency of the candidate term
- the frequency of the candidate term as part of other longer candidate terms
- the number of these longer candidate terms
- the length of the candidate term

The exemplary results of applying C-value method in order to identify terms in an unstructured text are shown in Figure 13[7].


Panorama Partners is a company specializing in business driven Identity Management. The company was founded in 2003.. In the beginning of 2008 Panorama was split into two separate companies , Panorama Partners and RM5 Software. Panorama Partners offers consulting in the Identity Management field and IdM implementations together with its partners.

Panorama has a thorough understanding of demanding IdM projects where organizations open their systems to external interest groups such as customers , partners , authorities etc. Panorama is well known for its strong know-how in defining and deploying identity management based on the business logic of its customers.

Panorama s consulting services include auditing , pre-study , analysis of requirements , functional and technical specifications , project work required for delivering IdM solutions , maintenance and support after delivery. At the moment Panorama helps more than 10 customer-organizations with their IdM challenges.

Panorama has implemented large , centralized IdM solutions for the Finnish Railways Group ( VR ) , The Finnish Centre for Pensions and a major Finnish financial services group. Panorama has extensive experience in different business areas , and strong know-how in various major technologies. Panorama Partners works together with partners , such as RM5 Software , Oracle and BMC.

*Figure 13: Term extraction based on C-value method*

### 2.2.3 TFIDF-value

In case, one would have a text corpus containing several documents, it is worth to mention the

7 http://www.nactem.ac.uk/software/termine/

TFIDF-approach. TFIDF (Zhang et al., 2005) is a standard keyword identification method in information retrieval tasks. It asserts preference to words that have high frequency of occurrence in a single document but rarely appear in the whole document collection. The TFIDF measure is often used in the vector space model together with cosine similarity to determine the similarity between two documents. The TFIDF is calculated as follows:

$$w_{ij} = tf_{ij} \cdot \log_2 \left(\frac{N}{n}\right) \quad (2)$$

where

$w_{ij}$ = weight of term $T_j$ in document $D_i$

$tf_{ij}$ = frequency of term $T_j$ in document $D_i$

$N$ = number of documents in collection

$n$ = number of documents where term $T_j$ occurs at least once

This method has not been yet tested on the text samples from SMEs. Nevertheless, we consider the implementation of this approach in the near future.

# 3 Current case studies

## *3.1 Terms and Concepts*

Term extraction is a prerequisite for all aspects of ontology learning from text. Terms are linguistic realizations of domain-specific concepts and are therefore central to further, more complex tasks. Term extraction implies more or less advanced levels of linguistic processing, i.e. phrase analysis to identify complex noun phrases that may express terms and dependency structure analysis to identify their internal semantic structure.

In our framework, we applied the TermExtractor tool[8] to the description of the Panorama Partners company (SME) for the purposes of terminology identification. The results are presented in Figure 14.

| R | Term | Acronym | Weight ▼ |
|---|---|---|---|
| ☐ | strong know-how | | 1.000 |
| ☐ | panorama | | 0.951 |
| ☐ | identity management | | 0.933 |
| ☐ | finnish centre | | 0.928 |
| ☐ | interest group | | 0.922 |
| ☐ | railway group | | 0.919 |
| ☐ | management field | | 0.919 |
| ☐ | service group | | 0.919 |
| ☐ | external interest group | | 0.919 |
| ☐ | financial service group | | 0.916 |
| ☐ | deploying identity management | | 0.915 |
| ☐ | identity management field | | 0.914 |
| ☐ | finnish railway group | | 0.912 |
| ☐ | finnish financial service group | | 0.911 |

*Figure 14: Term extraction from the Panorama Partners company description*

Our analysis of extracted candidates leads to the conclusion that automatic identification of relevant terms in business domain might be used for the recognition of taxonomic relationships as well. From the extracted candidates we were able to build a hierarchical structure as shown in Figure 15.

8 http://lcl2.uniroma1.it/termextractor/

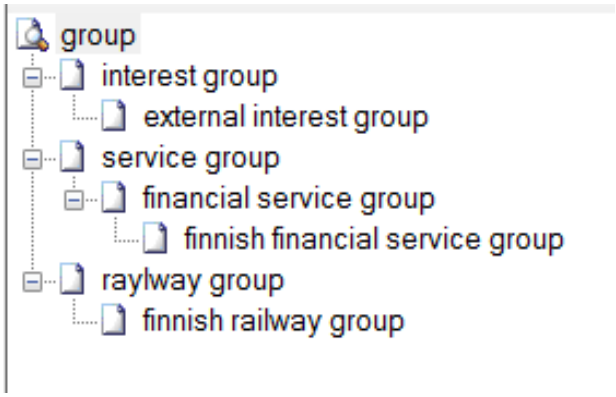


*Figure 15: Term hierarchy based on the results in Figure 14*

It is important to point out that the amount and "correctedness" of the extracted terms depend on several issues, such as involved contrast corpora, proper name exclusion, and statistical measures.

## *3.2 Relationship extraction*

Relationships in ontological as well as in taxonomic representations of information are essential. Therefore, one of the cases we consider in our investigations is related the subject of automatic identification of relationships between concepts in texts. We argue that some of the linguistic approaches could help to identify the relevant units.

Our research focuses on several aspects:

### 3.2.1 Extracting verbs as candidates for relationships between concepts

Nemrava & Svatek (2005) focus on finding verbs as simple POS category that usually occur with some product description in the product catalogue. They try to "construct ontologies containing relations labelled with extracted verbs" and "use these verbs for extracting further product categories from web pages" (p. 1). The process of verb extraction for purposes of finding ontological and/or taxonomic candidates requires a combination of different linguistic approaches (e.g. tokenizer, part-of-speech tagger, parser). Applying a parser based on the dependency grammar helps to identify not only the verbs, but also the dependency between a verb and related concepts. We analysed the sentences taken from the web descriptions of SMEs through the lens of dependency grammar. Figure 16 illustrates the possible outcome of the sentences:

> *Panorama Partners is a company specializing in business driven Identity Management. Panorama Partners offers consulting in the Identity Management field and IdM implementations together with its partners. Panorama's consulting services include auditing,*

*pre-study, analysis of requirements, functional and technical specifications, project work required for delivering IdM solutions, maintenance and support after delivery.* (Panorama Partners)

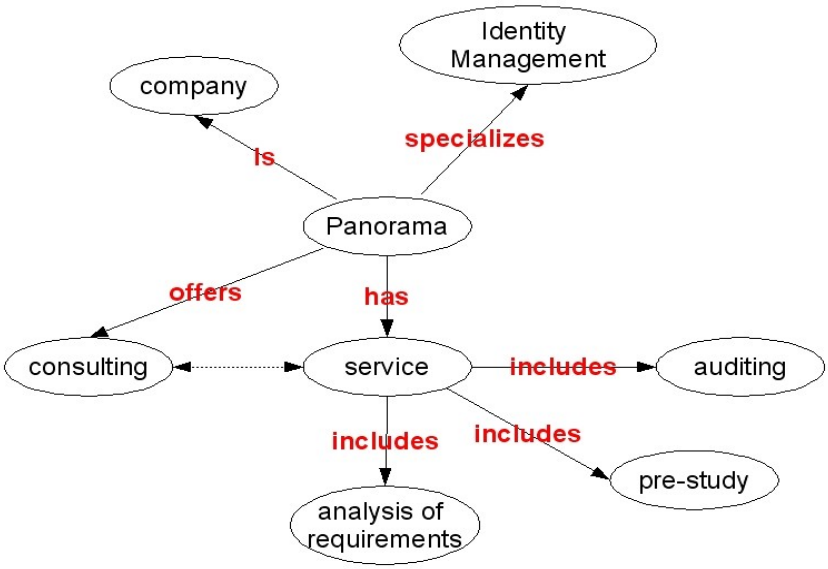


*Figure 16: Relation extraction between concepts*

Exemplary analysis of the sentences using dependency grammar provides interesting insights and opens a new dimension in the research on ontologies and their construction from the large amount of textual information. Further investigations are necessary in order to successfully implement the final step of extracting relevant relationships based on the dependency grammar parser.

### 3.2.2 Hyponymy and is-a relation

Hyponymy is one of the ways to explain the relations between words. It refers to a word with a more specialised meaning: *mouse* is a hyponym of *animal*. Contrary, the word *animal* is the hyperonym of the word *mouse*. Our investigations are related to finding the way of recognising this kind of relation in texts. Hearst (1992) identified different patterns which might help to find concepts with hyponymic relations. We applied different patterns to our samples (Appendix 6.1):

NP *such as* NP, NP, ... (and/or) NP

*Panorama has a thorough understanding of demanding IdM projects where organizations open*

*their systems to external interest groups* ***such as*** *customers, partners, authorities etc.* (Panorama Partners)

Using this pattern we are able to extract the following hyponyms:

*("interest group", "customer")*

*("interest group", "partner")*

*("interest group", "authority")*

Further example

*Panorama Partners works together with partners,* ***such as*** *RM5 Software, Oracle and BMC.* (Panorama Partners)

leads us to the following hyponyms:

*("partner", "RM5 Software")*

*("partner", "Oracle")*

*("partner", "BMC")*

The extracted information is context-specific, thus might help in identifying major actors referring, for example, to the concept "partner".

*Such* NP *as* NP, NP, ... (and/or) NP

This pattern has not been found in the samples, however, it is important to consider given a larger text corpus.

*Such fruits as oranges, nectarines or apples*

NP *including* NP, NP, ... and/or NP

*We have built systems in a variety of arenas* ***including*** *online banking, temperature monitoring,* ...(Redenet)

The extracted hyponyms are:

*("arena", "online banking")*

*("arena", "temperature monitoring")*

NP *is* NP

*Panorama Partners* ***is*** *a company specializing in business driven Identity Management.* (Panorama

Partners)

In this example, the hyponym is

*("company", "Panorama Partners")*

It is important to understand, that some of the patterns are more expressive than another. For example, the probability of the pattern *NP including NP, NP...* to represent a hyponymic relation is higher than one of the pattern *NP is NP.*

Using aforementioned patterns,we can extract *is-a* relations in texts. Considering the first two sentences (Panorama Partners), we might have following concepts and *is-a* relations between them (Figure 17).

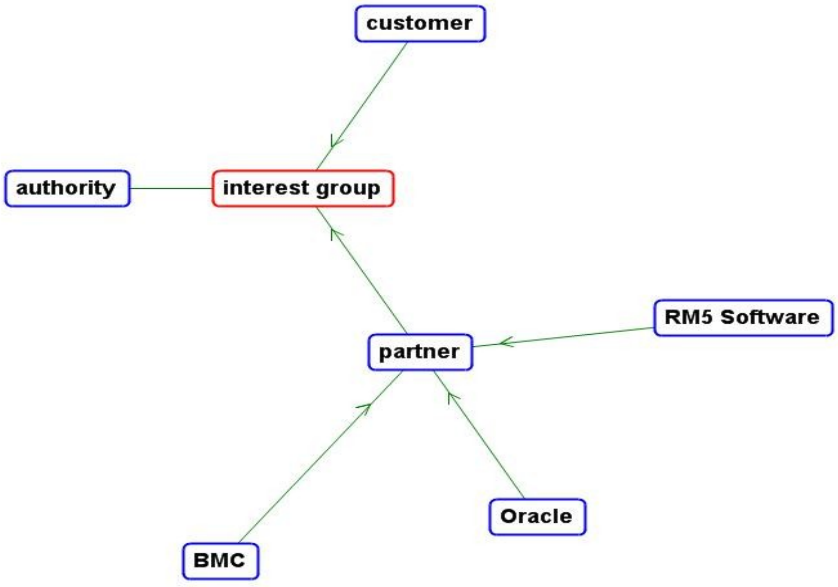


*Figure 17: Is-a relation between concepts*

The overall quality of this pattern extraction needs to be examined and analysed based on the relevant text corpora (for example, text corpora for SMEs).

### 3.2.3 Meronymy and part-of relation

Meronymy is referred to the part-whole relationship. For example, the concept *organization* has following meronymic relations in WordNet lexicon:

*Sense 2*

*administration, governance, governing body, establishment, brass, organization,*

*organisation -- (the persons (or committees or departments etc.) who make up a body for the purpose of administering something; "he claims that the present administration is corrupt"; "the governance of an association is responsible to its members"; "he quickly became recognized as a member of the establishment")*

*HAS MEMBER: advisory board, planning board -- (a board appointed to advise the chief administrator)*

*HAS PART: hierarchy, power structure, pecking order -- (the organization of people at different ranks in an administrative body)*

*Sense 3*

*organization, organisation -- (a group of people who work together)*

*HAS MEMBER: quorum -- (a gathering of the minimal number of members of an organization to conduct business)*

*HAS MEMBER: membership, rank -- (the body of members of an organization or group; "they polled their membership"; "they found dissension in their own ranks"; "he joined the ranks of the unemployed")*

According to Hearst (1992), following patterns might serve the purposes of finding whole-part relationships in texts.

NN ***'s*** NP

*Panorama's consulting services*

consulting services – *part-of* – panorama

NN ***of*** NP

According to the results of the linguistic analysis, we found that this pattern is frequently appears in the text sample. However, none of the extracted candidates reflect the part-whole relationship.

### *3.3 Metaphor and ontology*

The business language is not only rich of synonyms and homonyms, but also of metaphors. Being able to assist in solving the ambiguity issue of synonymy and homonymy by the aforementioned

techniques, let us consider metaphors as an example. The metaphors exist in our everyday life. Often, we don't even recognize them. Already in 1980 George Lakoff and Mark Johnson showed that metaphors exist in our life and are an important part of our conceptual system: „Metaphor is for most people a device of the poetic imagination and the rhetorical flourish - a matter of extraordinary rather than ordinary language...most people think they can get along perfectly well without metaphor. We have found, on the contrary, that metaphor is pervasive in everyday life, not just in language but in thought and action. Our ordinary conceptual system, in terms of which we both think and act, is fundamentally metaphorical in nature." (Lakoff & Johnson, 1992, p.3) Talking about metaphors in business domain, research on metaphors and several studies have shown that the metaphors not only exist in „business organizations“ but also important and relevant in studying knowledge systems in an organisation. In his book Images of Organization (1986), Gareth Morgan investigates the role and influence of metaphors in business organizations. He proposes that "all theories of organization and management are based on implicit images or metaphors that lead us to see, understand, and manage organizations in distinctive yet partial ways. Metaphor is often just regarded as a device for embellishing discourse, but its significance is much greater than this. The use of metaphor implies a way of thinking and a way of seeing that pervade how we understand our world generally"(Morgan, 1986, p. 4). He found out, for example, that in the business domain organizations are often discussed as if they were machines, designed to achieve specific goals and objectives. Recognizing and analysing metaphors existing in organizational life helps to understand organizations. Morgan found several aspects of viewing organizations in the light of metaphors:

- Organizations as organisms
- Organizations as brains
- Organizations as cultures
- Organizations as political systems
- Organizations as psychic prisons
- Organization as flux and transformation
- Organizations as instruments of domination

In the corporate world, metaphors are used as a tool for communication in terms of competition, policy and strategy. The source domain of such metaphors often roots in military or sporting domains: "target audiences", "advertising ammunition", "keeping the ball in play", "scoring points", etc. Thus, we argue that analysing conceptual metaphors in business language could assist in the

process of understanding domain-specific concepts, their relations to other concepts and their development. This leads to a more deep investigation in the area of knowledge representation[9]. In the context of ontological and taxonomic representation we recognise the relevance of metaphor in languages of business and consider to integrate results and insights gained from the research done and currently running within the task 6.2.

9 More information about theories/views of metaphor can be found in the Deliverable 6.3.

# 4 Conclusion

The report at hand explores some language related approaches to ontology building. Using information encoded in natural language facilitates the development of ontologies by skipping some stages in the evolutionary process of ontology building. Especially constituency and dependency grammar-models are well chosen candidates to ease the circular nature of knowledge storage in ontologies. The phrase structure of constituency grammars is important to define the scope of concepts. Dependency grammars, on the other hand, exploit the central role of verbs organizing the components of a sentence. So, this quality is important in defining the relation of concepts to each other. Both of them are needed for an ontological design. Furthermore, language also encodes information of subset/part-of and is-a relations that are not covered by dependency structures of verbs. These hyponymy/hyperonymy- and meronymy-relations are attributes within ontologies usually defined manually as formal categories. Techniques known as relationship extraction use language as a source of information to define taxonomies automatically drawn from the raw text input.

To achieve an ontological representation and before constituency and dependency information can be extracted, the text needs to be conveyed into a representation that makes it possible to apply language dependent patterns. That means, meta-information has to be added. Within the community of computational linguistics, these techniques are already well understood and we could present them in a nutshell: Tokenization, Stemming, Filtering, Tagging, and Parsing are prerequisites for an initial analysis. In addition, collocation and concordance statistics may help to reduce errors within these automated processes. Moreover, C- and TFDF-values provide some measures of evaluation that can be used to solve ambiguities. Last, the paper pointed at a challenging research issue, that is, metaphors. Metaphorical expressions might be misleading. While they cannot have the status of idioms and as such enlisted separately, metaphors cannot be marked as exceptions automatically. However, to understand them, one needs to mentally connect at least two concepts and apply exactly its relevant similarities on the given expression. This process involves world and semantic knowledge, making the process of metaphor identification (machine-based and even human-based) difficult.

We have examined existing methods (linguistic and statistical) to determine whether they can provide some additional insights into the problem of discovering the core units (concepts, properties and relationships) in ontology. Our investigations, whose first results have been presented in this

deliverable, are based on the assumption that introducing the linguistic view on ontologies should contribute to improve and facilitate the overall quality of research and practical interpolations related to knowledge representation and sharing in business environments (SMEs in Digital Ecosystem). In conclusion, we see a high potential of further research leading to establishing and strengthening the link between computer science, linguistics and their vision on ontologies. This collaboration could lead to a final stage of creating a platform for analysing the textual information and transferring the resulting information schemes to an ontology that is ready to be read by a machine.

# 5 Bibliography


1) Benson, M. (1990). Collocations and general-purpose dictionaries. *International Journal of Lexicography, 3*(1), 23—35.

2) Biber, D. (1993). Representativeness in corpus design. *Linguistic and Literary Computing, 8*(4), 243-257.

3) Borst, W.N. (1997). Construction of engineering ontologies For knowledge sharing and reuse. *Ph.D. Thesis*, University of Twente, The Netherlands.

4) Church, K. W. & Mercer, R.L. (1993). Introduction to the special issue on computational linguistics using large corpora. *Computational Linguistics, 19*(1), 1-24.

5) de Macken, C. (1996). *Unsupervised language acquisition.* Dissertationsschrift, MIT, Cambridge.

6) Euzenat, J. & Shvaiko, P. (2007). *Ontology matching*. Berlin, Heidelberg: Springer

7) Feldman, R., & Sanger, J. (2007). *The text mining handbook: advanced approaches in analysing unstructured data*. Cambridge: Cambridge University Press.

8) Frantzi, K., Ananiadou, S. & Mima, H. (2000). Automatic recognition of multi-word terms. *International Journal of Digital Libraries, 3*(2), 117–132.

9) Frege, G. (1862): *Funktion, Begriff, Bedeutung*, Göttingen.

10) Goldsmith, J. (2001). Unsupervised learning of the morphology of a natural language. *Computational Linguistics, 27* (2), 153-198.

11) Grünwald, P. (2005). *Advances in minimum description length.* Cambridge.

12) Gruber, T.R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, *5*(2), 199-220.

13) Hearst M. A.(1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the Fourteenth International Conference on Computational Linguistics.*

14) Hepp, M. (2008). Ontologies: state of the art, business potential, and grand challenges. In M. Hepp, P., De Leenheer, A., de Moor & Y. Sure (Eds.), *Ontology management: semantic web, semantic web services, and business applications* (pp. 3-24). New York, NY: Springer.

15) Jurafsky, D., & Martin, J. H. (2000). *Speech and language processing: an introduction to*

*natural language processing, computational linguistics and speech recognition*. New Jersey: Prentice Hall PTR.

16) Kennedy, G. (1999). *An introduction to corpus linguistics.* London.

17) Lakoff, G., & Johnson, M. (1992). *Metaphors we live by.* Chicago: University of Chicago Press.

18) Manning, C.D. & Schütze, H.(1999). *Foundations of statistical natural language processing, Cambridge.* MA: The MIT Press.

19) McEnery, T. & A. Wilson (1996). *Corpus linguistics.* Edinburgh.

20) McKeown, K.R. & Radev, D.R. (2000). Collocations. In R. Dale, H. Moisl & H. Somers (Eds.), *Handbook of natural language processing* (pp. 11—36). New York, NY: Marcel Dekker.

21) Milios, E., Zhang, Y., He, B., & Dong, L. (2003). Automatic term extraction and document similarity in special text corpora. *Proceedings of the 6th Conference of the Pacific Association for Computational Linguistics (PACLing'03)*, Halifax, Nova Scotia, Canada, 275-284.

22) Mitchell P. M., Santorini, B. & Marcinkiewicz M. (1993). Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics, 19*(2), 313—330.

23) Montague, R. (1970): Universal Grammar. In *Theoria*, *36*, 373-398.

24) Morgan, G. (2006). *Images of organization*. Thousand Oaks, CA: Sage Publications.

25) Nemrava, J., & Svatek, V. *Text mining tool for ontology engineering based on use of product taxonomy and web directory.* Paper presented at Workshop on Databases, Texts, Specifications and Objects, Ostrava.

26) Nirenburg, S. & Raskin, V. (2001). Ontological semantics, formal ontology, and ambiguity. In *Proceedings of the international conference on Formal Ontology in Information Systems (FOIS 2001),* Ogunquit, Maine, USA, 151 – 161.

27) Nugues, P. M. (2006). *An introduction to language processing with Perl and Prolog: An outline of theories, implementation, and application with special consideration of English, French, and German*. Berlin-Heidelberg: Springer.

28) Ogden, C.K. & I.A. Richards (1923). *The Meaning of Meaning*. New York: Harcourt, Brace

& World.

29) Palmer, D. D. (2000). Tokenisation and sentence segmentation. In R. Dale, H. Moisl & H. Somers (Eds.), *Handbook of natural language processing* (pp. 11—36). New York, NY: Marcel Dekker.

30) Partee, B. (1984): Compositionality. In F. Landman & F. Veltman (Eds.), *Varieties of the fourth Amsterdam Colloquium* (281-311). Dordrecht.

31) Porter, M.F. (1980). An algorithm for suffix stripping, *Program*, *14*(3), 130−137.

32) Sag, I.A., Baldwin, T., Bond, F., Copestake, A. & Flickinger, D. (2002). Multiword expressions: a pain in the neck for NLP. In *Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics (CICLING 2002)*, Mexico City, 1—15.

33) Somers, H. (2000). Empirical approaches to natural language processing. In R. Dale, H. Moisl & H. Somers (Eds.), *Handbook of natural language processing* (pp. 377—384). New York, NY: Marcel Dekker.

34) Sowa, J. (2000). *Ontology, metadata, and semiotics.* Paper presented at ICCS'2000, Darmstadt, Germany.

35) Stötzel, G. (1970). *Ausdrucksseite und Inhaltsseite der Sprache.* München.

36) Tesnière; L. (1959): *Èlements des syntaxe structural.* Paris.

37) Zhang, Y., Zincir-Heywood, N. & Milios, E. (2005). Narrative text classification for automatic key phrase extraction in web document corpora. In *Proceedings of the Seventh ACM International Workshop on Web Information and Data Management (WIDM'05)*, Bremen, Germany, 51- 58.

# 6 Appendix

## *6.1 Text material from SMEs*

Text material for the linguistic and statistical analysis is taken from the web sites of the following SMEs:

**Network Ltd**

http://www.networkltd.eu/about_us/about_us/history_2.html

**Redenet**

http://redenetdev.co.uk/redenet.co.uk/Default.aspx

**Joinex Oy**

http://www.joinex.com/index.php?page=home

**Panorama Partners Ltd**

http://www.panoramapartners.fi/index_eng.php

**Openscape**

http://www.openscape.co.uk/aboutus

**Excellis Consultants Network**

http://www.excellis.biz/ebchome/About.aspx

**INRAX**

http://www.meierpollard.co.uk/wa/1/17/372-Managed-Service.html

**DomainSolutions**

http://www.domsols.com/

**Izanet Global Services**

http://www.izanet.com/w3c/vista/index.php?iso639=EN

**Dialcom**

http://spontania.com/

## *6.2 Designations in Constituency and Dependency formalisms*

### 6.2.1 Brown Corpus

The following table explains some of the tags used in the Brown Corpus. The full list of tag sets can be found at http://www.scs.leeds.ac.uk/ccalas/tagsets/brown.html

| Tag | Description |
|---|---|
| **.** | sentence terminator(. ? ; ! :) |
| **:** | colon (:) |
| **ABL** | determiner/pronoun, pre-qualifier (quite, such, rather) |
| **ABN** | determiner/pronoun, pre-quantifier (all, half, many) |
| **ABX** | determiner/pronoun, double conjunction or pre-quantifier (both) |
| **AP** | determiner/pronoun, post-determiner (many, other, next, more, etc.) |
| **AP$** | determiner/pronoun, post-determiner, genitive (other's) |
| **AP+AP** | determiner/pronoun, post-determiner, hyphenated pair (many-much) |
| **AT** | article (the, an, a, every) |
| **BE** | verb "to be", infinitive or imperative be |
| **BED** | verb "to be", past tense, 2nd person singular or all persons plural were |
| **BED*** | verb "to be", past tense, 2nd person singular or all persons plural, negated weren't |
| **BEDZ** | verb "to be", past tense, 1st and 3rd person singular was |

| | |
|---|---|
| **BEDZ*** | verb "to be", past tense, 1st and 3rd person singular, negated wasn't |
| **BEG** | verb "to be", present participle or gerund being |
| **BEM** | verb "to be", present tense, 1st person singular am |
| **BEM*** | verb "to be", present tense, 1st person singular, negated ain't |
| **BEN** | verb "to be", past participle been |
| **BER** | verb "to be", present tense, 2nd person singular or all persons plural are art |
| **BER*** | verb "to be", present tense, 2nd person singular or all persons plural, negated aren't ain't |
| **BEZ** | verb "to be", present tense, 3rd person singular is |
| **BEZ*** | verb "to be", present tense, 3rd person singular, negated isn't ain't |
| **CC** | conjunction, coordinating (and, or, but, plus, &, either, neither, nor, etc.) |
| **CD** | numeral, cardinal (two one 1 four 2 1913 71 74 637 1937 8 five three million 87-31 29-5 seven 1,119 fifty-three 7.5 billion hundred 125,000 1,700 60 100 six ..). |
| **CD$** | numeral, cardinal, genitive (1960's, 1961's, 404's) |
| **CS** | conjunction, subordinating (that, as, after, whether, before, while, like, because, if, since, for, than, although, until, etc.) |
| **DO** | verb "to do", uninflected present tense, infinitive or imperative do dost |
| **DO*** | verb "to do", uninflected present tense or |

| | |
|---|---|
| | imperative, negated don't |
| **DO+PPSS** | verb "to do", past or present tense + pronoun, personal, nominative, not 3rd person singular d'you |
| **DOD** | verb "to do", past tense did done |
| **DOD*** | verb "to do", past tense, negated didn't |
| **DOZ** | verb "to do", present tense, 3rd person singular does |
| **DOZ*** | verb "to do", present tense, 3rd person singular, negated doesn't don't |
| **DT** | determiner/pronoun, singular (this, each, another, that, 'nother) |
| **DT$** | determiner/pronoun, singular, genitive (another's) |
| **DT+BEZ** | determiner/pronoun + verb "to be", present tense, 3rd person singular (that's) |
| **DT+MD** | determiner/pronoun + modal auxillary (that'll, this'll) |
| **DTI** | determiner/pronoun, singular or plural (any, some) |
| **DTS** | determiner/pronoun, plural (these, those, them) |
| **EX** | existential there (there) |
| **EX+BEZ** | existential there + verb "to be", present tense, 3rd person singular there's |
| **EX+HVD** | existential there + verb "to have", past tense there'd |
| **EX+HVZ** | existential there + verb "to have", present tense, 3rd person singular there's |
| **EX+MD** | existential there + modal auxillary there'll there'd |

| | |
|---|---|
| **HV** | verb "to have", uninflected present tense, infinitive or imperative (have) |
| **HV*** | verb "to have", uninflected present tense or imperative, negated (haven't) |
| **HV+TO** | verb "to have", uninflected present tense + infinitival to |
| **HVD** | verb "to have", past tense (had) |
| **HVD*** | verb "to have", past tense, negated (hadn't) |
| **HVG** | verb "to have", present participle or gerund (having) |
| **HVN** | verb "to have", past participle (had) |
| **HVZ** | verb "to have", present tense, 3rd person singular (has) |
| **HVZ*** | verb "to have", present tense, 3rd person singular, negated (hasn't, ain't) |
| **IN** | preposition (of, in, for, by, to, on, during, between, without, except, upon, out, ...) |
| **JJ** | adjective (recent, ambiguous, effective, ...) |
| **JJ+JJ** | adjective, hyphenated pair (big-large, long-far) |
| **JJR** | adjective, comparative (greater, older, further, earlier, later, etc.) |
| **JJS** | adjective, semantically superlative (top, chief, principal, northernmost, main, etc.) |
| **JJT** | adjective, superlative (best, largest, coolest, newest, worst, fastest, lowest, etc.) |
| **MD** | modal auxillary (should, may, might, will, would, must, can, could, shall) |
| **MD*** | modal auxillary, negated (cannot, couldn't, wouldn't, can't, won't, shouldn't) |

| | |
|---|---|
| **NN** | noun, singular, common (failure, court, fire) |
| **NN+NN** | noun, singular, common, hyphenated pair |
| **NNS** | noun, plural, common (irregularities, reports, years) |
| **NNS$** | noun, plural, common, genitive (taxpayers', children's, members') |
| **NP** | noun, singular, proper |
| **NP$** | noun, singular, proper, genitive (Smith's) |
| **NP+BEZ** | noun, singular, proper + verb "to be", present tense, 3rd person singular (Blackwell's) |
| **NP+MD** | noun, singular, proper + modal auxillary (John'll) |
| **NP+HVZ** | noun, singular, proper + verb "to have", present tense, 3rd person singular (Bill's, Tim's) |
| **NPS** | noun, plural, proper (Chases, Catholics, Congresses ...) |
| **OD** | numeral, ordinal (first, 13th, third, nineteenth, 2nd, 61st) |
| **PN** | pronoun, nominal (none, something, everything) |
| **PPL** | pronoun, singular, reflexive (itself, himself, myself, yourself, herself) |
| **PPLS** | pronoun, plural, reflexive (themselves, ourselves, yourselves) |
| **PPS** | pronoun, personal, nominative, 3rd person singular (it, he, she) |
| **PPS+BEZ** | pronoun, personal, nominative, 3rd person singular + verb "to be", present tense, 3rd person singular (it's, he's, she's) |
| **PPS+MD** | pronoun, personal, nominative, 3rd person |

| | |
|---|---|
| | singular + modal auxillary (he'll, she'll) |
| **PPSS** | pronoun, personal, nominative, not 3rd person singular (they, we, I, you) |
| **PPSS+BEM** | pronoun, personal, nominative, not 3rd person singular + verb "to be", present tense, 1st person singular (I'm) |
| **PPSS+BER** | pronoun, personal, nominative, not 3rd person singular + verb "to be", present tense, 2nd person singular or all persons plural (we're, you're, they're) |
| **RB** | adverb (only, often, generally) |
| **RBR** | adverb, comparative (further, earlier, better) |
| **RBT** | adverb, superlative (most, best, nearest) |
| **VB** | verb, base: uninflected present, imperative or infinitive (investigate, find, act) |
| **VBD** | verb, past tense (said, produced, took) |
| **VBG** | verb, present participle or gerund (modernizing, improving, purchasing) |
| **VBN** | verb, past participle (conducted, charged, won) |
| **VBZ** | verb, present tense, 3rd person singular (deserves, believes, receives, takes, goes) |
| **WDT** | WH-determiner (which, what) |
| **WPS** | WH-pronoun, nominative (that, who, whoever) |
| **WRB** | WH-adverb (however, when, where, why, how...) |
| **WRB+BER** | WH-adverb + verb "to be", present, 2nd person singular or all persons plural (where're) |
| **WRB+BEZ** | WH-adverb + verb "to be", present, 3rd person singular (how's, where's) |

### 6.2.2 Penn Treebank

The table below contains some of the tag descriptions from Penn Treebank (Mitchell et al., 1993)

| Tag | Description |
|---|---|
| CC | Coordinating conjunction |
| CD | Cardinal number |
| DT | Determiner |
| EX | Existential there |
| FW | Foreign word |
| IN | Preposition or subordinating conjunction |
| JJ | Adjective |
| JJR | Adjective, comparative |
| JJS | Adjective, superlative |
| MD | Modal |
| NN | Noun, singular or mass |
| NNS | Noun, plural |
| NP | Proper noun, singular |
| NPS | Proper noun, plural |
| PDT | Predeterminer |
| POS | Possessive ending |
| PP | Personal pronoun |
| PP$ | Possessive pronoun |
| RB | Adverb |
| RBR | Adverb, comparative |
| RBS | Adverb, superlative |
| TO | to |

| | |
|---|---|
| VB | Verb, base form |
| VBD | Verb, past tense |
| VBG | Verb, gerund or present participle |
| VBN | Verb, past participle |
| VBP | Verb, non-3rd person singular present |
| VBZ | Verb, 3rd person singular present |
| WDT | Wh-determiner |
| WP | Wh-pronoun |
| WP$ | Possessive wh-pronoun |

### 6.2.3 Dependency Grammar

The following table explains types of relations used in Dependency formalisms (Jurafsky & Martin, 2000).

| Dependency type | Description |
|---|---|
| subj | Syntactic subject |
| obj | Direct object |
| dat | Indirect object |
| pcom | Complement of a preposition |
| attr | Pre-modifying (attributive) nominals (e.g. genitive) |
| mod | Nominal post-modifiers (e.g. Prepositional phrases) |
| loc | Location adverbials |
| tmp | Temporal adverbials |