Two-level semantic analysis of compounds:
A case study in linguistic engineering[*]

Wilco G. ter Stal & Paul E. van der Vet

Knowledge-Based Systems Group
Department of Computer Science, University of Twente
Enschede, The Netherlands
Email: {terstal, vet}@cs.utwente.nl

Abstract

We describe an experiment of a two-level approach for the automated semantic analysis of N-N compounds. The first stage of the interpretation process consists of the translation of compounds into a representational format called Quasi Logical Form (QLF). The second stage consist of a mapping of QLFs onto domain-dependent, conceptual representations. Specifically, in the context of the Plinius project these QLFs are mapped onto relevant, formal expressions in terms of the so-called Plinius ontology KB. We briefly describe the linguistic analysis and then focus on a number of cases of deriving conceptual descriptions from QLFs. Our ultimate goal is to apply the method to a large (> 2000) number of compounds within a specific domain.

1 Introduction

Problems with processing N-N compounds have been described earlier in the literature, for instance by (Wachter and Provoost, 1993; Bouillon *et al.*, 1992; Isabelle, 1984). In languages such as Dutch and German, syntactic processing of compounds, that is *identification, segmentation and disambiguation*, is already a difficult enterprise. However, the most difficult issue related to compound processing concerns the interpretation task. In general, this task amounts roughly to the derivation of an expression in a suitable meaning representation language, based on the components of the compounds.

We have been studying English compounds in the context of the Plinius project, see § 3, which uses a large text corpus. From this corpus, we (manually) identified about 2200 compounds. In this paper we will focus on the particular module of the NLP engine used within Plinius that is responsible for the automated semantic analysis of these compounds.

[*] The authors are indebted to Franciska de Jong for her valuable comments on earlier versions of this paper.

First, we will briefly summarize the general, linguistic perspective on the semantic analysis of compounds. The purpose of this section is to see whether we can employ certain linguistically motivated distinctions within the interpretation task. In section 3 we will describe briefly the Plinius project in which our research is carried out. Specifically, we will discuss the linguistic processes within Plinius. In section 4 we will apply the two-level semantic analysis to Plinius compounds. In order to clarify the approach we will present a number of different cases. Section 5 contains a critical discussion of the introduced approach. Conclusions and suggestions for further research can be found in section 6.

2 Compound analysis: the linguistic perspective

A frequently recurring analysis concerning the semantics of compounds uses the notion of an implicit semantic relation between the components of a compound (Downing, 1977; Isabelle, 1984; Finin, 1986). For instance, compound (1) can be paraphrased as (2) and represented as a predicate-argument structure as in (3).

(1) *GM car*

(2) Car made by GM

(3) made_by(GM, car)

Traditionally, linguists have been trying to identify and to classify these semantic relations. We may distinguish the *descriptive* approaches, for instance Warren (1978), and studies within the *generative* framework such as Levi (1978); Selkirk (1982); Grimshaw (1991). The main objective of the generative approaches is to relate compounds systematically with predicate-argument structures. For instance, Levi (1978) proposes a number of so-called *Recoverably Deletable Predicates* (RDPs) which may be used to characterize the implicit semantic relation within compounds. In table I, we will give some examples of these RDPs.

RDP	Examples
CAUSE	cigarette smoke, drug death
HAVE	picture book, lemon peel
MAKE	daisy chain, coffee machine
USE	steam iron, water pipe
IN	house dog, kitchen table

Table I: Recoverably Deletable Predicates as semantic relations within compounds

A major problem of Levi's theory is the absence of a procedure to determine what RDP (or for that matter *semantic relation*) to select given a particular compound. For instance, *cigarette smoke* may be paraphrased as (4) or (5).

(4) *smoke that is* **caused** *by a cigarette*

(5) *smoke that is* **made** *by a cigarette*

Additionally, some predicates are semantically ambiguous. For instance, the IN-RDP may be used to specify a locative relation as in *kitchen table* as well as a temporal location as in *summer breeze*. Even if we introduce more fine-grained predicates, such as loc-IN and temp-IN, the problem remains that there is no syntactic clue, i.e. grammatical information, in the compound itself that suggests the selection of the proper RDP. Therefore, we claim that compounds bear an implicit semantic relation, the nature of which **cannot** be determined on the basis of grammatical knowledge alone. In fact, it seems that the interpretation of a compound is largely dependent on extra-linguistic or domain dependent knowledge.

Grimshaw (1991); Selkirk (1982); Isabelle (1984) appear to identify some exceptions. They distinguish a class of nouns which subcategorizes for other nouns. An important subclass of these nouns are *deverbal nominalizations*, that is nouns derived from a verb. In case a nominalization forms the head of a compound, the modifying noun can be semantically interpreted as an argument. In table II we have given some examples of nominalizations, the nominalization as head noun in a compound and a predicate-argument structure for such a compound.

Deverbal nominalization	Compound	Predicate-argument structure
giving	gift giving	give(_, gift, _)
observing	animal observing	observe(_, animal)
mixing	powder mixing	mix(_, powder)

Table II: Deverbal nominalizations in compounds

Compounds containing a deverbal head with the affix *-ing*, so with an internal structure as in $[[\mathrm{N}]_N\ [\mathrm{V\text{-}ing}]_N\ ]_N$, are referred to as *synthetic* compounds.[1] In table II we show that synthetic compounds trigger a different type of predicate-argument structure compared to standard, i.e. containing no nominalization, compounds, as (3). A consequence of this grammatical observation is that we should capture the difference between the two types at the linguistic level.[2]

In the following section we will briefly describe NLP research in Plinius, in order to sketch the context of our compound approach.

3 Processing compounds in Plinius

The Plinius project aims at developing a system which is capable of semi-automatically extracting domain-specific knowledge from the title and abstract of scientific publications in the field of ceramic materials. The knowledge base resulting from the Plinius

[1] The term *synthetic compound* is due to Grimshaw (1991). Note that there are a lot of other compounds containing a head derived from a verb. Examples are *truck driver*, *engine repair*, *oil pump* etc. However, if we consider these compounds, the underlying predicate-argument seems less straightforward, for instance ?DRIVE(driver, truck). In order not to complicate matters we limit the discussion to synthetic compounds.

[2] There are a number of other linguistic motivations for separating synthetic compounds from standard compounds. For instance, synthetic compounds do not pluralize. For more details, see Grimshaw (1991).

project should have economical potential, that is, the benefits of using the knowledge base should outweigh the costs of developing it. In order to obtain this ultimate goal we have made a number of design decisions among which are: (1) use of abstracts, (2) limitation to sublanguage, operationally defined by a corpus and (3) application of an ontology. For a detailed review of these design decisions we refer to Mars *et al.* (1993).

The automated semantic analysis of compounds contributes to the overall goal, since our input consists of abstracts containing information in a highly condensed format, which results in a frequent occurrence of compounds. Moreover, since the abstracts describe the results of innovative research, new compounds are used to denote (new) concepts.

In the following subsections we will first clarify the function of the ontology within Plinius. Subsequently, we will briefly explain the general NLP approach we are following. In particular we will focus on two-level semantic analysis, as proposed by van der Sloot and Rentier (1993).

3.1 The Plinius ontology as specification of semantics

In Plinius, a central role is played by a structured concept system (or *ontology* in current AI terminology). The Plinius ontology (van der Vet and Mars, 1993; van der Vet and Mars, 1991; Mars, 1993) is in first approximation a limitative list of concepts and relations between them. It currently contains concepts for materials, their chemical composition, processes to make materials, and properties of materials. The output of the Plinius process is to be expressed in terms of ontology concepts and relations only. In this sense the Plinius ontology indirectly specifies the semantics[3] of the texts that are processed. Although the ontology provides a framework for lexical semantic knowledge, it can not be compared directly with the model of a *generative lexicon* described in Pustejovsky (1991). In particular, Pustejovsky distinguishes four basic levels of semantic description whereas we only employ one.

In a more detailed account, the Plinius ontology is not a flat list of concepts but a structured system. It consists of *atomic* (primitive) concepts, distributed over several sets for clarity, and rules for making *complex* concepts. Any complex concept therefore is a particular combination of atomic concepts and thus coincides with its definition. We refer to this way of organising an ontology as the principle of the conceptual construction kit.

The principle of the conceptual construction kit can be illustrated by the concept for a particular chemical, the pure substance aluminium oxide (also known as alumina, chemical formula Al_2O_3). Concepts for chemicals are defined as sets of tuples. Each tuple has two arguments: a material ingredient and a number giving the proportion. The atomic concepts needed for constructing this concept are chemical elements and natural numbers. For disambiguation, concepts for pure substances do not consist of elements with their proportions but an intermediate level of concepts called groups is defined. For aluminium oxide, there are two groups and each group is a set of one

[3] Velardi (1991) uses the notion *technical semantics* for a similar approach where word senses are defined in terms of technical knowlegde concerning a domain.

tuple: $g_1 = \{\langle \mathrm{Al}, 1\rangle\}$ and $g_2 = \{\langle \mathrm{O}, 1\rangle\}$. The concept for aluminium oxide then is the set $\{\langle g_1, 2\rangle, \langle g_2, 3\rangle\}$.

In the account below, the level of detail achieved in the ontology is not needed and we will often use abbreviations. For instance, the concept for aluminium oxide just elaborated will be abbreviated as *alumina*. Further illustrations are provided below.

The particular language chosen to express these concepts is unimportant as long as the expressions are unambiguous. In this paper, we will write complex concepts as feature structures and relations as simple predicate-argument structures for clarity. Examples are given in § 4.3.

3.2 Grammar engineering in Plinius

The overall goal of the language-driven process is to convert natural language constructs into elements suitable for storage in a knowledge base. In order to attain this goal, we are developing and implementing an NLP system which currently consists of the following components:

Preprocess The task of the preprocess, as described in van Raalte *et al.* (1992), is to segment the abstracts of the Plinius corpus in such a way that the subsequent processes are not hindered by, for example, ambiguous end-of-sentence markers, record information, case conventions, unknown strings of characters representing chemical formulae, or other formulae.

Sublanguage Grammar Currently, our grammar consists of about 80 rules describing linguistic constructions specific for our texts. The formalism we employ is PATR (Shieber, 1986; Gazdar and Mellish, 1989). Thus, grammar rules are context-free phrase structure rules annotated with features. More details concerning the coverage and the organisation of the grammar can be found in Stefanova and ter Stal (1993); van der Vet *et al.* (1993).

In order to develop, test and debug the Plinius grammar rapidly, we developed a tool with a user-friendly, graphical interface, see Hofman and ter Stal (1994).

Head Corner chart parser The parser which operates on the Plinius grammar is in fact an extension of the Head-Corner (HC) parser described in Sikkel and op den Akker (1993). The main difference stems from the fact that the current parser is allowed to use a context-free grammar annotated with feature structures, *i.e.*, a PATR grammar. More details concerning the HC parser can be found in Verlinden (1993).

The core of the language-dependent process is formed by the sublanguage grammar. In combination with the parser, sentences are transformed into (I) a conventional parse tree representing the syntactic structure of the sentence and (II) a feature structure representing both (detailed) syntactic and semantic information. For instance, the feature structure for sentence (6) amounts to (7).[4]

(6) The material exhibits elongation

[4] Due to space limitations we leave out the complete value of the *object* feature.

$$
(7)\quad \left[\begin{array}{ll}
head & : \left[\begin{array}{ll} agr & : \left[\begin{array}{ll} num & : singular \\ per & : 3 \end{array}\right] \\ vform & : fin \\ tense & : present \end{array}\right] \\
args & : \left[\begin{array}{ll} subject & : \left[\begin{array}{ll} head & : \left[agr : \left[num : singular \right] \right] \\ content & : \left[\begin{array}{ll} para & : x1 \\ det & : the \\ relation & : material \\ arg0 & : x1 \end{array}\right] \end{array}\right] \\ object & : \ldots \end{array}\right] \\
mod & : null \\
content & : \left[\begin{array}{ll} para & : e1 \\ det & : E \\ relation & : exhibit \\ arg0 & : e1 \\ arg1 & : x1 \\ arg2 & : x2 \end{array}\right]
\end{array}\right]
$$

The language-dependent process is described in more detail in van der Vet *et al.* (1993).

3.3 Quasi Logical Form in Plinius

The values of the *content* features in (7) are used to construct a so-called QLF. The idea of QLFs as described in van Eijck and Alshawi (1992); van der Sloot and Rentier (1993); van der Vet *et al.* (1993) is that they form a suitable data structure for storing grammatical information relevant to further semantic and discourse processing. This means that in the Plinius context QLFs form the input for an additional process which maps expressions in QLF onto the final representation in terms of the Plinius ontology.

We are employing the notation for QLF as described in van der Sloot and Rentier (1993). The general format of a QLF expression amounts to: `det(parm, [restr, ...]`, where `det` functions as a kind of quantifier binding the parameter `parm` which is restricted by predicates in `restr`. The QLF for sentence (6) amounts to:

```
(8)  E(e1, [exhibit(e1, the(x1, [material(x1)]), zerodet(x2,
     [elongation(x2)]))])
```

Note that the 'quantifiers' of the NPs correspond to the normal linguistic determiners, like *the, a, all, most, every* etc. The quantifier `E` in (8) is an existential quantifier of **events** (kind of actions). In van der Sloot and Rentier (1993) it is explained why they employ so called event-semantics. Here it suffices to note that the event style analysis[5]is to be prefered over the standard first-order fashion, simply because it contains more information.

[5] Explained from a general, linguistic point of view in Parsons (1991).

Another feature, noted by Rich *et al.* (1987), of QLFs is that they enumerate the entities (referents)[6] referred to by the sentence as well as the surface functional relationships among those entities. Note that the QLF in (8) in fact is a linear notation of the collected *content* features of (7). For a further elaboration on the format and the theoretical background of QLF the reader is referred to van der Sloot and Rentier (1993).

In the following section, we will describe how compounds may be represented in QLF and how they relate to conceptual descriptions.

4 Two-level semantic analysis of compounds

In this section, we will explain our method for the two-level semantic analysis of compounds. As explained in § 3.3 the first phase of semantic analysis in Plinius consists of deriving a QLF from natural language input. It is important to note that the lexicon and grammar rules provide the necessary information to construct a QLF.

A QLF constitutes an intermediate, underspecified representation of a NL construct. This means that certain lexical semantic and conceptual aspects are only made explicit during the second phase. The result of this second phase is a representational structure in terms of the Plinius ontology, see § 3.1, which we will call Ontology Knowledge Base Representation (henceforth: OKBR).

First, we will present the QLF format for compounds. Subsequently, we will present a brief conceptual analysis of compounds. Finally, we will sketch the formalization of translating QLFs to OKBRs.

4.1 Representing compounds in QLF

In § 2 we showed that the class of synthetic compounds should be grammatically distinguished from the class of standard compounds. In addition, we explained that QLFs should capture relevant grammatical information. Based on these two observations we propose the following formats for compounds in QLF.
Standard compounds, such as *diesel engine*, are represented as follows:[7]

(9) diesel engine ⇒ `zerodet(x1, [engine(x1), REL(x1, zerodet(x2,`
`[diesel(x2)]))])`

The QLF in (9) contains the maximum grammatical information for a standard compound. It enumerates the entities involved in the compound and an underspecified semantic relation between them, viz. `REL(x, y)`.

If we follow the proposal of Grimshaw (1991) to treat synthetic compound differently, then we have to translate them into different QLFs. For instance, *animal observing* can be represented as:

[6] In case of example (8) these entities are indicated by *e1, x2, x3*.

[7] The QLF examples in fact represent compounds functioning as NPs with a ∅-determiner. However, (9) is a **count** compound noun for which such an analysis is unlikely. Despite this deficiency, we will accept the ∅ determiner reading of (9) since a proper treatment of the issue is beyond the scope of this paper.

(10) animal observing ⇒ `zerodet(e1, [observe(e1, Var, zerodet(x1, [animal(x1)]))])`

QLF (10) can be explained as follows. The noun *observing* triggers a ternary predicate `observe` in which the QLF for *animal* is going fill the *object*-argument. The *subject*-argument remains empty, notated with `Var`.

A negative consequence of distinguishing standard ad synthetic compounds is that both lexicon and grammar become more complex. First, we cannot specify nouns in a uniform way, because *event*-like nouns such as *forming, pressing, observing* etc., will receive a different *content* feature as compared to standard nouns. Moreover, we will have to define (at least) two compound rules in the grammar. In the next section we will try to relate QLFs for compounds to OKBRs.

4.2 OKBRs for compounds: conceptual preliminaries

Until now we implictly assumed that the semantic representation of a compound can be derived in a compositional fashion. However, linguists recognize a large class of compounds with a meaning not directly related to the meaning of its parts. For instance, the compound in (11) is not any 'bird that is black' but rather a particular kind of bird also known as *Turdus merula.*[8]

(11) *blackbird*

Following standard terminology, for instance Isabelle (1984); Jones (1982), we will refer to compounds of the former class as *productive* compounds and to (non-compositional) compounds as *lexicalized compounds.* Note that from an implementational point of view the distinction yields a difference in processing of the compound. A lexicalized compound will be treated as a single unit with a simple QLF as for instance in (12):

(12) *blackbird* ⇒ `zerodet(x1, [blackbird(x1)])`

Productive compounds will have to be analysed by means of compound rules in the grammar yielding a QLF as in (9) or (10).

In our system, too, a number of compounds are lexicalized. From an engineering point of view, there is no absolute contrast between lexicalized and productive compounds. For the analysis of the lexicalized compound (11) we can still imagine a system able to infer the meaning of (11) from the meanings of *black* and *bird.* Such a system will have to be equipped with a lot of background knowledge to have the inferences follow the meaning shifts that have occurred. The majority of this knowledge applies to this particular compound only. If (as will often be the case) this knowledge plays no role in the rest of the system, it is very impractical to store it. What this example makes clear is that for each compound there is a trade-off and we have to choose to lexicalize it or treat it as being productive. For certain compounds it just does not pay to infer their meaning, and therefore we lexicalize them.

[8] In Dutch: merel.

4.2.1 Why certain compounds are lexicalized in Plinius

The considerations relevant for deciding whether a particular compound is lexicalized or treated as productive can become quite complicated. They can be illustrated by means of examples from the Plinius corpus. We have chosen to lexicalize the following two compounds.

(13) *room temperature*

(14) *aluminium oxide*

In natural science, (13) means preci sely 25 degrees Centigrade. A process able to infer this meaning would have to make deductions involving a concept for room, its more specific interpretation of room in a laboratory, and the subsequent standardisation that has led to the precise meaning given above. All these concepts play no role whatsoever in the rest of the system. That is a high price to pay for the capacity to infer the meaning of (13) from the meanings of *room* and *temperature*. Thus, (13) is lexicalized.

We discuss the second example, (14), in more detail. Here, the ontology does contain the concepts needed to infer the meaning of (14) from the meanings of *aluminium* and *oxide*. The lexicon would have to include an interpretation in terms of groups of *aluminium* and *oxide*. One of the readings of *aluminium* translates it as a group that consists of one aluminium atom (that is g_1 of § 3.1). *Oxide* is translated as a group that consists of one oxygen atom (g_2 of § 3.1), but in this case (*oxide* rather than *oxygen*) it is obvious that we are dealing with an ingredient of a pure substance. Given this information from the lexicon, an inference process has to construct the meaning of (14) by combining the two groups into a concept for the pure substance aluminium oxide. To do that, the process has to calculate the proportions of the two groups as they occur in aluminium oxide. Extra information, either in the lexicon or in the background knowledge base, is needed: the valencies of the groups and valency rules. The outcome is that aluminium groups and oxygen groups constitute aluminium oxide in the proportion 2:3. (The concept for aluminium oxide is also given in § 3.1).

As was the case for (13), a lot of knowledge has to be added to make the inference possible. But in contrast to the former case, the knowledge needed to infer the meaning of (14) applies to many other pure substances and thus is more general. The problem this time is that we cannot make use of this generality because there is no equally general decision procedure to distinguish the regular cases from the many exceptions. We can still mark the case of aluminium oxide as being regular, either in the lexicon itself or in the background knowledge base. But among the exceptions in the domain of ceramics are pure substances involving aluminium, pure substances involving oxygen, and even pure substances involving both aluminium and oxygen besides other elements. To store a mark in the lexicon, (14) has to be an entry. But if we do that, it is more pragmatic to lexicalize (14).

4.2.2 Productive compounds

Due to the contents of our corpus, see § 3, new compounds are very likely to occur. Therefore, it is inevitable to develop a procedure which handles compounds in a compositional fashion. We now turn to three examples of compounds treated as productive compounds in our system:

(15) *compression stress*

(16) *alumina ball*

(17) *glass forming*

Example (15) will serve to explain the two-level approach in general terms. Examples (16) and (17) are instructive because they would have to be treated differently if we distinguish between productive and synthetic compounds. This distinction may be useful for systems poor in domain knowledge. Our approach, however, bears out that the distinction fulfils no purpose if extensive use is made of adequately represented domain knowledge.

4.3 Translating QLF to OKBR: the algorithm

In this section we will sketch the translation procedure from QLF to OKB for the compounds (15), (16) and (17). The corresponding QLFs are given in (18) and (19) and (20).

(18) `zerodet(x1, [compression(x1), REL(x1, zerodet(x2,`
`[stress(x2)]))])`

(19) `zerodet(x1, [alumina(x1), REL(x1, zerodet(x2, [ball(x2)]))])`

(20) `zerodet(e1, [form(e1, Var, zerodet(x1, [glass(x1)]))])`

A simple translation algorithm can be formulated as follows:

1. isolate the (non-variable) predicates from the QLF
2. find through a look-up in the QLF-predicate/Concept lexicon a (or more) conceptual description(s) for the predicates.
3. try to unify the concepts found in step 2.

The OKBR for compound (15) can be explained as follows. Stress is a quantity that measures the force applied to a sample. The related concept of strain measures the deformation undergone by the sample as a result of stress. The ontology conceptualises the relation between stress and strain as a property involving tensor quantities. Here, it it sufficient to note that *stress* can be written as follows:[9]

(21) $$\texttt{stress(x)} \Rightarrow \begin{bmatrix} quantity\_name & : stress \\ direction & : Var1 \\ magnitude & : \begin{bmatrix} value\!: \dots \\ unit\ : \dots \end{bmatrix} \\ time\_dependence\!: \dots & \end{bmatrix}$$

[9] For ease of explanation we are using a simple feature structure notation for OKBRs. However, in Speel *et al.* (1993) we present a small part of the actual formalisation of the Plinius ontology in CLASSIC.

Stress can be either static (constant in time) or dynamic; in the latter case, it can be applied at regular intervals with a specified frequency or at irregular intervals. This is expressed in the the time-dependence feature. It can have either the value *static* or have a feature structure specifying the relevant parameters of a dynamically applied stress.

In the lexicon, all values are empty. They are filled as other sentence or compound constituents are processed. In the present case, the word *compression* is interpreted in the lexicon as supplying a value for the *direction* feature, namely *inward.* Thus,

$$(22)\quad \texttt{compression(x)} \Rightarrow \begin{bmatrix} \textit{direction}\colon\ \textit{inward} \end{bmatrix}$$

It is easy to imagine how the resulting concept for (15) will be equal to (21) except that the feature *direction* will receive the value *inward.*
The QLF-concept translations for *ball* and *alumina* are respectively:

$$(23)\quad \texttt{ball(x)} \Rightarrow$$

$$\begin{bmatrix} \textit{sample}\colon & \begin{bmatrix} \textit{sample\_id}\colon\ Var1 & \\ \textit{material}\ : & \begin{bmatrix} \textit{chem\_composition} : Var2 \\ \textit{aggregation\_state} : \textit{solid} \end{bmatrix} \end{bmatrix} \end{bmatrix} \wedge \quad \textit{shape}(Var1, \textit{ball})$$

$$(24)\quad \texttt{alumina(x)} \Rightarrow \begin{bmatrix} \textit{material}\colon & \begin{bmatrix} \textit{chem\_composition}\colon\ \textit{alumina} \\ \textit{aggregation\_state} : Var \end{bmatrix} \end{bmatrix}$$

For Plinius, it has been decided to express the output conceptually as statements about samples and their properties. A sample is a particular and usually unique object that consists of a particular material. Any sample is identified by a unique label attached in the course of the language-dependent process. Often, we know more about a particular sample than just its material composition. For instance, a *ball* is a sample with a particular shape. This is written as an assertion involving a two-place predicate `shape`, with the sample identifier as its first argument and a characterisation of the shape as its second argument. Unifying the concepts in (23) and (24) yields:

$$(25)\quad \begin{bmatrix} \textit{sample}\colon & \begin{bmatrix} \textit{sample\_id}\colon\ 245.1 & \\ \textit{material}\ : & \begin{bmatrix} \textit{chem\_composition}\colon\ \textit{alumina} \\ \textit{aggregation\_state} : \textit{solid} \end{bmatrix} \end{bmatrix} \end{bmatrix} \wedge\ \textit{shape}(245.1, \textit{ball})$$

The concept in (25) represents a specific sample. The sample is labeled with a identifier (245.1). The material used in the sample is alumina with a solid aggregation state. The form of the sample is ball. In the next case will give the OKBR for QLF (17). The concepts for *glass* and *forming* are:

$$(26)\quad \texttt{form(e, x, y)} \Rightarrow$$

$$\begin{bmatrix} \textit{sample\_id}\colon\ Var1 & \\ \textit{material}\ : & \begin{bmatrix} \textit{chem\_composition}\colon \ldots \\ \textit{aggregation\_state} : \ldots \end{bmatrix} \end{bmatrix} \wedge \quad \text{process(Var1, Var2)}$$

$$\begin{bmatrix} sample\_id\colon Var2 & \\ material\ : & \begin{bmatrix} chem\_composition\colon \ldots \\ aggregation\_state : \ldots \end{bmatrix} \end{bmatrix}$$

$$(27)\quad \texttt{glass(x)} \Rightarrow \begin{bmatrix} sample\_id\colon Var & \\ material\ : & \begin{bmatrix} chem\_composition\colon \ldots \\ aggregation\_state : vitreous \end{bmatrix} \end{bmatrix}$$

Processes, used in (26), are conceptualised as being relations between two samples: the sample that constituted the starting point of the process, and the sample that is its product. A process can thus be written as a two-place predicate, with the starting sample and product sample as first and second arguments, respectively.
The result of integrating the two concepts amounts to:

$$(28)\quad \begin{bmatrix} sample\_id\colon 234.1 & \\ material\ : & \begin{bmatrix} chem\_composition\colon \ldots \\ aggregation\_state : \ldots \end{bmatrix} \end{bmatrix} \wedge \quad \mathrm{process}(234.1,\ 234.2)$$

$$\begin{bmatrix} sample\_id\colon 234.2 & \\ material\ : & \begin{bmatrix} chem\_composition\colon \ldots \\ aggregation\_state : vitreous \end{bmatrix} \end{bmatrix}$$

5 Discussion

The first issue we would like to address is whether it is essential to maintain the grammatical distinction between standard and synthetic compounds. In § 4.1 we proposed two different formats for compounds. In § 4.3 we discussed the translation from QLF to OKBR. In (28) we gave the conceptual, via QLF, translation for the compound *glass forming*. The examples illustrate that the actual semantic representation of a compound is generated at the second level. Therefore we would like to claim that the actual format of the QLF for the components is irrelevant **as long as they trigger the appropriate conceptual translation**. In Wachter and Provoost (1993) a brief discussion of compounds with a deverbal head renders a somewhat weaker, but compatible conclusion.

> ... predictions (*of the meaning depending on syntactic clues provided by the compound* (WtS/PV)) are possible to some extent, but very often they exhibit a tentative character. (Wachter and Provoost (1993), pp.19)

Therefore, we associate the word *forming* with `form(x)` and analyse all compounds as standard compounds yielding a QLF as in (9). A positive consequence, from an engineering point of view, is that all nouns are treated uniformly in the lexicon and compounds are captured with a single compound rule.

A problem which we did not discuss so far concerns the possibility of a word or QLF having more than one conceptual translation. For instance, if the word *compression* receives two OKBRs, the algorithm sketched in § 4.3 will become a little more complicated, since the two cases should be tested. This operation may prove to be very costly, so it may be beneficial to incorporate contextual knowledge, based on the previous processed sentences, to disambiguate between the two cases.

A third issue concerns the status of the ontology. In § 2 we argued that the decision whether we should treat a compound as lexicalized versus productive in fact depends on the granularity and scope of the ontology. However, if the amount of lexicalized compounds tends to become too high, an extension of the ontology should be considered. Such an extension, in terms of more fine-grained conceptual distinctions, would provide the means to analyse more Plinius compounds in a compositional fashion. At this moment it remains unclear how one should find an economical balance contents of the ontology and its application for semantic analysis.

6 Conclusions and further work

We discussed a two-level semantic analysis applied to compounds. We demonstrated that actual meaning representations of Plinius compounds are determined by the Plinius ontology. We also showed that in order to arrive at the final representation it is not necessary to capture grammatical information at the intermediate (QLF) level. Moreover, we argued that the distinction between lexicalized and productive compounds is a gradual one. The decision to classify compounds as belonging to the former or latter group is, in our view, motivated by pragmatic (viz. engineering) principles only.

Further work includes investigation of a larger sample of compounds from our corpus. This sample will also contain compounds consisting of three or more elements. Moreover, we will have to implement the second phase of our approach. This means that we have to specify a QLF to OKBR lexicon. Additionally, we will have to formalize and implement the inference and unification operations required to construct concepts from subconcepts.

References

Pierrette Bouillon, Katharina Boesefeldt, and Graham Russell. Compound nouns in a unification-based mt system. In *Proc. of the Third Conference on Applied Natural Language Processing*, pages 209–215, Trento, 1992.

Pamela Downing. On the creation and use of english compound nouns. *Language*, 53(4):810–842, 1977.

Timothy W. Finin. Constraining the interpretation of nominal compounds in a limited context. In Ralph Grisham and Richard Kittredge, editors, *Analyzing language in restricted domains: Sublanguage Description and Processing.* Lawrence Erlbaum, London, 1986.

G. Gazdar and C. Mellish. *Natural Language Processing in Prolog: an introduction to computational linguistics.* Addison-Wesley, Wokingham, 1989.

Jane Grimshaw. *Argument Structure.* Linguistic inquiry monographs 18. MIT Press, Cambridge, Massachusetts, 1991.

Arno Hofman and Wilco ter Stal. The plinius grammar engineering tool: A user manual. Memorandum ut-kbs-94-03, Enschede, The Netherlands, 1994.

Pierre Isabelle. Another look at nominal compounds. In *Proc. of the 10th International Conference on Computational Linguistics annd the 22nd Annual Meeting of the Association of Computational Linguistics*, pages 509–516, Stanford, 1984.

Karen Sparck Jones. So what about parsing compound nouns? In Karin Spark Jones and Yorrick Wilks, editors, *Automatic Natural Language Parsing*, pages 164–168. University of Essex, Essex, 1982.

Judith N. Levi. *The Syntax and Semantics of Complex Nominals.* Academic Press, New York, 1978.

Nicolaas J.I. Mars, Wilco G. ter Stal, Hidde de Jong, Paul E. van der Vet, and Piet-Hein Speel. Semi-automatic knowledge acquisition in plinius: an engineering approach. Memorandum ut-kbs-93-37, Enschede, the Netherlands, 1993. accepted for the Eighth Knowledge Acquisition Workshop, Banff, 1994.

Nicolaas J.I. Mars. The role of ontologies in structuring large knowledge bases. In *Proceedings of the International Conference on Building and Sharing Very Large-Scale Knowledge Bases '93*, pages 235–243, Tokyo, 1993. Japan Information Processing Development Center.

T. Parsons. *Events in the Semantics of English.* MIT Press, London, England, 1991.

James Pustejovsky. The generative lexicon. *Computational Linguistics*, 17(4):409–441, 1991.

Elaine Rich, Jim Barnett, Kent Wittenburg, and David Wroblewski. Ambiguity procastination. *AAAI*, 1987:571–576, 1987.

Elisabeth Selkirk. *The Syntax of Words.* MIT Press, Cambridge, Massachusetts, 1982.

Stuart Shieber. An introduction to unification-based approaches to grammar. Clsi lecture notes no. 4, Stanford, CA, 1986.

Klaas Sikkel and Rieks op den Akker. Predictive head-corner chart parsing. In *Proc. of the Third International Workshop on Parsing Technologies*, pages 267–276, Tilburg/Durbuy, 1993.

Piet-Hein Speel, Paul E. van der Vet, Wilco G. ter Stal, and Nicolaas J.I. Mars. Formalization of an ontology of ceramic science in classic. In Karen S. Harber, editor, *Proc. of the Seventh International Symposium on Methodologies for Intelligent Systems (ISMIS) Poster Session, Trondheim, Norway, June 15–18, 1993*, pages 110–124, Oak Ridge, TN, 1993. Oak Ridge National Laboratory. (ORNL/TM-12375).

Milena Stefanova and Wilco ter Stal. A comparison of PATR and ALE: practical experiences. In Anton Nijholt and Klaas Sikkel, editors, *Proc. of the TWLT 6. Natural Language Parsing: Methods and Formalims*, Enschede, the Netherlands, 1993. University of Twente.

Ko van der Sloot and Gerrit Rentier. The plus grammar. Esprit p5254, Tilburg, 1993.

Paul E. van der Vet and Nicolaas J.I. Mars. An ontology of ceramics. Ut-kbs-91-21, memoranda informatica 91-85, Enschede, the Netherlands, 1991.

P.E. van der Vet and N.J.I. Mars. Structured system of concepts for storing, retrieving and manipulating chemical information. *Journal of Chemical Information and Computer Science*, 33:564–568, 1993.

Paul E. van der Vet, Piet-Hein Speel, Wilco G. ter Stal, Hidde de Jong, Frank van Raalte, and Nicolaas J.I. Mars. Plinius intermediate report. Memorandum ut-kbs-93-36, Enschede, the Netherlands, 1993.

Jan van Eijck and Hiyan Alshawi. Logical forms. In Hiyan Alshawi, editor, *The Core Language Engine.* MIT Press, Cambridge, Massachusetts, 1992.

Frank van Raalte, Piet-Hein Speel, and Paul E. van der Vet. The plinius preprocess: intermediate report. Ut-kbs-92-26, Enschede, the Netherlands, 1992.

Paola Velardi. Acquiring a semantic lexicon for natural language processing. In Uri Zernik, editor, *Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon.* Lawrence Erlbaum, Hillsdale, New Jersey, 1991.

Margriet Verlinden. A head-corner parser for unification grammars. In Anton Nijholt and Klaas Sikkel, editors, *Proc. of the TWLT 6. Natural Language Parsing: Methods and Formalims*, Enschede, the Netherlands, 1993. University of Twente.

Lieve De Wachter and Jan Provoost. A computational interpretation of compounds. Working papers in natural language processing, Leuven, Belgium, 1993.

Beatrice Warren. Semantic patterns of noun-noun compounds. Gothenburg Studies in English 41, Göteborg, 1978.