Fachrichtung 4.7 Computerlinguistik
Universität des Saarlandes

Towards a Linking of FrameNet and SUMO

Diplomarbeit

Nils Reiter

Angefertigt unter der Leitung von
Prof. Dr. Manfred Pinkal und
Aljoscha Burchardt

Erklärung

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Saarbrücken, den 21. September 2007

Nils Reiter

Abstract

Most end-user applications of natural language processing such as question answering or information retrieval – and especially a research-oriented task like textual entailment – need to process entailment and contradiction in one way or another. In order to detect entailment and contradiction, a system needs to access very different kinds of resources, such as lexicons, semantic word nets or knowledge ontologies.

One such knowledge ontology is SUMO. SUMO represents upper-model knowledge in an abstract way, using a lisp-like format whose expressivity equals that of first-order logic. A direct connection between the syntactic level in the form of free or syntactically preprocessed text and the semantics in the form of SUMO does not exist. There is no obvious way to represent the information contained in a text or sentence using SUMO.

This thesis is concerned with this syntax-semantics interface. We will propose an algorithm that combines FrameNet and SUMO. Methods and tools to achieve FrameNet annotations of a given sentence do exist already. Our combination algorithm is based on this FrameNet annotation and links frames to SUMO concepts and frame elements to SUMO relations.

Acknowledgements

I would like to thank Prof. Dr. Manfred Pinkal for his support in this thesis. He created an inspiring and stimulating research environment at his department.

My deepest thanks go to Aljoscha Burchardt for his constant, fruitful and constructive input. He always knew how to make sense of my drawings and had encouraging words in moments of doubt.

Contents

1. Introduction

In this diploma thesis, we will describe an approach to link written text to knowledge contained in an upper-model ontology. The approach builds upon the existing resource FrameNet and several tools and methods used with FrameNet. FrameNet provides us with a semantic layer abstracting from the surface form of the sentence. This layer contains frames and frame elements, describing events and participants of events. Our approach links both frames and frame elements to appropriate concepts in the SUMO ontology.

In chapter 2, we will explain our motivation for this work and provide some background information as well as application scenarios. Chapter 3 describes WordNet, FrameNet, SUMO and the tools and algorithms needed for our approach. We will describe our own approach in detail in chapter 4 and the evaluation of our approach in chapter 5. We conclude this work in chapter 6 with a review and a brief outlook on possible future work.

2. Background and Motivation

In the first section of this chapter, we will explain textual entailment and why it is important for natural language processing. We will roughly categorize the knowledge needed in order to decide if textual entailment holds for various examples in the second section of this chapter. Some of the resources containing various kinds of knowledge are briefly presented in section three. We will discuss advantages and shortcomings of the resources and motivate why we aspire a linking between FrameNet and SUMO. A more detailed outline of this diploma thesis is presented in the last section of this chapter.

2.1. Textual Entailment

In recent years, different approaches to (end-user) applications of language technology have been developed and implemented in prototype systems. Any such approach has to deal with semantic processing in one way or another. An important part of semantic processing is the detection and handling of inferences and contradictions. In order to gain more insight in these phenomena and to compare the performance of systems addressing it, the general task of textual entailment has been defined.

The task of textual entailment is to detect entailment relations between complete texts, i.e., to decide if the information contained in one text is also contained in another text. In 2005, the RTE challenge (Recognising Textual Entailment, Dagan et al. (2005)) was established. In RTE, the performance of different systems is compared on 800 humanly annotated sentence pairs. The following definition of textual entailment is used.

> Textual entailment is a directional relationship between a coherent text T and a language expression, which is considered as a hypothesis H. We say that T entails H (H is a consequent of T), denoted by $T \Rightarrow H$, if the meaning of H, as interpreted in the context of T, can be inferred from the meaning of T, as would typically be interpreted by people.

Crouch et al. (2003) describe entailment (and contradiction) as "the key data of semantics". Full language understanding is more than entailment and contradiction, but nevertheless, entailment and contradiction are necessary parts thereof. For instance, if one fails to recognize the contradiction in example 2.1, then one has not understood the sentences.

(2.1) T No civilians were killed in the Najaf suicide bombing.

H Two civilians were killed in the Najaf suicide bombing.

$T \not\Rightarrow H$

Conversely, if one fails to recognize the entailment relation in example 2.2, one also has not understood the sentences.

(2.2) *T* Corrosion caused intermittent electrical contact.

H Corrosion prevented continuous electrical contact.

$T \Rightarrow H$

Textual entailment can therefore be seen as a task-independent formulation of certain aspects of the semantics of language. By focusing a challenge solely on these aspects, gained insights about entailment and inference become more clear and less interweaved with other issues of the task at hand. As a side effect, the competition allows – to a certain extent – the comparison of systems built for different tasks and the evaluation of approaches used in different systems. The actual relevance of textual entailment for different (end-user) applications is described by Dagan and Glickman (2004):

> A QA [question answering] system has to identify texts that entail the expected answer. For example, given the question "Who killed Kennedy?", the text "the assassination of Kennedy by Oswald" entails the sentential hypothesis "Oswald killed Kennedy", and therefore constitutes an answer. Similarly, in IR [information retrieval] the concepts denoted by a (non-sentential) query expression should be entailed from relevant retrieved documents. In multi document summarization[,] a redundant sentence or expression, to be omitted from the summary, should be entailed from other expressions in the summary. In IE [information extraction,] entailment holds between different text variants that express the same relation. And in reference resolution the antecedent typically entails the referring expression (e.g. *IBM* and *company*).

Approaches to these tasks all have to deal with the phenomenon of entailment. But, as Dagan et al. (2005) argue, there was no single, generic evaluation framework where different inference methods could be compared. This is exactly what textual entailment provides.

Several interesting points concerning textual entailment already have been observed and are discussed in the literature. One of them is the role of linguistic and extra-linguistic knowledge in textual entailment.

2.2. Levels of Knowledge

In order to decide if textual entailment holds, a system needs to have access to very different levels of knowledge. In the following paragraphs, we will look at a few examples and characterize the necessary knowledge. We are, however, aware of the fact that a precise definition is extremely difficult. It is not provided or even attempted here. We assume a huge intermediate area, where it is extremely difficult or even impossible to determine the "category" of a given piece of knowledge.

(2.3) *T* Bill murdered John.

H Bill killed John.

$T \Rightarrow H$

In example 2.3, it is sufficient to know that *to murder* is more special than *to kill.* This knowledge is considered to be included in the lexicon of a language and can therefore be called **lexical knowledge**. There is clearly no additional, e.g., legal knowledge or knowledge about Bill and John involved. Practically, this lexical knowledge is included in most dictionaries.

(2.4) *T* Deadly Nightshade is one of the most toxic plants found in the Western hemisphere.

H Belladonna is one of the most toxic plants found in the Western hemisphere.

$T \Rightarrow H$

The entailment relation between the sentences in example 2.4 is caused by the fact that *Deadly Nightshade* and *Belladonna* are synonyms. Crouch et al. (2003) argue that this is "a failure of botanical knowledge, not a lapse in language understanding" (p. 2). This piece of knowledge does not belong to the general lexicon of English, but rather to a domain-specific terminology. A speaker who does not detect the entailment relation would not be considered to have a poor understanding of the English language, but to lack some knowledge in the field of botany.

On a technical level, one might imagine a domain-specific lexicon for this example. It is still lexical knowledge, but **lexical domain knowledge** or ontological knowledge (the term *ontology* will be introduced later in this chapter).

(2.5) *T* Peter brings his car to the garage for repair.

H Peter's car is damaged.

$T \Rightarrow H$

In order to decide if textual entailment holds in example 2.5, one needs the knowledge that if something is repaired, it usually has been damaged or even broken earlier. This knowledge can be considered as a special part of the lexical knowledge, because it reflects a property of the word *repair*. In linguistic literature, this is known as **presupposition**. Sentence T in 2.5 presupposes that the car is damaged (for details about presuppositions, see, for instance, Gazdar (1979)). Even though presuppositions are considered to be part of the lexical knowledge of a language, most dictionaries do not contain this kind of knowledge.

(2.6) *T* Wal-Mart is being sued by a number of its female employees who claim they were kept out of jobs in management because they were women.

H Wal-Mart is sued for sexual discrimination.

$$T \Rightarrow H$$

More involved examples can be found in the RTE task. In example 2.6, legal knowledge as well as general knowledge about the world is needed in order to decide the entailment. One needs to know that *keeping female employees out of jobs in management* is (sexual) discrimination and that the female employees are the people suing Wal-Mart. One needs to know something about the world, especially including a deep understanding of the society or legal system. This goes beyond what can be found in currently available knowledge resources.

2.3. Resources

In the previous section, we observed the need for knowledge in order to detect entailment various RTE examples. We will now look at several resources that might provide such knowledge in a formalized form. We will also look at the "usability" of the resources, i.e., how such a resource can be embedded in an RTE system and how to connect its knowledge with the natural language sentences in RTE.

Resources containing lexical-semantic information (like presuppositions) are being developed and could be used for textual entailment. Many such resources are formulated as **ontologies**. Ontologies are formal descriptions of a set of concepts or classes and relations between these concepts. The most important relation in ontologies is usually the IS-A relation. A statement like "Car is-a Vehicle" asserts that every car is also a vehicle. The opposite does not hold, not every vehicle is a car. This relation is also called "subclass", because the set of cars is a subset of the set of vehicles. Ontologies also describe single objects, called "instances" of classes. While the concept or class "Car" denotes all cars, an individual car is called an instance of the class "Car".

So-called **upper model ontologies** describe general, cross-domain concepts. Three such upper model ontologies are Cyc, Dolce and SUMO.

Cyc Cyc (Lenat and Guha, 1989) is a very large knowledge resource. It contains more than 300,000 concepts and over 3 million assertions that use more than 26,000 relations. Cyc is not only an upper-model ontology, but includes a wide range of special ontologies for specific domains. Unfortunately, the Cyc ontology is a commercial product and was not available for research purposes until July 2006.

Dolce The DOLCE ontology ("Descriptive Ontology for Linguistic and Cognitive Engineering", Gangemi et al. (2002)) is the basic module of the WonderWeb Foundational Ontologies Library[1]. DOLCE is aimed at being a foundation for use in applications of the semantic web. A part of DOLCE (called DOLCE LITE) has been linked to WordNet, a large English dictionary (we will discuss WordNet in section 3.1). However, DOLCE is an extremely fine-grained resource and it is doubtful whether it is feasible to use it for inference resolution in textual entailment.

[1] http://wonderweb.semanticweb.org/

SUMO SUMO (Niles and Pease, 2001) stands for "Suggested Upper Merged Ontology". The SUMO ontology has been merged from a dozen different ontologies and is freely available[2]. 692 classes are contained in SUMO, but several domain ontologies exist that can be used in combination with SUMO. These ontologies cover domains like communication, economy, airports, transportation, ... SUMO also contains linkings to all of WordNet, i.e., every synset of WordNet 3.0 is linked to a SUMO concept (Synsets are the basic concepts in WordNet, see section 3.1 for details). We will discuss SUMO in detail in section 3.3.

Using SUMO, one could address some of the issues pointed out above (like presuppositions). However, in order to use SUMO, one needs to link the constituents in the sentences from a textual entailment item to the appropriate SUMO concepts. While we can link single words to SUMO concepts with the WordNet-SUMO-linking, we need to preserve or transform the semantic relations between the constituents, i.e., the predicate argument structure.

In example 2.5, for instance, it does not help to map the string *Peter* to the SUMO class HUMAN, the string *car* to the class CAR etc. We need to represent the information, that it is Peter's car, which is damaged. Missing at this point is an interface between the syntax (be it the surface form (the strings) or some syntactical representation of it) and the semantics in the form of a SUMO representation.

There are several resources available that already deal with that gap. All of them provide semantic information for sentences, including a formal representation of the predicate argument structure. Three well known resources are PropBank, VerbNet and FrameNet.

PropBank PropBank (Kingsbury and Palmer, 2003) is based on the Penn TreeBank, a large resource containing syntactic analyses of newspaper texts (Marcus et al., 1993). PropBank adds a layer of predicate argument structures to the syntactic annotation, labeling the semantic arguments with names like *Arg0*, *Arg1*, ... Even though the predicates in PropBank are grouped according to their valency, PropBank does not include a hierarchical structure of the predicates. Thus, an approach relying on such a hierarchy is not applicable with PropBank.

VerbNet VerbNet (Kipper et al., 2000) uses Levin classes (Levin, 1993) as basic structure of the resource. Levin classes provide a classification of English verbs according to various syntactic and semantic properties. However, a hierarchy of verb classes is also not provided.

FrameNet FrameNet (Baker et al., 1998) defines so-called frames that describe prototypical situations. The frames contain several lexical units thus allowing abstraction over single words. The frames also contain so-called frame elements. A frame element describes a semantic role, i.e., a participant of the situation. Frames are sorted in a hierarchy according to their specificity. We will discuss FrameNet in section 3.2.1 in detail.

[2] http://www.ontologyportal.org/

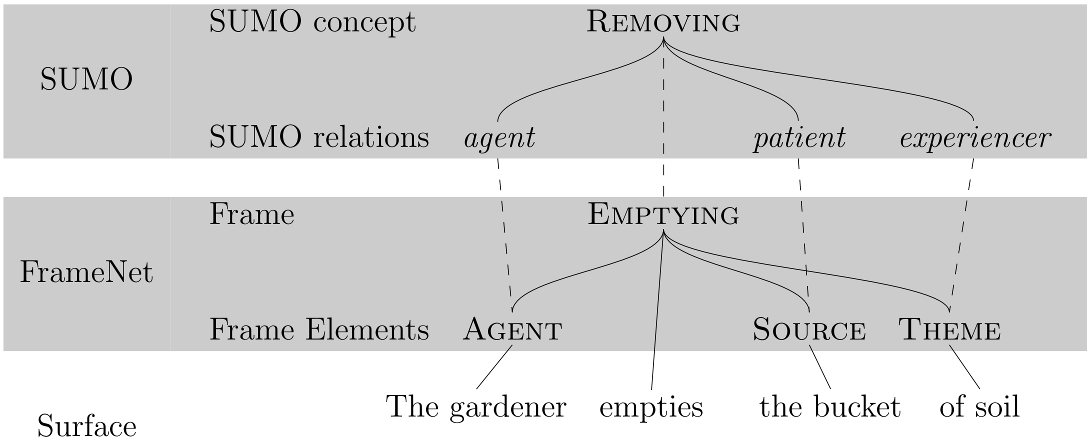

Figure 2.1.: Example SUMO and FrameNet analysis of the sentence *The gardener empties the bucket of soil*

In this diploma thesis, we use FrameNet as a syntax semantics interface. The tools that are available for FrameNet can be used to assign a FrameNet analysis consisting of frames and frame elements to free text. How to link these frames and frame elements to appropriate SUMO concepts and relations is the question we will discuss in this thesis.

In figure 2.1, we show a FrameNet analysis for the sentence *The gardener empties the bucket of soil.* The FrameNet analysis, consisting of the frame EMPTYING and the frame elements AGENT, SOURCE and THEME, can be generated automatically by using several already existing tools and methods, which we will discuss later. Figure 2.1 also shows the corresponding parts of a possible SUMO representation for the sentence. This diploma thesis aims at linking the FrameNet analysis to SUMO representation, shown by dashed lines in the figure. We will explore an algorithm that provides the basis for the automatic assignment of a SUMO analysis to a FrameNet analysis of a sentence.

2.4. Outline

In this section, we will give a brief overview on the following chapters, where we present not only the resources used in detail, but also several tools and methods used in conjunction with the resources and, of course, our own contribution.

WordNet, a large-scale dictionary for the English language, will be described in section 3.1. Section 3.2 covers the resource FrameNet as well as the techniques, tools and methods used with FrameNet.

In section 3.2.2, we will present an XML format to represent FrameNet annotation. SALSA/TIGER XML (Erk and Pado, 2004), as it is called, has been developed in the SALSA project (Erk et al., 2003). The format extends another file format used to store syntactic annotation by adding a layer of FrameNet specific information.

For the task of semantic parsing, i.e., assigning frames and frame elements to free text, we present two different approaches that can be seen as complementary. A rule-based semantic parser (Shi and Mihalcea, 2004), which extracts syntactic patterns from the annotations done in the FrameNet project and a statistical parser called Shalmaneser (Erk and Pado, 2006). Shalmaneser is a relatively open tool-chain, such that single components can be replaced or added easily. We will present both approaches to semantic parsing in section 3.2.3.

A problem of FrameNet is its low coverage. In order to address this issue, we present a system that uses WordNet as a detour and is thus possible to assign frames to words that are not included in FrameNet (Burchardt et al., 2005). The system called Detour will be presented in section 3.2.4.

The SUMO ontology is described in detail in section 3.3, followed by a description of the linking of WordNet and SUMO (Niles and Pease, 2003).

Our own contribution consists of developing and implementing an algorithm that links the frames and frame elements to appropriate SUMO concepts. We will describe the assignment of SUMO concepts to FrameNet frames in section 4.1. Our approach assigns a weighted list of SUMO concepts to each frame. The highest weighted concept is then selected as the concept corresponding to the frame.

The linking of frame elements to SUMO relations is described in section 4.2. The basic idea for this linking algorithm is to manually link only a small part of the frame elements to appropriate SUMO roles and to use the FrameNet inheritance hierarchy to extend the linking to more frame elements.

We will discuss the evaluation of both algorithms in chapter 5. As no SUMO-annotated corpus is available, we have to provide our own "gold standard" by grouping a sample of results manually into different categories, which reflect the appropriateness of a linking.

3. State of the Art

In this chapter, we will discuss the resources we will use in detail, together with several related methods, tools and algorithms. In section 3.1, we will discuss WordNet, section 3.2 discusses FrameNet (and related word) and section 3.3 will discuss the SUMO ontology.

3.1. WordNet

WordNet (Fellbaum, 1998) is a large scale lexicon. Version 3.0, which is used throughout this work unless indicated otherwise, contains over 155,000 words, the majority of them being nouns (roughly 117,000).

Synsets and Synset relations

The words are grouped according to *synonymy*. In WordNet, the following definition of synonymy is used:

(3.1) The words w and v are synonym, if they are interchangeable in some context without changing the truth value of the proposition in which they are embedded.

The word *police*, for instance, can in some contexts be replaced by the word *law*, without changing the truth value of the sentence:

(3.2) The police came looking for him.

(3.3) The law came looking for him.

Therefore, *police* and *law* are considered to be synonyms and included in one such group. These groups are called "synsets" and are the basic concept in WordNet.

Ambiguous words (words that can be used with more than one meaning) are part of more than one synset. *Law* in the phrase *the laws of thermodynamics* can not be replaced with *police* without changing the truth value of the proposition (or without making it absurd). Therefore, *law* occurs in one synset together with *police* and in another synset together with *law of nature*. For the word *law*, WordNet lists seven different synsets, each of them representing a different sense of the word. The synsets are enumerated and ranked according to the frequency by which each sense is used.

Throughout this work, the notation `word#pos#rank` is used to represent a synset. The synset containing *law* and *police* is represented by `law#n#7` (or `police#n#1`), while the synset containing *law* and *law of nature* is represented by `law#n#4`.

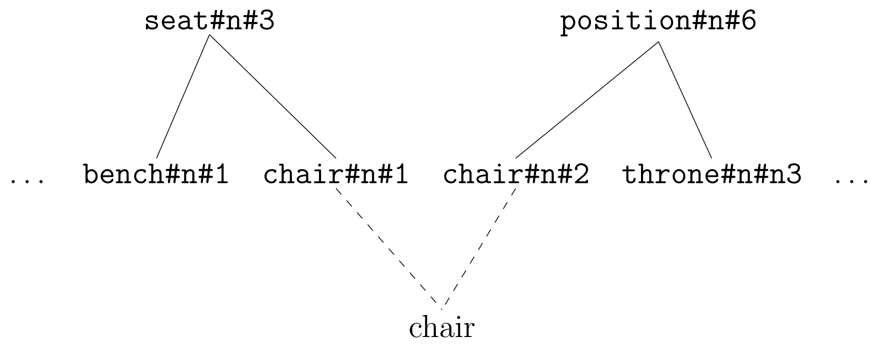

Figure 3.1.: An excerpt of the WordNet hierarchy

A gloss, a natural language description and often an example sentence, is also provided for the synsets. For the synset `law#n#4`, the gloss is *(a generalization that describes recurring facts or events in nature) "the laws of thermodynamics"*.

The WordNet hierarchy is formed by the hypernymy and troponymy relations[1]. They allow concepts to be set in relation with more general (or special) other concepts. Figure 3.1 shows an excerpt of the WordNet hierarchy as an example. The word *chair* belongs (among others) to the synsets `chair#n#1`[2] and `chair#n#2`[3]. The former is a hyponym (a more special concept) of the synset `seat#n#3`[4], while the latter is a hyponym of `position#n#6`[5].

The importance of WordNet as a resource is mainly due to its large coverage of the English language. It can be used as lexicon on a large scale that provides an approximate semantic characterization and as a reference for linking resources. There are also several ways in which semantic similarity can be measured using the WordNet hierarchy.

WordNet versions for other languages have been developed, for instance GermaNet for German (Hamp and Feldweg, 1997).

3.2. The FrameNet Environment

In this section, the FrameNet resource is introduced and several tools and algorithms for accessing and using it are described. The FrameNet database and corpus are described in section 3.2.1. Section 3.2.2 describes an XML-format to store annotated FrameNet data. Section 3.2.3 describes two different approaches for semantic role assignment and section 3.2.4 describes a system developed to deal with the coverage limitations of FrameNet.

[1] A number of other relations such as entails or antonymy further relate synsets or words, but since they are not used in this work, we will disregard them.

[2] Gloss: *(a seat for one person, with a support for the back) "he put his coat over the back of the chair and sat down"*

[3] Gloss: *(the position of professor) "he was awarded an endowed chair in economics"*

[4] Gloss: *(furniture that is designed for sitting on) "there were not enough seats for all the guests"*

[5] Gloss: *(a job in an organization) "he occupied a post in the treasury"*

3.2.1. FrameNet

The Berkeley FrameNet project (Baker et al., 1998) started in 1997 and is still ongoing. Its goal is to annotate a part of the British National Corpus (BNC) with a layer of semantic information. The semantic information is encoded in frames according to Fillmore's frame semantics (Fillmore, 1976). A frame represents a prototypical situation together with its participants. For each frame, a set of these participants (called *frame elements*) and a set of *lexical units* are defined in the FrameNet database. A lexical unit is usually a single verb (*avenge*), a phrasal verb (*give in*), an idiom (*kick the bucket*) or a noun (*lecture* in *Pat gave her a lecture*). We say that a lexical unit occurring in a sentence "evokes" the respective frame, the lexical unit in the sentence is then called "target" of the frame. Lexical units are not disambiguated with respect to WordNet.

The frame SELF_MOTION, for instance, is defined as when a "living being moves under its own power in a directed fashion" (documentation). 22 frame elements belong to this frame, from which six are considered to be central for the frame ("core frame elements", see below): SELF_MOVER, DIRECTION, GOAL, PATH, AREA and SOURCE. *hurry.v*, *swim.n* and *head.v* belong to its lexical units. Example 3.4 shows a FrameNet annotated sentence. The target lexical unit of the frame SELF_MOTION has been underlined in the text. Frame elements are marked with square brackets.

(3.4) [Therese]$_{\text{SELF\_MOVER}}$ <u>climbed</u> [down]$_{\text{PATH}}$ [from the stage]$_{\text{SOURCE}}$. (SELF_MOTION)

FrameNet 1.3 (which is used in this work) contains 795 frames and more than 10,000 lexical units. The following description is mostly based on Ruppenhofer et al. (2006).

Frame Relations

The frames in FrameNet are connected by frame relations. FrameNet 1.3 defines eight types of frame relations, which connect frames in specific ways:

Inheritance The inheritance relation corresponds to "is-a" relations known from ontologies. Frame X inherits from Y, if "anything which is strictly true about the semantics of [Y corresponds] to an equally or more specific fact of X" (Ruppenhofer et al., 2006, p. 104). We will use the term "root frame" for a frame that does not inherit from any other frame.

Perspective_on The Perspective_on relation (which was newly introduced in FrameNet 1.3) captures perspectivation in some frames. The frame MEASURE_SCENARIO, for instance, can describe both exact and relative measures ("John weights 7 pounds" vs. "John is heavy"). MEASURE_SCENARIO has therefore Perspective_on-relations to (among others) the two frames DIMENSION and QUANTITY.

Subframe Subframe relations are used to model complex scenes. A single frame may represent a whole sequence of events, each of which represented by a single frame. The frame CRIMINAL_PROCESS, for instance, has the four subframes ARRAIGNMENT, ARREST, SENTENCING and TRIAL. CRIMINAL_PROCESS itself is a subframe of the frame CRIME_SCENARIO.

Precedes Temporal precedence can be encoded between the different subframes of a complex frame. An ARRAIGNMENT usually follows an ARREST and is followed by a TRIAL.

Inchoative_of See below.

Causative_of Inchoative_of and Causative_of are used to mark the lexical aspect (Aktionsart) of verbs. CAUSE_CHANGE_POSITION_ON_A_SCALE, for instance is in Causative_of-relation with CHANGE_POSITION_ON_A_SCALE because it describes the cause of the change.

Using Several frames make references of some kind to more abstract other frames. This reference is marked by the Using-relation. The frame JUDGMENT_COMMUNICATION, for instance, uses the frames JUDGMENT and STATEMENT.

See_also The See_also relation is meant as a help for human readers. Frames that are somehow grouped together, should be seen in contrast to each other or carefully differentiated are related using See_also.

Frame Elements

The frame elements are defined per-frame and represent the participants of the situation described by the frame. The frame COMMITTING_CRIME, for instance, defines ten frame elements: PERPETRATOR, CRIME, PURPOSE, MANNER, MEANS, REASON, TIME, PLACE, INSTRUMENT and FREQUENCY, each of which representing a semantic role that occurs in real-life sentences describing a scene where a crime is committed.

A few properties are defined for the frame elements:

Coreness Each frame element has an annotated so-called coreness. This property of the frame element expresses how central the element is for that specific frame. FrameNet 1.3 distinguishes four different levels of coreness:

Core A core frame element is central for that frame. The core frame elements are necessary components in every situation described by this frame. The PERPETRATOR is a necessary component of the COMMITTING_CRIME frame: A committed crime necessarily implies that there is someone who committed it.

Peripheral Peripheral frame elements are not unique to the frame in question. Leaving out or adding a peripheral frame element does not change the event itself. COMMITTING_CRIME has six peripheral frame elements: PURPOSE, MANNER, MEANS, REASON, TIME and PLACE. In most of the situations described by the frames in FrameNet, one or all of these frame elements can be expressed.

Extra-Thematic Frame elements marked as extra-thematic conceptually do not belong to the frame they appear in. They are frame elements of other, usually more abstract frames. In COMMITTING_CRIME, the frame elements INSTRUMENT and FREQUENCY are extra-thematic.

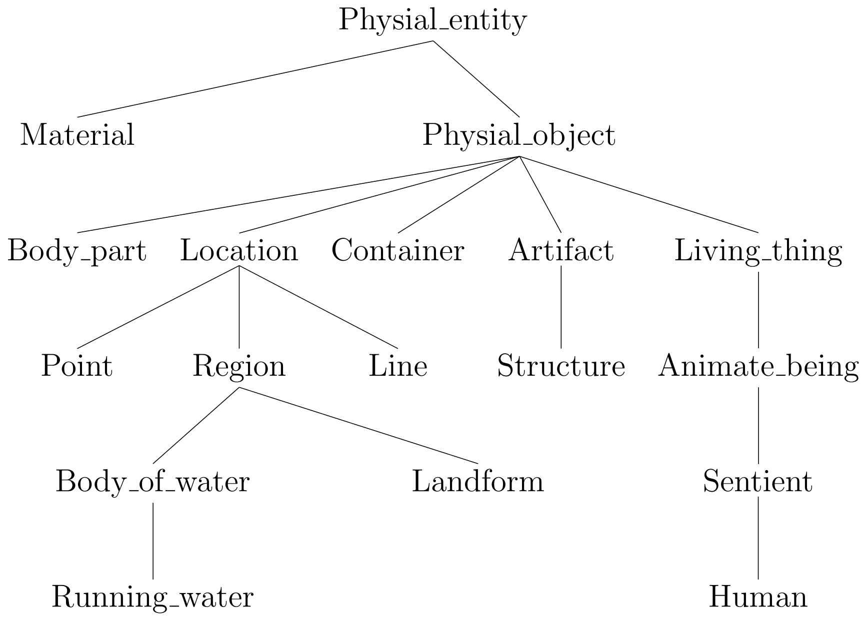

Figure 3.2.: The hierarchy of semantic types in FrameNet

Core-Unexpressed Core-unexpressed frame elements behave like core frame elements with the exception, that in inheriting, more special frames, this frame element might not be expressed. The frame element CRIME is marked as core-unexpressed, because in inheriting frames like THEFT, the crime itself is already covered by the lexical unit that evokes the frame.

Semantic Type Some of the frame elements are marked with a semantic type. The semantic types are types for the *filler* of a frame element. They are structured in a small ontology (see figure 3.2 for an excerpt).

Even if FrameNet does not provide a formally defined and complete mapping to an ontology, the design of the ontological types follows the principles of ontology building. Moreover, most of the ontological types are defined as equivalent to certain WordNet synsets. A mapping of these semantic types to concepts of the SUMO ontology by Scheffczyk et al. (2006) is briefly discussed in section 4.

Hierarchy The FrameNet frame relations do not only link frames, but also the respective frame elements. Every frame element of a specific frame has an identical or more special corresponding frame element in each more special frame (with the exception of core-unexpressed frame elements). Because frame elements are defined per frame, more specialized relatives can be named differently. Figure 3.3 shows an example. The frame element SELF_MOVER of the frame SELF_MOTION corresponds to the frame element AGENT in the frame INTENTIONALLY_ACT.

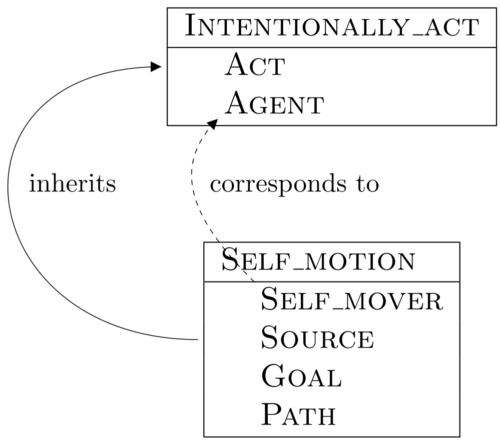

Figure 3.3.: Inheritance of frame elements

Obviously, some frames introduce entirely new frame elements which are not present in more general frames. SELF_MOTION, for instance, defines a frame element SOURCE, which simply does not make sense in a frame like INTENTIONALLY_ACT. Therefore, not every frame element has a corresponding frame element in a more general frame, even if the frame to which the frame element belongs has a has a more general frame (see figure 3.3). The coreness of a frame element is not passed along through the hierarchy. A frame element can be defined as core in one and as peripheral in an inheriting frame.

Grammatical Function FrameNet also includes an annotated corpus. The corpus consists of more than 135,000 sentences, which are annotated with frames and frame elements as well as grammatical function, i.e., a clearly syntactical feature. The following six different grammatical functions are used.

Ext External arguments are constituents that are outside of the maximal phrase headed by the target word. If the target word is a finite verb, this is mostly the subject. For adjectives in clausal predication and prepositions with copular predications, the subject of the copula verb is marked as external argument. For prepositions used as post-nominal modifiers, the noun is marked as external argument. Example: *[The physician]*$_{\text{Ext}}$ *performed the surgery.*

Dep The grammatical function "Dependent" is used for all adverbs, PPs, VPs and a small number of NPs that occur after governing verbs, adjectives, nouns or prepositions in declarative sentences. The constituents that are usually referred to as arguments and adjuncts are both included in this grammatical function. Example: *They want [to stay home]*$_{\text{Dep}}$.

Obj Any normal object, any wh-extracted object or any post-target-verb that controls the subject of a complement of the target verb. Example: *Voters approved [the stadium measure]*$_{\text{Obj}}$.

Head In sentences, where a pre-nominal adjective has a qualitative instead of a relational use, the noun, that is modified by the adjective, is marked with this grammatical function. This applies for phrases like *the small [children]$_{\text{Head}}$*.

Gen A possessive nominal phrase functioning as a determiner of a target noun is annotated as genitive determiner. Example: *[your]$_{\text{Gen}}$ book*.

Quant Pre-nominal determiners of target nouns that function as a number, are annotated as quantificational determiners. Example: *[two]$_{\text{Quant}}$ cups of coffee.*

Appos Post-target appositional named entities or nominal phrases are marked as appositives. Example: *Actor Robert Downey Jr. will walk down the aisle next year with girlfriend [Susan Levin]$_{\text{Appos}}$*.

3.2.2. SALSA/TIGER XML

SALSA/TIGER XML (Erk and Pado, 2004) is an XML format used to represent semantically annotated sentences in the SALSA project (Erk et al., 2003). The SALSA project – among other things – annotates the German Tiger Corpus (Brants et al., 2002) with FrameNet frames. SALSA/TIGER XML format consists of both a syntactic and semantic layer, which are conceptually separated. The syntactic layer was developed for the Tiger project and is described in Mengel and Lezius (2000).

The semantic layer stores frame and frame element assignments on the basis of the syntactic annotation. More than one frame can occur in a sentence, and a single node can be part of more than one annotated frame, either as target or as frame element.

(3.5) [Traffickers]$_{\text{SPEAKER}}$ demand [astronomical amounts]$_{\text{MESSAGE}}$ to smuggle their customers to the West. (REQUEST)

Figure 3.4 shows a graphical representation of the sentence shown in 3.5. As one can see, the graph includes syntactic (round labels) as well as semantic features (angled labels): The verb *demand* is part of the main VP and it evokes the frame REQUEST, for instance.

3.2.3. Semantic Parsing

In this section, we describe two different approaches to semantic parsing. Semantic parsing, in this case, is the process of assigning frames and frame elements to free text automatically. The two approaches described here – a statistical and a rule-based one – have been developed independently and can be seen as competitive. We present them here, because semantically parsed text is the basis for the use of the linking to SUMO that we present in chapter 4 in an application.

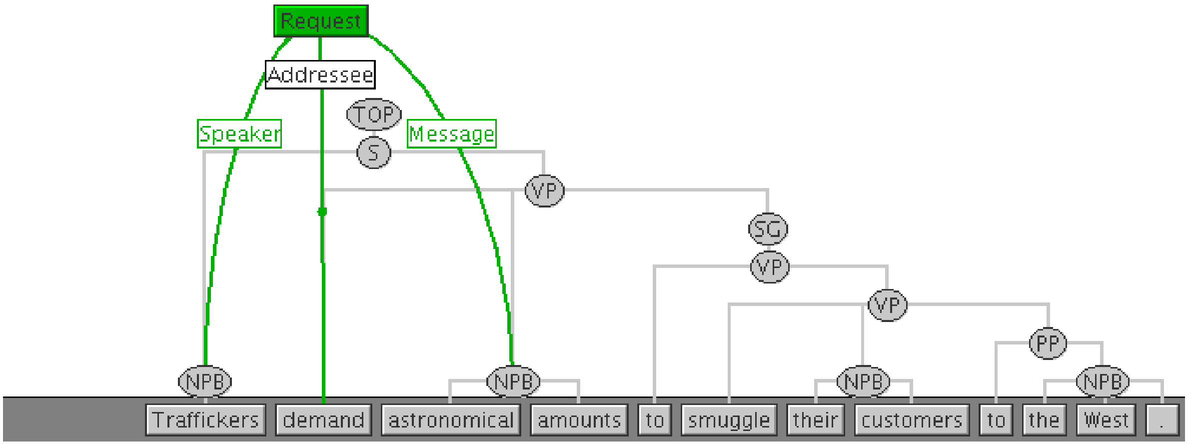

Figure 3.4.: Graphical representation of the annotation of sentence 3.5.

A Shallow Semantic Parser – Shalmaneser

Shalmaneser (Erk and Pado, 2006) is a tool chain of loosely connected modules with SALSA/TIGER XML as a common interface format. From start to end, the chain is able to assign frames to free text, identify the targets of the frames and the fillers of the frame elements.

The tool chain consists of three parts:

Pre-processing. In this step, the incoming text is tagged with parts of speech, lemmatized and parsed syntactically. The output of the parser is converted to SALSA/TIGER XML. The linguistic processing in this step is done by external tools, varying for different languages: Currently, the Collins (Collins, 1997) and Minipar (Lin, 1993) parsers are supported for English and the Sleepy parser (Dubey, 2005) for German. Lemmatization for English and German can be done with the TreeTagger (Schmid, 1994) and part-of-speech tagging with the TnT-tagger (Brants, 2000).

Frame assignment. In the second step, the constituents of the sentence are disambiguated and, if appropriate, one or more frame is assigned to them. For the disambiguation, a Naïve Bayes classifier is used. The English version is trained on the original FrameNet corpus, the German version is trained on the SALSA corpus (Erk et al., 2003). The features used in this classification include bag-of-words contexts, bi- and trigrams, grammatical functions and the target voice for verbs.

Role Assignment. The role assignment is done in two steps, the *argument recognition* and the *argument labeling*. Argument recognition tries to mark each syntactic constituent as role or non-role. Argument labeling identifies the role the constituent bears. Both parts work statistically and are based on around 30 features. As in the pre-processing step, the actual machine learning component can easily be replaced. Currently, Mallot (McCallum, 2002), TiMBL (Daelemans et al., 2003) and Malouf (Malouf, 2002) are supported.

Evaluation

Shalmaneser was evaluated for English and German. The frame assignment component achieved an overall accuracy of 0.932 for English and 0.79 for German. This difference is due to the different annotation modes in FrameNet (English) and SALSA (German): In the SALSA corpus, there are on average twice as many senses per lemma than in the FrameNet corpus.

The role assignment component was evaluated for both its sub tasks. Argument recognition achieved an f-measure of 0.751 for English and 0.6 for German. Argument labeling achieved 0.784 resp. 0.673. According to Erk et al. (2003), the lower results on German text reflect the smaller training set.

A rule-based Semantic Parser

Shi and Mihalcea (2004) present a system for rule-based semantic parsing. It uses FrameNet and WordNet as knowledge bases and is able to assign frames and frame elements to open (not preprocessed) text. In a first step (which is done only once and in advance), sentence- and word-level knowledge is extracted from the resources. The knowledge is then encoded in rules, which are used later on to decide frame and role assignments.

Knowledge Bases

Sentence Level Knowledge Patterns of the syntactic realizations of the frame elements are extracted from the FrameNet corpus and provide sentence-level knowledge. The patterns are built on syntactic features encoded in FrameNet (like grammatical functions) as well as features which are newly extracted from the sentence context (the relative position, the voice of the sentence and the preposition if it is a prepositional phrase).

Together with the semantic role that the constituent is assigned to, these patterns are listed in the same order as they appear in the sentence. The sentence 3.6, for instance, provides the rule shown in 3.7.

(3.6) $[\mathrm{I}]_{\mathrm{THEME}}$ had chased $[\mathrm{Selden}]_{\mathrm{GOAL}}$ $[\text{over the moor}]_{\mathrm{PATH}}$ (COTHEME)

(3.7)
```
[active, [ext,np,before,theme],
         [obj,np,after,goal],
         [comp,pp,after,over,path]]
```

All rules for all annotated sentences are extracted in this way and stored together with the respective frame name, the target word and its syntactic category.

Word Level Knowledge Word level knowledge is extracted from WordNet and contains the part-of-speech and further information with respect to the part-of-speech. An example entry for the adjective *slow* is shown in 3.8:

(3.8)
```
lex(slow,W) :- W = [parse:slow, cat:adj, attr:speed,
                    degree:base, type:descriptive]
```

The Semantic Parser

The actual parsing process is divided in three steps: (1) syntactic analysis with a feature-augmented syntactic parser, (2) assignment of semantic roles and (3) application of default rules.

Syntactic Analysis The feature augmented grammar identifies the target word(s), its arguments and groups the constituents of the sentence. These entities are marked with word level semantics (e.g., attribute, gender) and several shallow semantic features (e.g., modifier types). The analyzer is also able to detect ungrammatical sentences and to rule out sentences, which are grammatically or semantically incorrect[6].

For correct sentences, the analyzer returns a list of features for every constituent. For sentence 3.9, for instance, the analyzer returns the feature list shown in 3.10.

(3.9) I come here by train.[7]

(3.10)
```
[[ext,np,before,active],
 [obj,np,after,active],
 [comp,pp,after,active,by]]
```

Semantic Role Assignment According to the target word, which is often the verb or a predicative adjective, a set of rules that can be used is selected. A scoring algorithm that basically counts the number of matches between an analyzed sentence and a rule is used to select the best applicable rule.

[6] *The technology is very military*, for instance, is ruled out because *military* is not a descriptive adjective and therefore can not be connected to the degree modifier *very*.

[7] This example is taken from (Shi and Mihalcea, 2004).

```
(3.11) [[ext,np,before,active,theme],
        [obj,np,after,active,goal],
        [comp,pp,after,active,by,mode_of_transportation]]
```

```
(3.12) [[ext,np,before,active,theme],
        [obj,np,after,active,goal],
        [comp,pp,after,active,from,source]]
```

Consider, for instance, the rules in 3.11 and 3.12. The scoring function is applied to the sentence representation in 3.10. The scoring component compares all features but the last in the rules – which is the semantic role – and counts the matches. In this case, rule 3.11 gets 3, while rule 3.12 gets only 2 matches. Therefore, rule 3.11 is selected for the role assignment and the sentence is annotated as shown in 3.13.

(3.13) $[\text{I}]_{\text{Theme}}$ come $[\text{here}]_{\text{Goal}}$ $[\text{by train}]_{\text{Mode\_of\_transportation}}$. (Arriving)

Default Rules Default rules are used to handle the interpretation of constituents which are not covered in FrameNet. In sentence 3.14, for instance, the constituent *on the street* is not assigned a frame element from FrameNet, because it does not occur in the FrameNet corpus.

(3.14) I walk on the street.

One of about 100 manually defined rules states that *on something* modifies the location of an interaction and is used here. However, these rules are only applied if no other rule could be used in the previous steps.

Evaluation

Shi and Mihalcea (2004) evaluated their system on 350 randomly selected sentences from the FrameNet corpus. The test sentences were removed from the corpus before invoking the knowledge extraction procedure. Using this system for both frames and frame elements, 74.5% of the correctly identified elements were assigned to the correct role.

3.2.4. Dealing with Coverage Limitations of FrameNet – The Detour

Due to the fact that FrameNet is a relatively young resource, its coverage is limited. Burchardt et al. (2005) propose a way to circumvent this limitation by using WordNet as a detour to FrameNet. The algorithm has been implemented in Perl and is freely available at the Comprehensive Perl Archive Network (CPAN)[8].

[8] `FrameNet::WordNet::Detour`

The basic algorithm is designed to map a WordNet synset onto one or more appropriate FrameNet frames. As it uses WordNet, it is not dependent on a member of the input synset being annotated in FrameNet. Instead, the algorithm collects hyper-, hypo- and antonyms of the input synset and tries to find them in FrameNet. The resulting frames are weighted according to the similarity of the evoking synsets with the original synset. The weights of the frames are normalized, so that they are comparable over different runs and among different input synsets.

According to Reiter (2007), who did a detailed evaluation, the results of the Detour are as good as a gold standard in 70% of a randomly selected test set. The comparison ("as good as") was done manually on a subset of the test set. As gold standard, the linking of lexical units to WordNet synsets provided by Shi and Mihalcea (2005) was used (see section 4.1).

3.3. The SUMO Ontology

3.3.1. Classes and Axioms

The SUMO upper ontology ("Suggested Upper Merged Ontology", Niles and Pease (2001)) is discussed and developed since 2000. It is aimed at being standardized by the IEEE[9]. SUMO has been merged from a dozen freely available ontologies and is defined in a lisp-like format called KIF (Pease, 2004). KIF's expressivity equals that of first-order logic. The general part of SUMO[10] contains 1,116 terms, 692 of them are classes.

The classes are structured as a hierarchy using a *subclass*-relation. The class Guiding, for instance, is a subclass of the class IntentionalProcess. Other subclasses of IntentionalProcess are Keeping, Looking, Making, ... The subclass-relation in SUMO is defined as reflexive, i.e., every class is a subclass of itself. Classes can be instantiated using the *instance*-relation. An instance of the class Looking, for example, would then be the representation of a concrete event of that type. A concrete car, for example, would be represented by an instance of the class Automobile.

Apart from the simple class hierarchy, SUMO contains relations between events and participants like *agent*, *patient* or *destination*. In contrast to FrameNet, these relations can be used anywhere in SUMO, i.e., they are not bound to a specific class. Instead, there exist axioms which in a way define what relations are used in which classes.

Axioms

As can be seen in listing 3.1, KIF uses prefix notation, i.e., the first element in parentheses denotes the relation, the following elements denote the arguments. Variables in KIF are marked with a question mark. The listing shows an axiom for the class IntentionalProcess. It states, that every instance of that class is in *agent*-relation with

[9] Institute of Electrical and Electronics Engineers
[10] Merge.kif, version 1.75, which is the one used in this work.

Listing 3.1: An axiom of IntentionalProcess

```
(=>
    (instance ?PROC IntentionalProcess)
    (exists (?AGENT)
        (and
            (instance ?AGENT CognitiveAgent)
            (agent ?PROC ?AGENT))))
```

Listing 3.2: An axiom for Repairing

```
(=>
  (and
    (instance ?REPAIR Repairing)
    (patient ?REPAIR ?OBJ))
  (exists (?DAMAGE)
    (and
      (instance ?DAMAGE Damaging)
      (patient ?DAMAGE ?OBJ)
      (earlier
        (WhenFn ?DAMAGE)
        (WhenFn ?REPAIR)))))
```

an instance of CognitiveAgent. Even if axioms can be more complex (and include any number of logical connectives, including universal quantification), most of them resemble the one shown above. In these axioms, an event is set into relation with some participants of the event, while the relation describes the role of the participant. This of course resembles very much the notion of frame elements in FrameNet. A number of SUMO relations can be seen as describing the semantic roles of events, which is why we will call them role-like relations.

Most of these role-like relations are instances of the class CaseRole. It is defined as the "class of predicates relating the spatially distinguished parts of a Process" (documentation). The driver of a driving event, for instance, could be modeled by using an instance of CaseRole as relation. However, a number of role-like relations are not in the class CaseRole: The message of a conversation, for instance, can not be in any of the CaseRole-relations. Instead, an axiom of the class ContentBearingPhysical states the existence of a *represents*-role, which is no instance of CaseRole and apparently meant to be the relation to the message of a conversation (we will come back to this issue in section 4.2.1).

Listing 3.2 shows an axiom relevant for the class Repairing. It defines, that something that is being repaired had to be damaged earlier (we characterized this as presuppositions in chapter 2). Lines 3 and 4 state the requirements for this axiom to be applied: A Repairing event and a *patient* of this event, the thing that is being repaired. Line 5 states the existence of something else, which is a Damaging event (line 7). Line 8 defines the same object that is patient of the Repairing event to be patient of the Damaging event. Lines 9 to 11 state the temporal relation *earlier* between the two events.

3.3.2. The SUMO to WordNet linking

Niles and Pease (2003) linked the SUMO ontology to WordNet. At first, every synset from WordNet 1.6 was linked to a concept from a corresponding SUMO version. Modifications and enhancements to newer versions were carried out in the meantime. By the time of working on this thesis, a linking of WordNet 3.0 is available.

While manually[11] linking the SUMO concepts to synsets from WordNet, three different kinds of linkings were differentiated.

Synonymy A WordNet synset and a SUMO concept were marked as synonyms if they describe the same set of individuals. The synset `plant#n#2` e.g. is a synonym of the class Plant.

Hypernymy In this case, the synset is linked to a concept which denotes more entities than the synset. E.g. the synset `christian_science#n#2` is hypernymy-related to the concept ReligiousOrganization, because there is no direct counterpart for it in SUMO.

Instantiation A Synset is marked as an instance of a SUMO concept, if the entity it represents is a member of the class represented by the SUMO concept. Niles and Pease give the synset `underground_railroad#n#1` as an example. It is marked as an instance of the class Organization.

A concept from SUMO may correspond to more than one synset or vice versa. The synsets `life#n#10`[12] and `life_form#n#1`[13] are both linked to the concept Organism, because they "mean essentially the same thing". The distinction, that the first synset denotes a large number of living things while the second one refers to a living thing itself, i.e. to an instance of the class is, according to Niles and Pease (2003), not needed in knowledge engineering.

An example for a one-to-many-linking in the other direction is the synset `substitution#n#2`[14]. This synset involves both the removing and the putting of something. First, something is removed from a particular place. Second, something else is

[11]This is not stated explicitly, but can be concluded from Niles and Pease (2003).
[12]Gloss: *(living things collectively) "the oceans are teeming with life"*
[13]Gloss: *the characteristic bodily form of a mature organism*
[14]Gloss: *the act of putting one thing or person in the place of another: "he sent Smith in for Jones but the substitution came too late to help".*

put into that same place, substituting the thing that was in that place earlier. Since it is difficult to formulate the precise temporal and spatial restrictions for this substitution, the synset has been linked to both concepts Removing and Putting.

4. Linking FrameNet and SUMO

This section explains the algorithms and techniques used to provide a semi-automatic linking of FrameNet and SUMO. The linking consists of two parts: The assignment of SUMO concepts to FrameNet frames, covered in section 4.1 and the linking of frame elements to SUMO relations, covered in section 4.2. Two observations we made during the development of our algorithm are discussed in section 4.3. In section 4.4, we will briefly describe our implementation of the algorithm.

Related Work There has been a very recent approach on linking SUMO and FrameNet. Scheffczyk et al. (2006) manually linked each of the 40 semantic types of FrameNet to appropriate SUMO classes. They also investigated a half-automatic approach of assigning SUMO classes as semantic types for frame elements. This goal is complementary to our approach: Scheffczyk et al. (2006) focus on determining SUMO concepts for semantic types (see section 3.2.1) instead of the role fillers themselves. Moreover, they heavily rely on manual work.

4.1. Assigning SUMO concepts to FrameNet frames

Upper model ontologies like SUMO aim at describing general, domain-independent concepts. This includes physical objects, abstract ideas, processes and events. Especially the latter should contain prototypical situations. Because FrameNet frames represent prototypical situations, it is a reasonable assumption that all or at least most of the frames defined in FrameNet have some sort of counterpart among the concepts defined in SUMO.

Since SUMO includes a linking of SUMO concepts to WordNet synsets, WordNet can be used as an interface between SUMO and FrameNet. This linking of WordNet synsets to SUMO concepts is complete, i.e., every WordNet synset is linked to at least one SUMO concept.

Unfortunately, a formal connection between FrameNet frames and WordNet synsets does not exist. We use the lexical units defined in FrameNet to acquire such a connection automatically. For each lemma of a lexical unit in a given frame, we collect all synsets the lemma belongs to. For the lexical unit *contact.n* in the frame COMMUNICATION, for instance, we collect the nine noun senses found in WordNet for the lemma "contact".

Once the synsets of all lexical units of a frame have been collected, we use them to assign a list of SUMO concepts to the frame. This is done by looking up each synset in the WordNet-SUMO-linking and by counting how often each concept gets "triggered". We will refer to this number as *weight* of a SUMO concept. The SUMO concepts are ranked

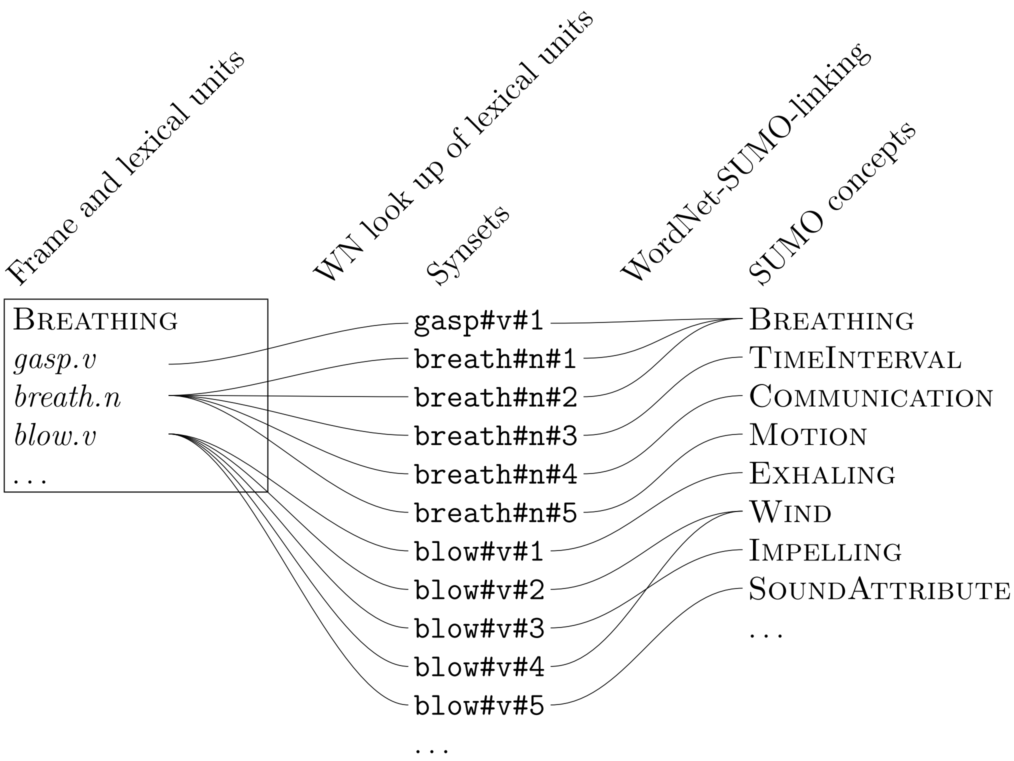

Figure 4.1.: Linking the frame BREATHING to SUMO concepts

Weight	Rank	Class
18	1	BREATHING
10	2	COMMUNICATION
3	3	*capability*, SOUNDATTRIBUTE, STATING
2	4	DESTRUCTION, ORGANISM, MOTION, TRANSFER, EXHALING, PUTTING, INTENTIONALPROCESS, SUBJECTIVEASSESSMENTATTRIBUTE, SHAPECHANGE, INTENTIONALPSYCHOLOGICALPROCESS, IMPELLING, WIND
1	5	LIQUIDMOTION, SMOKING, NORMATIVEATTRIBUTE, *attribute*, BODYMOTION, VOCALIZING, RADIATINGSOUND, PROCESS, CHEMICALSYNTHESIS, ORGANISMPROCESS, LEARNING, GIVING, LIVING, BUYING, BIOLOGICALLYACTIVESUBSTANCE, BIOLOGICALPROCESS, DEATH, EMOTIONALSTATE, POISONING, TIMEINTERVAL

Table 4.1.: Linking results for the frame BREATHING

Weight	Rank	Class
5	1	BREATHING
2	2	PUTTING
1	3	ORGANISM
1	3	SMOKING

Table 4.2.: Refined linking results for the frame BREATHING

according to their weight and the highest ranked concept is selected as the "target" concept for the given frame.

Figure 4.1 shows a graphical representation of the behavior of our algorithm for three of the twenty lexical units of the frame BREATHING. The lexical unit *gasp.v*, for instance, is not ambiguous at all. The WordNet look up for *gasp.v* relates it with `gasp#v#1`. The lexical units *breath.n* and *blow.v* are linked to multiple synsets: *breath.n* to the synsets `breath#n#1` to `breath#n#5` and *blow.v* to the synsets `blow#v#1` to `blow#v#22` (from which we show only the first five in figure 4.1).

One can see in figure 4.1 that synsets related to different lexical units are linked to the same SUMO concept, such as three synsets related to *gasp.v* and *breath.n* are linked to the SUMO concept BREATHING. The complete list of resulting SUMO concepts with their weights can be found in table 4.1.

4.1.1. Refined System

As one can see in table 4.1 as well as in figure 4.1, the results contain some noise. This noise is caused by the ambiguity of the lexical units (with respect to WordNet). Some of the lemmas of ambiguous lexical units of a frame belong to synsets which are not appropriate for this frame (`breath#n#3`, for instance, is glossed as *a short respite*, and therefore linked to the class TIMEINTERVAL).

A disambiguation of lexical units with respect to WordNet has been provided by Shi and Mihalcea (2005). They integrated FrameNet, VerbNet (Kipper et al. (2000), see section 2.2 for a brief presentation) and WordNet in order to improve both coverage and quality of the results of one semantic parser.

In order to combine the resources, Shi and Mihalcea (2005) manually labeled verbal lexical units of FrameNet frames with corresponding WordNet synsets. Due to the fine distinction between WordNet senses, most of the lexical units are linked to more than one synset. The list containing this linking has been released separately on the Internet[1]. However, it contains only the verbs and it is done with earlier versions of FrameNet (1.1) and WordNet (1.6). It is therefore no complete linking of all lexical units to WordNet synsets.

[1] `http://mira.csci.unt.edu/~spot/`

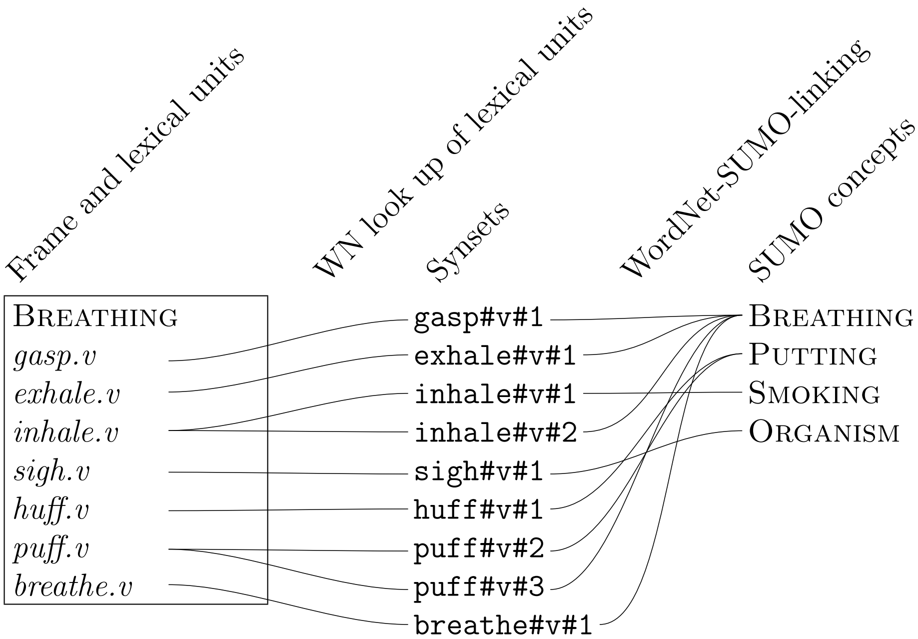

Figure 4.2.: Linking the frame Breathing to SUMO concepts using the refined system

In our approach, we use this disambiguation as a better connection of lexical units and WordNet synsets. Figure 4.2 shows the example from above – the linking procedure of the frame Breathing – with the refined system, table 4.2 shows the results. As one can immediately see, there is much less noise involved.

Our algorithm uses this refined linking by default, but falls back to the previous version for the frames whose lexical units are not included in the manual disambiguation. As in the previously described fallback case, we select the highest weighted SUMO concept as the result.

4.2. Linking Frame Elements to SUMO relations

We will now discuss the linking of frame elements and SUMO relations. As explained in section 3.2.1, frame elements describe participants of prototypical situations. This description includes the (semantic) type (a very rough categorization of potential participants) and, more importantly, the role it plays in this event. The frame element Speaker (semantic type: *Sentient*) in the frame Statement states the existence of a sentient being who plays the role of the speaker in this event. Another sentient being plays the Addressee in the same frame, he is the (intended) recipient of the message.

4.2.1. The role concept of SUMO

Similar information can be found in SUMO, but encoded in a completely different way. Classes representing situations do not have an associated list of participants, but instead one or more axioms. These axioms assert the existence of some entities and that the entities are in some role-like relation with the situation. The role-like relation is an instance of the class RELATION or – mostly – of a subclass of RELATION.

The conceptual difference between semantic roles in FrameNet and in SUMO is that FrameNet defines frame elements per frame. The frame elements of one frame do not have to be appropriate for all other frames. SUMO, on the contrary, defines general roles that are used across all classes. This definition of a role is not bound to a specific class or event. As we have seen in section 3.3, the relations are used in axioms, which often state the existence of an entity and what role this entity plays in an event.

Unfortunately, a common class representing just the semantic role relations in SUMO does not exist. A lot of the role-like relations are instances of the class CASEROLE, but CASEROLE's documentation restricts it to physical objects[2]. CASEROLE contains relations like *agent* and *patient*. *causes*, for instance, is a role-like relation which is not in the class CASEROLE, but in BINARYPREDICATE, which is a super-class of CASEROLE. The class BINARYPREDICATE also contains more technical relations like *entails*, which relates two SUO-KIF formulas.

In order to get an overview of the relations used in actual SUMO axioms, we semi-automatically extracted the axioms that state only the existence of an entity and its relation with an event. The relations have been collected and it turned out that only a relatively small inventory of 14 relations are used in axioms: *agent*, *attends*, *causes*, *destination*, *experiencer*, *instrument*, *located*, *origin*, *patient*, *realization*, *refers*, *represents*, *result* and *subProcess*. Table A.1 in the appendix shows the documentation for each of these roles.

4.2.2. The core linking algorithm

In this section, we present the algorithm that links a frame element to one (or more) SUMO relation(s). The basic idea of the algorithm is to manually link only the core frame elements at the top of the inheritance hierarchy to appropriate SUMO roles and to use the FrameNet hierarchy to link more special frame elements automatically via the top frame elements.

We will assume that we would manually link all 2,412 core frame elements to SUMO relations. This assumed linking will be considered as an "ideal" linking. As we have observed, the SUMO relations are much less fine-grained than the frame elements in FrameNet. Our assumed ideal linking will therefore contain a lot of cases, where multiple frame elements are linked to the same SUMO relation.

Figure 4.3 shows an example using the frames SELF_MOTION and INTENTIONALLY_ACT. SELF_MOTION inherits from INTENTIONALLY_ACT and the frame element SELF_MOTION. SELF_MOVER corresponds to the frame element INTENTIONALLY_ACT.AGENT. Figure

[2]This restriction is not found in the formal definition of CASEROLE.

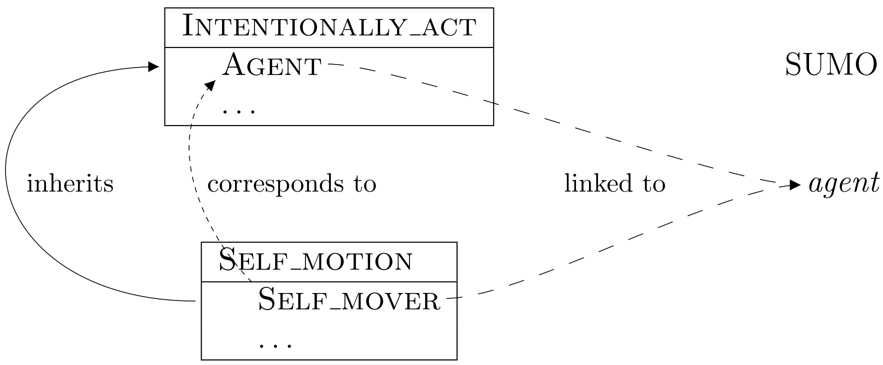

Figure 4.3.: Inheritance relation between the frames SELF_MOTION and INTENTIONALLY_ACT and "ideal" linking of frame elements to SUMO relations

4.3 shows the assumed ideal linking of frame elements to SUMO relations with a long dashed line on the right side of the figure.

As one can easily see, the two frame elements are linked to the same SUMO relation. Based on this observation, we can devise a semi-automatic approach that achieves similar results as the ideal linking, but with significantly less manual effort. Our basic idea is to manually link only the core frame elements in a root frame (a frame that does not inherit from any other frame) to a SUMO relation. To link a frame element in a non-root frame to a SUMO relation, we can then retrieve its corresponding frame element in a root frame and look up the SUMO relation it is linked to.

Figure 4.4 shows the complete linking for the frame element VICTIM of the frame ATTACK (which is no root frame). The frame element corresponds – through several steps – to the frame element DEPENDENT_ENTITY in the root frame OBJECTIVE_INFLUENCE. This frame element is then manually linked to the SUMO relation *patient*, shown by the dashed line.

Unfortunately, the FrameNet inheritance hierarchy is rather shallow. A lot of frames are not embedded in the inheritance hierarchy. We excluded their frame elements from our manual linking, because we expect them to be embedded in the inheritance hierarchy in future versions of FrameNet. 142 frame elements remained. They are linked to 1,256 frame elements in more special frames – we cover roughly nine times the frame elements that we link manually to SUMO relations.

142 frame elements have been linked manually to appropriate SUMO relations. The appropriate SUMO relation was chosen by the author of this diploma thesis, who discussed difficult cases with an expert. 69 frame elements could be successfully linked to a single SUMO relation. For the remaining 73, a SUMO counterpart could not be found.

Since FrameNet contains cases of multiple inheritance, a single frame element may correspond to multiple frame elements in more general frames. We therefore have to take into account that a single frame element may be linked to more than one SUMO relation. We will discuss these cases in the evaluation in chapter 5.

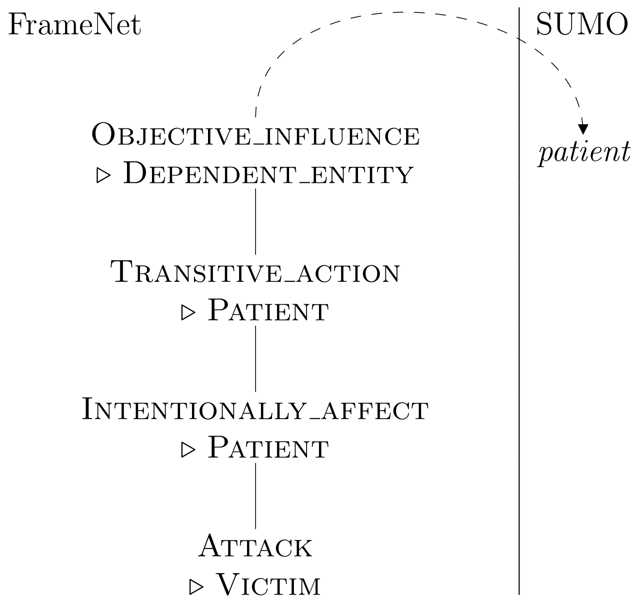

Figure 4.4.: Example for frame element Attack.Victim

We will exemplary look at the manual linking of the seven core frame elements belonging to the frame Bringing. Six of them have been linked more or less straightforwardly (e.g. Agent on *agent*, Goal on *destination*, ...). For the frame element Path, however, an appropriate SUMO role could not be found.

4.3. Issues

In this section, we will briefly discuss two observations that are of interest for future work on or with this linking of FrameNet to SUMO.

4.3.1. Partiality of the role linking

Our approach in linking frame elements to SUMO relations was deliberately designed to reduce manual effort. As a consequence, it remains a partial linking: Since the coreness of a frame element is not inherited along the FrameNet hierarchy, a core frame element may correspond to a non-core frame element in a more general frame, which is not linked manually. Also, if a frame element does not correspond to a frame element in a root frame (a frame that does not inherit from another frame), no linking is found.

There are two possible solutions to address this issue.

- Manually linking the root frame elements: It is technically easy to extract not the frame elements that belong to root frames, but the root frame elements. The root frame elements are the frame elements that have no corresponding frame element in a more general frame, even if a more general frame is defined. These frame elements belong not necessarily to root frames. Frame elements may be introduced at lower levels. Using root frame elements, one would have to link 480 frame elements manually.

- One could also link all the non-core frame elements in root frames which have a core frame element among their descendants.

Both steps sketched above could improve the coverage of our linking of frame elements to SUMO relations. However, since we aim at providing a proof of concept, they have not been implemented.

4.3.2. Specificity of the role linking

The inheritance relation in FrameNet is no identity relation. By following it "upwards", towards more general frame elements, information specific to a certain frame element is obviously lost. In the ATTACK frame, one of the core frame elements is WEAPON. This frame element does not describe the type of object (as is done by the semantic type), but rather the function in which it is used, the role it plays in this event.

(4.1) [Rioters]$_{\text{ASSAILANT}}$ attacked [one man]$_{\text{VICTIM}}$ [with pool cues]$_{\text{WEAPON}}$, breaking his fingers and smashing his cheekbone. (ATTACK)

In sentence 4.1, which is taken from the FrameNet corpus, *pool cues* are annotated as WEAPON, because they are used as a weapon. But of course, pool cues are no weapon. Their "category" or semantic type would not be a weapon, but a sports device.

The information that the pool cues are used as a weapon is lost if the frame element is mapped onto the more general frame element INSTRUMENT in the frame INTENTIONALLY_AFFECT or to the semantic role *instrument* in SUMO. This lack of specificity might lead to a misunderstanding: An instrument in an attack situation could be a weapon, but other instruments in an attack are easily imaginable (protective armor, for instance).

This is of course not a problem of this specific algorithm, but of the general question of how to describe lexical semantic meaning on an appropriate level of granularity.

4.4. Implementation

The algorithms described above have been implemented in Java 1.5. The implementation consists of four packages, which will be described here. Section 4.4.1 describes the package used to access the FrameNet database, section 4.4.2 describes the packages used to access WordNet and the SUMO ontology and section 4.4.3 describes the implementation of the linking algorithms itself. A technical documentation is provided in the respective package documentations.

4.4.1. FrameNet API

In this section, we briefly describe our implementation of a FrameNet API. The API is used to access the FrameNet database quickly and efficiently. The API is wrapped in the Java package `de.saar.coli.salsa.reiter.framenet`. It consists of seventeen classes, six of them being exceptions. The classes represent mainly the entities used by FrameNet. The class `Frame` represents a frame, the class `FrameElement` a frame element and so on.

The central class of the package is the class `FrameNet`. It reads the FrameNet data either directly from the FrameNet XML files or – for efficiency reasons – from a cache file. Using the `FrameNet` class, one can retrieve objects for single frames, semantic types, lexical units, etc.

An important part of the package is the handling of the FrameNet hierarchy. The hierarchy – stored in an extra XML file – is read in by the `FrameNet` class and handled by the classes `FrameNetRelation`, `FrameRelation` and `FrameElementRelation`. The hierarchy can for instance be used to "generalize" a frame or frame element by going upwards in the inheritance hierarchy.

The package can also be used to represent annotated texts by using the classes `RealizedFrame` and `RealizedFrameElement`. Both are basically connections between a piece of text or a node identifier from a syntactical analysis and a frame or frame element. Annotated text can be read in either from SALSA/TIGER XML (by using the class `SalsaTigerXML`; see section 3.2.2 for details about the format) or a very shallow format without any syntactic information (by using the class `FlatFormat`).

The FrameNet API has been released separately[3] under the terms of the GNU General Public License.

4.4.2. SUMO and WordNet API

The SUMO API builds upon the Java package `com.articulate.sigma`, which is released[4] under the terms of the GNU General Public License by Articulate Software and Teknowledge (see Pease (2003) for details). Our own package adds several methods combining already existing methods, but without any additional features.

We used the MIT Java WordNet Interface[5] as WordNet API. It is released under the terms of the "Creative Commons Attribution-NonCommerical Version 3.0 Unported" license, which allows the noncommercial use of the package.

4.4.3. Mapper

The linking program itself consists of two components. The first component of our program generates the list of all corresponding SUMO concepts for each FrameNet frame using the fallback algorithm (without disambiguation of lexical units). For the frames

[3] `http://www.coli.uni-saarland.de/~reiter/FrameNetAPI/`
[4] `http://sigmakee.cvs.sourceforge.net/sigmakee/sigma/`
[5] `http://www.mit.edu/~markaf/projects/wordnet/`

contained in the list of disambiguated lexical units (refined system, see section 4.1.1), the component adds the list of resulting classes using these disambiguations.

These lists are then stored in an XML format similar to the format of the FrameNet database files. The same format is used to store the manual linking of the frame elements to SUMO relations. This part of the implementation can be done once for each version of the involved resources. The XML files may then be shared or distributed.

The second component of the program is intended to run on-line. This component of the program reads the list of SUMO concepts stored for each frame and selects the based on certain criteria. In our implementation, the SUMO concept with the highest weight is selected, but our implementation is prepared to use other criteria. A user interested in specifying his own criteria has to implement a Java interface and load his implementation into the program.

For the linking of frame elements to SUMO relations, the program uses the FrameNet API to retrieve the corresponding frame elements in a root frame for a given frame element. The manual linking of frame elements to SUMO relations is read in and the resulting SUMO relation is returned.

Finally, the implementation can read in annotated FrameNet data (using SALSA/TIGER XML or the shallow format mentioned above) and return appropriate SUMO concepts and relations for all of the data. The program returns objects of the class `SUMOInstance`, which are basically connections between a SUMO concept and a string. The objects also include connections of SUMO relations and strings (the fillers or targets of the frame elements).

The complete implementation will be released on the Internet later in 2007.

5. Evaluation

In this chapter, we evaluate our approach on linking FrameNet and SUMO. The first section evaluates quantitatively: For the assignment of SUMO concepts to frames, we evaluate the coverage and the discriminative power of the weights associated to the classes. For the linking of frame elements to SUMO relations, we evaluate the coverage and look at frame elements that have been linked to more than one SUMO relation.

In the second section of this chapter, we make a qualitative evaluation based on a randomly selected sample of FrameNet annotations. For the assignment of SUMO concepts to frames, we manually rate the resulting SUMO concept of each frame according to certain criteria. The linking of frame elements to SUMO relations is evaluated by judging the appropriateness of each linking of a frame element to a SUMO relation manually.

5.1. Coverage

In this section, we will look at the coverage of our approach. We will investigate the relation between the number and part of speech of lexical units in a frame and the number of classes assigned to this frame by our algorithm. We will also look at the weight distribution and determine the discriminative power of the weights. We compare the results for the refined and unrefined systems. We also look at the number of successfully linked frame elements.

5.1.1. Assignment of SUMO concepts to FrameNet frames

FrameNet 1.3 contains 795 frames. Since the assignment of SUMO concepts to frames we propose is based on the lexical units, the distribution of the lexical units is of interest. On average, 12.82 lexical units belong to one frame. However, the variance is large. The "richest" frame has 179 lexical units, while 74 frames have no lexical unit at all. The standard deviation of the number of lexical units in one frame is $\sigma = 39.65$. This quite large deviation basically tells us, that the number of lexical units is very uncertain and heterogeneous.

Some of the lexical units in FrameNet are marked with parts of speech that are not used in WordNet: prepositions, numerals (*one.num*), determiners (*a few.art*), interjections (*sh.intj*) and complementizers (*while.c*). These lexical units are thus unusable in this approach, since we need to link them to WordNet synsets and WordNet uses only nouns, verbs, adjectives and adverbs. We excluded six frames which contained only such lexical units.

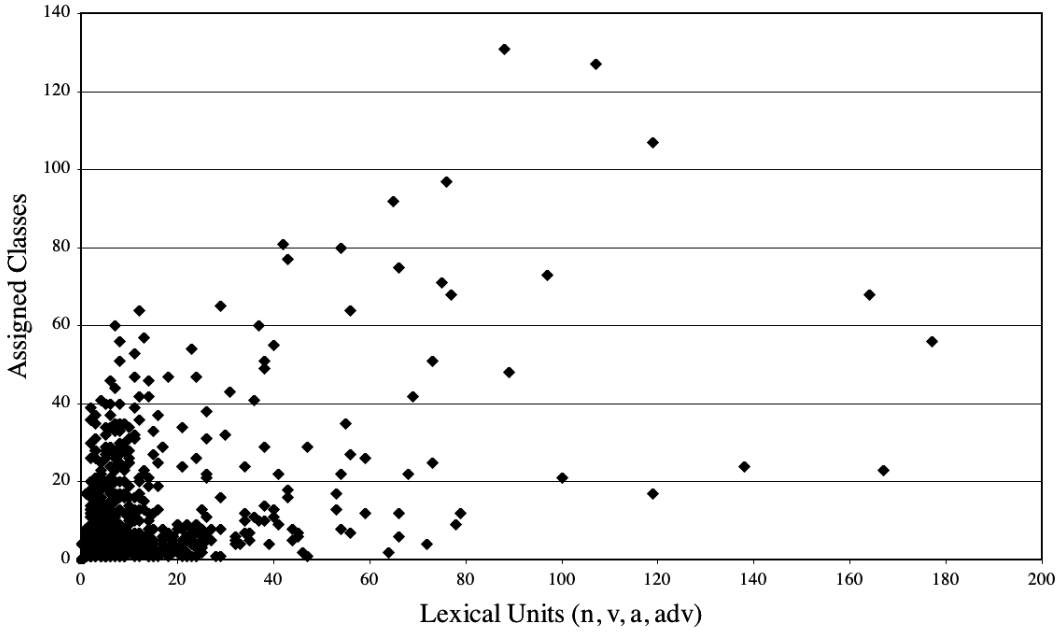

Figure 5.1.: Number of lexical units versus number classes that are assigned to the frame

All of the frames that have appropriate parts of speech are linked to at least one class by our algorithm. These are 715 frames. Looking at the relation between the number of lexical units in a frame and the number of classes that are assigned to that frame, we find that there is no clear correlation. Each dot in figure 5.1 represents one frame. As we can see, the frame with the highest number of lexical units is by far not the frame that has the highest number of assigned classes: A lot of its lexical units seem to be linked to the same class(es).

Refined System In the refined system, we use the disambiguation of lexical units with respect to WordNet provided by Shi and Mihalcea (2005) (see section 4.1.1). 308 (43%) frames use the refined system, 407 (57%) frames use the unrefined system. As expected, the numbers of (different) evoked classes are much smaller in the refined system. In fact, only five frames are assigned to more than 20 different classes (the maximum being the frame Body_movement with 26 classes).

Weights of the resulting SUMO concepts In order to use the linking in an application, the reliability of the weights of the assigned classes is of interest.

Our algorithm assigns a number of synsets to each frame and relates these synsets to SUMO concepts. In a worst case scenario, each SUMO concept would have the same weight. This can happen if each synset would be linked to a different SUMO concept,

Weight	Rank	Ratio	Class(es)
18	1	22.2%	BREATHING
10	2	12.3%	COMMUNICATION
3	3	3.7%	*capability*, ...
2	4	2.5%	DESTRUCTION, ...
1	5	1.2%	LIQUIDMOTION, ...

Table 5.1.: Assignment results for BREATHING, using the fallback system. The table shows the ratio of the number of synsets corresponding to the lexical units of the frame and the number of synsets linked to a single class

Weight	Rank	Ratio	Class(es)
5	1	55.5%	BREATHING
2	2	22.2%	PUTTING
1	3	11.1%	ORGANISM, ...

Table 5.2.: Assignment results for BREATHING, using the refined system. The table shows the ratio of the number of synsets corresponding to the lexical units of the frame and the number of synsets linked to a single class

which would result in a weight of 1 for each SUMO concept. In order to being useful, our linking algorithm should mark a single SUMO concept with a higher weight than the others.

We confirmed that by averaging the ratio of maximal possible weight and maximal observed weight over all frames. The maximal weight that can be assigned to a class is the number of synsets that are evoked by the frame. This "overall weight" is on average 27. The average maximal weight for a single SUMO concept is 8.11. Therefore, 29.71% of the synsets evoked by one frame are linked to the same SUMO concept, making it the highest weighted one. The second highest weighted SUMO concept gets on average only 10.53% of the synsets. This means that 40% of the synsets link to the first two resulting classes, leaving only small weights for the remaining SUMO concepts.

We compared these percentages separately for the refined and fallback system. In both variants, we observe a clear distance between the weight of the first ranked SUMO concept and the second ranked SUMO concept is . For the fallback system, 9.78% of the synsets are linked to the second SUMO concept, while 26.89% of the synsets are linked to the first SUMO concept. Both numbers are even higher in the refined system: 15.05% of the synsets are linked to the second synset and 46.73% are linked to the first synset.

Tables 5.1 and 5.2 show the ratio used above (the ration of the number of synsets corresponding to the lexical units of one frame and the number of synsets linked to a

Frame Element	Role	Role
Renting_out.Lessee	*agent*	*origin*
Commerce_pay.Buyer	*agent*	*origin*
Giving.Donor	*agent*	*origin*
Surrendering_possession.Surrenderer	*agent*	*origin*
Commerce_sell.Seller	*agent*	*origin*
Submitting_documents.Submittor	*agent*	*origin*
Supply.Supplier	*agent*	*origin*
Travel.Traveler	*patient*	*agent*
Cotheme.Theme	*patient*	*agent*
Fleeing.Self_mover	*patient*	*agent*
Intentional_traversing.Self_mover	*patient*	*agent*
Self_motion.Self_mover	*patient*	*agent*
Theft.Perpetrator	*agent*	*destination*
Taking.Agent	*agent*	*destination*
Becoming.Entity	*patient*	*experiencer*
Absorb_heat.Entity	*patient*	*experiencer*

Table 5.3.: Frame Elements that are mapped to more than one semantic role

single class) for the frame Breathing, which we already used as an example in chapter 4. As one can see not only from the absolute weight but also from the ratio, the distance from the first class to the second is large. Please note that the ratios are calculated per SUMO concept. They do not add up to 100%, because not all resulting SUMO concepts are listed in the tables.

5.1.2. Linking of Frame Elements to SUMO relations

The algorithm for the linking of frame elements to SUMO relations is heavily based on the FrameNet hierarchy. Unfortunately, almost half of the frames defined in FrameNet 1.3 are isolated: they neither inherit from other frames nor do they are inherited by other frames. For these 331 frames, the algorithm naturally does not work, which is why we exclude them from the following evaluation.

On average, each of the remaining 464 frames has 9.5 frame elements, 2.94 of them are core frame elements. Using the algorithm described in section 4.2, we can assign a specific SUMO role to 1.83 frame elements, the majority of the core frame elements.

Ambiguous Linkings Our algorithm includes a source for ambiguous linkings: A frame element may correspond to multiple frame elements in different more general frames. As a result, a single frame element may therefore be linked to two different SUMO relations.

This is the case for very few frame elements (16, on average 0.03 per frame). 7 of them are linked to *agent* and *origin*, 5 to *agent* and *patient* and two to each *patient* and *experiencer* as well as *agent* and *destination*. The table can be found in 5.3.

Looking for instance at the *agent*-and-*origin* cases, there is no problem with the double relations. Most of the frame elements *do* describe the agent of the event as well as the origin of something (for instance: Supply.Supplier, Giving.Doner, ...). But there is one exception: the frame element Lessee of the frame Renting_out. It is described as *The Lessee has Money and wants the Goods*. In this case, the assigned role *origin* does not seem to be appropriate. But this error is in fact caused by the FrameNet database, which lets Commerce_sell.Seller correspond to Lessee. The Seller is of course not the one who has money and wants the goods. We therefore suspect this to be an error in the FrameNet database.

5.2. Qualitative Evaluation

Unfortunately, a corpus with SUMO annotations does not exist. This makes it impossible to compare the results of our assignment with a gold standard. A computer program therefore randomly selected 50 annotations from the FrameNet corpus, applied the linking algorithm and we inspected the results manually. Again, we will first look at the frame linking and then at the frame element linking.

5.2.1. Assignment of SUMO concepts to frames

We divided the relations between a FrameNet frame and a SUMO class in seven categories which will be discussed in detail below. The basic idea is that both a frame and a class describe a number of situations. We compared these situations and divided the sample in parts according to the situations that "overlap". *Frame broader*, for instance, denotes the case where the set of situations described by the class is a subset of the set of situations described by the frame.

Figure 5.2 shows the distribution of four of the categories. We excluded the categories "attributes" and "relations" because in both cases, the result of the assignment algorithm is a SUMO class that does not describe an event (both will be discussed below). We did not find any occurrence of an intersection (see below). The remaining four categories that are included in the evaluation contain 28 different linkings.

As we will discuss, only one third of the cases are clearly inappropriate. One quarter of the linkings have a comparable coverage, i.e., the frame describes roughly the same situations as the class. The remaining cases are cases where the resulting class is either too general or too specific, but in both cases clearly related to an ideal result.

Relations

A significant amount of WordNet synsets are linked not to classes, but to relations in SUMO. This leads to assignments such as the relation *prevents* assigned to the frame

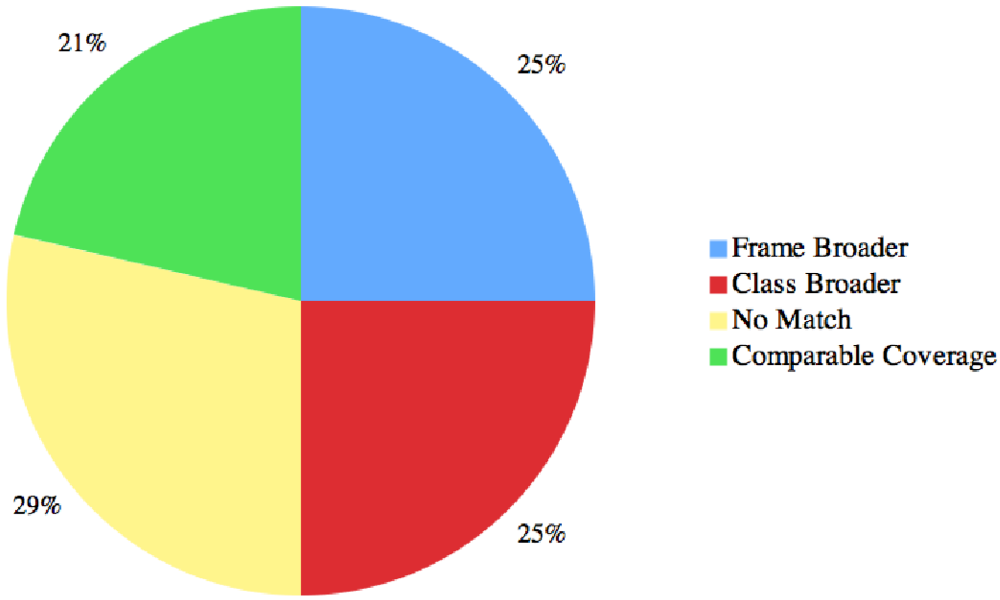

Figure 5.2.: Manual Comparison of linking results of 28 frames

Preventing. This assignment is intuitively not bad, but the problem is that the knowledge modeling on the SUMO-side would be different, making a simple use of the semantic roles impossible.

While the relation *prevents* directly relates (the SUMO-representation of) the preventing cause and the prevented event, the frame Preventing defines the preventing as a situation of its own, relating the preventing cause and the prevented event through relations that can bear more information. This discrepancy can be found in many cases, examples include the frame Existence linked to *exists* as well as the frame Increment linked to *greaterThan*.

We therefore decided to exclude these cases from our evaluation.

Attributes

In SUMO, the relation *attribute* is used to assign attributes to entities. The listing in 5.1 states that if something has the attribute SupremeCourtJudge and is employed by some organization, then this organization is the SupremeCourt.

It has to be noted that the description of attributes of entities is not a description of an event. A direct counterpart for such attributes can not be found in FrameNet, because such information is modeled differently in FrameNet.

The question remains, however, why frames are linked so often to certain attribute related classes and instances in SUMO. The class that is probably evoked most often is SubjectiveAssessmentAttribute. The documentation of the class indicates, why

Listing 5.1: SUMO Axiom with an *attribute*

```
(=>
    (and
        (attribute ?PERSON SupremeCourtJudge)
        (employs ?ORG ?PERSON))
    (instance ?ORG SupremeCourt))
```

it occurs so often: “This Class is, generally speaking, only used when mapping external knowledge sources to the SUMO. If a term from such a knowledge source seems to lack objective criteria for its attribution, it is assigned to this Class.” A large amount of WordNet synsets is linked to this “dummy class”, letting it appear quite often.

Attributes are also excluded of the evaluation.

No Match

In this case, the frame and the class simply do not match. They describe different events. The examples we found were probably caused by the difference in granularity between the different resources used. Education_teaching, for instance, is among others linked to the class Learning. The two do clearly not come from completely different domains, even though Learning is used not only in the case of educational learning, but for any process “which relate to the acquisition of information.” (documentation). However, they describe different events.

Another interesting example is the frame Intentionally_create. It is linked – among others – to the class ContentDevelopment. We classified this as a not matching case: Intentionally_create assigns the semantic type *Artifact* to the frame element Created_entity, which is a subtype of *Physical_object*. The created entity is therefore restricted to be a physical object. ContentDevelopment is applicable for content, i.e., for abstract things.

Both Intentionally_create and Education_teaching are linked with the refined system. The other cases of no match include the frames Activity_ongoing, Cardinal_numbers and Finish_competition and are with the exception of Cardinal_numbers all linked using the refined system.

The frame Cardinal_numbers is evoked whenever a cardinal number occurs. It has two core frame elements: Entity, which is counted, and the Number. It seems that there is no counterpart for this frame in SUMO on this abstract level. CardinalityFn may be the closest match in SUMO, but it has only a weight of 2 (instead of 6 for the class Device) and is a function returning the number of instances of a certain class or collection.

Frame Broader

This class of assignments consists of cases, in which the frame describes more events than the class. This means that an ideal class, which overlaps completely with the frame and may or may not exist in SUMO, is inherited by the resulting class.

In six out of seven cases (21.4% of all cases, see figure 5.2), exactly this is the case: POLITICAL_LOCALES, for instance, would ideally be linked to the class GEOPOLITICALAREA, which is also found in the result list, but ranked lower (rank = 3, weight = 9). This optimal result can actually be found among the lower weighted results in four of the six cases.

The optimal result is not found among the lower weighted classes for the frames MOVING_IN_PLACE and ORIGIN. In both cases, a class that represents the ideal result does not exist in SUMO. The concept of origin can certainly be modeled using functions like *BeginFn* and relations like *located*, but it does not exist as a single concept. MOVING_IN_PLACE is linked to the class ROTATING, which has MOTION as direct super class. A class in between is not present in SUMO.

The remaining case is the frame CHANGE_POSITION_ON_A_SCALE. A good result would be the class QUANTITYCHANGE, but it is not among the selected. A reason for this could be that the class has very few associated synsets, namely 14. On average, every concept has around 45 associated synsets.

Class Broader

In this category, we collect cases, where the SUMO concept describes more events than the frame it is linked to.

We found seven cases that fall in this category. Several concepts represented by frames are simply not present in SUMO. The frame JUDGMENT_COMMUNICATION is linked to the class COMMUNICATION, which is probably the best one can find in SUMO. The definition of the frame includes some of the content of the communication – that it includes a judgment – which should be linked to at least two different classes in SUMO. MOVING_IN_PLACE was already mentioned above, it is also linked to the class MOTION, which is too general.

In few cases, the class selected by the algorithm is actually a super class of an ideal class, for instance for the frame LEADERSHIP, were the class MANAGING would be most appropriate. EDUCATION_TEACHING, which has an equally ranked no match assignment, is linked to the class COMMUNICATION.

Comparable Coverage

For six assignments, we found a comparable coverage. We do not claim that the set of events described by the frame is strictly identical to the set of events described by the SUMO concept. However, there is intuitively more overlap in, for instance, linking the frame COOKING_CREATION to the class COOKING, than linking the frame POLITICAL_LOCALES to the class CITY.

Frame Element	Semantic role in SUMO
STATEMENT.MESSAGE	*represents*
STATEMENT.SPEAKER	*agent*
JUDGMENT_COMMUNICATION.EVALUEE	*experiencer*
CHOOSING.COGNIZER	*agent*
INTENTIONALLY_CREATE.CREATED_ENTITY	*patient*
POLITICAL_LOCALES.LOCALE	*located*
SELF_MOTION.GOAL	*destination*
SELF_MOTION.TIME	*time*
SELF_MOTION.SOURCE	*origin*
SELF_MOTION.SELF_MOVER	*agent*, *patient*
BECOMING.ENTITY	*patient*, *experiencer*
EXPECTATION.COGNIZER	*agent*
SCRUTINY.COGNIZER	*agent*

Table 5.4.: The list of linked frame elements in our evaluation sample

Apart from the frame COOKING_CREATION linked to the class COOKING, we considered the frame SCRUTINY linked to the class INVESTIGATING, the frame CHOOSING linked to the class SELECTING and a few more to describe similar sets of events.

Intersection

In theory, there might be cases where the class describes cases that are not described by the frame *and vice versa*, but there was no such case found among our test cases.

5.2.2. Linking Frame Elements to SUMO relations

It is much more difficult to find an appropriate evaluation measure for the linking of frame elements to SUMO relations. In the randomly selected sample of 50 annotations from the FrameNet corpus (the same sample as used above), we found 13 different frame elements that have been linked to one or more SUMO role.

As one can see in table 5.4, the linking results seem to fit the expectations. In this list, there is not a single linking that is particularly awkward. We will discuss two examples briefly.

STATEMENT.SPEAKER The frame element SPEAKER in the frame STATEMENT is defined as “the person who produces the Message”. The speaker is therefore the one who intentionally makes the situation a STATEMENT situation. He is an active contributor to the scene.

This matches the documentation of the *agent* relation in SUMO: "[the agent] is an active determinant, either animate or inanimate, of the Process [...], with or without voluntary intention". Even though this definition is much more general than the one of SPEAKER, it is nevertheless compatible.

BECOMING.ENTITY The FrameNet database defines this frame element to be the "Entity which undergoes a change, ending up in the FINAL_STATE or FINAL_CATEGORY" (Both FINAL_STATE and FINAL_CATEGORY are other frame elements of the frame BECOMING). By our linking, the frame element is linked to the two roles *patient* and *experiencer*. The role *patient* is defined as "a participant [...] that may be moved, said, experienced, etc.", *experiencer* as "[the experiencer] experiences the Process".

From these definitions, *experiencer* seems to be a better choice than *patient*. But if one looks at the use of the roles, the decision is not that clear. In an axiom of the CREATION class, for instance, the role *patient* is used to relate the created entity with the process, in the class COOLING, the *patient* role is used for the thing that is cooling, etc. Since there is no annotated corpus available and very little other guidance on how to use these roles, one can only speculate that both relations are acceptable.

6. Conclusions

In this diploma thesis, we have discussed an approach to link the resources FrameNet and SUMO, i.e., a lexicographically motivated resource containing frames according to (Fillmore, 1976) and one from the area of knowledge engineering containing abstract, domain-independent knowledge. We have described the resources in detail as well as the tools, algorithms and methods needed for our linking approach. We also made an evaluation that enables us to identify the strengths of this approach as well as remaining problems. We will now concentrate on the latter and discuss in the following section (6.1) some of the fundamental issues we discovered.

In the final section (6.2), we will give an outlook on possible future work and provide some starting points on using our work in an application.

6.1. Discussion

During the evaluation, we found that a significant number of frames have been (automatically) linked to instances of the class ATTRIBUTE. We decided to exclude them from the evaluation as the focus of our work was on relating *event* descriptions. However, a lot of these relations of a frame and a SUMO attribute are intuitively acceptable, because a lot of them contain a frame that describes an attribute or property of an individual and not an event.

The distinction in events and states does not make a big difference from the point of view of a lexicographer for FrameNet. Frames according to Fillmore (1985) are not restricted to be events: One of his main examples is the frame "kinship", containing words like *father* and *son*. The problem, however, is that the role assignment for attributes in SUMO is modeled differently than for events (this is merely a technical detail, we briefly showed this in section 5.2.1).

In order to include state-describing frames in the assignment of SUMO concepts to FrameNet frames, a more fundamental issue needs to be addressed: One would need a clear division between frames that represent states and frames that represent events. Even though all frames that inherit from EVENT describe events, not all frames that describe events inherit from EVENT (EDUCATION_TEACHING, for instance). A direct counterpart for states, i.e., a frame which is inherited by all frames describing states, does not exist in FrameNet 1.3.

If such a clear division would be available in the FrameNet database, the application using our approach could handle the SUMO concept resulting from the linking algorithm in different fashions according to the type of the frame. The application using the mapping results could use different axioms to feed them into a system reasoning over

Listing 6.1: Axiom for sentence 6.2

```
(exists (?F) (?G) (?H)
   (and
      (instance ?F Removing)
      (agent ?G ?F)
      (patient ?H ?F)))
```

the sentences, for instances (see below).

A slightly related problem is the rather incomplete FrameNet hierarchy. With one half of the frames completely isolated from all other frames, the hierarchy is of very limited use. However, we expect this to change in future versions of FrameNet.

6.2. Outlook

It would be very interesting to use our mapping of FrameNet to SUMO in an application such as RTE. In order to do this, a reasoner for first order-logic would be needed for interpreting the information stated in SUMO. There is, however, a technical gap between the results of the implementation of our algorithm and the use of a reasoner for SUMO. We will therefore give some information on how to use the results of our mapping with a reasoner.

The most obvious way is to formulate the results of our mapping as KIF expressions, such that they are defined in the same format as the SUMO ontology.

(6.1) [The gardener]$_{\text{AGENT}}$ empties [the bucket]$_{\text{SOURCE}}$ [of soil]$_{\text{THEME}}$. (EMPTYING)

Example 6.1 shows the sentence *The gardener empties the bucket of soil* with the annotation of a frame and several frame elements. The results of our mapping are shown in example 6.2 for the same sentence, using the corresponding SUMO entities instead of the frame and the frame elements.

(6.2) [The gardener]^{agent} empties [the bucket]^{patient} of soil. (REMOVING)

SUMO and its describing language KIF are not aimed at representing annotated sentences. It is, however, not complicated to represent the information resulting from our mapping with a simple SUMO axiom.

This would be the direct result of our algorithm. The EMPTYING-event is represented by the variable `?F`, which is an instance of the class REMOVING. The fillers of the semantic roles are represented by the variables `?G` and `?H` (and `?I`, if our mapping would have found a SUMO relation for the frame element THEME). There is, however, the question of how to bring together the information in the axiom and the role fillers, i.e., Listing 6.2: Axiom for sentence 6.2 with information about participants

```
(exists (?F) (?G) (?H) (?I)
   (and
      (instance ?F Removing)
      (agent ?G ?F)
      (patient ?H ?F)
      (instance ?G ?K)
      (instance ?H ?L)
      (instance ?I Soil)
      (subclass ?K Human)
      (subclass ?L Container)))
```

the participants of the sentences. Using a word sense disambiguation system that assigns WordNet synsets to words occurring in the sentence, one could, based on the linking of WordNet and SUMO, provide more information about the participants directly in the axiom.

Both the gardener and the bucket are, according to the WordNet-SUMO-linking, not direct instances of Human respectively Container, but instances of an unspecified subclass (`?K` respectively `?L`). The soil occurring in the sentence is modeled as a single instance of the class Soil, even if our linking was not able to relate it to the event.

Especially for debugging, annotation and evaluation purposes, it may be useful to include the surface string in the axiom, using a newly introduced relation called *string* (see listing 6.3).

However, in order to simply reformulate the results of our mapping for use with a reasoner, we propose the template axiom shown in 6.4. In this template, the starting frame is denoted by the variable f, the assigned class by f_c. We introduce the variable `?F` and state it to be an instance of the class resulting from the assignment. This represents the event itself.

The frame elements are denoted by the variables g, h (and more if needed). The results of the linking of frame elements is denoted by the variables g_r, h_r, $\ldots$. The variables then specify the relation between the event itself – `?F` – and a newly introduced variable representing the participant – `?G`, `?H`, and so on.

Using this template, it should be uncomplicated to use our mapping in conjunction with a reasoner in order to reason over natural language text. This reasoning could then be used to improve existing systems, for instance in RTE. Since SUMO provides knowledge that is not included in FrameNet or WordNet, problems with sentences like the ones shown in section 2 could then be addressed.

Listing 6.3: Axiom for sentence 6.2 with information about the participants and surface strings

```
(exists (?F) (?G) (?H) (?I)
   (and
      (instance ?F Removing)
      (string ?F "empties")
      (agent ?G ?F)
      (string ?G "the␣gardener")
      (patient ?H ?F)
      (string ?H "the␣bucket")
      (string ?I "of␣soil")
      (instance ?G ?K)
      (instance ?H ?L)
      (instance ?I Soil)
      (subclass ?K Human)
      (subclass ?L Container)))
```

Listing 6.4: Template axiom

```
(exists (?F) (?G) (?H) ...
   (and
      (instance ?F f_c)
      (g_r ?G ?F)
      (h_r ?H ?F)
      ...
   )
)
```

A. Appendix

Table A.1.: The set of semantic roles used in SUMO

Role	Documentation
agent	(agent `?PROCESS ?AGENT`) means that `?AGENT` is an active determinant, either animate or inanimate, of the Process `?PROCESS`, with or without voluntary intention. For example, Eve is an agent in the following proposition: "Eve bit an apple."
attends	(attends `?DEMO ?PERSON`) means that `?PERSON` attends, i.e. is a member of the audience, of the performance event `?DEMO`.
causes	The causation relation between instances of Process. (causes `?PROCESS1 ?PROCESS2`) means that the instance of Process `?PROCESS1` brings about the instance of Process `?PROCESS2`.
destination	(destination `?PROCESS ?GOAL`) means that `?GOAL` is the target or goal of the Process `?PROCESS`. For example, Danbury would be the destination in the following proposition: Bob went to Danbury. Note that this is a very general CaseRole and, in particular, that it covers the concepts of 'recipient' and 'beneficiary'. Thus, John would be the destination in the following proposition: Tom gave a book to John.
experiencer	(experiencer `?PROCESS ?AGENT`) means that `?AGENT` experiences the Process `?PROCESS`. For example, Yojo is the experiencer of seeing in the following proposition: Yojo sees the fish. Note that experiencer, unlike agent, does not entail a causal relation between its arguments.
instrument	(instrument `?EVENT ?TOOL`) means that `?TOOL` is used by an agent in bringing about `?EVENT` and that `?TOOL` is not changed by `?EVENT`. For example, the key is an instrument in the following proposition: The key opened the door. Note that instrument and resource cannot be satisfied by the same ordered pair.

Table A.1.: The set of semantic roles used in SUMO (continued)

Role	Documentation
located	(located `?PHYS` `?OBJ`) means that `?PHYS` is partlyLocated at `?OBJ`, and there is no part or subProcess of `?PHYS` that is not located at `?OBJ`.
origin	(origin `?PROCESS` `?SOURCE`) means that `?SOURCE` indicates where the `?PROCESS` began. Note that this relation implies that `?SOURCE` is present at the beginning of the process, but need not participate throughout the process.
patient	(patient `?PROCESS` `?ENTITY`) means that `?ENTITY` is a participant in `?PROCESS` that may be moved, said, experienced, etc. For example, the direct objects in the sentences 'The cat swallowed the canary' and 'Billy likes the beer' would be examples of patients. Note that the patient of a Process may or may not undergo structural change as a result of the Process. The CaseRole of patient is used when one wants to specify as broadly as possible the object of a Process.
realization	A subrelation of represents. (realization `?PROCESS` `?PROP`) means that `?PROCESS` is a Process which expresses the content of `?PROP`. Examples include a particular musical performance, which realizes the content of a musical score, or the reading of a poem.
refers	(refers `?OBJ1` `?OBJ2`) means that `?OBJ1` mentions or includes a reference to `?OBJ2`. Note that refers is more general in meaning than represents, because presumably something can represent something else only if it refers to this other thing. For example, an article whose topic is a recent change in the price of oil may refer to many other things, e.g. the general state of the economy, the weather in California, the prospect of global warming, the options for alternative energy sources, the stock prices of various oil companies, etc.
represents	A very general semiotics Predicate. (represents `?THING` `?ENTITY`) means that `?THING` in some way indicates, expresses, connotes, pictures, describes, etc. `?ENTITY`. The Predicates *containsInformation* and *realization* are subrelations of *represents*. Note that *represents* is a subrelation of *refers*, since something can represent something else only if it refers to this other thing. See the documentation string for names.

Table A.1.: The set of semantic roles used in SUMO (continued)

Role	Documentation
result	(result `?ACTION` `?OUTPUT`) means that `?OUTPUT` is a product of `?ACTION`. For example, house is a result in the following proposition: Eric built a house.
subProcess	(subProcess `?SUBPROC` `?PROC`) means that `?SUBPROC` is a subprocess of `?PROC`. A subprocess is here understood as a temporally distinguished part (proper or not) of a PROCESS.

References

Baker, Collin F., Fillmore, Charles J. and Lowe, John B. The Berkeley FrameNet Project. In *Proceedings of the conference of the Association of Computational Linguistics (ACL)*, pp. 86–90. Montreal, Canada, 1998.

Brants, Sabine, Dipper, Stefanie, Hansen, Silvia, Lezius, Wolfgang and Smith, George. The TIGER Treebank. In *Proceedings of the workshop on Treebanks and Linguistic Theories*, pp. 24–41. 2002.

Brants, Thorsten. TnT–a statistical part-of-speech tagger. In *Proceedings of the conference on Applied natural language processing*, pp. 224–231. Morgan Kaufmann Publishers Inc., 2000.

Burchardt, Aljoscha, Erk, Katrin and Frank, Anette. A WordNet Detour to FrameNet. In B. Fisseni, H.-C. Schmitz, B. Schröder and P. Wagner, eds., *Sprachtechnologie, mobile Kommunikation und linguistische Resourcen*, volume 8 of *Computer Studies in Language and Speech*, pp. 408–421. Peter Lang, Frankfurt am Main, 2005.

Collins, Michael. Three generative, lexicalised models for statistical parsing. In *Proceedings of the Conference of the Association of Computational Linguistics (ACL)*, pp. 16–23. Association for Computational Linguistics, Morristown, NJ, USA, 1997.

Crouch, Richard, Condoravdi, Cleo, de Paiva, Valeria, Stolle, Reinhard and Bobrow, Daniel G. Entailment, Intensionality and Text Understanding. In *Proceedings of the workshop on Text meaning on the joint conference of Human Language Technology and North American Chapter of the Association of Computational Linguistics*, pp. 38–45. Association for Computational Linguistics, 2003.

Daelemans, Walter, Zavrel, Jakub, van der Sloot, Ko and van den Bosch, Antal. TiMBL: Tilburg Memory-Based Learner. Technical Report ILK 03-10, Tilburg University, 2003.
URL `http://ilk.uvt.nl/downloads/pub/papers/ilk.0310.pdf`

Dagan, Ido and Glickman, Oren. Probabilistic textual entailment: Generic applied modeling of language variability. In *Proceedings of Learning Methods for Text Understanding and Mining workshop*, pp. 26–29. 2004.

Dagan, Ido, Glickman, Oren and Magnini, Bernardo. The PASCAL Recognising Textual Entailment Challenge. In *Proceedings of the PASCAL Recognising Textual Entailment Challenge workshop*, pp. 1–8. 2005.

Dubey, Amit. What to do when lexicalization fails: parsing German with suffix analysis and smoothing. In *Proceedings of the conference of the Association of Computational Linguistics (ACL)*, pp. 314–321. 2005.

Erk, Katrin, Kowalski, Andrea, Pado, Sebastian and Pinkal, Manfred. Towards a Resource for Lexical Semantics: A Large German Corpus with Extensive Semantic Annotation. In *Proceedings of the conference of the Association of Computational Linguistics (ACL)*, pp. 537–544. Sapporo, Japan, 2003.

Erk, Katrin and Pado, Sebastian. A powerful and versatile XML Format for representing role-semantic annotation. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*. Lisbon, Portugal, 2004.

Erk, Katrin and Pado, Sebastian. Shalmaneser - a flexible toolbox for semantic role assignment. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*. Genoa, Italy, 2006.

Fellbaum, Christiane. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

Fillmore, Charles J. Frame Semantics and the Nature of Language. In *Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech*, volume 280, pp. 20–32. 1976.

Fillmore, Charles J. Frames and the semantics of understanding. *Quaderni di Semantica*, 6 (2): 222–254, 1985.

Gangemi, Aldo, Guarino, Nicola, Masolo, Claudio, Oltramari, Alessandro and Schneider, Luc. Sweetening Ontologies with DOLCE. *Proceedings of the conference on Knowledge Engineering and Knowledge Management (EKAW)*, pp. 166–181, 2002.

Gazdar, Gerald. *Pragmatics: Implicature, Presupposition and Logical Form*. Academic Press, 1979.

Hamp, Birgit and Feldweg, Helmut. GermaNet-a Lexical-Semantic Net for German. *Proceedings of the workshop on Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications on the conference of the Association of Computational Linguistics (ACL)*, 1997.

Kingsbury, Paul and Palmer, Martha. PropBank: the Next Level of TreeBank. *Proceedings of Treebanks and Lexical Theories*, 2003.

Kipper, Karin, Dang, Hoa Trang, Schuler, William and Palmer, Martha. Building a class-based verb lexicon using TAGs. *TAG+ 5 Fifth International Workshop on Tree Adjoining Grammars and Related Formalisms*, pp. 147–154, 2000.

Lenat, Douglas B. and Guha, RV. *Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1989.

Levin, Beth. *English verb classes and alternations*. University of Chicago Press Chicago, 1993.

Lin, Dekang. Principle-based parsing without overgeneration. In *Proceedings of the conference of the Association of Computational Linguistics (ACL)*, pp. 112–120. Association for Computational Linguistics, Morristown, NJ, USA, 1993.

Malouf, Robert. A comparison of algorithms for maximum entropy parameter estimation. *Proceedings of the International Conference On Computational Linguistics*, pp. 1–7, 2002.

Marcus, Mitchell P., Marcinkiewicz, Mary Ann and Santorini, Beatrice. Building a large annotated corpus of English: the Penn treebank. *Computational Linguistics*, 19 (2): 313–330, 1993.

McCallum, Andrew Kachites. Mallet: A machine learning for language toolkit. 2002. URL `http://mallet.cs.umass.edu`

Mengel, Andreas and Lezius, Wolfgang. An XML-based representation format for syntactically annotated corpora. *Proceedings of the Language Resources and Evaluation conference (LREC)*, pp. 121–126, 2000.

Niles, Ian and Pease, Adam. Towards a standard upper ontology. *Proceedings of the conference on Formal Ontology in Information Systems*, pp. 2–9, 2001.

Niles, Ian and Pease, Adam. Linking Lexicons and Ontologies: Mapping WordNet to the Suggested Upper Merged Ontology. *Proceedings of the conference on Information and Knowledge Engineering*, 2003.

Pease, Adam. The Sigma Ontology Development Environment. *Working Notes of the IJCAI-2003 Workshop on Ontology and Distributed Systems*, 71, 2003.

Pease, Adam. Standard Upper Ontology Knowledge Interchange Format. 2004. URL `http://sigmakee.cvs.sourceforge.net/*checkout*/sigmakee/sigma/suo-kif.pdf`

Reiter, Nils. Detour – A CPAN module. Technical report, Saarland University, Software Project, 2007.
URL `http://www.coli.uni-saarland.de/~reiter/detour-cpan-module.pdf`

Ruppenhofer, Josef, Ellsworth, Michael, Petruck, Miriam R. L., Johnson, Christopher R. and Scheffczyk, Jan. *FrameNet II: Extended Theory and Practice*. ICSI Berkeley, 2006. URL `http://framenet.icsi.berkeley.edu/book/book.pdf`

Scheffczyk, Jan, Pease, Adam and Ellsworth, Michael. Linking FrameNet to the Suggested Upper Merged Ontology. In *Proceedings of the conference on Formal Ontology in Information Systems*. 2006.

Schmid, Helmut. Probabilistic part-of-speech tagging using decision trees. *Proceedings of the conference on New Methods in Language Processing*, 12, 1994.

Shi, Lei and Mihalcea, Rada. Semantic Parsing Using FrameNet and WordNet. In *Proceedings of the Human Language Technology Conference*, pp. 19–22. 2004.

Shi, Lei and Mihalcea, Rada. Putting Pieces Together: Combining FrameNet, VerbNet and WordNet for Robust Semantic Parsing. In *Proceedings of the Conference on Intelligent Text Processing and Computational Linguistics (CICLing)*, pp. 100–111. Springer-Verlag, 2005.