Automatic sortal Interpretation of German Nominalisations with -ung
Towards using underspecified Representations in Corpora

Manuel Kountz[1], Ulrich Heid[2], Kristina Spranger[2]

1 Introduction

In this paper we present work on using dependency structures in a process of automatic sortal interpretation of German nominalisations with *-ung*, such as *Messung* ('measurement') or *Zählung* ('count'). Many such *-ung* nominalisations are ambiguous with respect to their sortal interpretation (*cf.* Ehrich and Rapp (2000) - who lean heavily on McCawley (1968) and Lakoff (1972) - for the notion of sortal ambiguity). In section 2.1 a more detailed discussion on sortal ambiguity as regards German *-ung* nominalisations is given.

We are working towards a system for data extraction from corpus text that is able to carry out sortal disambiguation. Given the productivity of the *-ung*-formation process in German (*cf.* Esau 1971 and Scheffler 2005) and the high frequency of *-ung* nominalisations in text (*cf.* Knobloch 2002 or Osswald 2005), this ability is relevant, among others, for question answering or high quality information extraction[3].

In the first part of this paper (section 2), we discuss data for *-ung* nominalisations and the methodological bases of our work on their sortal interpretation. We present a preliminary case study on phenomena in the context of the *-ung* nominalisations *Messung*, *Zählung*, and *Schätzung* ('estimate'), and the potential of these contextual phenomena to constrain the sortal interpretation of *-ung* nominalisations (see section 2.2). From a descriptive point of view, such phenomona serve as "indicators" of sortal readings. For the automatic sortal interpretation process, knowledge about reading indicators is explicitly formulated as constraints which are applied to a given nominalisation. We model the sortal interpretation as a process of incremental specification where the context of a given nominalisation is used for its sortal interpretation (section 2.3).

The system we are conceiving will process tagged and parsed corpus data, thus we do not have any discourse representations available which go beyond the sentence level. This means that the largest context available in the interpretation process is the sentence context. The order in which different constraints are applied is crucial to the sortal interpretation of a nominalisation at the sentence level. We demonstrate (also in

[1]Graduate Programme 609, University of Stuttgart,
email: kountzml@ims.uni-stuttgart.de

[2]Institute for Natural Language Processing (IMS), University of Stuttgart,
email: {heid|sprangka}@ims.uni-stuttgart.de

[3]A more detailed discussion on the relevance of this ability in natural language processing systems is given in Reckman and Cremers (2007).

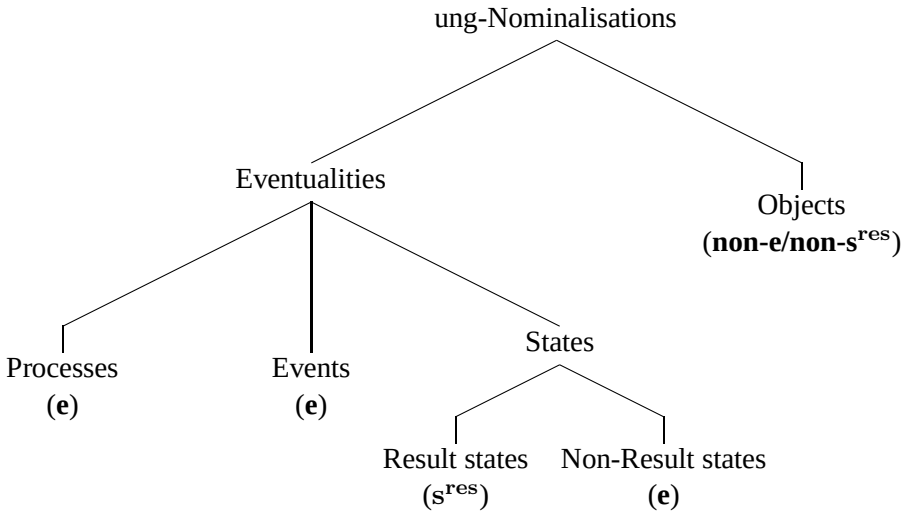

Figure 1: The Sortal Interpretation of German *-ung*-Nominalisations

section 2.3) that the sortal interpretation of a nominalisation depends on the syntactic analysis of the sentence in which it occurs.

In this paper we consider a dependency analysis as the syntactic analysis underlying the interpretation process. In section 3 we show by means of an example how the process of sortal interpretation can be operated on a dependency analysis.

With the sortal interpretation of a nominalisation in a sentence depending on the precise syntactic analysis, the process of its derivation is sensitive to syntactic ambiguity. We are developing an underspecified representation of ambiguous dependency structures which assembles all possible syntactic readings. Section 4 outlines options for an underspecified syntactic representation based on dependency structures.

In section 5, we propose an encoding of (underspecified) dependency structures in the framework of the *Linguistic Annotation Framework* (LAF, upcoming ISO standard, *cf.* Ide and Romary 2006): being able to annotate underspecified representations in the corpus will allow us to share the results our syntactic analysis with other corpus linguists.

2 Background

This section gives some background information on the sortal ambiguity of German nominalisations with *-ung* and for the interpretation method presented later.

2.1 Ambiguous German Nominalisations with *-ung*

To account for the sortal ambiguity of nominalisations, an ontology of reading types is commonly used, which distinguishes, according to Ehrich and Rapp (2000), between eventualities and objects.

Eventualities. Ehrich and Rapp subsume processes, events, and states under the concept of eventualities taken over from Bach (1986).

Events refer to telic actions whereas processes refer to atelic actions[4]. According to Moens and Steedman (1988) processes as well as events can be seen as event complexes that are an association of a goal event, or "culmination" with a "preparatory phase" by which it is accomplished and a "consequent state" which ensues.
States (result states as well as non-result states) refer to eventualities that do not have a dynamic preparatory phase. Result states (e.g. *Absperrung* 'roadblock'), in contrast to non-result states (e.g. *Bewunderung* 'admiration'), are caused by a preceding event. Therefore, we distinguish between result states and other eventualities (including non-result states).

In the following, processes, events and non-result states are referred to by **e**, result states are referred to by $\mathbf{s}^{res}$.

Objects. Objects refer to physical as well as abstract objects. They are referred to by **non-e/non-s**^{res}.

Except for non-result states and objects all classes of *-ung* nominalisations (*cf.* figure 1) refer to some phase in the event complex as it is described by Moens and Steedman (*cf.* Moens and Steedman 1988): result states refer to the post-culmination phase, and events and processes refer to the whole event complex. Thus, it is especially challenging to keep them apart. To this end, Ehrich and Rapp propose a number of distributional tests:

1. Only eventualities allow to refer to phases of the events (a) and can be combined with process modifying predicates (b):

   (a) Die Verfolgung des Täters / Die Absperrung des Geländes
   The pursuit the perpetrator / The cordon the area
   beginnt / **hört auf** / **wird unterbrochen**.
   starts / stops / is interrupted.

   'The pursuit of the perpetrator / The cordon of the area starts / stops / is interrupted.'

   (b) die **umständliche** / **vorsichtige** Verfolgung des Täters /
   the awkward / cautious pursuit the perpetrator /
   Absperrung des Geländes
   cordon the area

   'the awkward / cautious pursuit of the perpetrator / cordon of the area.'

2. Result states can be combined with stative predicates (a) and with predicates of perceptibility (b) (summed up as "static predicates"):

[4] According to Vendler (1967) events are "accomplishments" and "achievements", and processes are "activities".

(a) die **bestehende** Absperrung des Geländes
the existing cordon the area
'the existing cordon of the area'

(b) die **vorgefundene** / **kartographisch registrierte** Absperrung des Geländes
the found / cartographically registered cordon the area
'the cordon of the area found / cartographically registered'

3. Duration predicates can only occur together with processes and result states:

   die **tagelange** Verfolgung des Täters / Absperrung des Geländes
   the lasting for days pursuit the perpetrator / cordon the area

   'the pursuit of the perpetrator / cordon of the area lasting for days'

4. Events can go together with time frame predicates (a) and they allow to refer to the incremental progression of the event (b):

   (a) die **in zwei Tagen** erfolgte Absperrung des Geländes
   the in two days accomplished cordon the area
   'the cordon of the area accomplished in two days'

   (b) die **allmähliche** Absperrung des Geländes
   the gradual cordon the area
   'the cordon of the area completed step by step'

The distributional tests show that event nominalisations and result state nominalisations are distributed complementarily.

2.2 A corpus-based case study of *-ung* nominalisations: *Messung, Zählung, Schätzung*

On the basis of newspaper text from the *Stuttgarter Zeitung* (1992/93, total of *c.* 36 M words), we manually analysed the readings of a few (semantically related) nominalisations: *Messung* ('measurement'), *Zählung* ('count'), *Schätzung* ('estimate') and their compounds. For each nominalisation, at least 100 corpus sentences were sortally classified and sentence contexts analysed.

2.2.1 Sortal Readings of *Messung*

The nominalisation *Messung* ('measurement') is two-way ambiguous: it allows for an event interpretation (**e**), and for an object interpretation (**non-e**)[5].

[5]For the sake of convenience, we do without **non-s**^{res} since there is no result state interpretation of *Messung*.

The event reading of *Messung* refers to the process of measuring. Sentence (1) is a typical context for *Messung* as an event.

(1) die **Messung** des Erdumfangs durch Eratosthenes
the measuring the circumference of the earth by Eratosthenes

'the measuring of the circumference of the earth by Eratosthenes'

The object reading refers to the result of a measuring process, i.e. to data or figures. Sentence (2) is a context for *Messung* as an object.

(2) Die **Messungen** liegen unter dem zulässigen Grenzwert von 250 ppm.
The measurements lie under the acceptable critical value of 250 ppm.

'The measurements are lower than the maximum permissible value of 250 ppm.'

2.2.2 Indicators from the context for sortal disambiguation

To decide about the sortal interpretation of an *-ung* nominalisation, humans seem to use lexico-semantic and syntactic reading indicators from the context.
Many lexical indicators are combinatory constraints, ranging from preferences for general (ontological) classes, (e.g. `[human, agentive]` for the *durch*-phrase in (1)) over selection restrictions (the reading of *liegen* in (2) requires a subject of kind `[data]`), to lexeme-specific combinations such as the support verb construction *Messung+durchführen* (lit: 'execute measurement', 'carry out'). Some such combinatory constraints underlie the distributional tests proposed by Ehrich and Rapp (2000). We list more such indicators, derived from the *Stuttgarter Zeitung* data in tables 1 and 2.

In a given sentence, the indicators may appear in different syntactic structures; for example, the support verb construction *Messung+durchführen* may come as a verb+object pair, as a prenominal participle (*durchgeführte Messung*) etc.: automatic sortal disambiguation thus has to be based on syntactic parsing. Moreover, roughly synonymous indicators may belong to different word classes, *cf.* the duration predicates in table 1. This suggests that more abstract syntactic representations are more adequate for disambiguation.

2.2.3 Indicators in actual corpus data

Tables 1 and 2 contain, among others, verbs serving as reading indicators. However, not all verbs have sortal preferences, and there are sentences where the context does not provide any hints for the sortal disambiguation of the *-ung* nominalisation, i.e. where its sort does not matter and remains ambiguous (*cf.* 3)

#	Type	Examples
1	Reference to event phase	nominalization as a subject: *Messung geht weiter, beginnt, endet*
		nominalization as an object: *Messung aufnehmen, fortsetzen, abschließen*
2	Duration predicates	verbs: *Messung dauert (x lange)* adjectives: *fortlaufende, kontinuierliche M.,* *viertägige Messung* nouns: *Dauer der Messungen* temporal PP: *während der Messung*
3	Selection restrictions on the object of verbs of ordering	*Messung anordnen, vorschreiben,* *Messung veranlassen*
4	Lexical collocations	support verbs + object: *Messung + durchführen,* *Messung + vornehmen* verb + subject: *Messung findet statt*
5	Temporal/local adjuncts	*Messungen an Straßen, Messungen im Mai*

Table 1: Event reading indicators

#	Type	Examples
1	Static predicates	*Messungen liegen vor*
2	Selection restrictions on verbs indicating a value	subject: *Messung liegt bei x*
3	Use with verbs of proving	subject: *Messung belegt, beweist, zeigt, daß*
		instrument/cause: *jmd zieht aus Messungen den Schluß, daß ...;* *jmd beweist mit Messungen, daß ...*
4	Use in PP-adjuncts of type "citation"	*nach Messungen , laut M., M. zufolge*

Table 2: Non-event reading indicators

(3) Die Kürze der Haare spielt bei solchen Zählungen übrigens keine
the length the hair plays with such counts by the way no
Rolle
role

'Hair length is by the way irrelevant for such counts'

On the other hand, many sentences contain more than one indicator. Often these jointly suggest a given reading, but there are also cases where the indicators present in a given sentence do not support the same reading. Sentence (4) is an example:

(4) Wir beschreiben Messungen [**auf den Seychellen**]$_e$, [die Anzeichen
We describe measurements on the Seychelles, that indications
des Klimawandels **zeigen**]$_{non-e}$.
the climate change show.

'We describe measurements on the Seychelles that show indications of the climate change.'

auf den Seychellen is an indicator for the event reading: it is a local adjunct (*cf.* table 1). The relative clause *die Anzeichen des Klimawandels zeigen* with *zeigen* as predicate is an indicator for the object reading: *zeigen* belongs to the class of "proving verbs" (*cf.* table 2).
Nevertheless, the nominalisation does not (necessarily) remain sortally ambiguous at the sentence level. The human reader is perfectly able to interpret the nominalisation as an event or as an object – at the latest when he considers a larger context window than one sentence.

2.3 Incremental sortal Specification in Context

As example (4) shows, indicators may appear in different places in the syntax tree; consequently, the syntactic structure of a sentence may have an impact on the interpretation of *-ung* nominalisations it contains. In one reading of (4), the relative clause *die ... zeigen* modifies *Messungen*: as the proving verb *zeigen* requires a non-event subject, *Messungen* is then to be interpreted in its object reading. In another reading, the relative clause may be attached to Seychellen, and consequently, we get a complex PP *auf den Seychellen, die Anzeichen für den Klimawandel zeigen*, which is an adjunct to Messungen. As the nominalisation is then embedded under the main verb *beschreiben*, which has no preferences for event vs. non-event objects, the noun can receive either interpretation in this context.

These examples show that structural syntactic ambiguity may have an impact on the sortal interpretation of *-ung* nominalisations. We propose an incremental interpretation algorithm which starts from the bare nominalisation and takes increasingly larger amounts of context into account, by walking up the syntax tree of the sentence to be analysed.

The sortal interpretation of the nominalisation is "defeasible" as long as there is a larger context that is relevant for the interpretation process; thus, within a 'local context' (e.g. a noun phrase) a disjunction of all possible values (`e | non-e |` $<$`e` $\overset{+}{\cup}$ `non-e`$>$)[6] is available which carries a 'local preference' for one particular value. This preference is defeasible as long as the window of observation can grow[7]. Only at the

[6] $<$`e` $\overset{+}{\cup}$ `non-e`$>$ reads event or object.

[7] As regards our concept of "defeasible" and "indefeasible" sortal interpretations, we lean on Alshawi and Crouch's concept of "believed" vs. "unbelieved" in their monotonic semantic interpretation (*cf.* Alshawi and Crouch 1992).

sentence level, where the context for interpretation cannot grow any more, the interpretation is "believed" and thus becomes indefeasible. In this sense, our specification process is monotonic in so far as growing context and constraints introduced by this context lead to a specialisation of the sortal annotation from a smaller context up to the sentence context.

2.4 A Constraint-based Algorithm

The core idea of this specification process is that the reading indicators that enter the context while it grows incrementally introduce constraints that can be applied to a nominalisation in its current context. The specification process follows the algorithm given below:

1. The "bare" *-ung* nominalisation (i.e. the nominalisation in its null context) which, obviously, is sortally ambiguous gets the sortal type $<\texttt{e} \overset{+}{\cup} \texttt{non-e}>$.
2. Then, all sibling nodes are considered: before a sibling node is added to the "active" context[8], it is checked whether it dominates an indicator.
3. If so, the indicator introduces a constraint over the interpretation of the *-ung* nominalisation in its current context.
4. The constraint is applied, and the sibling node is added to the context of the nominalisation.
5. The procedure is repeated until the sentence node is reached.

See Spranger and Heid (2007) for more detailed examples for the application of the algorithm outlined above.

The Main Constraint and a Type Conversion Function. Supposed:

- $\texttt{U} = \{$x, x is a *-ung* nominalisation$\}$
- $\texttt{m} \in \texttt{U}$
- $\texttt{ung-sort} = \{$e, s^{res}, non-e/$s^{res}\}$
- $\alpha, \beta \in \texttt{ung-sort}$

We define a constraint $C_{<\alpha \overset{+}{\cup} \beta, \alpha>}$ that has the following two properties:

1. $C_{<\alpha \overset{+}{\cup} \beta, \alpha>}(m_{\alpha \overset{+}{\cup} \beta}) = m_{\alpha}$
2. $C_{<\alpha \overset{+}{\cup} \beta, \alpha>}(m_{\beta}) = m_{\alpha}$

[8] "Active" context is used in the sense of "active" edges in chart parsing.

In order that the constraint be applicable to m_β we define a type conversion function τ:

- $\tau(m_\beta) = m_{\alpha \overset{+}{\cup} \beta}$

2.5 Dependency Structures

A dependency structure describes a syntactic analysis of a sentence in terms of words linked by a directed, pair-wise relation of dependency between governor and dependent. Each such dependency link is labelled with the syntactic role[9] the dependent bears with regard to its governor. We depict a dependency structure as a directed acyclic graph with edge labels, which in turn can be seen as a set of dependency triples of the form `Role(Governor, Dependent)`.

As dependency is a directed relation, we can (somewhat informally) say that a governor is situated higher in the dependency structure than its dependents. In addition, we can extend pair-wise dependency to a transitive relation. This allows us to define substructures consisting of nodes which are transitively governed by a particular governor.

Valency controls which (classes of) words may be governed by a certain word. We extend the notion of valency to cover adjuncts (besides proper complements). This means that e.g. the valency of a verb also covers the possibility of attaching a PP, but excludes relative clause attachment. When we need to find possible points for attachment of complements, we exploit this knowledge.

In the dependency graphs pseudo-edges relate nodes of the dependency graph to words. Pseudo-edges have no theoretical status.

3 Using Dependency Structures to guide Context Selection in sortal Interpretation

As shown in 2.3 (and as also presented in Spranger and Heid (2007)), sortal interpretation of a German nominalisation with *-ung* in a sentence depends on the syntactic reading of the sentence. The syntactic structure determines the order of contexts from which indicators for the sortal reading are drawn. In this section, we will show how a dependency structure can be used for this purpose by giving details on the interpretation process.

The problem of role ambiguities is not addressed in this paper; we are aware of the fact that e.g. subject-object ambiguities may have an impact on attachment of relative clauses, but it is not yet clear in how far role ambiguities have an impact on the sortal interpretation process.

[9] We use the following role labels: `SB` – subject, `OBJ` – object, `PCOMP` – complement of a preposition, `DET` – determiner of a noun, `MOD` – adjunct to verb or noun, `GR` – postnominal genitive, `RSK` – separated verb particle in right sentence bracket.

All dependents of a word are equally attached, no matter what their linear order is or whether the relation of one dependent to the governor is 'closer' than that of another dependent. In these cases, we need additional grammatical knowledge in order to guide context selection. This knowledge can be added to the grammar by adding a separate layer which accounts for linear precedence (e.g. by employing topological fields, as in Duchier and Debusmann 2001). Another way might be to impose an ordering on the valency frame of a word, thus selecting a dependent first which is ranked higher.

We will show the interpretation process for the sentence in example 5, assuming that it has the syntactic reading with the structure given in figure 2.

(5) Die den Havelzander von unzulässigen Giftbelastungen
The the Havel zander from impermissible pollutant burdens
freisprechenden Messungen fanden im Januar statt.
clearing measurements took in January place

'The measurements clearing the Havel zander from impermissible pollutant burdens took place in January'

The sortal interpretation of *Messungen* in example 5 proceeds as follows:

1. We start with the nominalisation itself. Using the lexicon entry for *Messung*, we determine its sortal type as $\texttt{e} \overset{+}{\cup} \texttt{non-e}$.

2. The first context to be taken into account consists of *Messungen* and its dependent *den Havelzander von unzulässigen Giftbelastungen freisprechenden*. *freisprechen* is identified as a proving verb ('prove that not') and thus imposes the constraint $C_{<\texttt{e}\overset{+}{\cup}\texttt{non-e},\texttt{non-e}>}$ (*cf.* table 2). This constraint is applied to *Messungen*. The active context is now *den Havelzander . . . freisprechenden Messungen*.
   Next, we consider the article *die*, which does not impose any constraint. Thus we can simply add it to the context. *Messungen* has no further dependents.

3. Now we set focus on the immediate governor of *Messungen* – this node is *stattfanden*. *Stattfinden* indicates an event reading, as does the temporal adjunct *im Januar*.
   The event reading constraint obtained from *fanden im Januar statt* cannot be applied immediately. At this point in the interpretation process, *Messungen* has been found to bear a non-event reading. Thus, we apply the *type conversion function* τ (see section 2.3) and obtain the type $\texttt{e} \overset{+}{\cup} \texttt{non-e}$ for *Messungen*. Now the constraint imposed by *fanden im Januar statt* can be applied, and *Messungen* is interpreted as an event at the verb node.

4. As there is no higher node (and thus no larger context), interpretation stops, and *Messungen* is interpreted as an event at the sentence level.

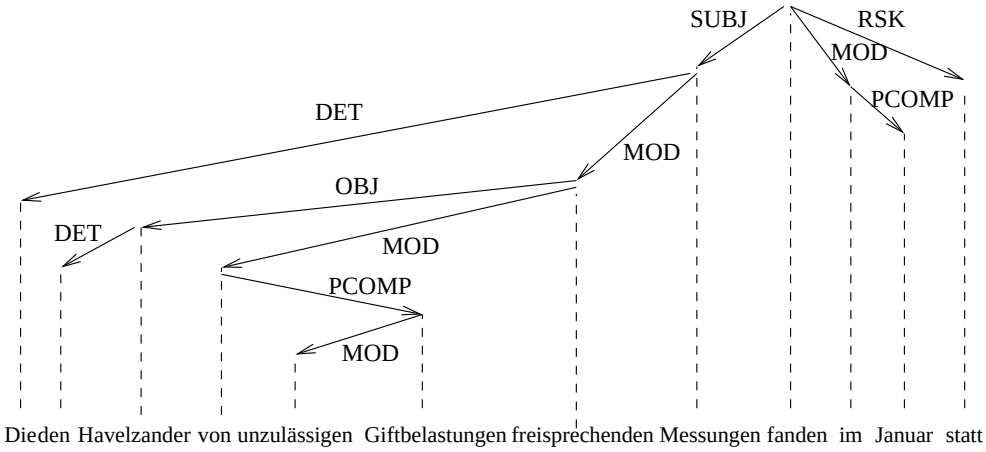

Figure 2: The reading of (5) used in section 3

4 Underspecified Representation and the sortal Interpretation Process

In order to efficiently store and process syntactic analyses of ambiguous material, we suggest an underspecified representation (USR) of dependency structures.

Our first step will be to discuss the basic assumptions and considerations behind the underspecified representation we devise. We will use an ambiguous example for this purpose, again focusing on structural ambiguities, and assume that we knew all possible readings and outline a way to represent them.
We proceed by showing how we can reconstruct all possible readings from the USR we propose, and how the interpretation process outlined in section 3 can be integrated into the reconstruction process.

To be usable in the process of constructing an underspecified representation like the one discussed here, a parser must meet a set of requirements; We will discuss these in section 4.3, along with the distribution of knowledge in the process.

4.1 Representation

We will use example 4 (repeated below as example 6) to present the considerations underlying the process of constructing an underspecified representation.

(6) Wir beschreiben Messungen auf den Seychellen, die Anzeichen für den Klimawandel zeigen.

Example (6) has the following four readings

1. The PP *auf den Seychellen* is adjunct to the main verb *beschreiben* and the relative clause *die . . . zeigen* is a modifier to *Messungen.*

2. The PP again modifies the main verb, but *die . . . zeigen* modifies *Seychellen.*

3. In the third reading, both the PP *auf den Seychellen* and the relative clause modify *Messungen.*

4. Finally, *Messungen* may be modified by the PP, while the relative clause again modifies *Seychellen.*

Fragments. If we compare the syntactic structures of the four readings of example (6), we observe that all of them can be partitioned into three *fragments* f_1, f_2, f_3 whose internal structures (i.e. the set of pair-wise dependencies and the respective roles) never change:

f_1 *Wir* and *Messungen* always depend on *beschreiben*;

f_2 *Seychellen* always governs its article *den* and depends on *auf*;

f_3 the relative clause *die Anzeichen für den Klimawandel zeigen* always has the same internal structure.

The dependency relation between a fragment governor (e.g. *auf* for fragment f_2) and its respective governor is not considered as being a part of the fragment proper. We depict the fragments assigned to example (6) as structures consisting of solid arrows in figure 3.

Constraints on Combinability of Fragments. The aim of constructing an underspecified representation of a set of syntactic structures is to encode all knowledge present in these structures as efficiently as possible. If there is no alternative for attaching a specific node to the whole structure (e.g. in *auf den Seychellen* we have only one way of attaching *Seychellen* to *auf*), encoding this pair-wise dependency immediately is efficient. However, if we have several options for attaching a specific node (e.g. the PP governor *auf* in example 6), we can either encode all alternatives explicitly or resort to a *constraint* which allows us to *reconstruct* all alternatives.

The approach to an underspecified representation of dependency structures presented here is based on the observation that each reading amounts to a specific arrangement of fragments (which are ordered with respect to the dependency relation). We formulate explicit constraints on the arrangement of fragments which cover all possible arrangements. Later, all possible readings are reconstructed by arranging fragments according to the constraint *and* by attaching the fragment governors in a way consistent with the grammar.

For the fragments f_1 and f_2, we can find the following constraints on fragment order:

- f_2 can be attached to the main verb *beschreiben* or to *Messungen* – in both cases the PP is attached to the fragment f_1. Thus we can formulate a constraint which indicates that f_2 must be positioned below f_1 (and leave it to the reconstruction phase to use grammatical knowledge to determine *Messungen* as a possible attachment point. In any case, the PP f_2 is a modifier (role: MOD) of the node it depends on.
  We write this as
  - *beschreiben* $>_{MOD}$ *auf*
- f_3 can be attached to *Messungen* and to *Seychellen.* As the fragment containing *Seychellen* is always attached to f_1, we can say that in all readings f_3 is situated *below the fragment* f_1. In both cases the role of the relative clause is MOD. This can be written as
  - *beschreiben* $>_{MOD}$ *zeigen*

In figure 3 these constraints are visualised as dotted arrows between the governors of fragments.

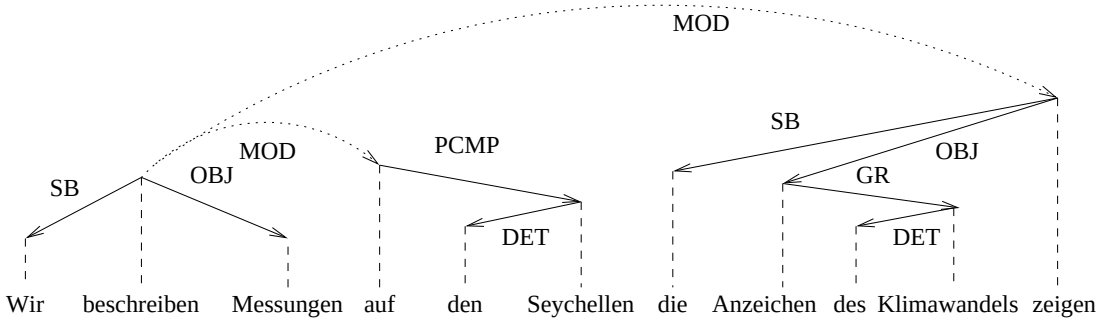

Figure 3: Fragments of the analysis of (3).

Dotted lines depict constraints on the ordering of fragments. Dependency triples which occur unchanged in all readings are shown as solid arrows.

4.2 Reconstruction

When reconstructing the structures of individual readings, all possible attachment points below the node given on the left-hand side of a particular constraint are determined using their (extended) valency: If the lexicon entry of a candidate governor (or a generic rule applying to it) allows for a modifier (in our example), the fragment on the right-hand side can be attached (as a modifier) to this particular governor.

Reconstruction proceeds in three phases:

1. Determine a topmost fragment:
   If for a fragment f_i there is no constraint of the form $\texttt{X} >_R \texttt{f}_i$, then it can be the topmost fragment.

2. Attach fragments according to constraints:
   For each fragment f_j which has not already been attached, identify all possible attachment points as follows:

   - If there is an attachment constraint of the form $\texttt{f}_k >_R \texttt{f}_j$ and f_k is already attached, identify all possible governors for f_j below f_k by means of extended valency.
   - If there is no constraint on attachment of f_j, identify all possible governors for f_j in the structure built so far (to which no other fragment has been attached).

   Attach f_j to a candidate governor to which it may attach without yielding a reading produced before.

3. If the resulting structure is valid (i.e. there are no unconnected fragments), output it as a reading.

Integrating the Process of sortal Interpretation. A node is a possible governor as long as its extended valency allows for more complements or adjuncts. After a fragment containing an *-ung* nominalisation has been added, and the nominalisation has ceased to be a possible governor, the sortal interpretation procedure outlined in section 3 can be started.

4.3 Construction from Parser Output

To construct an underspecified structures (of the kind discussed above) for a corpus sentence, at least a partial syntactic analysis is needed. The parser must be able to identify the links *inside* fragments (e.g. *subject(beschreiben, Wir)* in our example), that is, dependency triples which occur invariantly in all readings.
Furthermore, we must be able to determine the ordering between fragments. This can be derived from parser output, e.g. by a post-processing module which scans the fragments determined by the parser, and computes an order among them by applying knowledge about valency and rules for attachment of modifiers.

However, there is a drawback of almost certainly doubling knowledge between the parsing (or postprocessing) step and the reconstruction step – both employ knowledge about valency and attachment, and in almost the same way. One might argue that it is much easier to have the parser explicitly encode all possible attachments in its output and simply store this in the corpus, e.g. as alternative dependency triples.
There is certainly no "ultimate" answer to this question. The savings resulting from

storing constraints (e.g. as "expandable" dependency links) instead of explicitly coded attachment alternatives may outweigh the costs of duplicating knowledge and increasing processing time in one case. In another case, costs incurred by processing and duplication of (lexical and grammatical knowledge) may be dominant. When designing an NLP application, this question must be decided carefully.

5 Storing underspecified Representations using LAF

In this section, we will show how the underspecified representation described so far can be stored using the Linguistic Annotation Framework (LAF; Ide and Romary 2006). LAF proposes a generic data model for corpus annotation; an XML dump format for writing LAF annotations is being defined as well. We will show how fragments and constraints can be represented using the means of representation provided by LAF.

LAF assumes a 'primary segmentation' of corpus data (text in our case) by means of a set of edges demarking primary segments in the corpus data. We assume that this step has been completed by a tokeniser, and a primary segmentation denoting tokens is already available.

Dependency structures are encoded as a (so called) linguistic annotation which refers to a primary segmentation. Linguistic annotations as defined by LAF are directed graphs (Ide and Romary 2006, section 2), thus dependency structures, which are also directed graphs (*cf.* section 2.5) can be encoded directly. Edges in LAF's linguistic annotations may refer to other edges or to a primary segmentation. For dependency structures, only reference to primary segments (which denote words) is needed. Pair-wise dependencies are encoded as an edge from governor to dependent.

Edges are neither labelled nor typed in LAF. Information about categorisation of edges is stored in feature structures[10] (as is other information attached to edges). See figure 4 for an example, presented in the XML representation employed by Ide and Romary (2006).

In the underspecified representation, we distinguish two types of pair-wise relations between tokens: 'ordinary' dependency links which are invariant across readings and thus part of the internal structure of a fragment, and constraints relating a highest possible governor to a fragment. We can model both as category-bearing links between words (recall that constraints also encode a role), but have to encode the difference in relation type between link and constraint .

We suggest to encode this in LAF feature structures as follows:

- To a dependency link, a feature structure containing a feature `dep-rel` (dependency relation) is assigned; its value is the syntactic role of the dependent with regard to its governor.

[10] See ISO TC37 SC4 document N188, Feature Structures – Part 1: Feature Structure Representation (2005-10-01), available at `http://www.tc37sc4.org`.

```
<!-- primary segmentation: token edges -->
<edge id="t1" from="1" to="3"/>    <!-- Wir         -->
<edge id="t2" from="5" to="15"/>   <!-- beschreiben -->
<edge id="t3" from="17" to="24"/>  <!-- Messungen   -->

<!-- linguistic annotation: pair-wise dependency relations -->
<edge id="d1" ref="t2 t1">
  <fs>
    <f name="dep-rel" sVal="SB" />
  </fs>
</edge>
<edge id="d2" ref="t2 t3">
  <fs>
    <f name="dep-rel" sVal="OBJ" />
  </fs>
</edge>
```

Figure 4: LAF compatible representation of *Wir beschreiben Messungen*

The example is presented in the XML representation also used in Ide and Romary (2006). For the primary segmentation, we assume that primary corpus data is the string *Wir beschreiben Messungen*, with the characters numbered 1 through 24. The names `d1` and `d2` assigned to dependency edges are arbitrary.

- Constraints are encoded by means of a feature structure containing a feature `dep-constraint`. The value of this feature is a syntactic role; the governor of the fragment pointed to by the constraint edge will bear this role after attachment.

Figure 5 shows how the fragments *Wir beschreiben Messungen* and *auf den Seychellen* along with the constraint '*beschreiben* $>_{MOD}$ *auf*' in the underspecified representation of example (6) (*cf.* figure 3) can be encoded.

6 Conclusions

Sortal interpretation of nominalisations with *-ung* is highly context dependent; indicators drawn from the context can trigger a particular reading. We presented data from text corpora regarding indicators for the sortal readings of the nominalisation *Messung*. The exact course of sortally interpreting a nominalisation depends on syntactic structure and thus, due due syntactic ambiguity, also on the precise reading assumed (*cf.* Spranger and Heid 2007).

An algorithm for incremental sortal specification of *-ung* nominalisations has been presented; we also showed how this basic procedure can be applied to dependency

```
...
<edge id="t4" from="26" to="28"/>    <!-- auf   -->
<edge id="t5" from="30" to="32"/>    <!-- den   -->
<edge id="t6" from="34" to="43"/>    <!-- Seychellen  -->

...
<edge id="d3" ref="t4 t6">
  <fs>
    <f name="dep-rel" sVal="PCOMP" />
  </fs>
</edge>
<edge id="d4" ref="t6 t5">
  <fs>
    <f name="dep-rel" sVal="SPEC" />
  </fs>
</edge>

<!-- the constraint beschreiben >MOD auf -->
<edge id="c1" ref="t2 t4">
  <fs>
    <f name="dep-constraint" sVal="MOD" />
  </fs>
</edge>
...
```

Figure 5: One more fragment and a constraint added to figure 4

In this example, the XML code is shown which must be added to figure 4 to represent one more fragment and the first constraint of example (6).

structures. In order to account for structural ambiguity, we suggested an underspecified representation of dependency structures. This underspecified representation can be represented in a way compatible with the upcoming ISO standard for a linguistic annotation framework.

The precise selection of contexts (especially in the case of competing indicators being dependents of the same node) is based on grammatical rules. We will evaluate options e.g. based on topological extensions to dependency grammar (*cf.* Duchier and Debusmann 2001).

The underspecified representation devised here is one possible option (among others) of dealing with ambiguous data. For practical use, it must be evaluated against other options, e.g. the representation employed by the parser presented in Schiehlen (2003) or a representation based on Spranger (2006).

Beginning the process of sortal interpretation at a suitable point during the reconstruction of a reading is not the only option of integrating reconstruction and sortal interpretation. We aim at precomputing as much as possible and intend to derive the local preference for the sortal interpretation of a nominalisation (*cf.* section 2.3) whenever (a fragment containing) a reading indicator is added to the rudimentary reading reconstructed so far.

References

Alshawi, Hiyan and Richard S. Crouch (1992) 'Monotonic Semantic Interpretation.' *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, 32–38.

Bach, Emmon (1986) 'The Algebra of Events.' *Linguistics and Philosophy 9(1)*, 5–16.

Duchier, Denys and Ralph Debusmann (2001) 'Topological Dependency Trees: A Constraint-based Account of Linear Precedence.' *Proceedings of the ACL 2001 Conference.*

Ehrich, Veronika and Irene Rapp (2000) 'Sortale Bedeutung und Argumentstruktur: ung-Nominalisierungen im Deutschen.' *Zeitschrift für Sprachwissenschaft 19(2)*, 245–303.

Esau, Helmut (1971) 'Some facts about German nominalisation.' *Neophilologus 55(1)*, 150–156.

von Heusinger, Klaus (2002) 'The Interface of Lexical Semantics and Conceptual Structure: Deverbal and Denominal Nominalisations.' In: Nominalisierung. Zimmermann, I. and Lang, E. (eds.). Zentrum für Allgemeine Sprachwissenschaft, Berlin, 109–124

Ide, Nancy and Laurent Romary (2006) 'Representing Linguistic Corpora and Their Annotations.' *Proceedings of the Fifth Language Resouces and Evaluation Conference (LREC)*, Genoa, Italy.

Knobloch, Clemens (2002) 'Zwischen Satz-Nominalisierung und Nennderivation: -*ung*-Nomina im Deutschen.' *Sprachwissenschaft 27(3)*, 333–362.

Lakoff, George (1972) 'Linguistics and Natural Logic.' In: Approaches to Natural Language. Davidson, Donald/Harman, Gilbert (eds.), Reidel, Dordrecht/Boston, 545–665.

McCawley, James D. (1968) 'Lexical Insertion in a Grammar without Deep Structure.' Papers from the 4th Regional Meeting of the Chicago Linguistic Society, 71–80.

Moens, Marc and Mark Steedman (1988) ‘Temporal Ontology and Temporal Reference.’ *Computational Linguistics 14(2)* 15–28.

Muskens, Reinhard (2001) ‘Talking about Trees and Truth Conditions.’ *Journal of Logic, Language, and Computation 10(4)*, 417–455.

Osswald, Rainer (2005) ‘On Result Nominalization in German.’ *Proceedings of Sinn und Bedeutung 9*, 256–270.

Reckman, Hilke and Crit Cremers (2007) ‘Deep parsing semantic interpretation of nominalizations and their expressed and unexpressed arguments.’ *Leiden Working Papers in Linguistics 4(1)*, 40–55.

Scheffler, Tatjana (2005) ‘Nominalization in German.’ Unpublished Manuscript, University of Pennsylvania.

Schiehlen, Michael (2003) ‘A Cascaded Finit-State Parser for German.’ *Proceedings of the Meeting of the European Chapter of the Association of Computational Linguistics.*

Shin, Soo-Song (2001) ‘On the event structure of *-ung* nominals in German.’ *Linguistics 39(2)*, 297–319.

Spranger, Kristina (2006) Combining Deterministic Processing with Ambiguity-Awareness – The Case of Quantifying Noun Groups in German. PhD Thesis, University of Stuttgart.

Spranger, Kristina and Ulrich Heid (2007) ‘Applying Constraints derived from the Context in the Process of Incremental Sortal Specification.’ *Proceedings of the Workshop CSLP@Context 2007*, Roskilde.

Vendler, Zeno (1967) ‘Facts and Events.’ In: Linguistics in Philosophy, Cornell University Press, Ithaca, 122-146