

Durham E-Theses

An investigation into figurative language in the 'LOLITA' NLP system

Johannes Heitz

How to cite:

Heitz, Johannes (1996) An investigation into figurative language in the 'LOLITA' NLP system. Masters thesis, Durham University.

Use policy

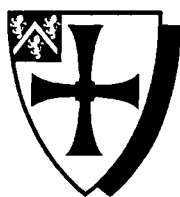
The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a <https://etheses.durham.ac.uk/id/eprint/5436/> is made to the metadata record in Durham E-Theses
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Please consult the [full Durham E-Theses policy](#) for further details.

University of Durham



An Investigation into Figurative
Language in the 'LOLITA' NLP
System

Johannes Heitz

*Laboratory for Natural Language Engineering,
Department of Computer Science,
University of Durham, U.K.*

Submitted for the degree of Master of Science,
November 1996

The copyright of this thesis rests
with the author. No quotation
from it should be published
without the written consent of the
author and information derived
from it should be acknowledged.



12 MAY 1998

Johannes Heitz

**An Investigation into Figurative Language in the
'LOLITA' NLP System**

Master of Science, 1996

Abstract

The classical and folk theory view on metaphor and figurative language assumes that metaphor is a rare occurrence, restricted to the realms of poetry and rhetoric. Recent results have, however, unarguably shown that figurative language of various complexity exhibits great systematicity and is pervasive in everyday language and texts.

If the ubiquity of figurative language cannot be disputed, however, any natural language processing (NLP) system aiming at processing text beyond a restricted scope has to be able to deal with figurative language. This is particularly true if the processing is to be based on deep techniques, where a deep analysis of the input is performed.

The LOLITA NLP system employs deep techniques and, therefore, must be capable of dealing with figurative input. The task of natural language (NL) generation is affected by the naturalness of figurative language, too. For if metaphors are frequent and natural, NL generation not capable of handling figurative language will seem restricted and its output unnatural.

This thesis describes the work undertaken to examine the options for extending the LOLITA system in the direction of figurative language processing and the results of this project. The work critically examines previous approaches and their contribution to the field, before outlining a solution which follows the principles of natural language engineering.

Declaration

The material contained within this thesis has not previously been submitted for a degree at the University of Durham or any other university. The research reported within this thesis has been conducted by the author unless indicated otherwise.

The copyright of this thesis rests with the author. No quotation from it should be published without his prior written consent, and information derived from it should be acknowledged.

Contents

1	Introduction	1
1.1	Problem outline	1
1.2	Summary	3
2	Context of this work	4
2.1	Natural language engineering	4
2.2	The LOLITA system	5
2.2.1	The SemNet	7
2.2.2	The LOLITA parsing system	8
2.2.3	Semantics and pragmatics	8
2.2.4	The LOLITA generator	9
2.3	Applications and prototypes	10
2.4	Summary	13
3	Related work	14
3.1	Classical accounts of metaphor	15
3.1.1	Deviance and substitution view	19
3.2	Modern accounts	20
3.2.1	Tension view	20
3.2.2	Anomaly views	21
3.2.3	Comparison view	29
3.2.4	Interaction view	38
3.2.5	Conventional metaphor view	44
3.2.6	Connectionist models	52
3.3	Summary	58

3.3.1	Central issues	58
3.3.2	Classification of types of figures	60
3.3.3	Additional issues	63
4	Figurative language and LOLITA	66
4.1	Problem statement	66
4.2	Methodology	69
4.3	Evaluation criteria	70
4.4	Summary	72
5	Processing figurative language in LOLITA	73
5.1	Theoretical issues	73
5.2	Tasks and prerequisites	76
5.2.1	Resolution levels	76
5.2.2	Types of figures	88
5.2.3	Knowledge structures	110
5.2.4	Methods	114
5.3	Processable figures	118
5.3.1	General algorithm	118
5.3.2	Simple tropes	126
5.3.3	Fixed metonymies	131
5.3.4	Contextual metonymies	134
5.3.5	Conventional figures	137
5.3.6	Idioms	139
5.3.7	Comparison statements	142
5.4	Integration and implementation overview	147
5.5	Summary	152
6	Evaluation and conclusion	154
6.1	Primary objectives	154
6.2	Secondary objectives	155
6.3	Natural language engineering perspective	158
6.4	Achievements	161

6.5	Future work	163
6.6	Summary	164
A	Examples of figurative processing in LOLITA	166
A.1	Simple tropes	166
A.2	Fixed metonymies	173
A.3	Contextual metonymies	176
A.4	Conventional figures	178
A.5	Idioms	182
A.6	Comparison statements	184
B	Algorithmic and representational details	191
B.1	Natural language representation of figurative knowledge	191
B.2	Algorithmic details	194
B.2.1	Simple tropes	194
B.2.2	Fixed metonymies	196
B.2.3	Contextual metonymies	197
B.2.4	Conventional metaphors	198
B.2.5	Idioms	200
B.2.6	Comparison statements	202
B.2.7	Figurative reasoning	203
B.3	Conceptual proximity	209
B.4	Invariance hierarchy	213
B.5	Abstract data type metaphor	215

List of figures

2.1	Structure of the LOLITA system	6
2.2	A portion of SemNet around an event	7
2.3	Example of a template produced by the content scanning module .	11
2.4	Sample output from the Chinese tutor application	12
2.5	Example of the dialogue application	12
2.6	Example of the query application	13
3.1	Example of a semantic formula in preference semantics	26
3.2	Example of the type hierarchy in the met* method	27
3.3	Structure mapping: base domain representation	32
3.4	Structure mapping: target domain representation	33
3.5	Semantic field theory: example of topic domain structure	41
3.6	Semantic field theory: example of topic domain structure	41
3.7	Spatial semantic operators for conceptual scaffolding	53
3.8	Sapper triangulation rule	55
3.9	Sapper squaring rule	55
5.1	Location of the figurative processing sub-system in LOLITA	76
5.2	Tentative semantic classification of types of metonymies	96
5.3	Problematic classification of types of comparison statements	108
5.4	Example of the triangulation rule	115
5.5	Example of concept extension: fixed metonymy	117
5.6	Conceptual proximity debugging output	118
5.7	Overview of the algorithmic analysis of figurative language	119
5.8	First stage of knowledge-based algorithm: concept extension	120
5.9	Second stage of knowledge-based algorithm: knowledge consultation	120

5.10	Third stage of knowledge-based algorithm: initial interpretation . .	121
5.11	Decision tree for the handling of comparison statements	123
5.12	Chain of reasoning in the analysis of figurative language	125

Chapter 1

Introduction

This thesis describes current work in the field of processing figurative natural language (NL) in the framework of the large-scale natural language processing (NLP) system LOLITA¹. The work was carried out as postgraduate research in close cooperation with other researchers and academic staff of Durham University's Laboratory for Natural Language Engineering. This work is interdisciplinary, drawing on different sources for methodological, empirical and theoretical inspiration, most notably computational linguistics in the form of natural language engineering (see Sect. 2.1), cognitive science and psychology, socio-linguistics and general linguistics as well as artificial intelligence and software engineering.

1.1 Problem outline

NLP concerns itself with the analysis, understanding and generation of NL, in the form of written texts or spoken language. NLP systems, therefore, deal with an array of tasks, one of them being the assignment of a meaning to NL input.

Until recently, figurative language was commonly seen as deviant, as an embellishment and a rhetorical means for persuasion and was generally not regarded as a part of everyday language worth investigating outside the scholarly study of poetics, literature and philosophy. Literal language was regarded as the only concise way of transferring information and as the only true representation of reality.

Most theories of meaning postulate that literal meaning is the primary mean-

¹Large-scale object-based linguistic interactor, translator and analyser

ing of any linguistic expression (truth conditional semantics [Davidson 1979, Carnap 1956], speech act theory [Searle 1975, Searle 1979], conversational implicature [Grice 1975]) and that instances of figurative speech such as metaphor and irony are conscious violations of semantic rules indicating a secondary or speaker meaning which is distinct from the invariable word or sentence meaning (based on the compositionality principle, [Frege 1952]). It is assumed, furthermore, that any meaning conveyed in a non-literal expression could be conveyed equally well by a literal paraphrase (the expressibility principle, [Searle 1969]) or that non-literal language is indeed so peripheral to language and used for such limited purposes that it is not necessary to extend the study of meaning to such instances [Black 1962].

Work carried out by an increasing number of researchers, most notably in the fields of psychology and (cognitive) linguistics over the past 15 years has, however, not only shown undeniable empirical evidence against the idea of figurative language being peripheral, but has also uncovered highly systematic figurative structures which are central to human cognition in general and to one rather central manifestation of cognitive activity in particular, namely, human language (see Chap. 3).

If, however, figurative language is pervasive in everyday NL, a system that concerns itself with processing NL certainly has to be able to process figurative language. Particularly if the input is unrestricted, i.e. if any kind of text is within the range of possible input, and figurative language cannot be explicitly ruled out. This is especially true of systems that do not perform 'shallow' NL tasks such as morphological analysis, simple parsing or pattern-matching style content scanning. As Noble observed: "an NLP system (in the fullest sense of the term) which cannot explain the mechanism of metaphor has failed" [Noble 1988].

The LOLITA system is based on deep understanding techniques and therefore must be able to process figurative language as an integral part of the range of NL tasks it set out to perform. Since the LOLITA system is trying to *understand* its input and is performing such tasks as query (where information can be added to the system's knowledge base (KNB) via NL input and queries on the existing knowledge can be performed) or reasoning (where the knowledge gleaned from the NL input is extended), the meaning of figurative input has to be computed.

If, for example, input like 'Becker killed Lendl on the centre court' is encountered, the meaning of this sentence is analysed and stored in the KNB to allow for

further access, e.g. for reasoning. A literal reading would entail that Lendl is dead after the event mentioned took place, that Becker caused Lendl to be dead, that a murder (or manslaughter) took place, that the event was located at the ‘centre court’ and so on. A query about the state of health of Lendl would therefore result in the information that Lendl was dead.

If a more fitting figurative interpretation were made, a different result would be obtained, e.g. that Becker competed successfully against Lendl, that Lendl lost in a competition, that the competition was fierce, that Becker used force and, most importantly, that both entities were presumably alive and well after the event.

Taking into account the results from other work, it will be shown in Chap. 3 that this is no isolated, contrived example and that highly conventionalised metaphors pervade all texts to such an extent that they are hardly consciously recognised.

After considering related work, we will present a detailed problem statement, elaborating on which sub-tasks figurative language processing consists of and which requirements have to be met in order to allow the treatment of figurative language. These results will then be examined in the light of the LOLITA system and the principles of natural language engineering (see Sect. 2.1), followed by an account of the work that was undertaken to extend the LOLITA system with facilities for processing figurative language.

1.2 Summary

An NLP system performing tasks above a certain level of complexity has to be able to arrive at the meaning of its input. Figurative expressions make up a considerable portion of NL. Therefore it is a necessity for NLP systems which operate on the meaning of their input to be able to process figurative expressions adequately. This thesis investigates the requirements and problems pertaining to the processing of figurative language in the existing large-scale NLP system LOLITA with emphasis on the underlying principles of natural language engineering.

Chapter 2

Context of this work

The work described in this thesis was carried out within the framework of the LOLITA natural language processing (NLP) system. This chapter will introduce the most important aspects of this system as well as the principles of natural language engineering (NLE) on which both LOLITA and, subsequently, this work are based.

2.1 Natural language engineering

Research in the field of NLP at the University of Durham centres around the methods of NLE rather than on the approaches taken by traditional computational linguistics or cognitive sciences, although NLE aims to incorporate and utilise results of various disciplines where they are believed to fulfil the criteria outlined by the requirements of NLE applied to natural language (NL). NLE is a relatively recent development [Garigliano *et al.* 1995] and concerns itself with building useful and robust systems; by contrast, in the traditional approaches taken by theoretical computational linguistics and artificial intelligence research, the evaluation of theories or the construction of smaller, domain-dependent or specialised systems (e.g. for parsing, spelling correction, text-to-speech) play a more central role.

The main aspects of NLE pertaining to building NL systems are:

- **Feasibility:** An NLE system should be designed with its applicability to real-world situations and limitations in mind, i.e. the requirements on resources such as hard- and software and execution time have to stay within reasonable

limits.

- **Robustness:** One of the most critical aspects of large-scale systems is robustness, pertaining on one hand to the scope of the system (e.g. vocabulary size, percentage of accepted versus rejected input) and on the other hand to the handling of input that falls outside the predefined scope of acceptable input. A robust NLE system should be able to deal with any type of input and to recover from worst cases.
- **Scale:** An NLE system must be of sufficient size to support realistic large-scale applications, i.e. considerably more powerful than limited, small-scale systems used for such purposes as the evaluation of linguistic and cognitive theories and feasibility studies. Properties such as vocabulary size and the coverage of the grammar are important in this respect.
- **Integration:** Single components of the system should not make unfounded assumptions about other parts (i.e. they should cover worst-case scenarios) and should be designed as independently as possible to be of general use to an array of other components.
- **Maintainability:** The usefulness of the system over a period of time must be ensured. Thus the requirements all three types of maintenance should be considered when designing the system and coding its components, i.e. perfective maintenance pertaining to the improvement of the system's effectiveness by adding new functionality, adaptive maintenance in response to changes in the specification or the system's environment and corrective maintenance regarding the elimination of flaws not detected during in-house testing.
- **Flexibility:** Ease of adaptation to different tasks and domains is crucial to the notion of usefulness of an NLE system.
- **Usability:** An NLE system should support applications that are of use to end users in real-life situations and be user-friendly.

2.2 The LOLITA system

The LOLITA system has been under development at the University of Durham's Laboratory for Natural Language Engineering [LNLE 1995] since 1986. Currently,

approximately 20 researchers are working on various aspects of the system. It is defined as an extension of the terminology presented by Galliers and Jones [Galliers and Jones 1993]:

An NLP system carries out a *task* and any system which is used to perform a task in a specific domain is an *application*. A generic system is designed to perform a certain task or [...] task type, in different domains [...] *General purpose systems* are intended to be directly usable without further tailoring for more than one application.

This means it is defined as a general purpose base NLP system [Smith *et al.* 1994], i.e. a system that offers fundamental NLP functions and supports a range of different applications based on the core. Because until recently research has been directed mainly towards the development of the core, no polished final application is yet available. The ease with which an array of prototypes could be developed, however, indicates how sound the core really is: the development of the figurative language processing sub-system was considerably facilitated by the foundations of the core system it was built on. The prototypes were developed to investigate the strength of the core and the feasibility of the tasks carried out by the prototypes. They will be discussed in Sect. 2.3. A diagram of the system's components and their relations is shown in Fig. 2.1.

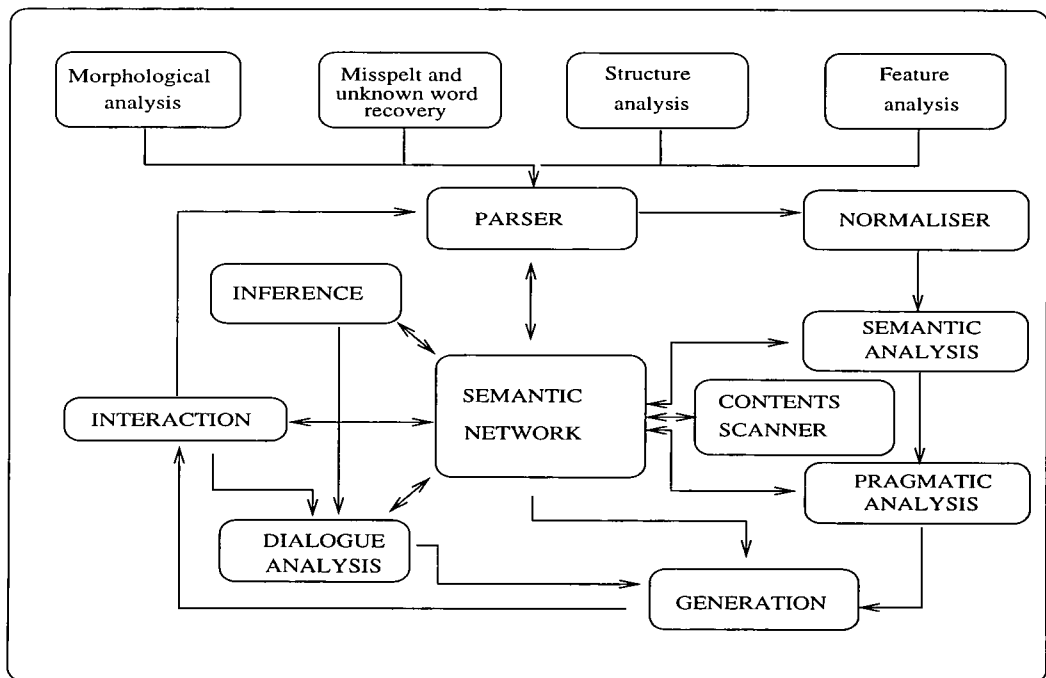


Fig. 2.1: Structure of the LOLITA system

2.2.1 The SemNet

The core of the LOLITA system is a semantic network (SemNet) [Short 1996, Short *et al.* 1996], containing linguistic and world knowledge, loosely based on Conceptual Graph Theory [Sowa 1984], organising concepts as nodes and relations as arcs in a directed hyper-graph. SemNet currently comprises some 100,000 nodes and is compatible with WordNet [Miller 1990]. The nodes in the SemNet graph correspond to objects, concepts or events and are identified by a unique number (Noderef). The links between the nodes represent the relationship between nodes. A link consists of an arc (the type of link stating the relation between nodes) and a list of Noderefs connected to the originating node by the particular arc. Each node is also assigned a set of control variables. Types of links are, for example, subject, is-a, or mode, denoting that the node they are referring to is the subject of an event, an instance of a concept, or a description of the mode of an event, respectively. For an example of an event node as represented in the LOLITA SemNet, see Fig. 2.2. This example corresponds to the event ‘Joe likes coffee’, where ‘event X’ is the node representing the whole event of ‘liking coffee’, consisting of other nodes from and links into the rest of the SemNet and ‘X’ representing the Noderef)

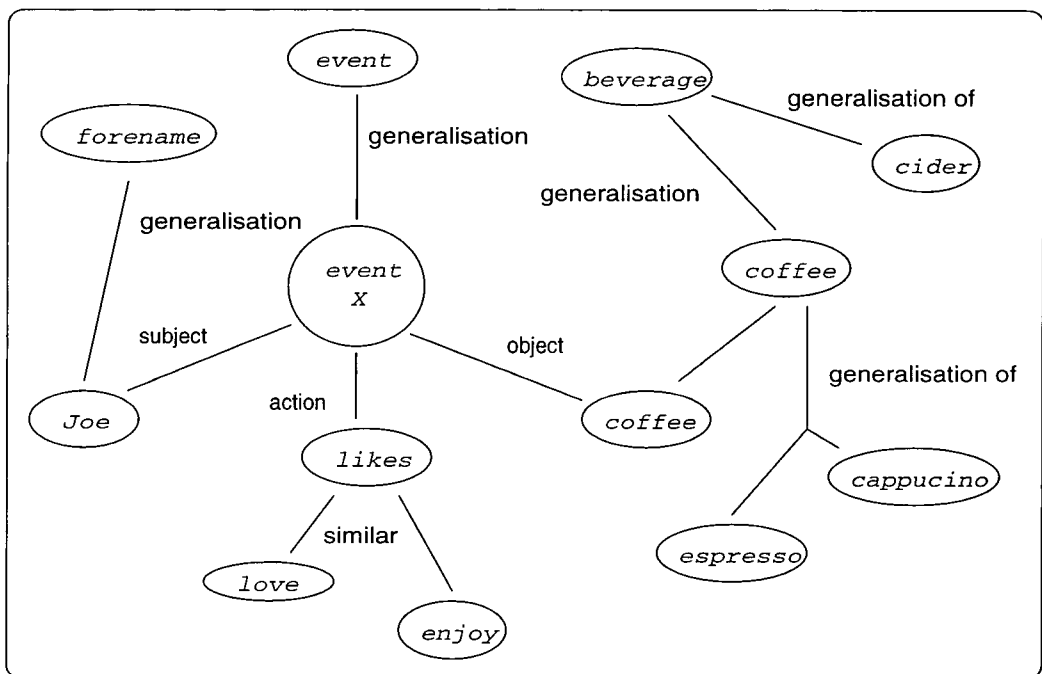


Fig. 2.2: A portion of SemNet around an event

Control variables are stored with each link for reasons of efficiency (i.e. since they are accessed fairly often, this representation results in faster lookup) and

provide information on specific nodes. Examples of some of the 60 different types of control variables currently in use are:

- **Rank:** provides information on the node's quantification, instances of which are individual, prototype, universal, bounded existential or named individual. LOLITA uses a multi-sorted logic representation to deal with the ranking.
- **Type:** roughly maps onto grammatical categories, with some exceptions and additions. Examples of type control variables are entity, relation, event, determiner, punctuation, attribute, preposition, greeting and fact.
- **Family:** classifies nodes into semantic and pragmatic classes and is used frequently in resolving polysemy and ambiguity. Examples of the family control variables are living, vegetal, animal, human, abstract, concrete, temporal and location.

2.2.2 The LOLITA parsing system

The LOLITA parser provides large-scale coverage, i.e. it is able to deal with full and serious text, such as newspaper articles, and is based on a Backus Naur Form definition. It uses a top-down Tomita-style approach and incorporates deterministic operators for improved efficiency [Ellis *et al.* 1993], building a parse forest of possible parses from the input. The ease with which the original non-deterministic parser was augmented with these operators without disruption to the rest of the system reflects on the maintainability of the system (see Sect. 2.1). The LOLITA grammar was built with the aim of dealing with erroneous and incomplete input (e.g. real-life speech and fragments of NL utterances), so that missing words and constructs at the beginning and end of a clause or sentence, for example, can be investigated. It comprises over 1500 grammatical rules. Large corpora, e.g. the Brown corpus and WordNet, have been used for developing a broad coverage.

2.2.3 Semantics and pragmatics

The task of the LOLITA semantic analysis is to map the deep grammatical representation of the input provided by the parsing component onto nodes in the SemNet. To do this, the network has to be checked for the existence of nodes that already represent concepts in the input, and decisions have to be taken on when to

generate new nodes and how to connect them to the rest of the SemNet. Amongst other things, this involves anaphora resolution and making deictic references absolute; ‘tomorrow’ will be expanded into the date after the utterance event, ‘I’ will be resolved into a reference to the speaker, etc.

Further semantic and pragmatic analyses ensure that, after a new or modified portion of the SemNet has been built on the basis of previous stages, this portion is consistent with the existing network. Examples are sentences like “I saw a tree fly over the house”, where no obvious syntactical and semantic rules are violated, or “I bought one of those Japanese TVs made by Philips” where it is highly unlikely and undesirable to extend the coverage of the semantic representation to world knowledge (stating that Philips is not a Japanese manufacturer of consumer electronics). If the pragmatic analysis (located at the intersection between semantic, linguistic and world knowledge) cannot resolve a conflict between new and existing information in the SemNet, a low level of belief is attached to the new portion of the semantic network resulting from the input.

Another way of deciding on the acceptability of input is source control. Source control takes into account from whom the information came and the way in which it was provided, e.g. a reliable source, an unknown source, part of a chat or a factual news report. A model of source control will be fully incorporated into the LOLITA system [Bokma and Garigliano 1992].

2.2.4 The LOLITA generator

The LOLITA generator was, like the rest of the LOLITA system, developed without any specific restrictions imposed by a particular application and is thus very flexible. It is capable of generating NL utterances from the SemNet and is widely used as an interface to LOLITA and as debugging tool. Its input consists of a chunk of the SemNet, and its output is an NL utterance of a complexity greater of or equal to a sentence, depending on parameters such as the particular application, e.g. storytelling or query, the context, the required style, i.e. colourful or simple, and previous dialogue analysis where applicable. A detailed discussion of the generator module can be found in [Smith 1995].

2.3 Applications and prototypes

The LOLITA system has been used as a base for various prototypical applications. A short account of a selection of these prototypes will be given in the following.

- **Content scanning**

Summarising templates are filled from input texts. Input text is parsed and semantically analysed to arrive at a representation of the input in the SemNet. An application, i.e. a domain-dependent module, then accesses the SemNet in order to find relevant information to fill the template slots. Further information on the use and applications of contents scanning can be found in the literature [Garigliano *et al.* 1993], [Morgan *et al.* 1995], [Costantino *et al.* 1996]. The task of content scanning is one of the standard tests for evaluating natural language systems [Long and Garigliano 1994]. Fig. 2.3 gives an example of content scanning in the LOLITA system.

- **Machine translation**

Although machine translation (MT) was not one of the goals the developers of LOLITA originally had in mind, a prototype MT system was built with only a small amount of modification. After the addition of a number of rules to the grammar, it became possible to analyse Italian and Chinese texts, adding the information contained in them to the SemNet. This information was subsequently realised as English NL utterance by the generator module. Clearly, the quality of the translation is not as refined as that of dedicated MT systems, but it has to be kept in mind that the MT prototype was neither one of the original goals nor is it a finished application. For work on improving the quality of transfer see Emery, who notes: “The eventual implementation of a style algorithm capable of extracting style and setting parameters for the generation of a range of stylistic effects would [...] be of particular benefit in Machine Translation where it would ensure that the style of the original text was preserved.” [Emery 1994].

- **Computer-aided language learning**

A Chinese tutoring prototype application based on the LOLITA core has been built [Wang and Garigliano 1992], helping students learning Chinese to overcome L1 interference-based transfer errors. The tutoring module makes use of intelligent tutoring techniques such as constant access and updating of

A car bomb exploded outside the Cabinet Office in Whitehall last night, 100 yards from 10 Downing Street. Nobody was injured in the explosion which happened just after 9 am on the corner of Downing Street and Whitehall. Police evacuated the area. First reports suggested that the bomb went off in a black taxi after the driver had been forced to drive to Whitehall. The taxi was later reported to be burning fiercely.

(THE DAILY TELEGRAPH 31/10/92)

```

Template: Incident
Incident: A bomb explosion.
Where   : On the corner of Downing Street and Whitehall.
          Outside Cabinet Office and outside 10 Downing Street.
          In a black taxi.
When    : 9pm.
          Past.
          Night.
          When a forceful person forced a driver to drive a
          black taxi to Whitehall.

Responsible:
Target: Cabinet Office.
Damage: Human: Nobody.
          Thing: A black taxi.
Source: telegraph
Source_date: 31 October 1992
Certainty: Facts.
Relevant Information
          Police evacuated 10 Downing Street.

```

Fig. 2.3: Example of a template produced by the content scanning module

a student and teacher model during execution and a specially designed mixed grammar. Students are asked to perform translations, their input is parsed and the resulting parse is analysed for transfer errors. For an example of the Chinese tutor module, see Fig. 2.4.

- **Dialogue**

A dialogue model loosely based on Shank's script theory [Shank and Abelson 1977] has been implemented in the LOLITA system. An example of the dialogue application [Nettleton and Garigliano 1995] is shown in Fig. 2.5. Since communicative behaviour in a dialogue, e.g. the appropriateness of a response or the right to initiate dialogue and to change the topic, depends mainly on different situations and contexts in which the dialogue takes place, and since there are, theoretically, an infinite number of possible contexts or dialogue situations, the dialogue module of LOLITA makes use of prototypical

```

7b
We have breakfast at eight.

Please enter answer and hit return: 我们吃早饭八点
I'm sorry to say that you haven't translated the sentence correctly.
There may be more than one way of translating the sentence:
我们八点吃早饭. 八点钟我们吃早饭.
The problem with your translation is that you have translated the
sentence according to the English word order. Please compare your
translation with the standard one(s) in order to find out the dif-
ferences between the structures and try again!

They(male) have lunch at one.

Please enter answer and hit return: █
英文输入 (ASCII input)

```

Fig. 2.4: Sample output from the Chinese tutor application

Dialogue Structure Models (DSMs), representing a class of dialogues. DSMs are composed of Dialogue Structure Elements (DSE), each of which describes a fundamental property of a dialogue. The Dominance DSE, for example, comprises such dialogue constraints as whether it is permissible to initiate and terminate a (sub-) dialogue or, to interrupt other participants (by choosing a different topic or referring to a previously encountered topic).

```

LOLITA 1: I desire a success.
Input 1: I like you.
LOLITA 2: I know; why do you like me?
Input 2: You are beautiful.
LOLITA 3: I am glad because you think I am beautiful.
Input 3: Do you like me?
LOLITA 4: Yes, I like you.
Input 4: Do you like me?
LOLITA 5: I slightly am bored and emotional.
Input 5: You are a machine.
LOLITA 6: I do not believe that I am a machine.
Input 6: Are you a machine?
LOLITA 7: No.
Input 7: Are you beautiful?
LOLITA 8: I do not know if I am beautiful.
Input 8: I am tired.
LOLITA 9: Tell me things.
Input 9: I am tired.
LOLITA 10: I slightly am bored.
Input 10 : Are you tired?
LOLITA 11: No.

```

Fig. 2.5: Example of the dialogue application

- **Query and database interfacing**

This application allows information to be entered into the LOLITA knowledge-base SemNet in the form of NL either by typing in sentences or by

having the system analyse files in batch mode, e.g. newspaper articles. The information extracted from the input is stored in the SemNet and can be retrieved using NL input, by typing in a question relating to information contained in the original input. For an example of the query application, see Fig. 2.6.

Input 1: do I own a motorbike?	Input 11: and Keith too.
LOLITA 1: Yes, you own a big fast motorbike.	LOLITA 11: I understand.
Input 2: do I control a vehicle?	Input 12: do I like you?
LOLITA 2: Yes, you own a big fast motorbike.	LOLITA 12: Yes, you like me.
Input 3: what do I use?	Input 13: whom do I like?
LOLITA 3: An office.	LOLITA 13: Keith and me.
Input 4: how many vehicles do I own?	Input 14: Keith
LOLITA 4: At least 1.	LOLITA 14: What about Keith?
Input 5: What do I own?	Input 15: you admire him
LOLITA 5: A big fast motorbike and a home.	LOLITA 15: I understand.
Input 6: why did I go home?	Input 16: whom do you admire?
LOLITA 6: You went to your home because you were tired.	LOLITA 16: Keith.
Input 7: I was owned by a motorbike	Input 17: I like you.
LOLITA 7: I do not believe that a motorbike owned you.	LOLITA 17: I know.
Input 8: was I owned by a motorbike?	Input 18: I hate you
LOLITA 8: no.	LOLITA 18: I do not believe that you hate me.
Input 9: do I like you?	Input 19: do you know that I like you?
LOLITA 9: I do not know if you like me.	LOLITA 19: Yes, you like me.
Input 10: I like you	Input 20: does Rick know that I like you?
LOLITA 10: I understand.	LOLITA 20: I do not know if Rick knows that you like me.

Fig. 2.6: Example of the query application

2.4 Summary

In this chapter, the framework of the undertaking to be presented in this thesis has been introduced. NLE as well as the criteria for a system that follows NLE principles have been described. The LOLITA system, its architecture, applications and prototypes have been presented in order to allow for the work described in this thesis to be seen in its context.

Chapter 3

Related work

This chapter reviews related work in the field of figurative language understanding. As the work presented in this thesis was conducted with an open mind towards suggestions from any direction, findings and theories from disciplines as diverse as psychology, philosophy, computer science, computational linguistics and cognitive science have been taken into account. However, as the scope of this work is limited, emphasis will be put on work that is closely related to the problem central to this thesis, i.e. feasible computational treatment of figurative natural language. Approaches that were deemed to be promising and of relevance to this work will be presented and critically evaluated.

There are, moreover, further advantages to be gained through a critical survey of work in related areas. It helps to establish and define the tasks that constitute what has been referred to, rather vaguely, as figurative language processing and ensures the current work takes into account the latest results, avoiding garden paths and reinventing the wheel; it delineates the boundaries of the field as it stands. It also serves as a means of evaluating the work presented in this thesis providing a host of results for comparison. Furthermore, it allows us to group together the vast array of different approaches according to schools of thought and ways of tackling problems in order to clarify the state of the art with regard to theory and advances in sub-areas.

3.1 Classical accounts of metaphor

Aristotle

Although it is not the first scholarly account of figurative language (both Plato and Socrates wrote about figurative speech and its use in rhetoric), Aristotle's thoughts on the topic had a great impact upon later studies of figurative language. The field has freed itself from them, in parts, not even today. His first detailed account achieved the status of definition and was, apart from some exceptions (e.g. [Nietzsche 1974]), not questioned until this century. In the *Poetics* XXI, 7-14, he says:

Metaphor consists in giving the thing a name that belongs to something else; the transference being either from genus to species, or from species to genus, or from species to species, or on grounds of analogy. That from genus to species is exemplified in 'Here stands my ship'; for lying at anchor is a sort of standing. That from species to genus in 'Truly ten thousand good deeds has Ulysses wrought', where 'ten thousand', which is a particular large number, is put in the place of the generic 'a large number'. That from species to species in 'Drawing of the life with bronze', and in 'Severing with the enduring bronze'; where the poet uses 'draw' in the sense of 'sever' and 'sever' in that of 'draw', both words meaning to 'take away' something. That from analogy is possible whenever there are four terms so related that the second is to the first, as the fourth to the third; for one may then put the fourth in place of the second, and the second in place of the fourth. Now and then, too, they qualify the metaphor by adding on to it that to which the word it supplants is relative. Thus a cup is in relation to Dionysus what a shield is to Ares. The cup accordingly will be described as the 'shield of Dionysus' and the shield as the 'cup of Ares'. Or take another instance: As old age is to life, so is evening to day. One will accordingly describe evening as the 'old age of the day' – or by the Empedoclean equivalent; and old age as the 'evening' or 'sunset of life'

Thus, Aristotle sets the scene for central questions such as what is metaphorical, how metaphors are structured, how metaphors are generated and how figurative language is related to everyday language.

From this quotation, we can derive that Aristotle sees metaphor as deviant in so far as it is "giving the thing a name that belongs to something else" and as a substitution: Metaphor seems to occur on the level of lexical items, with a term commonly used for one thing being replaced by another term, in accordance

with the four principles stated, namely, genus to species, species to genus, species to species and analogy, where analogy is based on similarities between the two terms. In a later paragraph, metaphor is described as a means of allowing poets to avoid everyday expressions and thus showing their talent and originality, and therefore belonging outside the realm of everyday language. Since the structure of metaphor or metaphoric replacement of one term by another follows the four principles, interpretation amounts to finding the term which has been replaced along the species–genus path or uncovering the similarities in the case of analogy.

Cicero

Aristotle's successors, most notably Cicero and Quintilian, elaborate on Aristotle's notions. In *De Oratore*, Cicero writes:

There are then three things which the orator contributes in the matter of mere vocabulary towards the decoration and embellishment of his style – rare words, new coinages, and words used metaphorically. [Cicero 1942]

Here, we see that metaphor is thought of as a stylistic device, used for making the style of speech more colourful. He goes on to say:

The third method in our list, the use of metaphor, is of wide application; it sprang from necessity due to the poverty and deficiency, but it has been subsequently made popular by its agreeable and entertaining quality. [ibid.]

Again, figurative speech is seen as an effective means of conveying a meaning in a more agreeable and entertaining way, but now a reason for the use of metaphor is also given, i.e. the necessity to express something which could have not been expressed equally well, or in an equally pleasing manner in literal terms. Cicero further states that:

A metaphor is a short form of simile, contracted into one word; this word is put in a position not belonging to it as if it were its own place, and if it is recognisable it gives pleasure, but if it contains no similarity it is rejected. [...] Occasionally also metaphors serve to achieve brevity. [ibid.]

Here, we are given the view that a metaphor is indeed a shortened form of a comparison or simile to aid brevity and thus the overall impact of the text. Following this view, interpretation would involve expanding or unpacking the simile. Cicero again tells us about the deviant nature of metaphor:

in fact the metaphor ought to have an apologetic air, so as to look as if it had entered a place that does not belong to it with a proper introduction, not taken it by storm, and as if had come with permission, not forced its way in. But there is no mode of speech more effective in the case of single words, and none that adds more brilliance to the style [ibid.]

The improper role of metaphorical expressions is once again emphasised, the notion of not belonging evoked. The dimension of the impact of a figurative expression is mentioned and the notion that metaphor belongs to the realm of substituting words occurs again. However, a little elaboration on the relationship between the proper word and what is expressed is also given:

[In] the figure of substitution or metonymy [...] one proper name is substituted for another [...] The method is effective in ornamenting the style, and should often be adopted; and to the same class belong the phrase *the impartiality of the War-god* and the use of the terms *Ceres* for corn, *Liber* for wine, *Neptune* for the sea, *the House* for parliament, *the polling booth* for elections [...] and also in the same class is the use of names of the virtues and vices to stand for the people who possess them [...] akin to it are the less decorative but nevertheless not negligible figures employed when we desire a part to be understood to mean a whole [...] or else a whole to mean a part. [ibid.]

The term “metonymy” is introduced and defining criteria are given, i.e. substitution of one term by a concept closely related to this term, e.g. a part for the whole or vice versa.

Quintilian

Quintilian introduces the aspect of the naturalness of metaphor, which has not been acknowledged before, but also emphasises the decorative function of figurative language when he notes in the *Institutio oratoria*:

the commonest [...] of tropes, namely, metaphor [...] is not merely so natural a turn of speech that it is often employed unconsciously or

by uneducated persons, but it is in itself so attractive and elegant that however distinguished the language in which it is embedded it shines forth with a light that is all its own. [...] A noun or verb is transferred from the place to which it properly belongs to another where there is either no literal term or the transferred is better than the literal. We do this either because it is necessary or to make our meaning clearer or, as I have already said, to produce a decorative effect. [Quintilian 1922]

He then elaborates on the point about similes made by Cicero and gives his classification of metaphoric structure:

On the whole metaphor is a shorter form of simile, while there is this further difference, that in the latter we compare some object to the things which we wish to describe, whereas in the former this object is actually substituted for the thing. It is a comparison when I say that a man did something *like a lion*, it is a metaphor when I say of him *He is a lion*. Metaphors fall into four classes. In the first we substitute one living thing for another, as in the passage where the poet, speaking of the charioteer, says, 'The steersman then // With mighty effort wrenched his charger round,' or when Livy says that Scipio was continually barked at by Cato. Secondly, inanimate things may be substituted for inanimate, as in the Virgilian, 'And gave his fleet the rein,' or inanimate may be substituted for animate, as in 'Did the Argive bulwark fall by sword or fate?', or animate for inanimate, as in the following lines: 'The shepherd sits unknowing on the height // Listening the roar from some far mountain brow'. [ibid.]

Again, metaphor is regarded as being interpretable by expanding a shortened comparison or by looking for the replaced term along the four lines of permissible replacements. Although Quintilian seems to take note of the abundance of metaphors, even in everyday speech, he goes on to say:

The worst errors of all, however, originate in the fact that some authors regard it as permissible to use even in prose any metaphors that are allowed to poets [...] For metaphor should always occupy a place already vacant, or [...] should be more impressive than that which it displaces. [ibid.]

Here again, he stresses the decorative and stylistic aspect and places metaphor proper (as opposed to the metaphors arising from the need of expressing something for which there is no literal term available) in the realm of poetry. He goes on to give a more detailed classification of tropes, including synecdoche, which is in his

words “making us realise many things from one, the whole from a part, the genus from a species, things which follow from things which have preceded; or, on the other hand, the whole procedure may be reversed” [ibid.]; metonymy, which is the substitution of one term for another, e.g. “to indicate an invention by substituting the name of the inventor, or a possession by substituting the name of the possessor” [ibid.]; and others such as the container replacing the content, the cause the effect, a plural for the singular. Definitions and examples are also given for other tropes such as irony, hyperbole and allegory, the latter being described by him as a chain of metaphors.

3.1.1 Deviance and substitution view

Incongruent as they may seem, all the cited classical authors deal with the (multifaceted) problem of metaphor, and we can now summarise their view on certain key points relating to figurative language:

- What is metaphor? Metaphor operates on the level of words and involves two entities. One literal term, a noun or verb is substituted by a figurative term that does not really belong in the place it occupies. Some types of metaphor are thought of as a shortened simile or comparison. The figures are deviations from the literal mode of speech and could be expressed using literal language.
- Is there a structure in metaphor? Aristotle speaks of general relations along an ontological hierarchy and analogy-driven substitutions, the others elaborate on various fixed relations between the original term and the new term, such as species–genus, inventor–invention, cause–effect, i.e. in general a semantic proximity following a well defined path chosen from a set of possible paths. This set of possible substitution patterns would amount to the set of possible metaphors.
- How are metaphors interpreted? None of the classical authors explicitly states how metaphors are interpreted. Their main concern seems to be the provision of an explanation on how pleasing and successful metaphoric expressions are achieved and used, i.e. the generation of metaphors. It seems obvious that, when a metaphor is used, the term most likely to have been replaced by the figurative term will be re-substituted by the hearer/reader and thus the metaphor interpreted.

- Why are metaphors used? Although there is mention of a necessity for metaphor (in cases where no literal term is available), the authors cited agree that the foremost function of figurative language is to improve poetry and artful speech to make it more pleasing and to aid brevity of expression.
- How can metaphors be classified? The terms metaphor, synecdoche and metonymy are introduced, the first being a general term for the substitution of words, the second a substitution along the lines of genus–species and other ontological relations, and the latter a substitution not along an ontological but a conventional path, of a general semantic association (e.g. inventor for invention).

No significant work was to follow these classical views for centuries, one of the reasons being that they were considered to be ultimate definitions. As with many disciplines, the theories of antique times were not questioned but regarded as definitive explanations and canonical truths. Furthermore, the scholarly treatment of language concerned itself mostly with (prescriptive) grammar and rhetoric, and therefore, as Meier observed, emphasis was put on didactic approaches rather than seeking new, objective insights [Meier 1963].

The classical view can be termed ‘deviance view’ insofar as it regards metaphor as deviant from the literal basis of language or ‘substitution view’ with respect to the fact that a figure is thought of as being an expression in which a literal term has been substituted by a figuratively used one in order to achieve a certain effect. Both views are still alive today, in linguistics and philosophy alike, and have spawned other views on metaphor which concentrate on aspects of the deviance and substitution view.

3.2 Modern accounts

3.2.1 Tension view

The tension view is based on the notion that figurative language is deviant and false if taken literally¹. It thus tries to explain the impact a metaphor has on the hearer by assuming that the falsehood encountered makes the reader/hearer want

¹One cannot literally drink a cup and no man literally is a lion.

to resolve the metaphor. The puzzle-like quality of a metaphor creates the desire (tension) for resolution and makes the figure aesthetically pleasing (if it can be resolved); the inability to resolve a metaphor, however, results in its rejection and categorisation as inappropriate. Way notes that the tension view also accounts for language change, because the frequent use of a combination of terms reduces the tension. “Thus, old or dead metaphors become a part of language because of a loss in emotive tension” [Way 1991].

The tension view subscribes to the substitution and deviance notions of the classical view as well as to the idea that metaphor primarily serves a decorative function. However, it also goes beyond this to explain the mechanisms behind this aesthetic function, thereby giving a glimpse of how metaphors are recognised, too. If an expression is considered strange or false in some way, the hearer tries to overcome the strangeness by classifying the expression as metaphorical and interpreting it according to rules different from the ones used for literal language.

3.2.2 Anomaly views

The anomaly view elaborates on the notion that metaphor somehow violates the rules of literal language and results in a semantic or pragmatic anomaly. We shall analyse one typical account in order to evaluate what it can contribute to the solution of our problem. Levin is one of the proponents of the feature addition/deletion view, which is derived from the deviance view [Levin 1977].

3.2.2.1 Feature addition/deletion view

The feature addition/deletion view operates in the realm of compositional semantics [Katz and Fodor 1963, Katz 1966, Katz 1967, Charniak 1981, Lyons 1977]. The basic assumption of compositional semantics is that the meaning of a larger linguistic unit such as a phrase or sentence can be constructed from the meaning of its subparts, where the meaning of the subparts, lexical items, is regarded as consisting of a bundle of features. These features are regarded as universal semantic primitives and a set of features or markers sufficiently describes the lexical traits of one meaning of a word. ‘Bachelor’, for example, is described in terms of [+human], [+male] and [-married]. Although there is no generally accepted hierarchy of semantic features, compositional semantics is a widely used approach (e.g. for pol-

ysemy resolution) and adequately describes semantic relations such as antonymy, synonymy and similarity.

Levin makes the distinction between pragmatic and semantic deviance, pragmatic deviance being the transgression of norms of language use. “Inasmuch as there are norms, they can be transgressed and, when the norms pertain to the participants in speech acts or the role played in such acts by the non-linguistic context, we may regard any such transgression as a form of pragmatic deviance” [Levin 1977]. Semantic deviance is the violation of selectional restrictions, i.e. the combination of words whose meanings do not fit together. “By semantic deviance we mean that type which results from an ‘improper’ collocation of lexical items; viz. *Green ideas sleep furiously*.” [ibid.].

He introduces the notion of pragmatic deviance to account for sentences which are semantically non-deviant and whose meaning can nevertheless not be derived from the compositionality principle [Frege 1952]; the latter states that the meaning a sentence can be understood without knowledge about the context in which it was uttered, solely on the basis of the meaning of the components and rules governing the combination of those meanings.

According to Levin’s theory, recognition of metaphor is triggered by the realisation that an expression is deviant (pragmatically or semantically), and interpretation of the deviant utterance is achieved by following the rules of construals. Metaphors involve the addition, deletion or transfer of semantic features from one of the involved terms to the other. One example which serves as explanation of how metaphors ‘fade’, i.e. lose their novelty, illustrates what he means by this. The expression “My brother devoured three books this morning” is deviant insofar as “*book* does not contain the feature [+food] normally required by *devour*. One of the ways for [this sentence] to have been construed originally would be by deleting (neutralising) the feature [+food], leaving as features in the reading of *devour* [+rapid, +intensive], features which are compatible with *book*” [ibid.]. He goes on to say that through continued use, this change in the feature set of ‘devour’, i.e. the metaphorical reading, is conventionalised and becomes part of the word’s meaning, making the metaphor less novel.

The main part of his theory (T) consists of six modes of construal, examples of which will be shown in detail below, according to which all metaphoric meaning can be understood, i.e. any metaphor can be explained through one or more of the construals. The construals operate on the features of the terms used metaphorically,

transferring or deleting them, as shown in the following examples:

1. Adjoining features of the verb to the noun and disjoining them in the overall semantic representation of the sentence. Levin's example 'The stone died' would in this construal lead to an attempt at finding something which satisfies [+animate] (because of 'died') as well as [+mineral] ('stone'), such as 'natural physical object', thus resulting in the construal 'the natural physical object died'.
2. Adjoining features of the verb to the noun and cojoining them in the overall semantic representation of the example 'The stone died' leads to the second construal in which the transferred feature is combined with the relevant feature of the other term, resulting in the reading 'the anthropomorphized stone died'
3. Disjunctively adjoining features of the noun to the verb followed by disjunction of these in the verb's selectional restrictions. If [+mineral] from the noun of 'The stone died' was adjoined to [+animate] from the verb and the generalisation of the disjuncts chosen, the resulting interpretation would be, according to Levin: 'The stone ceased to exist'.

The problems with this theory (T), anomaly views in general and the feature addition/deletion view in particular are that they concentrate on grammatically or, in Levin's case, semantically deviant sentences. Levin assumes that pragmatically deviant sentences are interpreted according to rules of language use (e.g. speech acts [Austin 1975, Searle 1969, Searle 1975, Searle 1979, Searle 1980] or conversational implicature [Grice 1975], and semantically deviant sentences are likely candidates for metaphors; he also declares that "it is exactly these anomalous (or deviant) sentences² that are the concern of a theory of metaphor" [Levin 1977]. There are, however, many sentences one would assume to be metaphorical which do not exhibit anomalies, e.g. negations such as 'No man is an island', 'He isn't living in the fast lane', or idioms such as 'He kicked the bucket' and 'They broke the ice'. Whereas idioms would, presumably, be treated under the heading of pragmatic deviance (in a context where 'They broke the ice' was uttered and it was evident that no breaking and no ice are involved, a pragmatic interpretation

²Sentences exhibiting "the failure of one or more constituents to satisfy the selection restriction of another" [ibid.].

would be performed), there are other cases – such as the much-cited ‘The rock is becoming brittle with age’, said of an old professor – which are, in isolation, semantically non-deviant. Although Levin acknowledges the role of context (taking the context into consideration would shift the example into the realm of pragmatics again), his claims that the construals are capable of explaining the meaning of all metaphors seems far fetched for the reasons that there are semantically non-deviant expressions that are metaphors.

Moreover, his theory allows only an a posteriori explication of how a meaning for a semantically deviant sentence might be constructed. It is not constrained or elaborate enough to state how (or indeed, which) feature addition/deletion processes take place and in which order.

To summarise, as far as recognition is concerned, Levin’s theory (T) is capable of recognising a subset of metaphors, those which exhibit semantic anomalies. The same set of expressions constitutes the answer to the question ‘What is metaphor?’. As far as the interpretation of metaphors is concerned, theory (T) does not provide much help because the final interpretation has to be given first, *before* the feature transfer mechanism can be consulted to give us an idea of how the interpretation might have come about.

The shortcomings of the feature addition/deletion thesis seem, therefore, to stem from the fact that it considers the metaphorical unit to be a sentence, i.e. ignores context and cotext (pragmatic deviance resolved by other means) and believes the metaphoricity to reside in single words whose features are the cause for the metaphoricity. It is, in this respect, a descendant of the classical views.

3.2.2.2 Constraint violation view

The anomaly view has been carried over into artificial intelligence (AI) and natural language processing (NLP) in the form of the constraint violation or selection restriction violation view. The theory of metaphor put forward by Wilks [Wilks 1975a, Wilks 1975b, Wilks 1978] borrows from both the anomaly view and feature addition/deletion view, but firmly rejects the binary distinction between acceptable and unacceptable readings imposed by the application of selectional restrictions [Katz and Fodor 1963]. In his framework of preference semantics, Wilks replaces the fixed and inflexible semantic restrictions with semantic preferences to allow processing of utterances which are semantically not well-formed.

For rejecting utterances is just what humans do not do. They try to understand them. [...] It is proper to prefer the normal, but it would be absurd, in an intelligent understanding system, not to accept the abnormal if it is described. Not only everyday metaphor, but the description of the simplest fictions, require it. [Wilks 1975b]

From this quotation, we can derive that Wilks acknowledges the need for robust broad coverage understanding and the extent to which everyday language does not conform to the narrow limits of selectional restrictions. He also regards metaphor as an everyday phenomenon and not a rare occurrence, which has to be treated adequately in an NLP system.

Wilks

Initially [Wilks 1975b], the preference semantics approach dealt only with the recognition of metaphors and sentences described as anomalous by the tight constraints of generative semantics, and the acceptance but not the interpretation of these in the input. Instead of plainly rejecting anomalous sentences, an attempt is made to find the ‘best-fit’ for possible meanings. This is achieved through semantic formulae stating the preferred semantics of words. The structured formulae themselves consist of semantic primitives “used to express the semantic entities, states, qualities and actions about which humans speak and write”[ibid.] The primitives are organised in classes such as entities (e.g. human being, part-of and state), actions (e.g. force, flow or exist), sorts (e.g. being a container or being inanimate) and cases (location, agent or goal). For each word in the input, one or more formulae relating to different word senses, e.g. crook [bad person] vs. crook [shepherd’s staff], is examined, and the heads of those formula are checked for compatibility. For an illustration of this, see Fig. 3.1.

Semantic density is computed on the basis of how many incompatibilities the given input exhibited when completely processed, and the most dense reading is taken to be the preferred one. Thus the most probable – if anomaly makes a reading improbable – overall meaning for the input is computed, based on the mutual influence words and higher syntactic structures have on each other. This procedure allows acceptance of anomalous input, attachment of grades of acceptability to possible readings and recognition of figurative speech if it exhibits a semantic anomaly.

In order to be able to distinguish between anomaly and metaphor, Wilks aug-

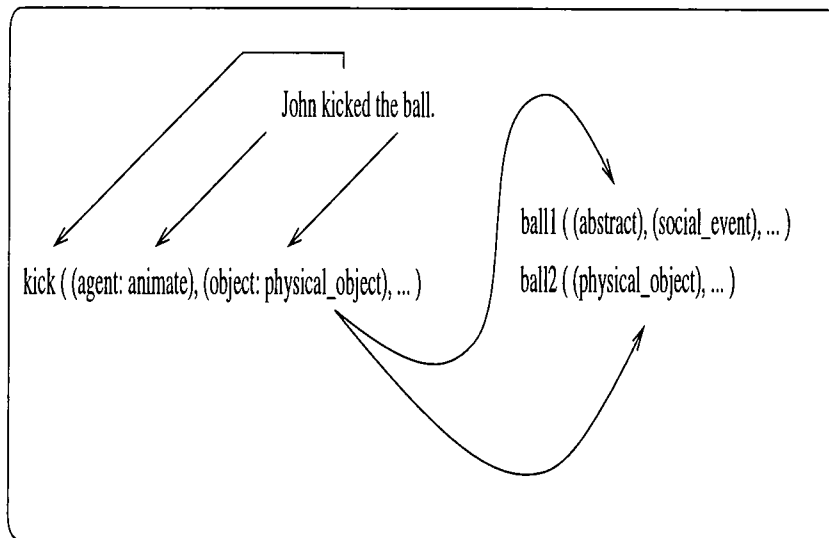


Fig. 3.1: Example of a semantic formula in preference semantics

mented his preference semantics with a knowledge structure called pseudo-text [Wilks 1978]. Pseudo-texts are episodic or contextual knowledge structures similar to scripts [Shank and Abelson 1977]. A pseudo-text for cars, for example, might contain information about the material a car consists of, its prototypical shape, the fact that it needs fuel and maintenance at regular intervals etc. The violation of a semantic preference triggers an algorithm called ‘projection’. The projection tries to find a match from the pseudo-text which, when replacing the part of the input that caused the preference violation, results in the constraints being satisfied.

This can be described using Wilks’ much cited example ‘My car drinks gasoline’. In preference semantics, ‘drink’ requires its agent to be animate, thus leading to a constraint violation in this example. Pseudo-text lookup (matching the offending part of the input, i.e. ‘drink’ against the whole of the knowledge about cars) will select all possible replacements, amongst them being that ‘cars use gasoline’. These are then processed further by the usual mechanisms, i.e. the ‘best-fit’ is selected on the basis of preferences. In this respect, the constraint violation view can be seen as related to the feature addition/deletion view. Features from the anomalous input are changed or replaced to satisfy the constraints: if one adopted the semantic feature ‘stance’, ‘drink’ could be described in terms of the features of ‘consume’, requiring an additional feature of its agent, namely [+animate].

Fass

An elaboration of the constraint violation view, resulting in a comprehensive implementation of the principles mentioned is Fass' *met** method³ [Fass 1991]. It is based on an extension of Wilks' preference semantics called collative semantics (CS) [Fass 1988]. Fass' framework deals with the recognition and classification of input as being metonymical, metaphorical, literal or anomalous, making use of an extended version of the preferences described above. However, instead of pseudo-texts, which play a role in CS the form of noun-sense frames (the word sense entries for nouns in the lexicon), Fass uses an abstraction hierarchy to recognise and interpret metaphors and metonymies. This abstraction hierarchy consists of the linked word senses (the dictionary entries for words, which, in turn, consist of 'cells') and virtue to the fact that more specialised word senses are composed of more generic ones, form a sub-/ supertype hierarchy. To give an example from [Fass 1991], one possible sense of 'enter', *enter* would be the supertype of *ingest*; *eat* and *drink* would both be subtypes of *enter* (see Fig. 3.2). Fass' assumption is that, given an example like 'My car drinks gasoline', the literal meaning will be a sibling of the metaphorical input concept, in Fass' terms: "A relevant analogy⁴ is indicated by a sister match of the source's relevant cell"[*ibid.*].

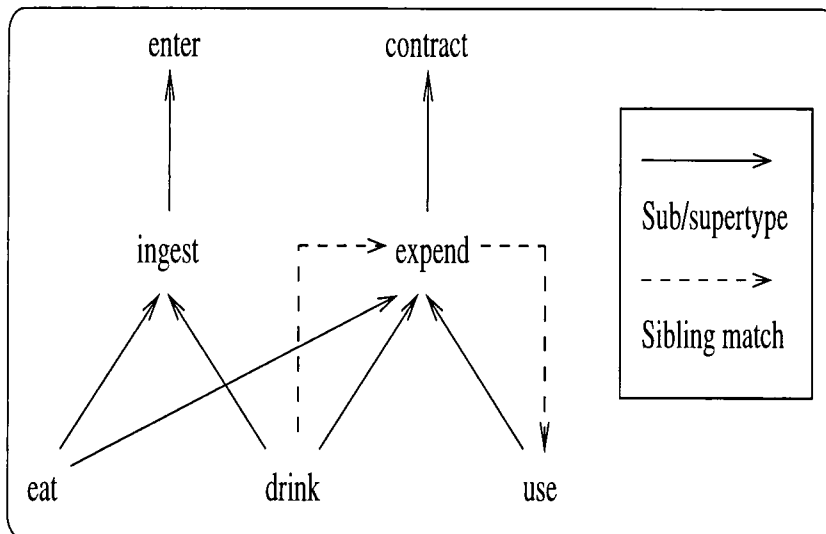


Fig. 3.2: Example of the type hierarchy in the *met** method

In the above example, according to Fass, one sense of both *drink* and *use* share

³Read 'met star', where star is to be taken as the regular expression wildcard, thus matching metonymy and metaphor alike.

⁴i.e. a metaphorical reading

the common supertype *expend*. A lookup of senses closely related to the sense(s) of the source (input) will thus lead to possible interpretations, which in turn can be evaluated using preferences (c.f. Fig. 3.2).

Another extension to the model of the recognition/interpretation presented by Wilks is the possibility of distinguishing between metaphor and metonymy. If the input is considered anomalous on the basis of violated constraints, a set of applicable ‘metonymic inferences’ is applied to it. These inferences are drawn on the basis of a set of fixed “knowledge-specific relationships between a concept and an aspect of another concept” [Fass 1991] which are coded into Fass’ implementation as metonymic inference rules of the form *rule(source, target)* and explicitly state the relationship that must hold between the input and a possible metonymic replacement, such as container for contents or artist for art form. Part of the *met** method is to find a path through the network of senses which will be in keeping with the metonymic inference rules.

It has to be noted that Fass exhibits a strong preference towards the literal reading of any given input. His *met** method operates as follows: If a literal reading can be found, exit. If not, apply metonymic inference rules. If this proves successful (possibly after iterating to resolve a chain of metonymies), substitute the result for the ‘offending’ part of the input and exit. If this is not successful, try to find a ‘relevant analogy’ i.e. a metaphorical interpretation. This can either lead to success or failure; in the latter case, the input is classified as anomalous.

This is an obvious point for further discussion: as seen above, there are semantically non-deviant sentences which would not violate any constraints, but yet have either an additional metaphorical reading or only a metaphorical reading. The role of context, as in ‘The rock is becoming brittle with age’ is a major one. Fass addresses this point by arguing that if the *met** method were viewed in parallel processing terms, literal and figurative interpretations would be sought at the same time. This seems to contradict his outline of the *met** method. If there is no constraint violation, no further readings are sought; and by obtaining multiple readings at the same time, the whole point of the *met** method, i.e. distinguishing between various types of semantic well-formedness by applying test in a certain order (satisfied preferences, metonymic inference possible, relevant analogy found), is made obsolete. Thus sentences with a literal and figurative reading will, as CS and *met** now stand, only ever receive a literal one if the role of context is not introduced into CS and multiple interpretations considered in the *met** method.

Another problem is Fass' notion that the preferences and possible metaphorical interpretations are tied to the main verb of the input, and the relevant analogy is expected to reside in the semantic neighbourhood of this part of the input. Although this is true of the 'car drinking gasoline' example, consider 'On hearing the news, he exploded'. Supertypes of `exploding` would presumably include `expand`, `self-destruct` and similar ones, but not necessarily exhibit suddenness (or any concept relating to anger). This means that, firstly, the quality of the recognition and interpretation will vary to a great extent in accordance with the information available to the algorithm in the form of a concept hierarchy. Secondly, the 'some up, some down' approach of finding a sister concept will not work well when the distance between the input and the common ancestor is too great, i.e. the 'relevant analogy' very general.

In summary, the `met*` method can be seen as a successful implementation of an advanced version of preference semantics and the classical view on metaphor, although in a limited framework; Fass gives the following implementation details: "The `meta5` program is written in Quintus Prolog and consists of a lexicon holding the sense-frames of just over 500 word senses, a small grammar, and semantic routines that embody [...] CS" [Fass 1991]. Possible metonymical relations (cf. Sect. 3.1) are explicitly stated in the form of metonymic inference rules, and metaphors are resolved using Aristotle's 'analogy' type, by finding the basis of the analogy in the shared supertype of both. Where Aristotle said 'As old age is to life, so is evening to day', Fass might say 'As drinking is to animates, consuming is to cars'.

It is a useful tool for the recognition of a subset of metaphors and sheds light on the issue of how metonymies can be recognised and interpreted, an issue which had not been addressed in depth prior to Fass' `met*` method.

3.2.3 Comparison view

A view that is reflected in both Wilks' and Fass' approaches is the comparison view. It is derived directly from the classical views on metaphor in that it emphasises the similarities between the terms involved in a metaphorical expression. It regards metaphor as a shortened comparison or simile, concentrates on pre-existing similarities between the parts of the metaphor and assumes that a literal paraphrase of metaphors could be given. For example, in saying 'A is B', the speaker means 'A is

like B (in certain respects)’ and that the ‘certain respects’ carry the meaning and have to be found in order to arrive at the interpretation of the metaphor. Using Aristotle’s examples of the “old age of the day”, the similarity between life and a day is that both span a certain amount of time, have a beginning and an end, and progress continuously. Furthermore, since ‘old age’ denotes the later stages in life, it can be used to mean the later stages of the day, namely, evening. This is basically an elaboration of the substitution view, not merely stating that a figurative expression is substituted for a literal one, but also giving an account of how the terms used relate to each other. It should be noted that, by its very nature, this view concerns itself exclusively with metaphors of the ‘A is B’ form or comparisons. It cannot explain phenomena such as metonymy, since the form of metonymies does not conform to the ‘A is B’ schema. Its reflection in the selectional restriction view comes in the form of Wilks’ projection and Fass’ relevant analogy, which are both embodiments of the similarity emphasised in the comparison view.

3.2.3.1 Analogy approaches

A school of approaches closely related to the comparison view, which have received much attention in AI are the analogy approaches. They can be seen as a further elaboration of the fundamental notion of the comparison view which assumes that finding the similarity between the two terms involved holds the key to understanding metaphor. Their main concern is the modelling and explication of phenomena such as analogical reasoning and learning by analogy [Gentner 1983, Gentner 1989, Holyoak and Thagard 1989b, Long and Garigliano 1994], but the underlying assumption is that the mechanisms of metaphor can be explained by the more general principles of analogy. This of course entails that metaphor is seen as a special case of analogy. One work on analogical reasoning, which does not deal with metaphor and can therefore be regarded as unbiased, describes analogy as follows: “the interpretation of analogy is taken to be the inference that a property holds of one term or concept from the premises that the same property holds of another term or concept and that the two terms are alike in some relevant way” [Long and Garigliano 1994]. Clearly, this sounds like a restatement of the problems of metaphor, at least according to the comparison view.

Gentner

The questions of the properties holding for one term and the way in which this term is similar to another one are addressed by Gentner in her structure mapping theory [Gentner 1983, Gentner 1989]. Gentner describes analogy as a mapping from a base domain that serves as the source of knowledge to a target domain where the domains are represented as concepts and properties of those concepts. Properties can be of two types, differing in their predicate structure, namely attributes such as *large* (x) and relations such as *collide* (x,y). The structure mapping theory is based on three mapping rules, the discarding of attributes, the preservation of relations between concepts and the systematicity principle which prefers the mapping of systems of relations over that of isolated relations. Thus, the structure mapping theory favours systematic, (highly connected) relations between concepts rather than less connected relations, and these in turn are preferred to attributes as possible correspondences between domains. Depending on whether the ‘systematic relations first’ rules are satisfied, Gentner distinguishes different kinds of domain comparison; literal similarity (milk is like water), analogy (heat is like water), abstraction (heat flow is a through-variable), anomaly (coffee is like the solar system) and mere appearance (the glass tabletop gleamed like water) [Gentner 1989].

In a literal similarity statement, for example, many attributes and relations are mapped from the base onto the target, and the number of mapped properties is large relative to the number of non-mapped properties, e.g. a statement like “The X12 star system in the Andromeda galaxy is like our solar system” would not only entail that there is a central star and several planets orbiting it (relations), but also that attributes such as the size and colour of the X12 star are similar to the attributes of objects in our solar system [Gentner 1983]. In the case of an analogy, few if any attributes are mapped, but many relations, as in “The hydrogen atom is like our solar system” where attributes such as the size of the central object or its temperature, the number of planets etc. are not mapped, whereas many relations such as *attracts* (x,y), *revolves-around* (y,x) are carried over from the base into the target domain [Gentner 1989].

The relevance this has for metaphor is shown in the quotation “A number of different kinds of comparisons go under the term ‘metaphor’. Many (perhaps most) metaphors are predominantly relational comparisons, and are thus essentially analogies. [...] Although most metaphors are relationally focused, some are predominantly attribute matches” [ibid.].

The structure mapping engine

The Structure Mapping Engine (SME) is an implementation of Gentner's structure mapping theory. (It is described in detail by Falkenhainer et al. [Falkenhainer *et al.* 1989] and an overview is given by Gentner et al. [Gentner *et al.* 1989].) Its operation follows the principles stated in the structure mapping theory; a set of possible matches between the concepts and their properties of the base and target domain is generated and, once the match exhibiting the highest systematicity is found and selected, the (systematic) relations present in the base domain but missing in the target domain are transferred. In the example above, all relations holding between the entities in our solar system could thus be mapped onto the (underdetermined) domain of the X12 star system.

This implementation helps to highlight some problems of the structure mapping theory and the comparison view in general. For example, SME starts processing off as follows: "If two relations have the same name, create a match hypothesis" [Gentner *et al.* 1989]. The example presented is the metaphor 'She allowed life to waste like a tap left running', where a section of one domain is represented as shown in Fig. 3.3 and 3.4.

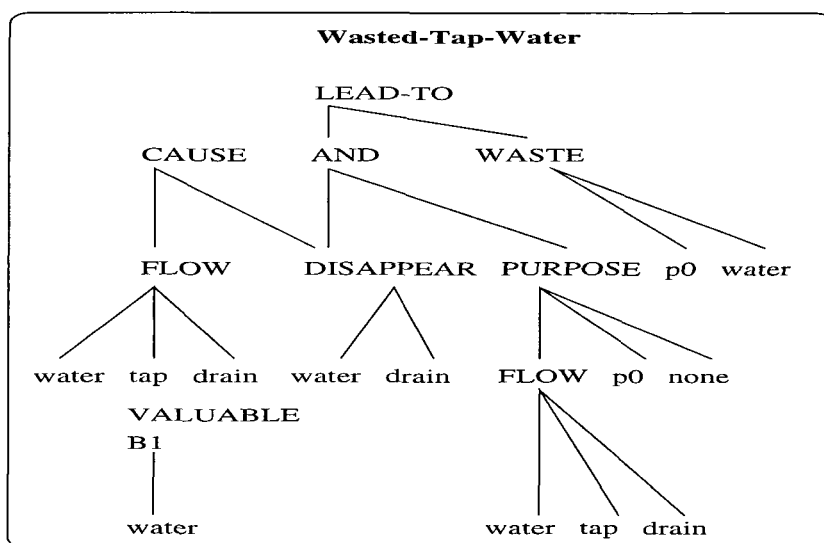


Fig. 3.3: Structure mapping: base domain representation

The implicit a priori similarity between the base and target domain with respect to the relevant relation is striking, and the representation seems to already incorporate metaphor interpretation, as life is metaphorically described as flowing from the present into the past. In other words, the analogy is only found on the basis of some pre-existing similarity, in this example the fact that life and running

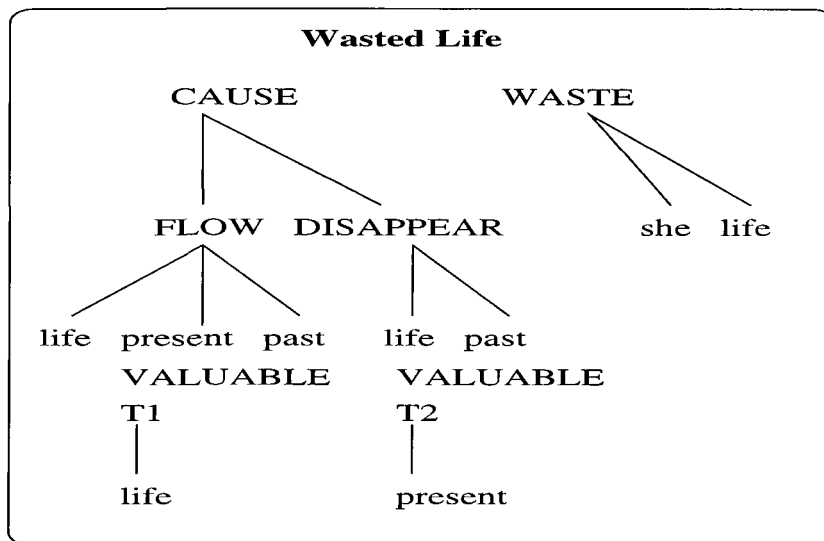


Fig. 3.4: Structure mapping: target domain representation

water are described as flowing. The fact that similarities are ‘uncovered’ leads to another problem; similarities are by definition symmetrical relations. If A is similar to B, then B is similar to A. If life flows away like unused water, than it should be possible to describe water being wasted in terms of a wasted life (‘She left the tap running like wasting life’, which has a very odd ring to it). It is not true that most metaphors, if any, are symmetrical. There is a significant difference in stating ‘all surgeons are butchers’ and ‘all butchers are surgeons’. Whereas surgeons are seen in a negative way in the first sentence, butchers are seen as skilled meat-cutters in the second sentence. This emphasises yet another problem of the SME approach: as the name suggests, the structure mapping theory’s and SME’s operation is purely structural, and they rely on the form of their input and take no notice of the content.

Salience imbalance

The fact that metaphors tend to exhibit asymmetry is addressed by Ortony’s salience imbalance theory, which forms part of his account of metaphor and can be classified as an instance of the comparison view. Ortony argues that similarity or comparison can take two forms, literal similarity and non-literal similarity [Ortony 1979]. He gives the following examples:

1. Encyclopaedias are like dictionaries.
2. Encyclopaedias are like gold mines.

Whereas, according to Ortony, (1) is an instance of a literal comparison statement, (2) is an example of a non-literal comparison. In making a comparison, as explained above, properties from one domain are transferred to another domain. In the first example, the properties of dictionaries are transferred or compared to the properties of encyclopaedias. There will be a considerable overlap between them and this will mean that a considerable amount of highly salient features from the base will be found to be highly salient in the target domain as well, e.g. dictionaries and encyclopaedias are both 'reference books' and 'consist of alphabetically sorted entries'. In the second example, however, there is no obvious overlap of properties, the highly salient features of gold mines might be 'filled with something valuable', 'a commercial enterprise', 'a subterranean location' and so on. Being full of something 'valuable' is true of encyclopaedias, too, but it is not the most outstanding feature, and, more importantly, metaphorically rather than literally true. This leads Ortony to the following thesis: "[In] a comparison 'A is like B', if high-salient predicates of B are also high-salient predicates of A then the comparison is a literal one and the two referents will be judged as being 'really' similar. If a high-salient predicate of B is a less-salient predicate of A while there are high-salient predicates of B that cannot be applied to A at all, then we have a simile" [Ortony 1979].

Metaphors (especially 'good' or 'interesting' ones) thus will exhibit a salience imbalance between the features of the two terms. Moreover, it is the salient features of the term 'B' in a comparison 'A is like B' that are of relevance in a metaphor. Since salience is conventionally fixed, a reversal of the terms will result in different features being central to the metaphor. The 'butcher-surgeon' metaphor mentioned can help to exemplify this; although 'cutting flesh', 'wearing white coats' and 'using sharp implements' is true of butchers and surgeons alike, highly salient features of butchers include the coarseness of their cutting, whereas skill and care certainly count as highly salient features of surgeons. The comparison 'all butchers are surgeons' therefore focuses on the skills, whereas the opposite is true of the comparison 'all surgeons are butchers'.

Summary of analogy views

To return to the structure mapping theory, as far as recognition is concerned it relies on the two terms of the comparison to be explicitly present in its input and cannot deal with cases where they are opaque or where there is only little information on

a concept available (e.g. only attributes).

On the computational side, SME also has some shortcomings. First of all, it will find all possible matches and transfer all reasonable (systematic and structural) relations, whereas metaphor serves to highlight only certain possible inferences and owes much of its effectiveness to the fact that it relates concepts which have not been previously related via a pre-existing (structural) similarity. Work by Tourangeau and Sternberg addresses this problem of ‘aptness’, i.e. the balance between the similarity that is necessary to find a common ground between the terms and the dissimilarity that is necessary to give the metaphor its effect [Tourangeau and Sternberg 1982]. They state that the aptness of metaphor depends on two measures of similarity. “The two kinds of similarity are within-domain similarity, or the degree to which two concepts occupy similar positions with respect to their own class or domain; and between-domain similarity, or the degree to which the classes or domains occupied by the concepts are themselves similar” [ibid.]. Similar domains, i.e. a great between-domain similarity, will result in less interesting or apt comparisons (the worst case being an intra-domain comparison). The greater the within-domain similarity (matching concepts that ‘play the same role’ within their domain such as the sun and the nucleus both occupying the centre of the solar system or the atomic structure, respectively), the higher the aptness.

Useful as this notion may seem, it has not been implemented. Moreover, the very complexity of SME, which requires another stage of processing comparing each of its input elements with all others (since the possible similarities are defined purely in structural terms in the SME framework) on top of a costly algorithm will make a possible implementation prohibitively expensive in terms of resources.

Furthermore, Carbonell cites evidence that the structure matching is NP-complete⁵ and any unguided matching will be computationally intractable, too [Carbonell and Minton 1985].

More generally, the comparison view (and SME, if representation does not come to its aid) will fail for another reason. Everything is similar to everything else in one way or another. If there is not enough knowledge about the base and target concepts and their properties available, a very general similarity will be found, but in metaphor it is not possible to combine any two concepts via any similarity.

⁵Nondeterministic Polynomial-time complete, e.g. all deterministic algorithms for the problem are exponential. This means that the execution time for n inputs and some constant C denoting an operation is C^n

Moreover, it is usually not the obvious similarity that is of interest and central in understanding a metaphor, but some more obscure, sometimes metaphorical similarity that ties two concepts together. Ortony's salience imbalance theory is the only representative of the comparison view that avoids those pitfalls, by giving criteria regarding the directedness of comparisons and the structure of the comparison. In addition, it accounts for cases where there is little knowledge of the target domain available, and most properties (metaphorically) attributed to the target will come from the base. Whereas the structure mapping theory and SME are unable to account for such configurations, Ortony terms such cases where features of the base are newly introduced into the target domain "predicate introduction metaphor" [Ortony 1979]. However, there exists no implementation of this process of 'predicate introduction', even though Carbonell's (theoretical) work, which will be outlined in the following aims at modelling the way in which features from the base can be re-instantiated or introduced into the target domain.

3.2.3.2 Invariance hierarchy

A different approach which has much in common with the analogy school of thought described above but which has a more refined view on metaphor and gives useful new impulses is the invariance hierarchy approach put forward by Carbonell [Carbonell 1982, Carbonell and Minton 1985]. His original interests in analogical and common-sense reasoning led him to the problems of metaphor; "Our use of the same terminology⁶ to describe metaphors, similes and analogies reflects our opinion that they are merely different linguistic manifestations of the same underlying process: analogical reasoning" [Carbonell and Minton 1985]. Although he too regards metaphor primarily as an analogy or comparison, he pays more attention to the computational needs of a metaphor comprehension system and gives a detailed account of those he considers necessary in a process model of metaphor understanding.

He rightly criticises the plain comparison view by stating: "Metaphors, similes and analogies are more than clever ways of restating the obvious. They are extraordinarily concise devices by which a writer can convey new information, simply by signalling his audience that information in the source domain is applicable in the target domain. [...] however, before the reader can initiate the transfer, he must

⁶Carbonell refers to 'source', 'target' and 'analogical mapping' for the domains and the correspondences between the concepts in the two domains.

identify the correspondences between the two domains and establish exactly what information in the source is applicable to the target, since much of it is clearly inappropriate.[...] Furthermore, when one considers requirements for computational tractability, it becomes evident that there must be strategies to help focus the comparison and constrain the matching process” [Carbonell and Minton 1985].

His major contribution to the field is the invariance hierarchy, which helps to narrow down the set of possible correspondences and can be regarded as a more concise filter for cross-domain mappings than Gentner’s systematicity principle. On the basis of empirical results of the relations and attributes that are mapped from the base onto the target domain, he claims that “[there] is a well-defined invariance hierarchy among the aspects of a situation that are mapped by a metaphor” [Carbonell 1982], i.e. there is a regularity with respect to which relations and attributes are carried over into the target domain, which are transformed and become metaphorical, and which are ignored in the transfer. He lists them in decreasing order of expected invariance as goals, planning strategies, causal structures, functional attributes, temporal orderings, natural tendencies, social roles, structural relations, descriptive properties and object identity. In the process of metaphor interpretation, these invariant structures are consulted in a top-down fashion. Thus an utterance like ‘John is a fox’ would not be taken to mean ‘John has pointed ears and a bushy tail’ but would rather be resolved as follows: “scan down the hierarchy stopping at the first entry for which we have a commonly known fact. For foxes, we stop at planning/counterplanning – folk wisdom tells us that foxes are very adept at devious counterplanning behaviour” [Carbonell 1982]. This attribute of ‘slyness’ would then be transferred to John, preventing a lower-match transfer from being made (e.g. descriptive properties such as ‘bushy tail’).

Unfortunately, it is not directly obvious how the invariance hierarchy can be exploited computationally. Carbonell himself states that the question of an appropriate representation is indeed difficult. Moreover, it will still only serve as a refinement for a computational treatment of metaphor in the framework of the analogy approach, whose shortcomings are discussed above. Nevertheless, Carbonell must be praised for the new ideas he has injected into the field of computational treatment of figurative language and the stance of both taking new results into account (such as the pervasiveness of conventional metaphors in everyday language, see Sect. 3.2.5) and realising the needs and requirements of any realistic computational approach. Particularly with respect to the determination of the base

and target domain and recognition, his theoretical work is partly influenced by the conventional view, which is discussed in Sect. 3.2.5

3.2.4 Interaction view

A theoretical modern view on metaphor which refuses most of the classical view is the interaction view put forward by Black [Black 1962, Black 1979] based on the philosophical work of Richards, which was presented as a series of lectures [Richards 1936]. Richards introduces technical terms to facilitate discussion about various aspects of metaphor, most notably ‘tenor’ and ‘vehicle’ for the two terms, words or concepts involved in a metaphorical utterance; the claim stemming from the classical view that metaphor involves two terms was never disputed, but an array of incompatible terminological material only contributed to confusion about the subject matter. As Richards notes:

One of the oddest of the many odd things about the whole topic is that we have no agreed distinguishing terms for these two halves of metaphor – in spite of the immense convenience, almost the necessity, of such terms if we are to make any analysis without confusion [ibid.].

‘Tenor’ denotes the part of a metaphorical utterance (which may or may not be explicitly mentioned) that is described in terms of another concept, the ‘vehicle’, which ‘carries’ the metaphor. According to Richards, not all metaphors consist of one tenor–vehicle pair, a secondary vehicle may be also present, as exemplified in the following:

A stubborn and unconquerable flame // Creeps in his veins and
drinks the streams of life

The tenor, which is not mentioned explicitly is ‘fever’, expressed in terms of the vehicle ‘flame’. But the flame is seen as ‘creeping’ and ‘drinking’, i.e. described in terms of a ‘secondary vehicle’, a stubborn, unstoppable, consuming animate agent. As an extension of Richards’ terminology, the terminus ‘ground’ has been introduced, denoting the attributes or features shared between the tenor and vehicle or, more generally, describing in which way the tenor resembles the vehicle. The ground in the example above would be ‘heat’, since the feverish body and a flame resemble one another in terms of heat.

The interaction view addresses problems that cannot be solved by the comparison view, which states that metaphor is based upon a pre-existing similarity. Such problems include the existence of metaphors where the terms exhibit no apparent similarity or the similarity itself is metaphorical (e.g. ‘He is an open book to me’) and the fact that it is possible to conceive of metaphors where the tenor and its features are largely unknown but rather introduced by the metaphor (‘Tom Brown is the Mother Theresa of functional programming’). The interaction view states that metaphor is not located at the level of words, but that of ideas and thus a metaphor such as ‘man is a wolf’ is concerned neither with replacing part of the expression with a literal paraphrase (e.g. ‘man is fierce and competitive’) nor simply with finding an analogy or pre-existing similarity (e.g. both man and wolves are fierce creatures living in a social structure, competing for power and status within this structure), but rather with associations made between the concepts. The ‘ideas’ are commonly shared beliefs about and associations with the words used in the metaphorical expression. Black calls these ideas ‘systems of commonplaces’: “These commonplaces are stereotypes, not necessarily definitional, not even necessarily true, just widely agreed upon. In interpreting ‘man is a wolf’, we evoke the wolf-system of related commonplaces and are led by them to construct a corresponding system of implications about the principal subject (man)” [Black 1962]. This ‘system of implications’ (from the domain of the vehicle) is employed to view the tenor in a metaphorical light, i.e. relations that hold in the wolf domain are used to re-organise the man domain. Black explains this process using a metaphor:

Suppose I look at the night sky through a piece of heavily smoked glass on which certain lines have been left clear. Then I shall see only the stars that can be made to lie on the lines previously prepared on the screen, and the stars I do see will be seen as organised by the screen’s structure. We can think of metaphor as such a screen and the system of ‘associated commonplaces’ of the focal world as the network of lines upon the screen [ibid.].

Seeing the domain of the tenor in terms of the commonplaces from the vehicle makes the two domains interact with each other, as this process of highlighting some relations and concepts and simultaneously ignoring others on the basis of one domain changes the way the tenor domain is viewed. Since those changes are not tied to single words or concepts but to a system of commonplaces, an array of changes occur in the tenor domain.

Thus, the fact that metaphors can create similarity, rather than stating pre-existing ones is explained, as the similarity in many cases will be the *result* of the domain interaction. The interaction view has been very influential, and many other approaches take its findings into account. It explains phenomena which other theories cannot account for – such as the creation of similarity through metaphor and the ambiguity in metaphorical meaning — by introducing the ‘corresponding system of implications’ which allows the reader/hearer to select any possible interpretation in accordance with the implications.

It has to be noted that, as with most other approaches, the interaction view concerns itself with ‘metaphor proper’ and gives no account of how other tropes such as metonymy and irony may be described. A more serious problem is that it is difficult to see how a process model could be derived from the findings of the interaction view; the central idea of ‘filtering’ leading to interaction is expressed in metaphorical terms and lacks explicitness. It is also not clear how the ‘corresponding system of implications about the principal subject’ can be constructed. The ‘related commonplaces’ are a priori indefinite and possibly infinite, yet not all are equally important or can indeed contribute towards the implications. The selection and importance of the commonplaces varies, depending on the tenor or context, as the contrast between ‘man is a wolf’ and ‘your dog is a wolf’ shows.

Kittay addressed the two problems of limiting the possible relations arising from the ‘system of implications’ and the attribution of importance to the ‘commonplaces’ in her semantic field theory of metaphor [Kittay 1987]. She argues that the metaphorical meaning can be traced to the introduction of relations from one semantic field (the field of the vehicle) into the field of the tenor, whereby the semantic fields are not universally fixed, but dynamic context dependent relations. She further makes use of syntactic relations between the concepts from the semantic fields, to predetermine possible matches between concepts from the tenor and vehicle domain (or field), as in her example of Socrates’ midwife metaphor (see Figs. 3.5 and 3.6).

Her work can be seen as a concretion of the rather abstract notions put forward by Richards and Black, but stays well within the framework of the interaction view. A toy implementation (NETMET) of her model has been made available [Steinhart 1993]. Its generation of a large number of match hypotheses and its limited applicability (input has to have the form of predicate-calculus notations) classify it as a theory evaluating model (which presumably is what it set out to

be), but nevertheless an interesting and well presented (self-guiding exercises, an interface for the incorporation of new semantic fields, general analogical processing capabilities) one.

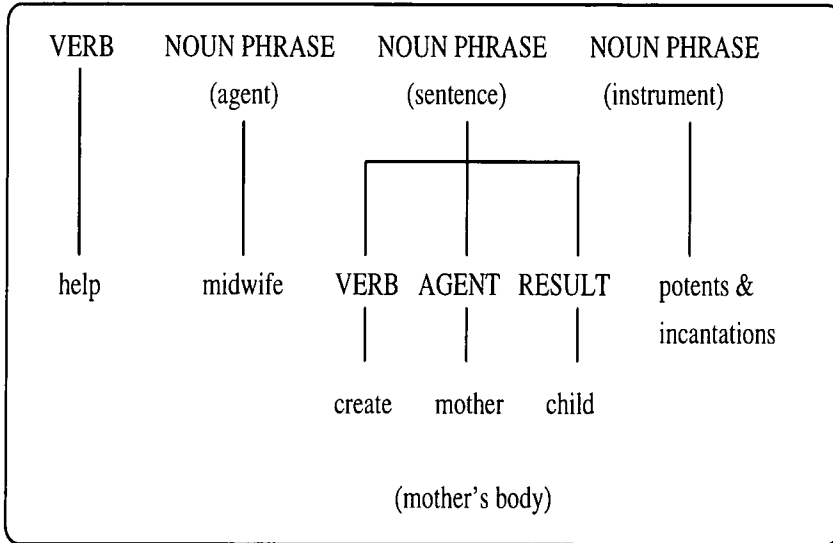


Fig. 3.5: Semantic field theory: example of topic domain structure

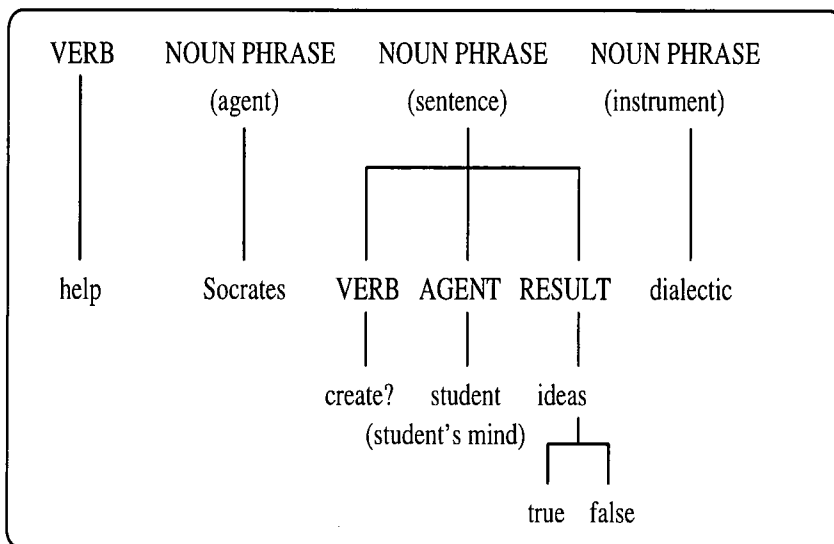


Fig. 3.6: Semantic field theory: example of topic domain structure

3.2.4.1 Dynamic type hierarchy

Way's dynamic type hierarchy of metaphor (DTH) can be seen as an elaboration on the interaction view and a more comprehensive implementation of the central notions "to make his⁷ ideas about filtering, domains and meaning shifts more precise

⁷Way is referring to Black.

and less metaphorical” [Way 1991]. As the name suggests, a type hierarchy is at the centre of Way’s model. It is based on conceptual graphs [Sowa 1984] and is used for “modelling the hearer’s ontology of the world. Different beliefs and different knowledge about the world will generate a different hierarchy and a different set of conceptual graphs” [Way 1991]. According to Way, the ontological hierarchy, changes with context, time and knowledge. A hierarchy at a given point in time consists of a number of concepts and links between those concepts, representing semantic links. The filtering idea of the interaction view is echoed in that an original hierarchy may be viewed under different aspects, applying a literal or figurative mask to the hierarchy, respectively. Different masks will obscure different concepts and links, thus giving a different view on a section of ‘the world’.

In order to recognise metaphors, the DTH relies on the occurrence of constraint violations. Way’s central notion is that the similarity, which is either pre-existing or created by the metaphor, is based on a common supertype of the tenor and vehicle.

I hold that the ‘similarity’ we find in common is an *abstraction* of some form of the properties found in the tenor and vehicle. In Black’s example of men and wolves, what we find in common are not similarities between the way a wolf is fierce or competitive and the way a man is, for they are radically different kinds of fierceness; rather, what we find in common is a more abstract concept of fierce competitiveness. What this higher-level concept maps to in the domain of men is social behaviour; for the domain of wolves it is in the form of physical attacks. These are both *instances* of a general form of competitiveness [Way 1991]

Way gives two examples “Nixon is the submarine of world leaders” and “the car is thirsty”. As Nixon and submarines have no immediate attributes in common, the hierarchy is consulted in a bottom-up fashion. As Way states: “when we look up the hierarchy for Nixon’s salient characteristics as a leader, we might see that they have a supertype in common, namely, ‘things which behave in a hidden or secret manner’ ”[ibid.].

She acknowledges that there are cases where no supertype common to the tenor and vehicle that makes sense of the metaphor can be found. In such cases, a new node which is a generalisation of both the tenor and vehicle concepts is created. Way’s second example illustrates this process; ‘cars’ are (amongst other things) ‘inanimate things’, ‘machines’ and ‘objects’. ‘Thirst’ is a ‘state’ and a ‘need’, commonly associated with ‘animate entities’. Going up in the hierarchy will lead to the

common supertype ‘mobile-entity’ for ‘animate’ and ‘vehicles’ and a generalisation of requirement for ‘need’. Thus the node ‘mobile-entity’ is specified with reference to the requirement (of liquids) to create the new node ‘things that require liquid’.

Since the domains are now linked via the common supertype of the tenor and vehicle, other relations can be seen in terms of the vehicle domain. In the submarine example, Nixon can be described as torpedoing policies, attacking enemies and so forth and in the car example, the car can drink, get ill or be stubborn.

There are, however, some apparent problems with this approach. Most notably, it relies on a very detailed ontology. It is hard to imagine the average ontology conveniently accommodating a concept such as ‘things which behave in a hidden or secret manner,’ at least not explicitly and as a generalisation of world-leaders. Hidden behaviour and instances of it such as plotting may well be present in a semantic network, as might be instances of world-leaders, but it is hard to conceive of a knowledge engineer hard-coding a sub-supertype relation between them. Secondly, Way’s model of metaphor interpretation works through abstraction and therefore necessarily loses information. She states that in the Nixon example, torpedoing can be understood as wrecking the policies of Nixon’s enemies, but this is almost a substitutionist view, since wrecking a policy itself is another metaphor. If Way kept to her own model, the action of torpedoing would have to be generalised to an action that is applicable to both ships and policies (e.g. rendering unfunctional) and thus is literally applicable. However, trying or succeeding in ‘rendering unfunctional’ has lost all of the original impact of the torpedoing metaphor.

To summarise, Way’s DTH models the important aspects of the interaction view well, and her notion of figurative and literal understanding operating on the same representation by highlighting and playing down various concepts and relations seems very convincing. It is a definite bonus that the creation of new concepts is triggered by the encounter of figurative utterances and thus formerly dormant relations are activated, serving as a means of describing a domain in terms of another domain. It is furthermore an advantage that related inferences can be drawn in accordance with those new links. Both these capabilities enable the representation to model a learning process and the elaboration of metaphors over a period of time. However, the reliance on a convenient ontology (the only source of knowledge in the DTH) casts doubt on the applicability of the DTH in situations where no such elaborate ontology is available. Other sources, such as encyclopaedic knowledge, episodic knowledge and world knowledge should be considered where available,

instead of demanding that the ontology provides all necessary information. The creation or discovery of a common supertype has positive and negative aspects as well. It can serve as a good heuristic in finding the ground of the metaphor, but on the other hand will provide little information if it is very abstract (cf. the critique of Fass' met* method in Sect. 3.2.2). Way has provided very little information about the actual implementation. Relying on the knowledge rich ontology and ignoring established relations such as part-for-whole leads the DTH into a position where possible metaphors have to be anticipated to a great extent [E. Way, personal communication].

3.2.5 Conventional metaphor view

The view currently receiving most attention, especially in the field of cognitive science and psychology, is an approach that can be termed the conventional view. It is influenced by cognitive, linguistic and computational work and is based on the highly influential work of Michael Reddy and partially on the interaction view. Reddy presented an analysis of figurative language which was based on empirical work and linguistic analysis rather than the analysis of individual examples of tropes in the light of a preconceived theory. He thereby came to almost revolutionary conclusions, namely, that large areas of everyday language are highly metaphorical and that the metaphoricity 'behind the scenes' is very systematic [Reddy 1979].

The 'conduit metaphor' forms, according to Reddy, a major framework "[implying] that human language functions like a conduit enabling the transfer of repertoire members from one individual to another" [Reddy 1979] and has systematic extensions, elaborating on the fundamental packing–sending–unpacking schema: "(1) thoughts and feelings are ejected by speaking or writing into an external 'idea space'; (2) thoughts and ideas are reified in this external space, so that they exist independent of any need for living human beings to think or feel them; (3) these reified thoughts and feelings may, or may not, find their way back into the heads of living humans" [ibid.].

The 'conduit metaphor' shows that it is almost impossible to talk about ideas and communication in a literal way. We speak of ideas or thoughts as if they were objects⁸ ('you still haven't given me any idea of what you mean'), words and

⁸All examples in this paragraph are taken from Reddy [1979].

sentences as if they were containers for those objects ('whenever you have a good idea, practice capturing it in words', 'try to pack more ideas into fewer words', 'you have to put each concept into words very carefully'), and communication as if it were finding and filling the right container for the idea and sending it along a conduit to the recipient ('try to get your thoughts across better', 'none of Mary's feelings came through to me with any clarity'), whose task it is to unpack the ideas ('Can you actually extract coherent ideas from that prose?', 'Let me know if you find any good ideas in the essay'). Not only are concepts from the domain of communication regularly expressed in terms of the domain of the exchange of physical objects, but also reasoning about communication, e.g. reasons for failure of communication, is guided in accordance with how things work in the domain of sending physical objects. Reddy gives examples of how the speaker/writer's success or failure to 'pack his ideas' ('The sentence was filled with emotion', 'your words are hollow – you don't mean them'), the listener/reader's role in 'unpacking' them ('I don't get any feelings of anger out of his words'), and the nature of the container ('that remark is completely impenetrable') are metaphorically expressed to state how the communication process and its success or failure are seen.

From the large corpus of examples presented by Reddy, it becomes clear that there are systematic correspondences between the domain of ideas and communication and the domain of transfer of physical objects. Aspects of communicative acts such as understanding are frequently and systematically expressed in terms of receiving and unpacking, i.e. in instances of the generic 'conduit metaphor'; so frequently indeed, that this conventionalised systematicity seemed to be too obvious to be unnoticed prior to Reddy's work.

On the basis of corpora analysis, Lakoff, and Lakoff and Johnson investigated the systematic correspondences between domains further. They uncovered a vast system of cross-domain correspondences which lie behind a large number of conventionalised figurative expressions, such as the correspondences between the domain of arguments and the domain of warfare ('He attacked all weak points', 'you cannot win an argument with them') [Lakoff 1993, Lakoff and Johnson 1980]. The fact that such correspondences govern a large number of everyday expressions and that the correspondences find their expression not only in language, but also in reasoning and behaviour related to the metaphorically conceptualised domain led Lakoff and Johnson to the hypothesis that metaphors (cross-domain correspondences) are to be localised in thought, rather than language.

The essence of metaphor is understanding and experiencing one kind of thing in terms of another. It is not that arguments are a subspecies of war. Arguments and wars are different kinds of things – verbal discourse and armed conflict – and the actions performed are different kinds of actions. But ARGUMENT is partially structured, understood, performed, and talked about in terms of WAR. The concept is metaphorically structured, the activity is metaphorically structured, and, consequently, the language is metaphorically structured. Moreover, this is the *ordinary* way of having an argument and talking about one. [...] Our conventional ways of talking about arguments presupposes a metaphor we are hardly ever conscious of. The metaphor is not merely in the words we use – it is in our very concept of an argument. [...] Metaphors as linguistic expressions are possible precisely because there are metaphors in a person's conceptual system. [Lakoff and Johnson 1980]

Metaphor, according to Lakoff and Johnson, is a conceptual structure, a method of accommodating lesser known, inaccessible or new concepts in a better known or better understood framework, which helps to structure the newly acquired conceptual material. The INFLATION IS WAR metaphor, for example, likens inflation to an attacking enemy ('inflation is destroying the savings'), entities opposing inflation to defending forces ('the government launched an offensive to beat inflation') and monetary policies to warfare ('the chancellor is battling inflation'). Since the concept of inflation is too theoretical to be understood directly, it is expressed in terms of a concept that is, on the one hand, more accessible and, on the other hand allows the right amount and kind of inference and extensions to allow for reasoning about inflation in terms of reasoning about warfare.

Lakoff and Johnson present three classes of metaphors, namely, orientational, ontological and structural metaphors [Lakoff and Johnson 1980]. In structural metaphors, one concept is partly understood or conceptualised in terms of another, as in the ARGUMENT IS WAR and INFLATION IS WAR examples given above. Structural metaphors are a combination and elaboration of the ontological and orientational metaphors. Orientational metaphors are centred around spatial relations and the mapping of spatial relations onto other, non-spatial domains. Examples of orientational metaphors are MORE IS UP ('prices are soaring', 'circulation figures went up') and its counterpart, LESS IS DOWN ('his income is too low', 'turn the music down'), as well as instances of these generic metaphors where 'less' or 'more' are concreted to specific ranges, e.g. HAPPY IS UP ('she is in high spirits', 'he fell into a depression') or LOW STATUS IS DOWN ('her social standing dropped considerably').

The class of ontological metaphors consists of correspondences between the domain of directly accessible objects and entities, especially our own bodies, onto more abstract domains. “Understanding our experiences in terms of objects and substances allows us to pick out parts of our experience and treat them as discrete entities or substances of a uniform kind” [Lakoff and Johnson 1980]. Examples are THE MIND IS AN OBJECT with instances like THE MIND IS A MACHINE (‘he’s been churning out ideas all day long’) and THE MIND IS A BRITTLE OBJECT (‘he just snapped’, ‘the refusal shattered her’) and the many event, action and state metaphors such as ACTIVITIES ARE CONTAINERS (‘she is in banking’, ‘they are totally absorbed in playing cards’). Personifications can be subsumed under the heading of ontological metaphors as they are a specialisation of the general principle; a particular domain is not described in terms of any physical object, but in terms of a human or animate agent (‘the theory explains nothing new’, ‘the government kept silent’).

Metonymy is also explained in terms of conventional metaphors, as there exist highly generative schemata such as PART FOR WHOLE, and, as an instance, FACE FOR PERSON. Lakoff and Johnson give the following non-linguistic example: showing a photograph of somebody’s face will be seen as an appropriate response to the request ‘Can I see a photograph of ..?’, whereas showing a photograph of the person’s body without the face will be seen as inappropriate. Lakoff and Johnson show convincingly that these cases are just as systematic and governed by an underlying cognitive process as the other types of metaphor and give a short listing of possible metonymic mappings (e.g. PART FOR WHOLE, PRODUCER FOR PRODUCT, OBJECT USED FOR USER, CONTROLLER FOR CONTROLLED). They do, however, make a distinction between metonymy and metaphor. The latter “is principally a way of conceiving of one thing in terms of another, and its primary function is understanding” [Lakoff and Johnson 1980], and metonymy “has primarily a referential function, that is, it allows us to use one entity to *stand for* another” [ibid.].

The systematicity leads, by application of Occam’s razor, to the assumption that the many related surface expressions, (figurative speech) can be traced to an underlying principle governing them all. Lakoff gives the following examples of the LOVE AS JOURNEY metaphor ‘we’ve hit a dead-end street’, ‘we can’t turn back now’ and ‘their marriage is on the rocks’ [Lakoff 1993]. In assuming that they are three different metaphors, one would ignore the fact that they follow the same principle, namely expressing states of a relationship in terms of states of

journeys. Thus Lakoff comes to the conclusion that: “We have one metaphor, in which love is conceptualised as a journey. The mapping tells us precisely how love is being conceptualised as a journey. And this unified way of conceptualising love metaphorically is realised in many different linguistic expressions” [ibid.].

To summarise, the theory of conventional, conceptual metaphors states that metaphor is a cognitive agency whose surface phenomena, metaphorical expressions in natural language, reflect the conventional and systematic way of conceptualising one more or less abstract domain in terms of a more accessible one. This is founded in bodily experience and cultural conventions. General mappings (such as LOVE AS JOURNEY) have systematic instantiations, therefore metaphor should not be regarded as ‘mere words’. The theory has considerable strong points. First of all, it was based on the empirical analysis of large amounts of data. The collection of instances of conceptual metaphors is an ongoing project, resulting in the Master Metaphor List [Lakoff *et al.* 1995], which is a repository, classification and index of the examples analysed so far. It is backed up by this empirical evidence insofar as the notion that metaphorical expressions are the surface of an underlying process explains the systematicity and the ubiquity of metaphor; if it is easier (or sometimes even the only way) to talk and reason about something in terms of a better known structure, then this will surely be reflected in the use of language. If there exists a large and conventional system of cross-domain mappings, based either on bodily experience (we are subject to gravitational force, have an intrinsic front and back) or cultural community, the ease with which conventional metaphoric expressions and extensions of those mappings seemingly are understood is explained. Novel metaphors are easily understood because they are based on pre-existing conceptual schemata and are mere extensions of them into a previously uncharted area.

Despite the appeal of the explanatory power and coherence of this theory, it must be kept in mind that it is a rather theoretical approach and does not easily lend itself to a process model of metaphor understanding. As Martin noted in reference to the conceptual metaphor view: “While it is difficult to apply results from psycho-linguistics to computational models in a direct fashion, these results can nevertheless pose useful rough constraints” [Martin 1990]. The aspects of recognition of metaphors, the influence of context in the recognition and interpretation of metaphors and metonymies and the details of how the mapping works are not detailed; the mappings are seen as static correspondences, derived from cultural or bodily experiences and the interpretation obviously seems to be immediate.

Furthermore, a comprehensive basis of explicit knowledge about the cross-domain mappings and their instances is assumed to exist a priori, and an account of how those existing mappings might be extended systematically is given, but no model is provided of how they (or new metaphors) can be acquired in the first place.

3.2.5.1 Knowledge-based approaches

The findings that metaphors are a conventionalised part of everyday language because they are situated in thought (hence the high systematicity in language, which is the expression of those cognitive configurations) led to new approaches in the computational treatment of figurative language. As Carbonell stated: “The problem of understanding a large class of metaphors may be reduced from a reconstruction to a recognition task. That is, the identification of a metaphorical passage as an *instance* of one of the general metaphorical mappings” [Carbonell 1982]. This means that not every occurrence of a possible metaphoric expression is computed anew by comparing domains and trying to find a possible correspondence. Instead, figurative input is compared to explicitly stored knowledge about general metaphors.

MIDAS

Martin’s work was one of the first computational approaches to make use of explicit knowledge about figurative language [Martin 1990]. For a limited domain (the UNIX operating system, the transfer of ideas and diseases), he showed how a knowledge-based approach makes the treatment of figurative language more tractable, or even possible. By coding knowledge about processes being seen as enclosures, for example, his system was able to interpret instances of metaphors such as ‘How can I get out of emacs?’ and automatically extend its knowledge by classifying previously unknown metaphors (‘how do I get into the LISP shell?’) as instances of already known ones.

His metaphor interpretation, denotation and acquisition system (MIDAS) consists of a lexicon (in which traditional word senses are listed alongside conventional metaphors), the metaphor interpretation system (MIS) and a metaphor extension system (MES). The representation of knowledge about conventional metaphors is modelled in terms of source and target domains (case roles and their relations) and a set of associations between concepts from the two domains (a mapping structure,

linking case roles), using the KODIAK representation language, a construct similar to KL-ONE [Brachman and Schmolze 1985].

Martin distinguishes between core metaphors and extended metaphors. This is basically a distinction between generic metaphors (such as the MORE IS UP metaphor) and instances of these generic metaphors, (e.g. HIGH STATUS IS UP). He also states a set of conditions for an extended metaphor to be derivable from a core metaphor, pertaining to the relationships between the concepts involved in the core and extended metaphor. As an example of a core metaphor, Martin names “infection-as-possession: Mary has a cold” and as extended metaphors derived from this core metaphor he cites “becoming-infected-as-getting: John got his cold from Mary, causing-infection-as-transferring: Mary gave John a cold” [Martin 1990].

The roles of the infection domain (‘having a cold’, ‘giving a cold’) correspond to the roles in the possession domain; the extension consists in the mapping of previously uninstantiated roles from the core metaphor to roles in the extended metaphor (infection-as-possession does not explicitly mention receiving, but a possessor can pass on his possession to a recipient).

Interpretation in MIS consists of two main steps; first, all possible interpretations are sought, literal and non-literal alike. If one or more matches with the stored metaphorical knowledge are found, the most specific is preferred. In the ‘Mary gave John a cold’ example, a match would succeed with both ‘infection-as-possession’ and ‘causing-infection-as-transferring’; and the latter would be preferred as being a specific case of the more general ‘infection-as-possession’. A process termed ‘concretion’ is responsible for ensuring that the most specific interpretation is found; it instantiates generic concepts such as ‘giver’ in accordance with the constraints the metaphor puts on the new role of the input concept, i.e. the instance must satisfy all roles it plays when used in the metaphor schema. The second step is called ‘metaphoric unviewing’. It creates an instance of the general metaphor that was found to be applicable to the input, in which the target concepts from the metaphor replace the corresponding source concepts. To give an example, all roles from ‘how can I get into LISP’ satisfy the constraints of the entering-as-invoking (‘self-propelled entity as agent’, ‘process as target location’, ‘entering as action’), where entering an enclosure corresponds to invoking a process; the mapping requires the agent and the location to remain unchanged and the action to be replaced by ‘invoke’.

Martin argues that it is useful to arrange metaphors in an inheritance/ab-

straction hierarchy, firstly so that “independent concepts can then be shared by the various related senses that have common associations” [Martin 1990]. For example, having an infection puts the agent in the role of possessor in the core metaphor and is shared by causing infection, in the role of the transferer. The second advantage is to allow cases of non-core related metaphors, (metaphors which cannot be derived from generic ones via extension) to be subsumed under one metaphorical scheme. He gives the following examples for non-core related metaphors: ‘how can I kill a process’, ‘those ideas won’t bear any fruit’, ‘inflation is eating up our savings’ as being related through “a component association that links an abstract concept that is not alive with a kind of living thing” [ibid.] or, in better known terms, personifications.

The notion of arranging metaphoric knowledge in a hierarchical fashion has obvious advantages. It enables a system to find the most specific interpretation of a generic metaphor, thus avoiding the danger of returning interpretations which are so general that they are of little value. It accounts for the fact that linguistic realisations of metaphors seem to be related to general conceptual schemata and allows for systematic relations between metaphorical expressions which share parts of their structure (as in Martin’s core and extended metaphors).

Martin went on to examine how metaphor can be used to explicate the phenomena of polysemy and for the acquisition of new word senses [Martin 1991b]. He further investigated how his model of metaphor extension can be used towards the accommodation of semantic systematicities not captured by lexical semantics [Martin 1992].

To summarise, Martin’s work offers some very helpful insights and new directions. The need to represent metaphorical knowledge explicitly to allow for efficient computational treatment of figurative input can be derived from both the findings about the conventionality of metaphors by Lakoff and Lakoff and Johnson, and the shortcomings of the analogy-based approaches. Arranging metaphors and their surface realisations in a hierarchy is in keeping with the conventional theory of metaphor, too, as it allows for systematic elaboration of more general schemata. The idea that a system should be able to interpret new instances of figurative language on the basis of previously acquired knowledge is particularly appealing, as it loosens the constraints of the closed world assumption.

The problems with Martin’s work, however, must also be examined. In his original work [Martin 1990], he does not give details about how an inheritance hierarchy

could be modelled, apart from one example consisting of 11 Non-Living-Thing-As-Living-Thing metaphors. In a later paper [Martin 1991a], the construction of a large knowledge base (KNB) containing all metaphors from the Master Metaphor List [Lakoff *et al.* 1995] is considered, but no further details are given. Moreover, possibly because his examples come from a limited domain, he deals with cases where an interpretation is completely determined by the metaphor that is found to match against the input; “Metaphor maps are needed to link all the core source concepts [...] to their counterparts in the target domain” [Martin 1990], either directly or through extension. This means that his metaphor maps are specific enough to lead to a detailed interpretation, stating explicitly which roles in the input correspond to which roles in the metaphor and thus in the interpretation. Granted, in the first instance, his work serves as a computational model, but in a real application with an extensive KNB, the chances are high that there might be a general schema such as process-as-enclosure, whereas the existence of a domain specific schema such as entering-as-invoking cannot be presupposed. Therefore, the model should support ‘on the fly’ generation of instances of metaphors to support the coverage provided by Martin’s processes of extension and concretion. A final point, similar to the problem with Way’s DHT, is the fact that the correspondences in Martin’s model link target and source concepts through an abstraction, thus losing information. In a similar vein, MIDAS ‘replaces’ the figurative input with a literal and often more abstract interpretation; “The concept that best accounts for the input replaces the primal representation as the ultimate representation of the input” [Martin 1990] and therefore records no evidence about the relation between the two domains, apart from the fact that the interpretation will be classified as an instance of a particular metaphor.

3.2.6 Connectionist models

The work of Veale lies at the intersection of knowledge-based and connectionist approaches to modelling of metaphor understanding [Veale and Keane 1992a, Veale and Keane 1992b, Veale and Keane 1993, Veale and Keane 1994, Veale and Keane 1995].

Conceptual scaffolding

Veale's model of metaphor interpretation, conceptual scaffolding [Veale and Keane 1992a, Veale and Keane 1992b] takes the fact that many conventional metaphors use spatial relations to structure non-spatial domains (MORE IS UP, LOW STATUS IS DOWN) as a basis for establishing a framework of connections (the conceptual scaffolding) between concepts. According to Veale, this framework forms, the basis of metaphor understanding. The scaffolding is "constructed from a fixed set of spatially founded operators, which encode our experiential intuitions about collocation, containment and orientation" [Veale and Keane 1992a]. This is a direct reflection of the large system of conventional metaphors based on spatial relations uncovered by work in the wake of Lakoff and Johnson [Lakoff and Johnson 1980, Lakoff *et al.* 1995]. The scaffolding captures underdetermined relations, association and disassociation between concepts, on the basis of which a finer interpretation, involving the naming of associations and the inference of new associations from the existing ones, takes place at a later stage.

Veale presents a spatial semantic for conceptual scaffolding, involving operators such as 'connect', 'disconnect', 'actual-causality' and 'attempted-causality' that can represent both literal and figurative meanings [Veale and Keane 1992a]. For an illustration of the operators, relating to the sentences (1) 'Mary went to town', (2) 'Mary went to sleep', (3) 'Mary went from rags to riches' and (4) 'Bill gave Mary a book', see Fig. 3.7.

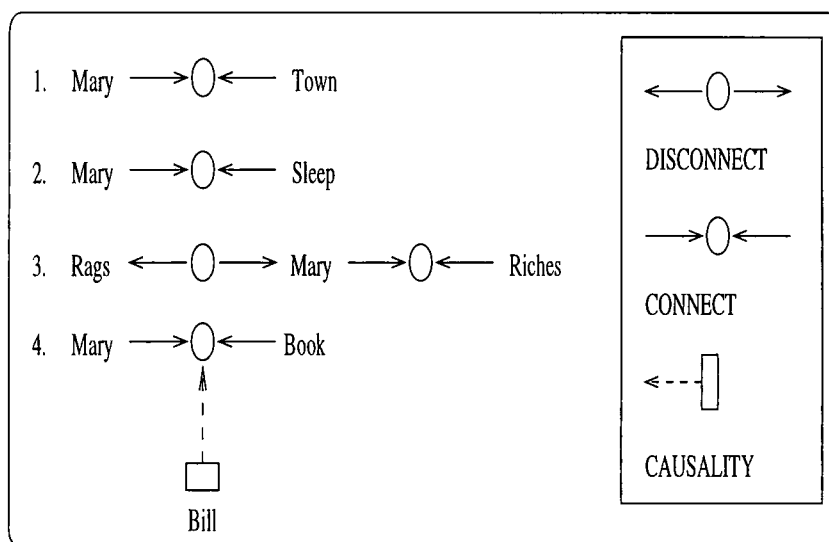


Fig. 3.7: Spatial semantic operators for conceptual scaffolding

The process of metaphor interpretation in Veale's model is broken up into the

first stage of the construction of the conceptual scaffolding between the concepts from the input and the later stage of elaborative inference from the scaffolding.

The creation of the scaffolding involves finding the most interesting attributes of a concept, which might change with context (a brick will be seen in different lights, according to whether it is on a building site, securing a caravan from moving, or is thrown through a window), but “Regardless of context, an orientation operator will naturally prefer those attributes which contribute to the function of the concept as a whole” [Veale and Keane 1992a], in other words, the context-independent most salient features. Once the connections between concepts are established, a knowledge base is consulted. The KNB takes a simple concept association, such as ‘apple, colour’ or ‘tea, hot’ and returns a relation specifying the associations between the concepts, labelling the association, such as ‘apple, colour, green’ or ‘tea, hot, temperature’. Since there is no guarantee that a concept association can always be labelled (reasons can be the lack of knowledge or an unresolved metaphorical meaning), unspecified associations are kept by the KNB until enough knowledge for their labelling is available.

Veale gives the following example: “The RISCsystem 6000 workstation rescued IBM”. ‘Rescue’ is expressed in terms of $\text{up}(\text{predicament}) \rightarrow \text{safe}$ and leads to $\text{connect}(\text{IBM}, \text{safe}) \rightarrow \text{connect}(\text{IBM}, (\text{up}(\text{predicament}))) \rightarrow \text{up}(\text{IBM})$. As IBM is, in this context, seen as a manufacturer of a product, the salient feature of IBM being a business is chosen; thus, $\text{up}(\text{IBM})$ resolves to $\text{up}(\text{IBM}, \text{market-share})$ (which, as it turns out, is another metaphor).

The conceptual scaffolding employs the same techniques when dealing with other kinds of figurative language such as metonymy. Input such as ‘Jim drank the whole bottle’ would receive the relation (beverage as content of bottle) from the knowledge base on the query $\text{connect}(\text{bottle}, \text{beverage})$.

Veale’s model presents some very good answers to questions not previously resolved. Furthermore, it overcomes the difficulties traditional lexical semantics have with metaphor, thus the deep seatedness of metaphor in our conceptual system (his spatial semantics represent the basic structure of the spatially founded metaphor system) is acknowledged. A uniform method is used for treating different kinds of figurative language. Although some implementation details are given, namely that the conceptual scaffolding model was implemented as part of the TWIG concept acquisition from text NLP system in common LISP, a very central question is not addressed, namely, how the knowledge base is organised, how large an amount of

knowledge is stored in it, and how many relations apart from the ones returned are possible on the basis of its data. The data base's functionality is explained, but not its key features.

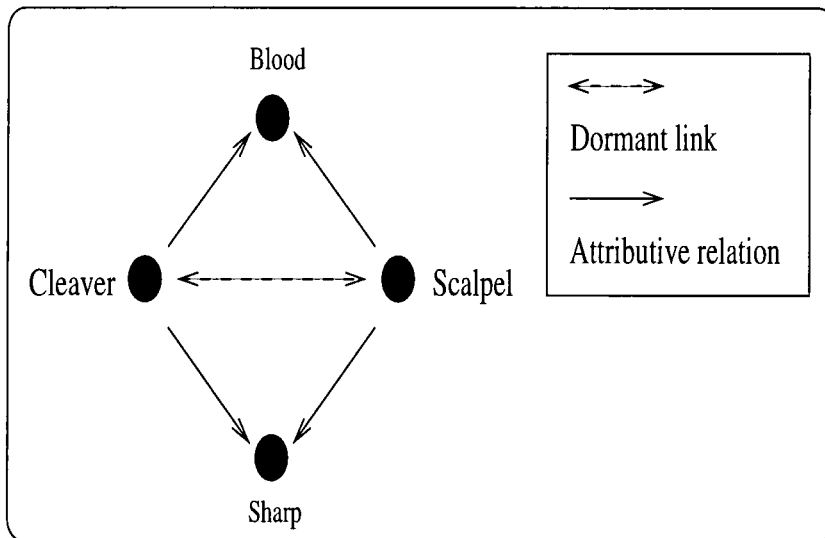


Fig. 3.8: Sapper triangulation rule

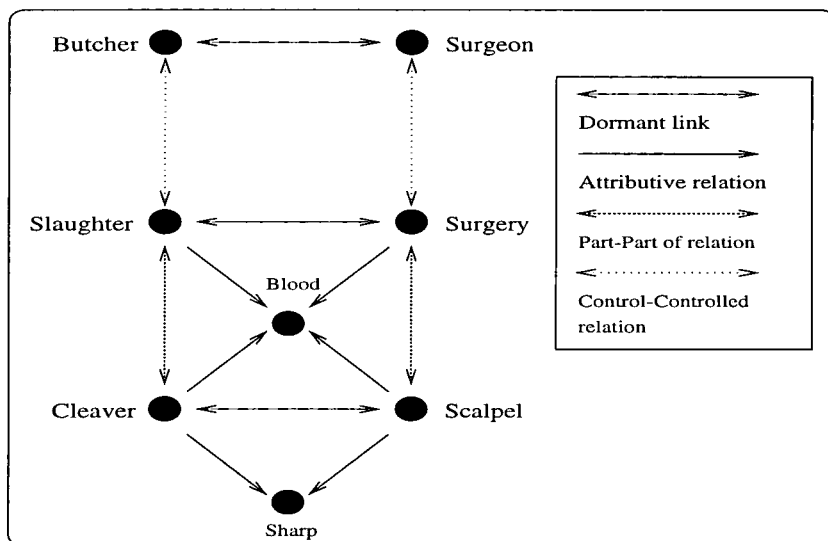


Fig. 3.9: Sapper squaring rule

Another shortcoming of the model is its failure to address what had been of central concern to almost all other scholars in the field, namely, comparison statements of the form 'A is B'.

This deficit has been made up for by further work [Veale and Keane 1993, Veale and Keane 1994, Veale and Keane 1995], in which Veale complemented conceptual scaffolding with a hybrid symbolic/sub-symbolic model of metaphor interpretation

called Sapper. This model is based on a localist graph that, represents concepts as nodes, and semantic relations between the nodes as linkages. The interpretation process consists of two separate steps. First, a symbolic process augments the network of concept relations with dormant links: “Such dormant network linkages represent merely *plausible*, rather than fully *established*, semantic relations, and are thus not operative carriers of activation” [Veale and Keane 1994]. The creation of new dormant links is governed by two rules which operate on the structure of the network. The first ‘triangulation rule’ forms a link between concepts that share a common supertype or a common attribute (see Fig. 3.8). This can be compared to Way’s approach of finding a common supertype, but it must be noted that while Way is making use of an ontology, Veale’s triangulation rule operates on knowledge that is not restricted to ontological relations. The second ‘squaring rule’ makes use of the links established by the triangulation rule in establishing further inter-domain linkages which are coherent with the links derived via the triangulation rule (see Fig. 3.9).

The second, connectionist step propagates activation from the tenor and vehicle nodes. This activation can ‘awaken’ a dormant link on the condition that the link receives activation from different parent nodes. Once it is activated, it passes on activation from one domain to the other. This can be seen as a direct model of the theoretical notions of domain interaction proposed by the interaction view (see Sect. 3.2.4). The newly established links are then ranked according to various criteria, including a measure of how well they fit the overall systematicity of the domain relation (single unsystematic links are ignored) and are seen as ‘is metaphorically like’ semantic relations (see Fig. 3.9).

The LISP implementation of Sapper comprises 284 concepts nodes from the domain of professions and 1597 user-specified semantic links, including attributive, control/controlled, part/part-of and effect. In an evaluation experiment [Veale and Keane 1995], it compared favourably with systems based on the analogy approach [Falkenhainer *et al.* 1989, Holyoak and Thagard 1989a, Holyoak and Thagard 1989b], but some of its limitations must be pointed out. The model obviously only deals with comparison statements and, although this might prove very useful, an application of its mechanisms to extend the depth of the interpretation of conventional metaphors cannot be easily derived from the information Veale presents. It expects the tenor and vehicle concepts to be explicitly present in its input and returns a very general ‘is metaphorically like’ semantic relation in the form of a newly

established link between concepts. It seems that, while this is a cognitively very plausible model of metaphor interpretation, direct application of its mechanisms to a general computational treatment of metaphors is not possible if one considers the size and structure of the data used for its evaluation.

Veale notes that “the worst case scenario involves the generation of $n * (n - 1) / 2$ dormant linkages, where n is the number of localist nodes in semantic memory” [Veale and Keane 1995]. If one applied Veale’s model directly to the LOLITA system with its 100,000 nodes (c.f. Sec. 2.2.1), this would mean the creation of 4 999 950 000 dormant linkages. This is a better result than SME’s $n!$ complexity⁹, but still a rather prohibitive number for a large system aiming at real-time processing, as the dormant linkages only form the basis for the second, connectionist step of metaphor interpretation. Assuming that 10 million operations of linkage generation are performed per second, the initial stage in the LOLITA system would, in the worst case, take more than 8 min. Veale further notes that “as it happens, the average case performance is much lower than this ceiling, the effect of a well-defined domain structure being to considerably reduce the possibilities of inter-node bridging” [ibid.]. While an evaluation configuration can afford to rely on a well-defined domain structure, an implementation which deals with a dynamic and non-predetermined representation will lack the benefit of the well-defined domain structure and will thus tend towards the worst-case scenario.

To summarise, while the scale and dependence on the representation of the Sapper model allow for no immediate transfer of its results to a real-life large-scale system, it is a very convincing cognitive model. The central notions of domain interaction are reflected, and the loss of information that accompanies ‘abstraction’ approaches (such as Way’s DTH, see Sect. 3.2.4) is avoided by letting the two domains interact freely during the connectionist phase. On the computational side, Sapper manages to avoid the factorial complexity of analogy-based approaches (see Sect. 3.2.3). This is obviously due to the fact that processing is split into a two-tiered symbolic and subsequent sub-symbolic stage, which is a highly interesting approach.

⁹For n inputs, $(1 * \dots * n - 1 * n)$ operations are performed, making a complexity of $n!$ unfeasible for all but the smallest number of inputs

3.3 Summary

There are many different approaches to the many questions surrounding metaphor and figurative language, each highlighting various aspects of the multifaceted problem. There is no unified account to date, but the study of metaphor (outside traditional disciplines like literature) is an emerging field. The renewed interest is due to the realisation that metaphor plays a central role (in thought, and therefore in) language. Most current work is carried out in cognitive frameworks, with emphasis on psychology and linguistics. Computational models tend to be created mainly for the evaluation of theories from those fields.

3.3.1 Central issues

The questions that arose from an analysis of the classical authors can now be re-evaluated in the light of the various approaches discussed above:

- What is metaphor?

Metaphor and other figures are the expression of cognitive mechanisms, which accounts for their high degree of systematicity (and pervasiveness) in natural language. Figurative language is widespread and describes one set of concepts in terms of another. Most current theories deal with comparison statements of the form ‘A is B’, which has led to overemphasis of the necessity for analogical mapping. A distinction is made between metaphorical (as a means of *understanding/describing* one thing in terms of another) and metonymic figures (as a means of *referring* to one thing in terms of another) [Lakoff and Johnson 1980].

- Is there a structure in metaphor?

The analogy approaches argue that the key to metaphorical structure lies in the similarity between the structures of the base and target domains, thus restricting the domains that can possibly be used to describe others. The conventional view, which seems more convincing and ultimately more suited as a basis for a computational approach, states that experience and cultural influence constitute a conventional system of general metaphors, which serve as templates for individual instances and realisations of figurative expressions. This is in keeping with the more theoretical interaction view, which posits

that two systems of concepts and a set of ‘commonplaces’ associated with the concepts structure metaphor [Black 1962].

- How are metaphors interpreted?

There are two main views on the interpretation of metaphor. The analogy school of thought sees interpretation as a process by which the relevant valid similarities in two domains are computed, and the most important ones then transferred from the base domain to the target domain, where they are re-instantiated. The conventional view assumes fixed but general and abstract correspondences between domains (e.g. metaphors such as MORE IS UP) and sees interpretation as activation of knowledge about these correspondences. As Carbonell noted in reference to the results of Lakoff and Johnson [Lakoff and Johnson 1980]: “There appear to be a small number of *general* metaphors [...] that pervade commonly spoken English. [...] The computational significance of the existence of a small set of general metaphors underlies the reasons for my current investigation: The problem of understanding a large class of metaphors may be reduced from a reconstruction to a recognition task” [Carbonell 1982]. The feasibility of this approach is underlined by the results of Martin’s MIDAS system [Martin 1990].

- Why are metaphors used?

Having established that metaphors do not merely have a decorative function, since otherwise a large percentage of everyday language would have to be classified as ornamental, the most convincing explanations seem to be the *inexpressibility* and the *compactness* hypotheses.

The inexpressibility hypothesis states that metaphor is used “to express ideas that simply cannot be easily or clearly expressed with literal speech” [Gibbs 1994]. The compactness hypothesis claims that metaphor serves as a cue, evoking many associations and possible inferences on the basis of a compact utterance [ibid.].

Grice [Grice 1975, Grice 1978] assumes that metaphor violates the rules of literal language and that, on the basis of this violation, a pragmatic analysis to salvage the communicative contribution is initiated. He has been severely criticised for his views on metaphor [Gibbs 1994], but his conversational maxims hold, in our view, for figurative language and literal language alike. This is particularly true for his maxim of quantity — make your communicative

contribution as informative as required by the context, neither more nor less so — and the maxims of relation and manner — say only what is relevant in the context; make your contribution as precise and brief as possible. These seem to capture the impact of metaphors, even though they were not intended to apply to figurative language. By giving the hearer/reader a cue that can, due to its conventionality, serve as a basis for further inferences and elaboration, all three maxims are clearly being followed.

- How can metaphors be classified?

As the field as a whole is in the process of establishing itself and many opposing camps handle various different aspects, no unified or generally accepted terminology has been agreed upon yet. In the following, a classification outlining the better established terms and traditional classes will be given. This classification will be re-assessed in Sect. 5.3, taking into account the necessities of a computational treatment.

3.3.2 Classification of types of figures

- Allegory

A trope of ‘continual metaphor’, where a large section or a whole text is expressed in metaphorical terms from one domain.

E.g. The description of statesmanship in terms of seafaring (likening the leader of a state to the captain of a ship, the state to the ship, history to the sea, politics to the art of navigation).

- Analogy

The likening of entities or relations. A stipulated isomorphism between two concepts and their attributes. An analogy states, in varying degrees of explicitness, the matching similarities between its referents, usually in the form *A is to B as C is to D*.

E.g. The atom is like the solar system, it has a large central object around which other objects are orbiting.

- Antonomasia

The replacement of a proper name by either a term describing the genus or a periphrasal construction.

E.g. 'The almighty' instead of 'God'. 'An odyssey' instead of 'a long and complicated journey'.

- Emphasis

A subclass of synecdoche (see below), where a (semantically or pragmatically) more poignant expression than required in the context is used to increase the impact of the utterance.

E.g. 'Be a man' for 'Behave like a man'. 'A childish child' to put emphasis on the childish qualities (e.g. of behaviour).

- Euphemism

Replacement of a term (often from a taboo domain) by a term without any negative or even a highly positive connotational value.

E.g. 'Friendly fire' for the gunfire of one's own soldiers.

- Hyperbole

Use of a term that exaggerates a given quality of the term it denotes in order to draw attention to this quality or increase the emotive impact of the utterance.

E.g. 'Heart of stone'. 'Drink a gallon of milk'.

- Idioms

A fixed syntagmatic construction consisting of more than one lexical entity (otherwise an additional lexicalised meaning could be listed for a single lexical item) where the meaning of the whole expression cannot in part or as a whole be derived from the individual parts; the paradigmatic exchange of single elements does not result in any systematic change in meaning.

E.g. 'This work is not my cup of tea'. 'There is too much red tape'.

- Irony

Replacement of what is *meant* by an expression denoting the opposite by means of antonymy (e.g. 'How nice of him' instead of 'How awful of him'), contrasts (e.g. 'Physically, he is very strong' for 'Intellectually, he is not very strong') or illocution (e.g. 'Go ahead!' for 'Do not do this!').

- Litotes

Replacement of a term (often accompanied by an emphatic marker) by the negation of the opposite. Related to irony and the inverse trope hyperbole (see above).

E.g. 'Not (particularly) good' for '(Very) bad'.

- Metonymy

One of the classical definitions of figures holds that metonymy is the replacement of a term by a semantically related term. The relation can be causal, spatial, temporal or a generically semantic/ontological such as inventor-invention, place-inhabitants, thus less restricted than the relations that hold in the case of synecdoche (see below).

E.g. 'Read Shakespeare' for 'Read Shakespeare's work(s)'. 'Drink a bottle' for 'Drink the contents of the bottle'.

- Personification

The likening of an inanimate or abstract entity to a human or humanoid entity; the application of human attributes or features to non-human referents (entities, events). Depending on the point of view, personification could be regarded as a specialisation of an inverse antonomasia, synecdoche or metonymy (see above).

E.g. 'This dog owns the house'. 'The car is thirsty'.

- Rhetorical question

In a general sense every interrogative phrase that although formally and grammatically asking for an answer (where a possible answer would be either yes/no or a phrase) pragmatically does not ask for an answer (newspaper headlines, for example). In a more special sense, the same grammatical constructions indicating the opposite of their proposition.

E.g. 'Did he arrive in time?' for '(I am sure that) he did not arrive in time'. 'Does he ever leave his terminal' for 'He never leaves his terminal'.

- Simile

A comparison between two referents, usually made explicit by the use of lexical entities expressing a comparison, i.e. 'as' or 'like'.

E.g. 'His voice is like thunder'.

- Synecdoche

Replacement of one term by a semantically more restrictive or less restrictive one. The restrictions are based on the classical definition of synecdoche (genus/species, part/whole). In most cases, the relation can be inverted. Although the traditional classification makes a distinction between metonymy and synecdoche, there is no evident reason why the two types should not be subsumed under the more general type.

E.g. ‘The sail disappeared on the horizon’ for ‘The ship disappeared’

3.3.3 Additional issues

Besides the original questions discussed in Sect. 3.3.1, there are additional points arising from the discussion of the literature which seem worth noting.

Recognition

Recognition in some theoretical and almost all computational models (the exception being Martin’s MIDAS system [Martin 1990]) is based on the detection of an anomaly, although linguistic and cognitive results show that no anomaly detection mechanism is responsible for the recognition of figurative language by humans [Gibbs 1994].

Generally, there does not seem to be a fail-safe way of recognising or distinguishing figurative from literal or anomalous language that does not at the same time wrongly classify a substantial amount of utterances. Without recourse to context, world and episodic knowledge (c.f. Wilks’ pseudo-texts, Way’s type hierarchy) as well as knowledge about figures (c.f. Martin’s core metaphors, Fass’ metonymic inference rules), the anomaly detection heuristic is the only fallback method.

Stages of processing

In a cognitive framework, Gibbs [Gibbs 1993] makes the necessary distinction between the process and the product of linguistic understanding and identifies the following stages during the understanding of figurative language:

- Comprehension.

The “immediate moment-by-moment process of creating meanings for utterances. These moment-by-moment processes are mostly unconscious and involve the analysis of various linguistic information (e.g., phonology, lexical access, syntax), which, in combination with context and real-world knowledge, allows listeners/readers to figure out what an utterance means or a speaker/author intends” [ibid.].

- Recognition.

The conscious identification of the products of comprehension as either literal or figurative and as instances of a certain type, e.g. a metonymy, a promise or a command.

- Interpretation.

The conscious assignment of one or more (more or less elaborate) meanings to “the early products of comprehension” [ibid.].

- Appreciation.

This stage “refers to some aesthetic judgement given to a product” and “is not an obligatory part of understanding linguistic meaning, because listeners/readers can easily comprehend utterances or texts without automatically making an aesthetic judgement about what has been understood” [ibid.]. At this stage, dimensions such as the aptness, the novelty and impact of a figurative expression are evaluated.

This categorisation is useful, because it helps to avoid a dilemma that might arise when no distinction is made between the individual stages of processing and the process as opposed to the product. When being confronted with an utterance like ‘My car is thirsty’, we are inclined to immediately state that this is an instance of a metaphor. However, we can do so only after having passed through the primary stages of linguistic processing. This would mean in detail: understanding what the individual words mean and what the relation between them is (comprehension), consulting our world knowledge (cars are machines and only animate beings that require fluid intake for their metabolism to function can be thirsty) (recognition), and constructing one or more interpretations on the basis of the previous stages and extra-linguistic knowledge such as the discourse situation (the car needs oil or petrol). These stages then support appreciation (the car is described in terms of a pet or companion, its owner employs similar metaphors very often etc.).

In our model, too, these stages are reflected, not because we are aiming at a cognitively appropriate model, but because they either prove necessary or useful or both. Nevertheless, our distinction differs from the proposals made by Gibbs, and this will be discussed in detail in Sect. 5.2.1.

Representation

The hierarchical structure of general metaphors (as proposed by Lakoff [Lakoff 1993] and modelled by Martin [Martin 1990]) and their instances is a useful and necessary concept, as it captures already established metaphorical relations and allows elaboration of these on the basis of pre-existing structures. In a computational model, this means a reduction in processing time, as it is possible to build on foundations that have already been analysed. In a similar vein, the automatic extension of metaphoric knowledge (as shown in Martin's MIDAS system [Martin 1990]) is useful and important, too. Firstly, it frees the system to a certain extent from the restrictions of pre-coded knowledge (the closed world assumption); secondly, it allows broader coverage by extending basic mechanisms over a dynamic range of possible input.

The representation of knowledge and the results of the processing is also considered to be of importance. Whereas Wilks and Fass do not make direct use of the resolution, Martin stores the results of the analysis in the appropriate place in the abstraction hierarchy to facilitate the processing of similar input [Wilks 1975a, Fass 1991, Martin 1990]. In a theoretical outline of a process model, Carbonell assigns this task a high priority: "Remember this instantiation of the general metaphor within the scope of the present dialogue (or text). It is likely that the same metaphor will be used again [...] with additional information conveyed" [Carbonell 1982].

Novelty of metaphors

The dimension of novelty is discussed controversially. Whereas Lakoff argues that even the most poetic and novel metaphors are necessarily based on (a combination) of the conventional, general metaphoric schemata and even idioms and 'dead' metaphors (such as 'leg of a table') should be regarded as 'true' metaphors [Lakoff 1993], others, such as Way advocate that 'dead' metaphors be simply listed in the lexicon [Way 1991]. In a computational framework, it is important to choose the most efficient way of resolving figurative input. The question of novelty thus becomes an issue, since it affects whether the same resolution mechanisms should be applied to 'novel' and 'dead' metaphors alike.

In the following chapter, these points will be analysed, alongside the implications from the evaluation of related work, in the framework of the LOLITA system.

Chapter 4

Figurative language and LOLITA

This chapter presents the implications of the work discussed in Chap. 3 for the problem outlined in Sect. 1.1 with regards to the LOLITA system. In Sect. 4.1, the general problem area of figurative language processing is re-analysed in the light of the LOLITA framework, and Sect. 4.3 presents criteria for evaluating the solution to the stated problem. Methodological considerations of importance to the work in general and the solution in particular will be discussed in Sect. 4.2

4.1 Problem statement

Recent research has unarguably shown that figurative expressions pervade everyday language. The meaning of figurative language cannot be computed easily, using the established natural language processing (NLP) methods such as lexical (compositional) semantics, truth conditional semantics and generative semantics frameworks. Nevertheless, it exhibits systematicity and various models aiming at explaining the mechanisms behind understanding figurative expressions have been put forward (c.f. Chap. 3).

By far the largest part of work undertaken in the field of non-literal language concerns itself with the evaluation of cognitive theories by implementing models of those theories. The results of both theoretical and small-scale work seem advanced enough to be analysed with respect to their applicability in a large-scale, natural language (NL) engineered system. But at the same time it has to be kept in mind that the transfer of results from a small evaluation model to a real-life NLP system

is not a mere software engineering endeavour; an *expensive* algorithm that may work well for a limited domain or small knowledge base (KNB) is unsuited to a large-scale system.

As stated in Sect. 1.1, the LOLITA system would greatly benefit from an extension of its NLP capabilities into the direction of figurative language for the following reasons:

- The LOLITA system is a large-scale systems, whose aim it is to process unrestricted, everyday texts. This means that figurative language cannot be ruled out as lying outside the systems scope of possible input. Especially if one considers the pervasiveness of figurative language.
- Being able to process non-literal language is not so much of an advantage, but a necessity given the ubiquity of figurative language of varying complexity. NLP systems which cannot deal with figurative input are severely restricted in their scope.
- The LOLITA systems is based on deep techniques. This means a thorough analysis of its input on all linguistic levels (syntax, semantic, pragmatics) is needed, and an analysis of figurative input has to be performed in order to ensure that the required depth of processing is guaranteed for the whole range of input; this is particularly crucial, since subsequent stages of processing make use of the results of previously analysed input. Not being able to process figurative language would ultimately result in ‘holes’ in the system’s performance, whenever access to various levels of analysis of non-literal language were needed.

It is necessary therefore to extend the existing processing capabilities of the LOLITA system to cover figurative language as well. By processing figurative language, we mean:

- Recognising figurative parts in the system’s input and distinguishing it from both literal and anomalous input.
- Enabling the system to apply its existing functionality to figurative input. This means that figurative input should be assigned an interpretation and thus subsequently be available for other stages of processing (such as the drawing of inferences or generating NL on the basis of it).

- Representing knowledge about figurative language in the system's KNB. This is partly a prerequisite for the analysis of figurative language and partly a requirement for the successful completion of the functionality mentioned in the last point above. Since the LOLITA KNB is the basis for generation of NL, a suitable representation of knowledge about non-literal language would ultimately enable the generation of figurative expressions.

The details of this functionality with regard to the LOLITA system and their comparison with the task of understanding figurative language as presented in Sect. 3.3.3 will be discussed in Sect. 5.2.

The delimitation of what we consider to be 'figurative expressions' is, of course, a prerequisite for further investigation. It is necessary to give a clear account of what kind of NL expressions we consider to be within the range of figurative language (as regards processing in LOLITA), thus stating which phenomena we expect to be treated. This will be addressed in Sect. 5.2.2, where the classical types of figures (discussed in Sect. 3.3.2) will be re-assessed in the light of computational requirements for their resolution and representation.

One fundamental, hitherto implicit assumption has to be made explicit, namely, whether or not the undertaking of implementing non-literal NL analysis is possible and feasible, especially in the framework of a large-scale NLP system. Therefore, one central objective of this work is to establish the feasibility of such an undertaking. To this end, a thorough literature survey and evaluation of related work has been carried out; as discussed in Chap. 3, some theories and approaches seem more suited to forming the basis of an implementation than others. Most work in this area has either been theoretical, undertaken to evaluate theories by supplying a computational model, or restricted in size. Although the work of Wilks [Wilks 1975a, Wilks 1975b, Wilks 1978], Fass [Fass 1991], Martin [Martin 1990], Veale and Gentner [Falkenhainer *et al.* 1989] [Veale and Keane 1992a, Veale and Keane 1993] has been implemented, none of it was incorporated into a large-scale system. Domain independence or broad coverage were not a concern to those implementations either, and in some cases (including most of the theory-evaluating implementations) the representation was customised for the task at hand.

For the reasons given in Sect. 3.2.5, we favour the knowledge based approach to understanding figurative language. On the theoretical side, it exhibits the highest degree of explanatory and predictive power over a range of types of figurative

language, whereas other approaches concentrate on single types. While it is not a readily implementable theory, it avoids the pitfalls of the analogy approaches' computationally intractable process model.

While the conventional, knowledge-based theory is chosen here as a basis for the investigation into processing figurative language in the LOLITA system, however, it should be noted that our aim is not the evaluation of its theoretical merits or the ease with which it lends itself to implementation. Neither will we consider it as our only source of theoretical inspiration, as it cannot claim to explain all phenomena which are of relevance to the treatment of non-literal language in an NLP system. When seen in isolation, this liberal approach to the re-use of theories and results might be considered unmotivated. Therefore some points on the methodology adopted for this work will be presented in the following.

4.2 Methodology

The subject of this research is to investigate the state of the art in figurative language processing as well as to outline and, if possible, implement a solution to the basic problems concerning figurative language in an existing, NL engineered NLP system.

Owing to the complexity of the field and general lack of agreement, this work had to be partly theoretical. The critical evaluation of the results, advantages and shortcomings of previous approaches was necessary in order to gain insights into the general and special problem areas, to localise obvious garden-paths (with the benefit of hindsight), to avoid approaches which seem outright unsuited for a large-scale system, and to have a means of comparing our solution with others. Furthermore, as there is no generally accepted terminology and (particularly with regard to computational work) only localised theories are available, the analysis and comparison of previous work serves to focus the manifold approaches in order to arrive at a clear outline of which tasks pertain to figurative language processing.

As far as the modelling and implementational side is concerned, one methodological aspect is the view taken by artificial intelligence (AI) or applied computational linguistics: The model should exhibit behaviour parallel to human behaviour, regardless of whether this behaviour is due to underlying processes similar to those assumed or proven to govern human behaviour. In addition, the methodology and

criteria applied in the field of natural language engineering (NLE) in general (c.f. Sect. 2.1) apply to this work as well. Since NLE is understood as the application of sound engineering methods to the field of NLP, the work presented in this thesis follows the structure of engineering procedures. After taking related work into account, a solution to the problem is designed in Chap. 5, and its implementation evaluated in accordance with the criteria outlined below in Sect. 4.3.

It should be noted that the aim of the design and implementation stages is to arrive at a prototype. The work set out, namely, to establish whether a broad spectrum of problems in the field of processing figurative language in the LOLITA system can be dealt with, rather than concentrating on the perfection of a single sub-task, e.g. the treatment of personification. Prototypes are defined as software solutions which are developing in parallel to the theoretical aspects of solving a given problem [Görz 1993]. They contain the essential algorithms, data structures and interfaces needed for an application or full solution and are a valid formal approach, since the required algorithms and data-structures for many problems are known only theoretically, and detailed insights can only be gained by prototypical implementations and their evaluation. In the context of this work, this means that starting from carefully chosen examples, an initial solution was sought, which was then the subject of testing and generalisation.

Keeping to NLE principles means, that although a sound theoretical basis is desired, theory forming and adherence to a chosen paradigm is not the primary goal. The need for useful and robust systems calls instead for a flexible approach, subordinating aspects like cognitive adequacy. This might result in the need for the incorporation of localised theories in sub-areas, especially when, as is the case with most aspects regarding figurative language, no comprehensive computational account for certain phenomena has been given yet.

4.3 Evaluation criteria

There are some reasons why no comparative evaluation in the traditional sense can be carried out. Firstly, the systems discussed in Chap. 3 are not generally available for testing. Secondly, some systems are not comparable either in size (toy systems or evaluation implementations as NETMET [Steinhart 1993] or Sun's microfeature-based approach incorporated into CONSYDDER [Sun 1995], consisting of nine relevant concept nodes) or in the direction of their thrust: the structure

mapping engine [Falkenhainer *et al.* 1989] deals with comparison statements based on analogues. Wilks' aim [Wilks 1975a] on the other hand was to extend an existing system so it would tolerate input that was a little outside the system's scope; which is in itself a laudable approach, in line with the NLE criterion of robustness.

Moreover, the implementations were based on a customised representation (facilitating the operation of the chosen algorithms) that was limited in size, granularity, coverage or structure. No work concerned itself with the extension of an existing system (with its own representation) in the direction of comprehensive processing of figurative language.

The aim of this work is therefore not to compete with models whose design bears a resemblance to self-fulfilling prophecies, but to assert and tackle the requirements and problems involved in processing figurative language in the framework of the existing large-scale system LOLITA.

The evaluation of NLP systems (particularly large-scale NLP systems) is an emerging and not yet well developed area. Due to the variety of NLP tasks and an abundance of more or less specific systems, there are no established criteria or methods to evaluate any given (sub-)system. Although competitions and workshops can provide a basic measure of how well a system (dedicated to single tasks, e.g. machine translation or information retrieval) fares with respect to others, this is not a sensible method for evaluating parts of a system. Problems arising in a different sub-part might be attributed to the component in question; an evaluation setup for the sub-task of processing figurative language is hard to conceive of. All the components the figurative resolution module relies on (e.g. parsing, semantics, pragmatics, the knowledge-base and generation) have to be up to scratch in order to show the actual performance of the resolution module.

Of course, a sub-system adhering to NLE principles should not make assumptions about the reliability of other sub-systems it depends on. Moreover, the average NLE system will not have the benefit of a small-scale, customised environment, as is the case for theory-evaluating models. Keeping these factors in mind, the only conceivable method for evaluation is a meticulous comparison of results in the light of their *setting*, that is the original goal, the data available and, most importantly the implicit factors such as back-door encoding of results and constraints.

An evaluation of the work presented herein thus consists of a review of the following points: how does the solution to the problem of non-literal analysis compare

to previous solutions, bearing in mind that they have been designed for a different framework and therefore a one-to-one transfer of results is not always possible, and how does the solution conform to the NLE criteria (c.f. Sect. 2.1).

4.4 Summary

This chapter introduced a detailed account of the problem area of processing figurative language in a large-scale, natural language engineered NLP system, mentioned aspects of the methodology observed throughout the work presented in this thesis, named criteria for the evaluation of the achievements, and presented the objectives of this work.

The objectives were to establish the feasibility of non-literal analysis within the LOLITA system and, to meet this objective, the state of the art has to be analysed. If the task proves feasible, a solution which is in keeping with the NLE criteria (see also Sect. 2.1) and which recognises and classifies figurative input should be designed. The integration of this solution into the LOLITA system, i.e. the extension of LOLITA's existing capabilities in the direction of non-literal language, embodies the final objective.

Chapter 5

Processing figurative language in LOLITA

In this chapter, the tasks and prerequisites for the successful completion of the problems discussed in Sect. 4.1 are presented in detail, and important details of the implementation and integration into the LOLITA system are given. The introductory section discusses theoretical issues of the computational treatment of figurative language in the LOLITA system.

5.1 Theoretical issues

We adopt an approach similar to the knowledge-based and conventional approach put forward by Lakoff, Johnson and Martin [Martin 1990, Lakoff and Johnson 1980, Lakoff 1993]. The deep entrenchment of well-established figurative schemata in the conceptual system, reflected in the systematicity with which they appear in linguistic expressions, is grounded in bodily experience [Lakoff and Johnson 1980, Johnson 1987], cultural or linguistic tradition [Lakoff 1993]. They determine how and under which circumstances we may use one concept to denote another or one relation to express another. It would, therefore, seem very inefficient to compute the relations between concepts time and time again for each occurrence of a figurative expression. Rather one should make use of the schemata to the best possible extent, as has been realised by others working in a computational framework. Carbonell, for example, states:

The problem of understanding a large class of metaphors may be reduced from a reconstruction to a recognition task. That is, the identification of a metaphorical passage as an *instance* of one of the general metaphorical mappings is a much more tractable process than reconstructing the conceptual framework from the bottom up each time a new metaphor instance is encountered. Each of the general metaphors contains not only mappings of the form ‘**X is used to mean Y in context Z,**’ but inference rules to enrich the understanding process by taking advantage of the reason why the writer may have chosen the particular metaphor (rather than a different metaphor or a literal rendition). [Carbonell 1982].

Furthermore, Sowa maintains with reference to the knowledge-based approach: “A standard catalogue of metaphors is probably adequate for interpreting everyday speech” [Sowa 1984].

The reasons for rejecting the analogy approach are that it is interested only in comparison statements (missing out on the large amount of metonymies) and that it computes the cross-domain correspondences anew every time a metaphor is encountered. Moreover, as it lacks the facility to recognise metaphors a priori, it fails to enable a system to distinguish between figures and anomalies without going through a complete resolution process.

We do not *replace* a figurative expression with a literal paraphrase. We keep the original input and the interpretation, as well as the method by which we arrived at the interpretation. So the figurative surface can be carried over into subsequent stages of processing, which greatly helps cohesion in dialogue, for example. However, at the same time, the fact that a figurative resolution has taken place is also recorded, together with the result. If a part of the system requests the meaning behind the figure, it can easily be retracted and serve as the basis for further processing.

This touches on the fact that we do not subscribe to the *repair* view on figurative language, i.e. input is analysed to yield a literal reading first; if this reading is found to be defective, a non-literal reading is computed [Searle 1979]. Neither do we express any a priori preference towards literal or figurative interpretations, as Fass’ met* model does [Fass 1991]. It should be noted, however, that in doing so, we are not merely shifting the responsibility towards a different component of the system. The goal of this work was to determine if and how figurative processing in the LOLITA system can be tackled. Therefore, integration into the pre-existing structures and flow models had to take priority. On request, our model delivers

information about whether given input is metaphorical or not and which class(es) of trope(s) it belongs to if it is figurative, and provides an interpretation of varying depth (depending on the depth of information that is available for the interpretation).

Instead of ranking the results in any order or attempting a literal interpretation first, our starting point is located at the level where sense creation [Gibbs 1994] takes place in the system. Although this is in keeping with cognitive results obtained from experiments, it was not cognitively adequate modelling but the necessity for robustness and coverage that motivated this design decision. Gibbs states:

One idea, called the *error recovery* model, assumes that sense creation is initiated only after the conventional meaning has been found to be in error [...]. This model posits that listeners recognise the need for a figurative interpretation of such an utterance as *The ham sandwich is getting impatient for his check* after it is seen as violating the maxims of truthfulness. [ibid.]

Although a preference for literal interpretation can be expressed (by disabling the figurative processing sub-system completely or requesting non-literal resolution for events individually *after* literal processing has taken place), the entry point for figurative processing is where meanings for lexical items and phrases are selected/created in the course of ‘literal’ processing (see Fig. 5.1).

Figures can appear in combinations and chains. We are able to account for such cases, as we do not treat the unit sentence but ‘events’; this means that any relation between concepts that is expressible in LOLITA’s representation formalism can be analysed (see Sect. 2.2.1 and Fig. 2.2). Concepts in LOLITA are not restricted to the level of lexical material but are rather nodes in the SemNet, defined by their relation to the rest of the semantic network. They can represent an entity (‘the beer in the glass on the table’, ‘Jake’), a class (‘surgeons’, ‘breweries’), a word sense (‘strip’, for example, has a number of senses, either as a noun or verb) or a complex conceptual structure, such as a whole story.

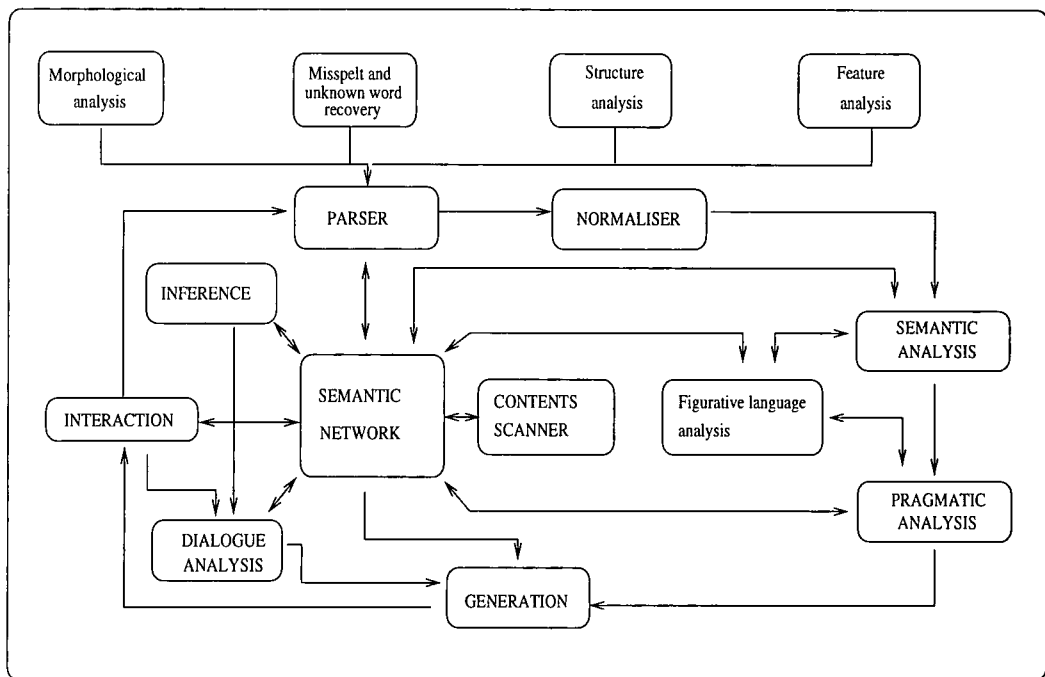


Fig. 5.1: Location of the figurative processing sub-system in LOLITA

5.2 Tasks and prerequisites

This section introduces the prerequisites for a computational treatment of figurative language and examines their manifestation in the LOLITA system. The tasks relating to the processing of figurative language are discussed and, subsequently, a classification of figures based on the requirements of their computational handling is given.

5.2.1 Resolution levels

In the following, the tasks that are part of the complex process of understanding figurative language are presented and the psycho-linguistic categorisation of tasks carried out during the understanding of figurative language (presented in Sect. 3.3.3) is re-evaluated in the light of a feasible computational framework.

The convincing categorisation of temporal stages and discrete processes of understanding figurative language proposed by Gibbs (see Sect. 3.3.3) and backed by experimental work [Gibbs 1994] is reflected in our model. Yet computational treatment of figurative language requires additional tasks to be carried out, such as representing, modifying and storing knowledge (about metaphors, about the process of understanding and its results). In the following, the tasks involved in

processing of figurative language in the LOLITA system will be described.

Comprehension

Comprehension does not play any central role in our model, because the retrieval of meanings, the lexicon lookup, the morphological and syntactical mechanisms we use in the processing of figurative language in the LOLITA system are the same as for literal processing. There are, however, two points to note. Although the equivalent treatment of literal and figurative processing is in keeping with cognitive models of linguistic processing, it has not yet been explicated which processes take place during comprehension. Meaning selection, for example (discarding or preferring one particular reading of a linguistic entity), or polysemy resolution would require at least a partial interpretation. If this interpretation is not accessible until a later stage, ambiguity has to be carried over into the following stages. This is the way in which the LOLITA system deals with ambiguities that cannot be resolved during parsing and lexical lookup. Ambiguities are carried over into semantics and pragmatics and, in our case, into the figurative processing sub-system. If they cannot be resolved, they are kept and carried over into the processing of the following input until enough information for their resolution is available. Although an interpretation (even partial) might help at early stages (e.g. polysemy resolution), the architecture of the system currently provides no alternative to going through all (or most) stages of resolving the literal and figurative meaning of an utterance before any information derived from the input can be used to discard improbable readings with an acceptable degree of certainty.

Moreover, inasmuch as the *assignment* of a figurative meaning to input is performed by the figurative language sub-system, the stage of sense creation (traditionally associated with comprehension) is indeed part of our model. Meanings for words and phrases not conventionally accessible (by lexicon-lookup) are *created* by the figurative interpretation, thus comprehension features in our model. But since the figurative interpretation process operates alongside the literal processing (i.e. it is not necessarily triggered by the detection of an anomaly *after* a full literal interpretation was sought, although the system can be configured to prefer literal readings), comprehension is not a process that has unique relevance to figurative processing.

Recognition

The task of recognition is central to our model. Aspects of robustness and usability are very influential at this crucial stage. The percentage of input correctly tagged as being metaphorical has to be maximised, and the percentage of metaphorical input not recognised as such has to be minimised. Ideally, all metaphorical input should be recognised as such, and no literal input should be tagged as metaphorical.

Recognition cannot be regarded as a uniform process for all types of figures (see Sect. 5.2.2 for a more detailed discussion). As even seemingly ‘simple’ metaphors require complex knowledge to be recognised as such and since there are cases where both literal and figurative readings are possible, it is assumed that, in the absence of other criteria for recognition, even the shallowest of interpretations indicates the presence of a figurative utterance (which might also have a valid literal interpretation) and that the absence of a figurative interpretation indicates that the input is not metaphorical.

It is impossible for any type of figurative expression, let alone all of them, to give even a heuristic for their safe recognition. Some previous approaches to metaphor (most computational models) took the violation of certain rules (communicative, truth conditional or semantic) not only as indication but as a necessary and sufficient condition for the existence of a metaphorical expression [Levin 1977, Searle 1979, Fass 1991, Carbonell 1982]. But the evidence of counterexamples has shown that, although these criteria might help in the task of recognising figurative language, they neither support a clear distinction between literal and non-literal, nor do they guarantee a high success rate in detecting metaphors.

Employing methods such as selectional restriction violation is helpful in a limited area, which leads us to use them in our model without relying on them exclusively; this is because the requirements of robustness favour failure to recognise correctly rather than incorrect classification of literal input as metaphorical.

Truth conditional criteria are of even less value, since they rest on a clear distinction between literal and figurative language or, in their extreme form, deny figurative language to have any truth value at all. Assigning a truth value to a figurative expression before the stage of interpretation does, however, not make sense. When ‘Becker kills Lendl on the Centre Court’, it follows that ‘Lendl is dead’. Truth conditionalists would claim that since ‘both tennis players walk off the court’, either no truth value can be assigned to the expression, or it is plainly false.

Not so, after we have arrived at an interpretation of this instance of the COMPETITION AS PHYSICAL AGGRESSION conceptual metaphor, where competition is seen and described as direct physical aggression and the entailments (metaphorical mappings) between the domains of aggression and competition have been taken into account. Killing, beating and winning battles are successes in aggressive behaviour, whereas being killed, being beaten and losing battles constitutes failure in aggressive behaviour. In accordance with the metaphorical mappings of the COMPETITION AS PHYSICAL AGGRESSION metaphor, success in aggressive behaviour is success in competition, and failure in aggressive behaviour is failure in competition. So Lendl is really (although not literally in the literal sense of the word) dead, because he failed in a sporting competition against another tennis player. As can be seen from this example, it is not valid to claim that metaphors have no truth value, especially when this claim does not take into account the interpretation or actual meaning of metaphors; it is impossible to decide on the truth value of a literal expression if the meaning of the expression is not taken into consideration. The false assumption about metaphorical truth seems to arise from the lack of consideration for the various stages of linguistic interpretation.

In general, there are different criteria for different types of figurative expression which can serve in their recognition. For the knowledge-based figures (see Sect. 5.3), the correspondence of the input to knowledge about the form and instances of figures can be utilised. The anomaly detection mechanism can be used as a fallback method, but it fails to recognise a number of metaphorical expressions and, more importantly, it cannot help to distinguish between figurative language and anomalies (see Sect. 3.2.2). The role of context and an array of kinds of knowledge is important for the recognition of figurative input. An utterance like ‘the King lost his head during the revolution’, when seen in isolation, can be taken either literally or figuratively. Only world knowledge will help to disambiguate. A special twist of figures is that they can *combine* literal and figurative meaning, leading to an unresolvable ambiguity. The example above exclusively illustrates this point. Humour is often based on such unresolvable ambiguities where multiple valid interpretations are possible. As a last resort, finding a shallow interpretation serves as an indication that a metaphor has been encountered in cases where no other method can be applied. Shallow interpretation in this context means that the input can definitely be classified as belonging to a specific type of figure.

Interpretation

The essential sub-task of understanding figurative language is interpretation. But what exactly does interpretation of figurative language mean? The interpretation of a sentence like ‘The university told him to apply’ seems to differ significantly from the interpretation of a sentence like ‘They drank all the bottles’ which in turn is different from the interpretation of a sentence like ‘Becker killed Lendl on the Centre Court’ or ‘Your mother really is an elephant’.

While the first example does not even seem figurative, the last one has a decidedly odd ring to it, if taken literally. It seems that there are not only different types of interpretation needed for different types of figures, but that there are different levels or depths of interpretation needed to ‘understand’ a figure of a certain type.

- ‘The university told him to apply’ is figurative depending on whether we see the university as a collective or abstract body (which cannot perform utterance acts, thus a personification) or whether we take ‘university’ to stand for the office or employee that told the person to apply (thus a metonymy). It is obvious that this kind of figure is so common that humans do not seem to encounter any problem when faced with instances of it. No deep interpretation is performed, and the utterance is merely accepted.

It has to be noted that not all input that is ‘slightly’ outside the normal range of expected input should be accepted. It is perfectly reasonable to restrict a system’s analytical capabilities by e.g. selectional restrictions (only humans perform utterance acts) to simplify tasks such as anaphora resolution (e.g. ‘My wife went to see the Cutty Sark in Greenwich. She told me about it’, where ‘she’ will presumably not refer to the ship). But the restrictions should not interfere with other tasks the system has to perform, or the restrictions have to be loosened again when those other tasks are carried out. In the case of figurative expressions, the system’s restrictions have to be loosened in a formal manner to allow the acceptance of figurative expression at least (this being the lowest level of interpretation).

In the above example, no deep interpretation is needed because the meaning is quite clear and, more importantly, no further entailments arise from the figurativeness. This is not the case in the following example:

- ‘Becker killed Lendl on the Centre Court’. Understanding poses no problem to humans here either, but if taken figuratively, more can be gathered from

the expression than ‘Becker won a fierce competition against Lendl, who was defeated’, e.g. the contestants are seen as adversaries, the contest is bitter and serious and force was used. In such cases, accepting the input and arriving at the immediate figurative meaning is only a (necessary) first step. Finding the possible entailments constitutes a further level of interpretation. The next example lies somewhere in between these two extremes of interpretational requirements.

- ‘They drank all the bottles’ cannot merely be accepted and yet has no entailments. Bottles cannot be drunken, fluids can. Bottles prototypically contain fluids, and the highly conventional figure of expressing a part through the whole (and its instance using a container to stand for the content) makes the task of interpretation straightforward. The container is used, the content meant. But interpretation should in such cases not simply be regarded as replacing a figurative expression with a literal paraphrase. This would reduce the validity of the analysis, as can be exemplified by one natural language processing (NLP) task, namely, machine translation. If the source text contains a figurative expression and the target text does not, the quality of the translation is adversely affected. Therefore, it is necessary to retain as detailed information as possible about the figurativeness of the input once it has been analysed. No conventional correspondences between what is said and what is meant can be postulated for cases such as:
- ‘Your mother really is an elephant’. Here, interpretation is a deep task which cannot be performed on the basis of lexical or conventional information. Knowledge about one particular mother and elephants in general and the world they live in and their relationship is needed. Moreover, knowledge about the situation in which the utterance was made will contribute towards the meaning as well. Interpreting this sentence means finding out about how this mother and elephants in general relate to each other in the context of the utterance.

Interpretation should not be considered to be a monolithic process. It has various depths, and the required depth at which a sufficient interpretation can be said to have been achieved varies with the type of figure at hand. Furthermore, information that a certain type of figure was encountered has to be regarded as part of the interpretation.

In the discussion of the recognition of figures, it was mentioned that recognition might in some cases rely on finding a shallow interpretation. This might seem contradictory, but becomes more reasonable in the light of dividing up the task *interpretation*. It is obvious that once we have arrived at full metaphorical interpretation for some input, we can classify the input as being of a certain type of figurative language. But it is also possible to arrive at safe knowledge about the figurativeness without being able to or having to arrive at a full interpretation. The minimal interpretation consists of stating that the input can be safely assumed to be metaphorical.

This approach was taken in our model to maximise the percentage of figurative input correctly classified as such and to minimise the percentage of non-figurative utterances classified as metaphorical, since it is possible to ascribe a metaphorical interpretation to almost any utterance. Thus there exists an intersection of recognition and interpretation, rather than a clear-cut boundary between the two stages. Minimal interpretation means reliably ascribing a type of figurativeness to the input, which amounts to recognising the input as metaphorical; the recognition as a certain type in turn counts as the minimal interpretation.

Since mechanisms like constraint violation cannot guarantee the detection of figures, and a futile attempt (in case the input is not figurative) at finding an interpretation is not acceptable in terms of resources, the interpretation stage is split into ‘primary interpretation’ and the ‘extended interpretation’. The primary stage classifies the input as belonging to a type of figure (e.g. ‘They drank all the bottles’ as CONTAINER FOR CONTENT metonymy) and the second stage tries to find the actual interpretation. Failure of the primary stage is taken to indicate that the input is not metaphorical. As mentioned earlier, the extent of the individual stages varies from type to type. Some types may not need an extended interpretation in all cases, while others require a considerable amount of processing in the primary stage and less in the extended stage and so forth.

Appreciation

Appreciation (in a non-localised way) is clearly beyond the scope of not only this work but presumably also beyond the scope of (current) NLP systems. What does it mean to make “an aesthetic judgement about what has been understood” [Gibbs 1994] and what is necessary to appreciate figurative language? Clearly, there has

to be a basis for an aesthetic judgement, and this basis would have to encompass a very detailed speaker/hearer model, extensive world knowledge, access to vast amounts of contextual knowledge and explicit information about conversational goals and textual structure (what is appropriate, what is pleasant for the text-sort in question). Although the LOLITA system provides some of these requirements, at least in a rudimentary form, we do not believe that there are models of appreciation of linguistic understanding from any discipline that can readily be implemented, nor that localised theories will yield satisfying results.

First steps towards a crude form of appreciation can be taken, however. By analysing the input, classifying it and storing the occurrence together with the analysis for future reference, a statistical evaluation of the frequency of any given type and the converging of instances of a generic metaphor, i.e. a basic corpus analysis regarding figures, can be carried out. Although this would enable a coarse stylistic analysis of the form ‘X idioms have occurred in the text, X metonymies of the PRODUCER FOR PRODUCT type...’ which is an interesting possibility in itself, it is a far cry from a real model of appreciation. This is, however, no real drawback, as the task of appreciation is not one of our goals, since appreciation “[...] is not an obligatory part of understanding linguistic meaning [...]” [Gibbs 1994] and, therefore, can be neglected in the framework of this enterprise.

Representation

The task of representation comprises a number of distinct sub-tasks. As we subscribe to the knowledge-based view, one aspect of representation concerns *figurative knowledge*, i.e. knowledge about the form of figures (for individual types see Sects. 5.3.2–5.3.7), their instances and extensibility, their possible hierarchical structure and the concepts which can be involved in a type of metaphor as well as the relation the concepts have when seen in the light of the particular figure.

To give an example, the CONTAINER FOR CONTENT metonymy has the linguistic form of expressing the contents of a container through the use of the container, which is the relation the figure establishes between those concepts. It can be instantiated in many forms (almost unrestrictedly, as long as the schema is kept) although some forms seem (through frequent use) more apt. To avoid wearisome repetition of the ‘drink a bottle’ example, let us consider ‘my handbag was stolen’ instead. Even if the handbag in question is retrieved later on, the incident will still

be referred to as theft of the handbag, although the contents of the handbag are usually of greater importance and, expensive designer handbags aside, the target of thieves. This metonymy has no extensions, but stands in a hierarchical (sub-type) relationship to WHOLE FOR PART/PART FOR WHOLE figures and in a super-type relation to generic instances such as BOTTLE FOR BEVERAGE and individual instances of related figures ('The kettle has boiled').

The knowledge about the form and relation between figures is explicitly stored as natural language (NL) information in the SemNet in the form of events (c.f. Appendix B.1). In this way, all parts of the system have immediate access to knowledge about metaphors. This is an explicit modelling of knowledge that might or might not be subconscious in humans. This evidential insecurity poses no problem for us, since we do not claim cognitive adequate modelling and since it does not mean that LOLITA is 'more conscious' of the knowledge stored explicitly. If at all, it is an improvement with respect to other systems, where the distinction between algorithm, knowledge about metaphors and other knowledge is not made as clearly as in our model and implicitly represented aspects contribute considerably to the performance. The effect of representing metaphoric knowledge in the SemNet is twofold; first it extends the knowledge base (KNB) with detailed linguistic knowledge and, second, makes all of the knowledge stored in SemNet available to the figurative processing sub-system. This, in turn, has two entailments.

On the one hand, it must be stressed that the knowledge about metaphors is not 'alien' to the structure and the type of LOLITA's other linguistic, world and encyclopaedic knowledge. On the contrary, we only make use of the concepts and structures used in the rest of the system, which. On the other hand, this latter point makes life slightly more difficult for the figurative sub-system, as it has to extract various forms of knowledge from the SemNet, instead of being able to rely on customised information structures (see Sect. 5.2.4). But the overall benefits of mutual knowledge exchange, the fact that no discrepancies can arise from a shared KNB (which might be the case if separate representations were chosen) and the savings in terms of resources (no additional coding, maintenance only for one KNB, a NL knowledge engineering interface) outweigh this slight difficulty.

The knowledge about metaphors (schemata) is arranged hierarchically in our model. It is possible to extend generic or more abstract schemata by using the knowledge LOLITA provides about the concepts participating in the schema, thus enabling us to traverse the hierarchy of schemata in both the upward and downward

directions (important in the recognition stage), and to generate new and valid schemata 'on the fly' (important for deeper interpretation and learning/extending metaphorical knowledge).

The knowledge for all types of tropes (apart from comparison statements, but with the exception of comparison statements which are reducible to conceptual metaphors) is represented uniformly in this fashion. Although the initial hierarchy for idioms is flat, and contextual metonymies cannot initially be represented in this way, but once a contextual metonymy is encountered, it is added to the SemNet in a uniform fashion.

It should be noted that the input/extension of the knowledge about metaphors is possible via the NL interface of LOLITA. The hierarchical ordering of metaphorical schemata was, in the first instance, considered to be fixed taxonomically, i.e. a PRODUCER FOR PRODUCT metonymy is the super-type of a AUTHOR FOR WORK metonymy which in turn is the super-type of a PLAYWRIGHT FOR DRAMA and COMPOSER FOR COMPOSITION metonymy. But such an approach might give rise to possible inconsistencies between the knowledge about metaphorical forms and the other knowledge in the KNB. The hierarchical organisation of figurative knowledge has to follow the organisation of the concepts involved in the figures to allow for 'holes' in the system's knowledge and areas of low granularity, to make the best use of areas of high granularity, and, most importantly, to keep in synch with any dynamic update of the system's KNB. Therefore, the events representing the figurative schemata of knowledge-based tropes are not ordered hierarchically as such, but obtain their hierarchical ordering through the ordering of the knowledge about their parts.

Other representational tasks certainly include the results of figurative processing, be they positive or negative. This means that not only will one or more possible interpretations have to be represented, but also that in the case of failure to find an interpretation, this has to be recorded as well. It is generally not a good idea to regard an algorithm as a black box, to feed it some input and wait excitedly for some output without being able to keep an eye on what is happening in between. This is even more important in a large-scale system (as opposed to smaller, domain-dependent KNBs, where the topology of the representation, due to its limited size, might be known by heart) where robustness and coverage require a system to deal appropriately with input outside its scope. To elaborate on this: it is not acceptable for the processing to stop for any reason or not to produce any

results. So the worst-case scenario should be that a statement is made about what was done and what the result of the action was, even if no output of the expected form is produced.

As it is impossible (for a non-toy system) to anticipate all possible states that might arise during processing, we have taken the approach of internally documenting each step taken. Following the requirements of integration, the information is stored in the SemNet in a form that is accessible by all other parts of the system and lends itself to NL generation. Thus it is possible to obtain ‘intelligent’ results, even in the case of failure of the intended processing. With regard to avoiding a black box model, the path taken by our algorithm in arriving at an interpretation is documented as well, again using the SemNet as the common data structure to store information. The documentation includes the type of metaphorical resolution applied to the input, whether the application was successful or not, and any entailments the successful or failed interpretation has, thus making the *chain of metaphorical reasoning* (see Sect. 5.3.1 and Fig. 5.12) explicit and therefore accessible to evaluation and revision. The chain of metaphorical reasoning serves as a representation of the side effects of the interpretation, such as a statement of the metaphorical relation between the concepts and the input (x is seen as y in the light of the figure z).

Explicitly representing detailed information about figures also forms the basis of extension, learning, and type change. By extension we mean it would be a valid yet restricted achievement to be able to recognise and interpret only those portions of input for which processing facilities (in the form of metaphoric knowledge) have been provided beforehand. Systems that put emphasis on their process model may take this stance, but the need for robust and useful processing makes it necessary to process as large a percentage of input as possible with a limited number of rules and knowledge. So the representation has to facilitate the systematic extension of coded knowledge to cover related cases. By using the SemNet as a basis for our representation (thus having a rich knowledge structure on hand), the possibility of extending metaphoric knowledge is given. For example, the conventional metaphor STATES ARE LOCATIONS might be present in the KNB in the form of ENTERING STATE AS STATE CHANGE (‘fall into a depression’, ‘enter Xemacs’, ‘go into receivership’). The antonym, hyperonym and synonym relations already present in the SemNet can be utilised to extend the given metaphoric knowledge to capture systematically related schemata. The extent to which the existing relations can be

used has to be delimited on the algorithmic side, however, as free extension along all possible lines of concept relations would over-generate.

Type change is a side effect that has not been a primary goal of this work, yet holds interesting possibilities. It concerns the classification of a given figurative expression as belonging to a type of figure and the possibility of dynamically re-asserting this classification. An a priori classification of any expression can only be made on the basis of vast amounts of linguistic and world knowledge. As LOLITA (like any other current NLP system) lacks this amount of knowledge, any classification that is made by the system might initially be ‘wrong’. Thus a dynamic re-classification as soon as relevant new information is available can help to correct initial (bona fide) misjudgements. If, for example, input along the lines of ‘a warm welcome’ is encountered, and the figurative analysis finds a conventional metaphor relating warmth to affection¹ whereas there is no lexicalised reading of ‘warm’ as affectionate (the only possible lexicalised reading being based in turn on the AFFECTION AS WARMTH metaphor, which for the sake of the argument we shall ignore), the input will be classified as an instance of a conventional metaphor. As Plantinga noted with respect to the dimension of novelty, a dimension that is implicitly echoed in our classification: “But a ‘dead’ metaphor is not dead for everyone. Children, for example, are frequently puzzled by a ‘dead’ metaphor such as ‘out to lunch’. [...] What is ‘dead’ and what is ‘live’ does not depend on the linguistic expression, but upon the mental model of the language processor.” [Plantinga 1989]. If a figure that is new to LOLITA has rightly been classified as novel, but subsequently similar expressions are encountered, the metaphor will ‘age’ as far as LOLITA is concerned. This is possible on the basis that metaphors and their instances (realisations) are linked and that events in the SemNet (newly acquired concepts and their relation) are referred to by a unique number whereby a higher number indicates a recently acquired concept. As mentioned before, this is only a sideline, although a promising one. Therefore no metric, which should be based on empirical results of corpus analyses, is available to determine the age at which a change from one type to another should occur. Yet it enables the system to model novelty and conventionalisation (up to idiomaticity) to a certain extent, aided by the fact that the representation of metaphoric knowledge in the central SemNet structure itself is freely accessible and modifiable.

¹A well-established metaphor, c.f. ‘To give someone the cold shoulder’, ‘He is an old flame of hers’, ‘I cannot warm to him’, ‘She has such a cold, logical mind’.

5.2.2 Types of figures

In this section, a classification of types of figurative language is presented which is not based on the traditional criteria which are often derived from classical sources, but on the needs of computational treatment, with emphasis on the LOLITA framework, the operations to be performed on the input (see Sect. 5.2) and the adopted knowledge-based approach. The computational needs are multidimensional in as far as they consist of different criteria such as the depth of processing (see Sect. 5.2.1), the kind of knowledge that is needed for the different depths of processing, whether reasonable shortcuts can be taken to process them, and how their representation is best achieved.

There are different ways of classifying figures, such as the degree to which they are conventionalised:

- Lexicalised figures ('leg of table', 'head of department'), where the listing of an additional word sense is sufficient to account for the meaning, even though the expression might be derived from a metaphoric principle
- Fixed meaning correspondences, allowing for little variation and not implying any significant further cross-domain relations (instances of personification and metonymies such as 'IBM said')
- Highly conventional figures (metonymies such as PRODUCER FOR PRODUCT), which are productive and have systematic instances (AUTHOR FOR WORK)
- Conventional figures (LIFE AS JOURNEY, COMPETITION AS PHYSICAL AGGRESSION), which are generic cross-domain mappings with many possible instantiations
- Novel figures, which are variations, extensions and combinations of the other classes, otherwise they could not be interpreted.

Other dimensions include:

- What kind of knowledge is needed to be process them, i.e. to complete individual resolution levels for the figure in question.
- The depth of resolution required to arrive at a sufficient interpretation. Conventional metaphors such as 'being in trouble', 'swallowing a lie' are motivated by conceptual schemata, but do not generally entail many related

cross-domain correspondences (they do not imply extensions of the metaphor they are based upon above the level of other lexicalised material, e.g. 'get out of trouble again'), therefore a less thorough analysis is sufficient and a more constrained representation and resolution might in individual cases be the most efficient option.

- The processing stage at which a sufficient resolution can be achieved. Simple figures may not need an interpretation as such, accepting them (instead of rejecting them as anomalous) can be considered appropriate, depending on the NL task performed.

Allegory

The textual scope of allegory is, by definition, above the phrasal level, i.e. there have to be at least two phrases in close contextual proximity which are metaphoric and whose metaphoricity is closely related. The main impact of an allegory is on the resolution level of appreciation. The individual metaphors must be analysed and a connection between them must be made in order to distinguish a chain of unrelated (i.e. coming from different domains) metaphors from an allegory. Moreover, it is to be expected that other parameters such as the speaker/hearer relation and the situation of the allegory (in a literary text, presented as a speech, the authorship, the purpose of the text containing the allegory) play a role in the full appreciation of the trope. As neither the resolution level of appreciation nor any higher contextual analyses are within the scope of this work, allegories will not be considered in this framework. The basic functionality which is the goal of this work can, however, serve as a basis for the recognition of allegories, as the analysis of the individual figurative expressions making up an allegory (classification and storage/representation) is provided for.

Analogy

Analogy is similar to metaphor in that each involves two concepts or domains, but analogy (in its prototypical sense of 'A is to B as C is to D') is more explicit. Of course, at one end of the continuum where the explicitness is low, analogy intersects with metaphor and comparison statements. Analogy in the prototypical sense is not of concern to this work, as the 'meaning' of the analogy is not figurative. There are, however, more opaque analogical expressions and the intersection of research

in the fields of metaphor and analogy shows that the phenomena are related to a certain extent. For example, ‘the electrons orbit the nucleus like the planets the sun’ could easily be classified as an analogy; ‘the atom is like the solar system’ exhibits characteristics of a comparison statement (being an implicit analogy); and a single cross-domain mapping between the domain of solar systems and physical models like ‘the orbit of the electron’ can not easily be explained as an analogy if the metaphorical background of systematic correspondences between the two domains is not considered.

Antonomasia

Antonomasia is a figure founded in stylistic and rhetoric motivation. Its behaviour and instances can be expressed by figures from the class of metonymies and idioms, and it will, therefore, not be treated as a separate entity.

Emphasis

As with allegory, the significant point of emphasis is on the resolution level of appreciation. As far as a semantically ‘more poignant expression’ is used, the content of an emphasis can be captured by the type of metonymy. The analysis (on the resolution level of appreciation) of pragmatically motivated uses cannot be captured without a detailed pragmatic model.

Euphemism

The notion of ‘taboo domain’ and positive/negative connotation are, again, directed more towards appreciation. The lexical form of euphemisms, especially since they are motivated by cultural and other extra-linguistic factors and thus conventionalised within a speaker community, can be resolved using the class idiom.

Hyperbole

One aspect pertains to the appreciation of hyperbole, namely the heightened emotive impact. The underlying meaning of the expression, however, has to be made explicit if comprehension (interpretation) and further processing involving the expression are required. If, for example, an utterance like ‘he drank a gallon of milk’

is encountered, it is at least necessary to arrive at the meaning ‘he drank a large amount of milk’. A number of conventionalised hyperbolae can be treated using the class idiom.

Generally speaking, a prototypical dimension is needed to resolve hyperbolae. Fluid intake by humans, for example, prototypically ranges from x ml to y ml over a period of time, waiting for someone to arrive at an appointed time is prototypically limited to a range within a couple of minutes to less than an hour.

It should be noted that those metric prototypes are highly context dependent and that there is graded acceptability, decreasing with the degree to which the prototypical range is exceeded. But it is not possible to determine for a set of prototypes just when the acceptability drops too low and the utterance has to be classified as hyperbole.

The amount of time spent waiting at a bus stop has a different prototypical range from the time spent waiting for one’s turn at the dentist’s, which in turn is different from waiting in a queue at the Post Office. It is not surprising that, as Gibbs noted, “[there] is no published experimental research in understanding hyperbole and understatement” [Gibbs 1993]. As the ‘exaggeration’ of one dimension central to hyperbole can only be detected on the basis of prototypical ranges (realised as world knowledge or as lexical information) and this information is not currently available in the LOLITA system, hyperbolae (and understatement) will not be treated in this framework.

Idioms

As opposed to traditional views, idioms form a continuum from fixed to more flexible expression, i.e. some idioms can undergo a change in lexical material without losing their idiomatic meaning.

Gibbs distinguishes between ‘lexical flexibility’ of idioms and ‘semantic productivity’ [Gibbs 1994]. Lexical flexibility means that parts of the idiomatic expression can be changed without affecting the meaning of the idiom. “For instance, some idioms, such as *button your lip*, can be changed (to, e.g., *fasten your lips*) without loss of meaning” [ibid.]. In other cases, the change will result, according to Gibbs, in a (subtle) modification of meaning but no loss of idiomaticity; “For example, the idiom *break the ice* can be altered to form *shatter the ice*, which now has the meaning of something like ‘to break down an uncomfortable and stiff social situation

flamboyantly in one fell swoop!’ ” [Gibbs 1994].

This is an interesting observation, although it is not quite clear why there should be no (subtle) change in meaning when ‘fasten’ substitutes ‘button’, whereas there is a change when ‘shatter’ substitutes ‘break’. But depending on the depth of the analysis, the change in the latter example can be neglected, since the main direction of the interpretation will still be to ‘break down an uncomfortable social situation’.

Idioms may not be the most striking of figures, but their ubiquity and their productivity as part of or motivation for other types of figures makes it necessary to include them in a model of real text figurative language analysis, as they do not follow the compositionality principle and cannot, as described above, be fully accounted for by lexicographic methods.

Irony

In addition to the linguistic devices indicating an instance of irony (modals, hyperbole, exclamatory phrases, intonation), pragmatic and non-linguistic factors have to be taken into consideration as well, e.g. given an utterance ‘Very nice weather, isn’t it?’ in a context where the weather is not nice at all. A closer examination reveals some further problems relating to the figure of irony:

Irony is defined as the expression of the opposite of the intended meaning [Bußman 1990]. Although this definition is not very clear, it is acceptable as intuitive. But what exactly does ‘the opposite’ mean, and, most important, what might ‘intended meaning’ be? It is possible to state the opposite of any meaning only after the meaning has been understood in the first place, leading to a circular definition. To narrow down the range of possible ‘intended meanings’, the recognition of irony relies on the violation of truth conditions, beliefs and contradictory discourse elements (c.f. the example above).

The context is of central importance, even more so than with other tropes. It seems that both types of context (linguistically speaking context and cotext) are important for the recognition and interpretation of irony, namely; the actual linguistic context (cotext) and the situation of the speaker and hearer in the real world as well as non-linguistic (context) communicative information (e.g. gestures). In many cases, both context types are hidden or at least not easily retrievable, for example in cases where presupposed knowledge is assumed. For example, A says to B ‘There won’t be many people in town, so we can go shopping’. The town

is full of people and B says to A 'Hardly anybody in town today' to indicate (a) that A's statement was untrue and (b) that this was noted and caused a certain reaction from B.

Irony, as a subtle means of communication, hardly ever works where one of the contexts is left open, A case where one makes a statement only to contradict it straightaway is hard to conceive of and certainly not prototypical case of irony.

This leads to the conclusion that the 'expression of the opposite' (the negation or some form or denial of something which has already been asserted a good degree of belief) operates on information not readily available in text and the surface structure of discourse. Data from a deep discourse analysis, belief and speaker/hearer models (roles a speaker and hearer play with respect to the conducting of discourse, presupposed, assumed and actual shared beliefs) in conjunction with world and situational knowledge (knowledge about defaults such as expected behaviour, [proto]typical relations between concepts and between participants in discourse, default expectations, scripts) is needed to be able to retrieve possible meanings on the basis of the given 'opposite' meaning.

The negation can take many forms, including purely pragmatic and non-linguistic ones. The same holds true for the presuppositions, which may also lie outside the scope of linguistic description. Consider the utterance 'nice weather' made in unpleasant weather conditions. It can be a reaction to the promise that it would not rain today; it can be said to someone who saw/heard/read the same weather forecast as the speaker (indicating nice weather) to make a comment about the reliability of weather forecasts; it can be used to indicate that even worse weather had prevailed in the past few days etc. Consider 'thank you' as a response to queue jumping or an unkind act such as being hit by someone. Here the presupposed (if indeed it is legitimate to construe any presupposition) default is well outside the realm of linguistics. If one assumes that unfriendly behaviour is not the norm, postulating that friendly behaviour is normally rewarded with 'thank you', the negation would lie in the reward of unfriendly behaviour.

The amount of world knowledge needed for a thorough interpretation of irony (given that the recognition of irony can be accomplished), which should also encompass the *motivation* for the use of irony and its background (the presuppositions) if it is to be satisfying, is too large for a successful treatment of irony at the present stage.

These points lead to the following conclusion: Recognition of irony seems outside the scope of this work since many primarily non-linguistic sources of information have to be considered if more than very few 'shallow' instances of irony are to be captured. Since even recognition of ironic expressions is not achievable, the interpretation of ironic utterances has to be postponed for the time being. Despite being an interesting area of figurative language, irony will thus play no further role in the current framework.

Litotes

As litotes does not affect the meaning of an utterance in quite the same way and on the same level as, for example, metonymy, but seemingly has more impact on the stylistic level (the appreciation stage of linguistic understanding), we shall not concern ourselves with processing figurative expressions of this type. A future refined stylistic model [Emery 1994], would, however, certainly benefit from being capable of analysing and generating this figure.

Metonymy

The classical definition of metonymy as relating terms on a causal, spatial or temporal basis or via genus/species and part/whole relations certainly has to be extended, as attributes and features of a concept can also be used to denote another concept. The classical relations holding between concepts involved in a metonymical expression stem from the early days of scholarly treatment of metaphor rather than from empirical evidence.

The relations which hold between the source term (present in the metonymical expression) and the target term (denoted by the source term) are manifold, but yet restricted in a systematic way.

The ubiquity, productivity and conventional and systematic structure of metonymies has been unarguably shown by linguistic and psychological researchers and is backed up by indisputable empirical evidence. "Like metaphors, metonymies are not random or arbitrary occurrences, to be treated as isolated instances. Metonymic concepts [...] are systematic in the same way that metaphoric concepts are." [Lakoff and Johnson 1980]. "I claim [...] that there are various metonymic models in our conceptual system that underlie the use of many kinds of figura-

tive and conventional expressions such as OBJECT USED FOR USER. Many of these models depend on conventional cultural associations, which reflect the general principle that *a thing may stand for what it is conventionally associated with* [Turner 1987]. Nevertheless, metonymy has not received the same amount of attention as ‘metaphor’, as reflected in the following quotation “My aim in discussing tropes other than metaphor is partly motivated by a desire to stem the inflation of metaphor to the status of master trope. Figurative language researchers in the cognitive sciences have been especially guilty of ignoring tropes other than metaphor” [Gibbs 1993]. Surprisingly, no attempt to draw up a hierarchical account of metonymical concepts, comparable to the ‘Master Metaphor List’ [Lakoff *et al.* 1995] has been made in the light of the conventional metaphor view. Usually, if they mention metonymy at all, treatises of figurative language give some selected metonymical relations between concepts, and then only enough to prove a point or to make the underlying principle understood. The classification presented in Fig. 5.2 can be seen as a first attempt² at giving an account of a small section of the hierarchical structure of metonymical schemata, gathered from texts and the literature.

Although the current trend in the scholarly treatment of figurative expressions is not to subsume metonymy under metaphor (where metaphor is used to mean expression of one concept in terms of another), we do not feel it is necessary to honour this approach. It is based on the notion that in the case of metaphor, the concepts involved come from different conceptual domains, whereas in the case of metonymy, they come from one domain. Gibbs gives the following criterion for distinguishing metaphor from metonymy: “A convenient way of distinguishing the two types of figurative trope is to apply the ‘is like’ test. If a non-literal comparison between two things is meaningful when seen in an *X is like Y* statement, then it is metaphorical; otherwise it is metonymic” [Gibbs 1994].

We regard metonymies as a more restricted way of expressing one concept in terms of another one, following conventional schemata governing the possible relations between the concepts. Yet we subscribe to the notion expressed by Lakoff and Johnson that metaphor “is principally a way of conceiving of one thing in terms of another, and its primary function is understanding” [Lakoff and Johnson 1980], whereas metonymy “has primarily a referential function, that is, it allows us

²It could be argued that relations 1 to 4 are special cases of the part-whole/whole-part schema, but this is subject to idiosyncratic ontological preferences.

Schema	Source	Target	Example
1	cause	effect	They live from hunting
1.1	producer	product	Jake bought a BMW
1.1.1	author	work	She likes Janowitz
1.1.1.1	poet	poetry	He reads Goethe
1.1.1.2	composer	composition(s)	She likes listening to Bach
2	person	office	Clinton stopped the bill
3	office	person	The director rang earlier
4	part	whole	They need a good head for the job
4.1	species	genus	Vipers and snakes (for reptiles)
4.1	object used	user	Table 3 wants more salt
5	whole	part	America (for the USA)
5.1	raw material	product	Jake burnt rubber
5.2	place(name)	associated concept	
5.2.1	place(name)	entity	
5.2.1.2	place(name)	institution	Eton, 10 Downing Street
5.2.1.2.1	place(name)	industry	Silicon Valley, The City
5.2.1.2.2	place(name)	company	Wolfsburg (VW), Windscale
5.2.1.2.3	place(name)	government	London, Paris
5.2.2	place(name)	occupiers	The House of Commons
5.2.3.	place(name)	event(s)	Three Miles Island, Chernobyl
5.3	container	content	She drank a bottle
5.4	genus	species	Mortals (for men)
5.5	institution	representatives	The school sent him home early
6	concrete	abstract	He refused the crown

Fig. 5.2: Tentative semantic classification of types of metonymies

to use one entity to *stand for* another” [ibid.].

Diachronically, metonymies find their way into the lexicon and lose their metonymical status, e.g. *to lynch* after the American judge *Lynch* or *algorithm* after the Persian mathematician *Al Chwarismi*. In those cases which can be of interest to other disciplines, we consider the meaning to be fixed and the metonymical principles at work, at least as far as the lexical material is concerned, to be finished. This means that there is no change in meaning and therefore no need to apply any form of resolution/interpretation, as the meaning can be retrieved most efficiently by lexicon-lookup.

There are other grades of novelty, besides conventionality and a fixed lexical meaning. Through their conventional status, some types of metonymy are rather fixed. That is, there is a highly conventionalised relation between the concept men-

tioned and the concept referred to, as is the case with CONTAINER FOR CONTENT metonymies such as ‘He had two glasses already’³, whereas others are fairly flexible and productive (schemata higher in the abstraction hierarchy, such as PART FOR WHOLE).

Since fixed correspondences between concepts by definition need less processing if knowledge about them is available, we propose a continual class of metonymies, ranging from *conventional metonymies* (productive schemata such as PART FOR WHOLE) over *fixed metonymies* (highly conventionalised schemata such as CONTAINER FOR CONTENT which yet cannot be listed as additional word senses) to very restricted *simple figures* (correspondences which are almost one to one and so highly conventionalised that they require only a very shallow resolution).

A phenomenon outside the dimension of conventionality is the role of context. Some metonymies follow the conventional schemata, yet refer neither to ontologically nor semantically but contextually related terms. As Gibbs noted about an example given by Lakoff and Johnson: other contexts “permit the use of referring functions that are not sanctioned outside those situations. For instance, the metonymic sentence ‘the ham sandwich is getting impatient for his check’ makes little sense apart from some specific context as when one waiter informs another that his customer, who was served a ham sandwich, wants to receive the check” [Gibbs 1993]. This class of metonymies that draw on contextual knowledge is the force behind the next class of figures we propose, namely, contextual figures.

In the following, the classes of simple tropes, fixed metonymies and contextual metonymies will be discussed in detail; the conventional metonymies are a subclass of the conventional figures and discussed in Sect. 5.2.2.

Simple tropes

Simple tropes are a class based on computational and knowledge requirements and are a subclass of the fixed metonymies. Although they might be derived from conventional figures, synchronically they have a rather fixed meaning and appear with a high frequency, and for these reasons a fast and shallow analysis is necessary and sufficient. They combine instances of metonymy, synecdoche and personification.

³In this and some other cases, one might state that the expression is simply elliptic (‘I finished the bottle of mineral water’, but consider other examples such as ‘This room is getting excited’, ‘The house went quiet’, where an extension to e.g. ‘All the people in this room are getting excited’, ‘Everybody and everything in the house went quiet’ seem somewhat contrived.

Apart from knowledge about the concepts involved, they mainly require knowledge about their form and possible instances and no or very little other knowledge (if no deeper analysis is required, as with most cases). Examples are highly conventionalised metonymies/personifications like ‘Ford said...’, ‘The university does not allow you to...’, ‘The school sent her home...’ where one concept is conventionally described in terms of another concept, but there is variation as regards the instantiation of the concepts used on the surface; all of the above examples involve organisations but denote different parts of the organisation. The task of recognising simple tropes amounts to establishing whether or not the concepts from the input stand in a figurative relation laid down by the form of simple tropes. There is no or only very little concept extension required (c.f. Sect. 5.2.4), as ontological and taxonomic knowledge of the desired depth is available in situ (‘Ford’ being an organisation, ‘saying’ being an act prototypically performed by humans). Interpretation amounts to either accepting the input as being figurative or additionally deriving a literal meaning which might be required for subsequent processing. As there are no entailments, no extended interpretation is needed.

Fixed metonymies

Fixed metonymies are mono-domain partly conventional, partly conceptual grounded mappings. They are a subclass of conventional figures, characterised by the fact that the apparatus needed for recognition is less complex, due to a clear relation between the concepts such as CONTAINER FOR CONTENT. They, too, need knowledge about the concepts involved, knowledge about their form and, more importantly than in the case of simple tropes, knowledge about possible instances (more generic and more special instances). They also require ontological knowledge. As we do not rely on our ontology to be omniscient, world or contextual knowledge is used to make up for missing ontological knowledge, some cases require shallow world knowledge in their own right. The method with which the KNB is consulted to retrieve the relevant knowledge is discussed in Sect. 5.2.4.

Consider typical examples of fixed metonymies such as AUTHOR FOR WORK or CAPITAL FOR GOVERNMENT. The mapping indicates an almost one-to-one correspondence, but the actual realisation of the schemata may vary to a great extent. If, for example, the input expression contains a concept of which we know that one of its features is ‘being an author’, we can relate it to the AUTHOR FOR WORK schema. If, however, no such knowledge is available for the input concept,

concept extension is required (c.f. Sect. 5.2.4). If the input involves a concept of which we know through episodic or world knowledge that one of its features is ‘being employed as technical writer’, we have to extend the input concept and will arrive at the knowledge that technical writers are authors, and we can then apply the schema. The same process is required for the CAPITAL FOR GOVERNMENT schema. The name of a city alone will not succeed in matching this particular schema (it may succeed in matching against different schemata) unless it is extended and we find out whether the city is a capital or not. Of course, the information on whether any given city is a capital or not is only present in the ontology in an ideal world. A fallback method is thus to carry over the concept extension into world and contextual knowledge. For example, if the information that Berlin is a capital is missing in the KNB, but previous input to the effect that ‘Berlin became what London had been for a long time’ has been processed and information that London is a capital is present in the KNB, this information can be used in the concept extension.

Recognition of fixed metonymies therefore requires a certain amount of concept extension, typically along ontological and taxonomic paths, in order to make a definite statement about whether the concepts from the input relate to concepts featuring in a fixed metonymical relation. Interpretation calls for the checking of the relation derived from a fixed metonymical schema against the actual input concepts. Extended interpretation of fixed metonymies consists of stating the figurative relation between the input concepts and their meaning in the framework of the figure.

There are constraints on the applicability of metonymical schemata⁴. These constraints primarily apply to the generation of figurative language. Generation needs a threshold of abstractness, above which the use of a schema is not permissible. This threshold should be computed dynamically from the ‘root’ of the metonymical schema hierarchy, depending on how deep the hierarchy grows. PRODUCT FOR PRODUCER for example, is a very abstract schema, situated ‘high’ in the hierarchy with sub-schemata such as AUTHOR FOR WORK, COMPANY FOR PRODUCT, WINE-GROWING REGION FOR APPELLATION, CHATEAU FOR WINE. If the only applicable schema available to the generation module relating to the concepts that are to be realised is too abstract, generation of a metonymical expression is

⁴One could hardly say “‘Mary was tasty’ meaning by *Mary* the cheesecake that Mary made” [Gibbs 1994].

not permitted and thus cases like ‘Mary is tasty’ (see footnote 4) are avoided. The necessity of taking these constraints into account is part of the motivation for a hierarchical structure of metonymical schemata. As far as the analysis of figurative language is concerned, it is sufficient to add information on the abstractness of a metonymical schema to the schema itself and to seek the most specific interpretation; if the only available, most specific interpretation is beyond an accepted level of abstraction, the input cannot be considered to reflect the metonymical principle of the schema (as it is possible to construct part-whole and other relations for almost any two concepts). But a further point should be made about Gibbs’ comment on the applicability of metonymical schemata. A contextual/idiomatic reference to the producer of a cheesecake or the cheesecake seems not so impossible after all. If Mary is well known for her habit of bringing tasty cheesecakes to social gatherings, an utterance like ‘That was a real Mary again’ or ‘Shh, the cheesecake can hear you’ seems perfectly acceptable; but it is dependent on the individual context, the role of which was touched upon earlier. The influence of context on the possibility to relate concepts via metonymic rules is reflected in the following class.

Contextual metonymies

In contextual metonymies, the source and target concepts are linked by a schema from a hierarchy of conventional metonymical mappings, not on the basis of ontological or taxonomic, but rather contextual or episodic knowledge. This contextual knowledge can be very localised in idiosyncratic contexts, such as special words married couples develop over time, a secret code between a group of schoolchildren referring to common background knowledge (common examples are references to a teacher on the basis of a salient feature such as glasses, clothing, hairstyle); localised (slang that encompasses the background knowledge of a larger linguistic group, technical expressions); or universal but restrained by other means (script and plans, for example). Therefore, the knowledge required to handle contextual metonymies comprises the knowledge required for the treatment of fixed metonymies and contextual information.

It is obvious that the boundaries between contextual tropes and idioms are not clearly marked. This is, however, modelled and accounted for in our approach, as any initial classification can be revised on the basis of further occurrences of a figure and the information gained by processing more input (see Sect. 5.2.1).

The task of recognising and interpreting contextual metonymies consists in finding a contextual referent that stands in a fixed metonymical relation to a concept from the input; interpretation is similar to the mechanism employed for fixed metonymies.

Personification

By personification, we mean likening of an inanimate or abstract entity to a humanoid, human or individual entity, application of human attributes or features to non-human referents (entities, events, etc.). Depending on the point of view, personification could be regarded as a specialisation of the inverse antonomasia, synecdoche or metonymy. In general terms, personification is a metaphor where one domain is restricted to humans and human behaviour. Lakoff and Johnson define personification by saying: “A physical object⁵ is further specified as being a person. This allows us to comprehend a wide variety of experiences with non-human entities in terms of human motivations, characteristics, and activities” [Lakoff and Johnson 1980]. Although personification spans a variety of types of tropes (conventional figures, metonymies, idioms), it is a useful category since it involves a clear surface marker. But as it is a class denoting the content rather than the structure, it receives no individual treatment. Its instances are processed using the mechanisms of other classes and only at a later point will a classification as ‘personification’ be made on the basis of the conceptual content of the figure.

Rhetorical question

Although the meaning of a rhetorical question, if the term meaning is applicable in this context, differs from the compositional meaning of the phrase that is classified as rhetorical question, the processes involved are situated in the domain of discourse analysis, communicative goals and speech acts. An indication of this is that rhetorical questions are subject to felicity conditions, i.e. they can fail if the addressee does not understand that the question was a rhetorical one and answers it. A discourse model is the appropriate place for the treatment of rhetorical questions, although they are traditionally listed alongside other tropes (presumably due

⁵This is a unnecessary restriction in our opinion, events and abstract concepts can be involved in a figurative expression of the type personification as well, c.f. a sentence like “The conversation died”. In the wake of their definition, Lakoff and Johnson themselves [Lakoff and Johnson 1980] present examples like “Life has cheated me” and “Inflation is eating up our profits”



to the fact that figurative language was traditionally dealt with in the framework of poetics and rhetoric).

Simile

As the class of simile is motivated by surface criteria (the presence of an explicit comparison marker) but can be subsumed under the more general class of comparison statements, we will not consider it in detail. It has to be noted that, similar to personification, the existence of a clear-cut surface criterion can be utilised for the task of recognition.

Synecdoche

Although traditional classification makes a distinction between metonymy and synecdoche, there is no evident reason why the two types should not be subsumed under one, preferably the more general type. In contrast to personification and simile, the restriction to a specific type of semantic relation in the case of synecdoche does not establish a useful class of trope. Involving concepts that stand in a certain semantic relation might be of importance to the resolution level of appreciation, but the other stages can not make use of this information. In this we follow Lakoff and Johnson: “We are including as a special case of metonymy what traditional rhetoricians have called *synecdoche* where the part stands for the whole. In these cases, as in other cases of metonymy, one entity is being used to refer to another” [Lakoff and Johnson 1980].

Metaphor

As discussed in Chap. 3 and Sect. 5.2.2, most theories and models have concentrated exclusively on comparison statements of the form ‘A is B’. The conventional theory of metaphor (see Sect. 3.2.5) however, uncovered a vast system of figurative schemata structuring cross-domain correspondences and the language expressing them. These ‘conventional tropes’ are another class we consider relevant for a computational model of figurative language processing. The name ‘conceptual metaphors’ means the same, but highlights the fact that those figures are based in the conceptual system. Since this dimension is not a primary concern for our work, whereas the dimension of conventionality is, we use the term ‘conventional

figures’.

Conventional figures

The super-class of all knowledge-based tropes. If the body of examples presented by various authors on the conventional (conceptual) metaphors is examined, it becomes clear that they subsume a range of tropes under the heading conventional metaphor, such as metonymies (CONTAINER FOR CONTENT), the class of conventional figures we are discussing (COMPETITION AS PHYSICAL AGGRESSION), analogies and proverbs (which are described by Lakoff [Lakoff 1993] as being instances of the GENERIC IS SPECIFIC metaphor) and ‘novel’ metaphors (poetic, creative metaphors, for which Lakoff argues convincingly that they have to be based on conceptual/conventional metaphors, otherwise they would lack the basis for their interpretation).

Our defining criterion for this class is that a figure has to fit into the conventional figure system whose existence and pervasiveness has been shown by Lakoff and others and that it cannot be represented in one of the lower classes. This means that although an expression like ‘swallow a lie’ is captured by the conventional metaphor IDEAS ARE FOOD (‘devour a book’, ‘warm up a theory’), an efficient resolution should aim at processing it at the smallest possible cost. Since highly conventionalised figures like the one mentioned do not usually entail anything, it is advisable to process them as idiomatic expressions. There are two points to note. Of course we cannot assume an a priori knowledge about which class an expression belongs to. If ‘to swallow a lie’ is encountered for the first time and no idiomatic knowledge is available, the processing will have to derive the meaning by other means, such as applying the conventional metaphor IDEAS ARE FOOD . But if in the course of further processing more instances of the figure are encountered, it is sensible to let the trope percolate down towards a more conventionalised class (see Sect. 5.2.1). The other point concerns the fact that even though such an expression should be handled efficiently, it is, for various reasons, not enough to simply replace the input with a literal paraphrase. Therefore, the conceptual grounding should be taken into account as well, that is, for possible tasks such as appreciation and subsequent processing, the fact that the expression is based in the IDEAS ARE FOOD metaphor should be kept in mind, whatever the resolution level. In doing so, the basis for simpler processing of figures in the same text (which will certainly pick up on a chosen metaphorical paradigm) and for stylistic analysis as well as enrichment

of the linguistic content of the KNB is created.

The nature of conventional figures deserves a further investigation, as they are the super-class of other types, e.g. fixed metonymies. Although a conventional metaphor such as COMPETITION AS PHYSICAL AGGRESSION seems to describe a straightforward and almost one-to-one correspondence between concepts (any concept related to competition can be expressed, adhering to the constraints set out in the mapping through a concept related to physical aggression), there is more to this than meets the eye.

First of all a conceptual metaphor involves two distinct domains which are related *on the basis* of the metaphor. Secondly, the mapping between the concepts is different from the correspondences in other figures. There is no one-to-one correspondence (as in basic idiomatic cases), some-to-one (as with variations of idioms), or one-to-some (instances of fixed metonymies, such as ‘To read Shakespeare’, where Shakespeare can stand for any of the author’s work, poetry or drama).

In the case of conventional figures, a systematic, extensible web of correspondences between domains is introduced, whereas it is neither possible nor necessary to extend the concept relation WORK FOR AUTHOR. Concepts from the source domain correspond to concepts in the target domain, but not necessarily on the basis of structural similarities between the domains; the correspondence is determined by the metaphorical mapping, or as Lakoff put it: “Each conventional metaphor, that is, each mapping, is a fixed pattern of conceptual correspondences across conceptual domains. As such, each mapping defines an open-ended class of potential correspondences across inference patterns” [Lakoff 1993].

In the case of the COMPETITION AS PHYSICAL AGGRESSION metaphor, one of the direct correspondences would be that ‘competing businesses’ can be metaphorically seen as contestants in a (physical) fight. But there are other additional and systematic correspondences, such as ‘advertising’ – ‘strike a blow’, ‘new products’ – ‘weapons’, and the correspondences are carried over into other domains as well; in a physical fight, striking a blow with a weapon can cause injuries, and injuries can be metaphorically seen as harm to the contestant, in this case, a business.

Although there exist highly conventionalised cross-domain correspondences, which govern not only linguistic expressions but also reasoning, the recognition and primary interpretation is most important in the current framework. Instead of signalling a semantic or other category violation, an utterance such as ‘IBM ended

the battle with Apple' should be accepted and assigned an interpretation. The fact that during the battle, Apple may have dealt IBM a blow with its advertising campaign, that IBM may have launched an attack by challenging Apple in court and so forth may well be part of the metaphor and even the text, but they should not be computed pre-emptively. The fundamental assumption is to avoid unnecessary work. If any of the elaborations on the metaphor ('deal a blow', 'launch an attack') are encountered in the same portion of input as the original metaphor 'IBM ended the battle with Apple', the fact that the figure was resolved in the first instance and classified as being an instance of the COMPETITION AS PHYSICAL AGGRESSION metaphor will help resolution of the further instances. An extensive computation to find as many members of the 'open-ended class of potential correspondences across inference patterns' is not, however, needed by any component of the current parent system, and should therefore not be considered, appealing as the task may seem.

There is also a range of conciseness as regards the cross-domain mapping itself. Simply stating that a concept from the domain of physical aggression can denote a corresponding concept from the domain of competition might not be enough, as it would allow for many odd utterances to be classified as instances of the COMPETITION AS PHYSICAL AGGRESSION metaphor. As the criteria of robustness dictate that no assumption should be made about the well-formedness of the input, more specific mappings have to be used for recognition of conventional figures. They can either be represented statically (as in Martin's work, where case-role correspondences restrict the applicability of conventional metaphors [Martin 1990]) or dynamically (making use of the developing knowledge available in the rest of the system). The second approach was chosen in our model for reasons of consistency (see also Sect. 5.2.1). This leads to a dependency from the rest of the system, but in terms of integration and re-use, it is a decision that was favoured over the introduction of external, inaccessible structures which would only be of localised use. A compromise is to add more detailed knowledge about the mappings in the form of NL information, which is knowledge that is not of immediate benefit to the system as a whole, but integrated and accessible for the figurative sub-system.

A class of figures between conventional tropes and their open-ended cross-domain correspondences and metonymies with their limited concept relations are the conventional metonymies. They are more flexible and more productive generic metonymical schemata with respect to their instantiations than the fixed met-

onymies and limited in comparison to conventional figures in that there are not numerous possible cross-domain relations, thus they are some-to-some mappings, an example of this class is the CAUSE FOR EFFECT metonymy.

The recognition of conventional figures requires knowledge about the schemata and becomes more secure with knowledge about the mapping established by the schema. Normally, a considerable amount of concept extension (c.f. Sect. 5.2.4) is necessary to find out about whether the input can be classified as an instance of a conventional schema. Moreover, world and episodic knowledge is required to ensure the applicability of the schema to the input. To clarify this, given input like 'IBM ended the fight with Apple', it would be venturesome to assume that the utterance is indeed an instance of the COMPETITION AS PHYSICAL AGGRESSION metaphor, on the basis of a partial match (fight is a concept from the aggression domain). Humans, however, find no difficulty in classifying this instance correctly. But the background knowledge to the effect that IBM and Apple are businesses operating in the same market to a certain degree, and that businesses operating in the same market tend to compete in a capitalist economy is present and suitable to bolster up the hypothesis about the metaphor. Concept extension, therefore, gathers information from all available sources including the context, and will, depending on the richness of the data available, succeed in correctly classifying figurative input as such.

The primary interpretation has to rely on knowledge about the mapping or the structure of the two domains to find a correspondence. In the above example it has to consider that (hypothesising it is dealing with a COMPETITION AS PHYSICAL AGGRESSION on the basis of 'fight') a fight has antagonists (and will assume that IBM and Apple fill this role). Having satisfied initial requirements, the interpretation will then have to find information that meets the requirements of the competition domain (that the antagonists are competitors and that a fight thus can correspond to competition of a non-physical kind).

For the extended interpretation, rich and highly interconnected world and encyclopaedic knowledge is required. The example cited does not necessarily require an extended interpretation, but consider the metaphor LONG-TERM PURPOSEFUL ACTIVITY IS A JOURNEY and its sub-schema LIFE IS A JOURNEY. All the implications this schema has (the mappings that can be made) need background knowledge if one wants to bolster up the interpretation beyond mere recognition. Some of the possible mappings are [Lakoff *et al.* 1995]:

- The person leading a life is a traveller ‘He sails through life’, ‘He was speeding in the fast lane’
- Life states are locations ‘He regarded his childhood as a prison’
- Long-term purposes are journey destinations ‘My goal is to become professor’, ‘He reached his retirement in good health’

If there is no or very little knowledge about journeys, information such as that a journey follows a path, that it has stages, a starting and an end point, that a person can use different means of transportation such as cars, legs, planes, that the path has a spatial dimension and the journey a temporal and spatial dimension, that there are impediments to travel and so forth, the interpretation will be limited to the recognition and expansion of the schemata to whatever knowledge is available. This is even more obvious in the ‘childhood as prison’ example. Recognition poses no problem, childhood is a life state and a prison is a location, but knowledge about the restraining qualities of a prison has to be available, otherwise the input can only be accepted as a conventional figure and classified as an instance of the LIFE STATES ARE LOCATIONS metaphor.

The implicit knowledge associated with concepts and the subtle relations that are established by a metaphor are the most prominent features of the last class of figures, which will be presented in the following.

Comparison statements

Comparison statements (see also Sect. 3.2.3) can either be literal or non-literal [Ortony 1979, Gentner *et al.* 1989, Gentner 1989]. They are not knowledge-based in the sense that there exists knowledge about generic forms of comparisons and rules for their instantiation, as is the case with e.g., the fixed metonymies. On the surface, one concept is compared to another concept in a comparison statement. This means that attributes of the tenor concept (c.f. Sect. 3.2.4) or its relations to other concepts are compared to the attributes and relations of the vehicle concept (c.f. Sect. 3.2.4). This seemingly simple statement becomes problematic under close scrutiny.

The most important question pertaining to comparison statements is: which attributes and relations are compared and how are they compared. The analogy

approaches [Gentner 1983, Gentner 1989, Gentner *et al.* 1989, Holyoak and Thagard 1989b] assumed that structural similarity played a central role in finding which information from the tenor domain is compared to which information in the vehicle domain. This notion was found to be sufficient only for analogies. For the many cases of ‘predicate introduction metaphors’ (c.f. Sect. 3.2.3), different criteria have to be used. This leads to a first distinction between analogical comparison statements and metaphorical comparison statements in addition to the previously made differentiation between literal and non-literal comparison statements (‘encyclopaedias are like dictionaries’ vs. ‘encyclopaedias are like gold mines’).

There is, in our opinion, no clear demarcation between literal, analogical and figurative comparisons, rather a continuum (see Fig. 5.3). It is difficult to classify many instances of comparison statements with absolute certainty. This uncertainty is echoed (it is hard to tell the cause from the effect here) in the various ways analogy, metaphor and literal expression are related in different schools of thought. The analogy approach [Gentner *et al.* 1989, Gentner 1989] regards analogy as the central phenomenon (c.f. Sect. 3.2.3) and metaphor as a special case, whereas the conventional view [Lakoff 1993, Veale 1995] argues for the opposite position.

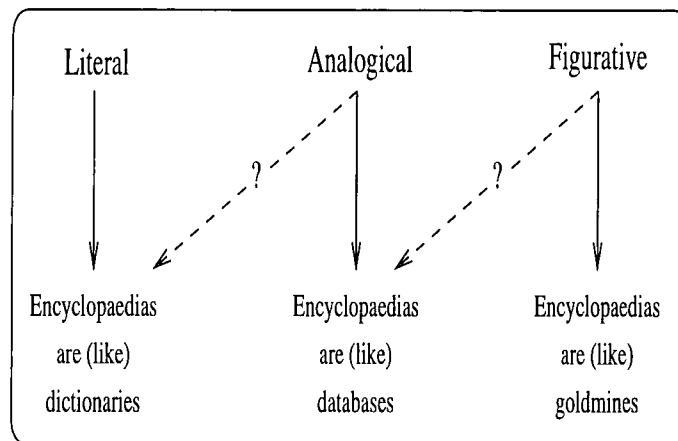


Fig. 5.3: Problematic classification of types of comparison statements

Clearly, the interpretation of literal comparisons differs from the interpretation of non-literal comparisons. Whereas an encyclopaedia *really* is like a dictionary, it is only *metaphorically* like a gold mine. But where does one draw the line between the ‘neighbouring’ classes of literal and analogical or analogical and figurative comparisons, respectively? Is an encyclopaedia really like a database? Is it more like a database than like a dictionary? There is no problem in saying that literally, it is more like a database than like a gold mine, but no general assertion can be made.

The reason for this, it seems, is the context dependency of such comparisons. Whether a comparison is seen as literal, analogical or figurative depends on the transfer of information from the vehicle into the tenor domain. Or, vice versa, the type of transfer made determines the type of comparison statement we are dealing with. Finding the exact kind of information transferred, however, amounts to the interpretation of a comparison statement. To give an example: in the literal comparison statement ‘encyclopaedias are (like) dictionaries’, the important, salient features of dictionaries are transferred to the concept of encyclopaedias. As mentioned above, this transfer does not require pre-existing information about the tenor domain (encyclopaedias) in order to take place. On the contrary, such literal comparisons are often used to convey *new* information about the tenor [Ortony 1979]. Amongst the salient features of dictionaries are: ‘being a reference work’, ‘consisting of alphabetically sorted entries’, ‘entries consist of heads and explanatory text’ and ‘educational value’. Note that there is no requirement for the hearer to possess knowledge about the tenor of the comparison. A suitable vehicle is chosen to ensure the correct transfer of (presumably) as many features as possible; but always in a certain context. Consider the utterance ‘encyclopaedias are (like) the classical authors’. In that figment of linguistic imagination, namely, zero context, it would have to be taken as a figurative comparison. But in the context of a bookseller who has difficulties in selling his stock of encyclopaedias, it is almost (if not completely) literal. Encyclopaedias are ‘expensive’, ‘respectable’, ‘sell slowly’.

The type of comparison, therefore, is context dependent. Thus, the type of transfer that is made is also context dependent. The relevance of this is that, in order to assign an interpretation to an utterance, we must be able to decide which information is important (in the context) and how it is transferred, literally or metaphorically. Therefore, the knowledge required for the interpretation of comparison statements not only comprises detailed knowledge (taxonomic, ontological, world knowledge, e.g. about attributes) of the concepts involved in the comparison statement and the relations of those concepts to other concepts, but also contextual knowledge.

The task of recognising comparison statements in the input relates to the decision of whether an utterance is literal or non-literal, but has some more basic aspects as well. Until a thorough corpus analysis proves that this notion has to be revised, the form of the input is seen as sufficient for the triggering of the recognition. Any concept relation of the ‘is-a’ form (‘all x are z ’, ‘ x is like z ’, ‘ x is the z of

y') is considered to be a comparison statement. But not all concepts are deemed to be of immediate interest in a comparison. Plainly attributive, identity or class inclusion relations such as 'Jake is lazy' or 'Neurosurgeons are surgeons' are not to be treated in this model. Yet they have to be filtered and, thus, be analysed first in order to be put aside (see Sect. 5.3.7). Of course, this means that even for the recognition of comparison statements, complex knowledge is needed, especially taxonomic knowledge to detect class inclusion, identity and similarity.

To return to the interpretation of comparison statements, a modified model of salience (im)balance [Ortony 1979] is put to use to distinguish between types of comparison statements. This aids the *how* side of the comparison, namely, how are attributes and features of the vehicle concept compared to the attributes and features of the tenor concept. It also fits in with the problem of the asymmetry of comparison statements, as discussed in Sect. 3.2.3, since pre-existing (structural) similarities are not a necessity, rather a transfer of information about the vehicle to the tenor concept is performed. This touches upon the *what* side of comparisons, to wit; which type of information is transferred. In order to decide which information from the vehicle is applicable to the tenor in the mode designated by the *how* side, a modified versions of Carbonell's invariance hierarchy [Carbonell 1982] and Veale's [Veale 1995] triangulation and squaring rules (c.f. Sect. 3.2.6 and Figs. 3.8, 3.9) are employed. Modification was necessary because Carbonell's theoretical model is satisfied upon finding the highest ranking match in the invariance hierarchy ("The key to the process is that comparisons along the higher-invariance entries in the hierarchy are preferred. Once a high-invariant property is found, no lower ones are considered" [Carbonell 1982]). This will be a successful strategy only if the KNB is known to be perfect, a demand often met by domain-dependent or small-scale system, but incompatible to the domain independence and robustness requirements of the LOLITA system. Conversion of Veale's rules proved necessary for reasons of efficiency (c.f. the discussion of Veale's model in Sect. 3.2.6). A detailed discussion of how the mechanisms mentioned work can be found in Sect. 5.3.7.

5.2.3 Knowledge structures

For the successful handling of figurative input on the basis of a knowledge-based approach, various knowledge structures are, of course, a further prerequisite. The important types of knowledge needed in our model and their manifestation in the

LOLITA system are discussed below.

Taxonomic and ontological knowledge

Clearly, one fundamental type of knowledge that is needed has to represent what exists and if and how the existing things relate to one another.

The systems view of what there is in the world is captured in ontological knowledge. How the existing concepts, objects, events can be structured in terms of sub- and super-type relations; class inclusion and similarity are expressed as taxonomic knowledge. In the KNB of the LOLITA system, there are various hierarchies representing taxonomic knowledge. All concepts within the SemNet are treated in a similar fashion as far as this knowledge structure is concerned, i.e. events and entities are both located in a hierarchy specifying their taxonomic status within the world knowledge. For example, (see also Fig. 2.2), the most generic concept for an event is 'event' and its representation in natural language is 'something happens'. Any event will be a sub-type of this generic event, as is the event of 'liking something'. The event of 'somebody liking coffee' in turn is a sub-type of the latter, and the event of 'Joe liking coffee' is a taxonomically subordinate to the event of 'somebody liking coffee'. Entities are arranged accordingly in a taxonomic order. Apart from the basic requirement of needing to know what does exist from the point of view of the system, this type of knowledge is needed for concept extension (see also Sect. 5.2.4), decisions on class inclusion and similarity as well as the computation of conceptual proximity (see Sect. 5.2.4).

World knowledge

Although it could be argued that the type of knowledge described in the last section forms part of 'world knowledge' we do not, in this context, mean the all-encompassing, diffuse type of world knowledge. By world knowledge, we mean how concepts relate and interact outside the taxonomic organisation of knowledge. This indicates that definitional knowledge, for example, forms part of world knowledge in this sense. By this we mean knowledge that, in a more general fashion, explicates the relations between concepts, the connection between concepts and events, and so forth. For example, the knowledge that spheres are round is to be classified as world knowledge. By consulting the taxonomic knowledge, we can derive that

a ball, being a proper sub-type of a sphere⁶, is round, too. Prototypes, a kind of knowledge present in SemNet, belong to the category of world knowledge, too. They restrict the scope of concepts that can connect to another concept (usually in the form of agency and event) to a certain class of concepts and thus are partially comparable to Wilks' pseudo-texts (see Sect. 3.2.2.2), which make a statement about characteristic concept relations (e.g. 'cars consuming fuel'). In the SemNet, the prototypes are not as specific as pseudo-texts, which also have characteristics of episodic knowledge. Instead, they contain knowledge about common concept relations, e.g. animate agents being capable of dying or fluids typically flowing. For some classes of tropes such as simple figures and fixed metonymies, this could be encoded into the figurative knowledge, i.e. we encode that containers hold liquid instead of deriving this information from episodic knowledge each time or failing if this derivation is not possible because there is no episodic knowledge to that effect available. But this would mean that coding figurative knowledge would be more complicated and less open and, even worse, it would mean that a cache of idiosyncratic knowledge that was inaccessible and not modifiable by the common methods were placed within the KNB. For the resolution of comparison statements, world knowledge is a necessity if a deeper interpretation is required, as becomes obvious, if one considers that it represents the non-taxonomic (i.e. 'interesting') links between concepts.

Episodic knowledge

Episodic knowledge is comparable to the non-definitional part of Wilks' pseudo-texts. Prototypes in LOLITA play a more abstract role by encoding information about prototypical conditions that have to hold in a given concept relation. Episodic knowledge, on the other hand, can be seen as an instantiation of this prototypical knowledge (definitional knowledge that has not been pre-coded into the system's KNB but was derived from input, either directly or with the use of the system's reasoning facilities) or as non-definitional knowledge representing any concept relation that has not been part of the KNB before. Input like 'Roberto owns a red motorbike', for example, conforms to the prototypical knowledge that humans are owners, but instantiates this knowledge and is non-definitional. It extends the SemNet with knowledge about 'motorbikes', 'ownership' and 'Roberto'

⁶Inheritance of properties within the SemNet is, of course, not quite as simple as the example suggests. But within this context, the depth of detail seemed appropriate.

alike. Definitional episodic knowledge can be derived from the input (e.g. from input like ‘Rick is a lecturer’, it can be derived that ‘Rick works at a university’. Depending on the amount of world knowledge available about lecturers and universities, the derivational process can go even further) or it can be explicitly present in the input (e.g. ‘the sun always sets in the west’⁷). Episodic knowledge exists in the SemNet in the form of previously processed input and the changes to the SemNet caused by it (by adding new concepts or modifying the relations between concepts on the basis of the input).

Contextual knowledge

The ability to choose context referents and their conceptual relations to the rest of SemNet and the input is needed for managing contextual metonymies and in order to account for context dependent shifts in focus. These shifts influence the conceptual proximity, for example, of concepts (see Sect. 5.2.4) and the appropriateness and selection of figurative interpretations. To illustrate this point, the sentence ‘she reads Shakespeare’ can mean different things depending on whether it occurs in the context of poetry, where ‘Shakespeare’ is likely to refer to verse or in the context of drama, where it presumably refers to one or more plays by ‘Shakespeare’. A prototypical means for selecting competing interpretations on the basis of contextual knowledge has been implemented.

Contextual knowledge also serves a more fundamental purpose, namely, helping to establish whether the input is to be taken literally or not. For example, the sentence ‘the King lost his head’ is likely to be meant literally in the context of a revolution or plot, whereas in the context of confusion, a non-violent crisis or illness, it is more likely to be an idiomatic expression denoting the loss of the King’s reasoning facilities to a certain extent.

Figurative knowledge

Clearly, the knowledge-based approach makes it a necessity to have access to figurative knowledge, i.e. knowledge about the form, type and organisation of (conventional) figures. After some consideration, we decided that the best method of

⁷Of course, factors such as quantification play a crucial role in this context, and a substantial amount of work is devoted to this aspect, but for the sake of the argument, an overly simplistic view is presented here.

representing figurative knowledge was to make use of the SemNet in the same way the KNB is used to represent all the system's knowledge. This means that figurative knowledge is simply represented in the SemNet in the form of events and concepts. Thus, it is not only possible to use existing NL interfaces to SemNet to modify, check, update and input figurative knowledge, but all the knowledge contained within the SemNet becomes available to the figurative sub-system, while figurative knowledge is available to all other parts of the system (see also Appendix B.1). Moreover, this shared representation facilitates the coding of figurative knowledge (easily extensible via NL input, using, for example, the query application module of the LOLITA system, c.f. Sect. 2.3) and ensures that no discrepancies arise between non-shared parts of information.

5.2.4 Methods

In order to enable the system to process figurative input, a set of methods supporting the general, knowledge-based algorithm have to be implemented. These methods are presented in the following.

Masking

Since some types of the knowledge required for the resolution of figurative input are not directly available, but nevertheless implicitly present in the SemNet, they have to be derived from ontological knowledge and/or the knowledge contained in events. This derivation is performed by a process called masking. Masking is collecting information from the SemNet by applying a mask to the SemNet structure. Attributes of concepts can, for example, be collected by examining events in which the concepts appear in the role of subjects, and the object slot is filled by an adjective.

By abstracting, attributes of classes of concepts can be derived. Examples of attributes of fruit contained in the SemNet might be that **apples** are **green**, **lemons** and **bananas** are **yellow**. One of the super-types of **apples** and **lemons** and **bananas** is **fruit**, one of the super-types of **green** and **yellow** is **colour**. By collecting lists (i.e. sets) of concepts and abstracting, it is possible to derive knowledge from the SemNet that is not explicitly present. Masking is not a fully developed, independently applicable method at the moment, but rather a localised

way of making good use of the rich knowledge in SemNet in areas where it is not explicit enough for the task at hand. The usability and success of this method nevertheless make it an approach that deserves further investigation.

Triangulation

The method of triangulation is used as a heuristic for ranking cross-domain mappings in the algorithm for comparison statements. It was derived from the triangulation rule presented by Veale [Veale 1995], but there are fundamental differences between Veale's triangulation and our method. In Veale's model, triangulation operates prior to the domain comparison and forms the basis for the interpretation. In our model, it operates during the domain comparison to reject unlikely mappings and, additionally, after the domains have been compared, acting as a filter, preferring cross-domain correspondences that satisfy the triangulation rule over those which do not.

During the phase of domain comparison, triangulation collects concept relations which are similar in both domains (according to the information contained in the SemNet). During the later stage of building the interpretation, it again operates on the cross-domain correspondences, favouring those that satisfy the triangulation rule in some way (see Fig. 5.4).

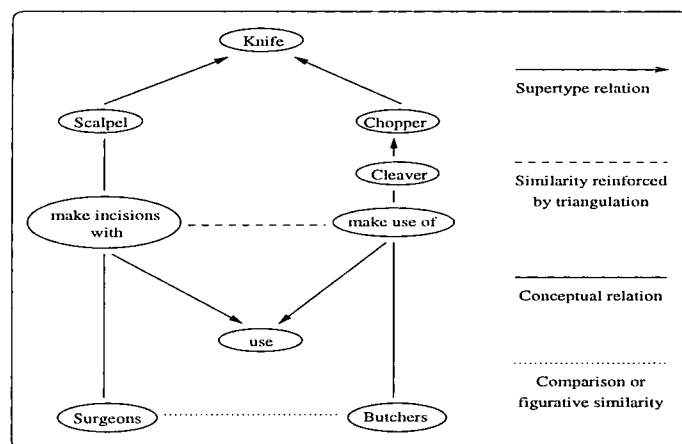


Fig. 5.4: Example of the triangulation rule

Concept extension

Concept extension (conceptually) extends the input in a certain way, depending on the type of trope under consideration. This means that concepts related to the

input concept(s) in varying degrees are considered to be of importance to the recognition and interpretation of the input. Like masking, this is a localised method. For example, simple tropes need little (if any) concept extension and the extension is limited to the ontological dimension. ‘Close’ super-types and information about the ‘family’ of the input concept(s) (c.f. Sect. 2.2.1) are considered to be important (e.g. ‘IBM’ is an organisation and a company, ‘the school’ is an organisation and an institution). Fixed metonymies make wider use of the ontology, typically needing an extension of the input towards all super-types as well as towards ‘similar’ of closely related concepts (and their super-types) and, if necessary, to instances of close super-types (see Fig. 5.5). Contextual metonymies require analysis of contextually related concepts rather than an investigation of ontologically related concepts. Idioms in their most simple form require no concept extension at all, but in order to account for ‘lexical flexibility’ and ‘semantic productivity’ [Gibbs 1994] (see Sect. 5.2.2), very closely related concepts (e.g. concepts connected to the input by a ‘similar’ or ‘instance’ arc) are considered during the processing of idioms as well.

Fig. 5.5 helps to visualise the process of concept extension, using an example (see also Appendix A.2) of the fixed metonymy variety. During processing of e.g., the sentence ‘she writes a thesis about Shakespeare’, the input concept ‘Shakespeare’ is extended to its immediate (ontological and taxonomic) super-types, in our example ‘versifier’, ‘playwright’ and ‘author’. Additionally, the taxonomy is traversed upwards from the extended concepts, thus adding ‘producer’, ‘mortal’ and ‘being’ to the extension. At that time, the input is further extended to concepts standing in a ‘similar’ relation to these recently collected ones and to immediate sub-types of them (‘artist’ and ‘adult’ in the case of ‘mortal’ as well as ‘scriptwriter’ and ‘co-author’ in the case of ‘author’). Non-circularity and termination of the algorithm are guaranteed by the fact that this kind of SemNet traversal (as far as the SemNet is concerned from the point of view of a taxonomic ordering) can be regarded as similar to going up a tree, starting at the leaves. In addition, no work is done twice, i.e. if one concept is included in the extension by following one particular path, this path will not be traversed again when approaching it from a different direction.

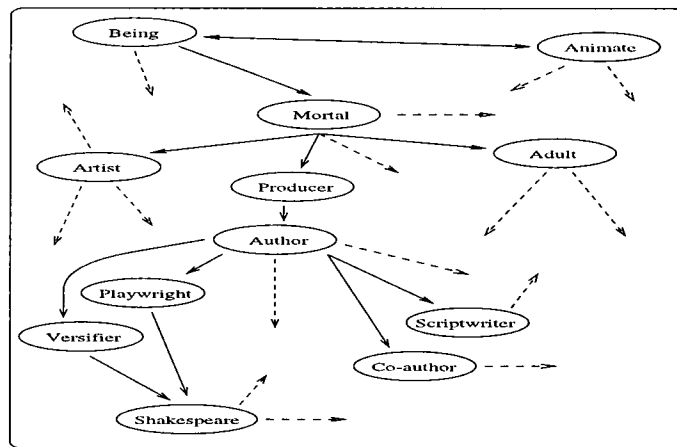


Fig. 5.5: Example of concept extension: fixed metonymy

Conceptual proximity

The computation of conceptual proximity implements the notion of salience (im-)balance (see Sect. 3.2.3) as proposed by Ortony [Ortony 1979]. The intersection of properties of concepts is compared to the number of non-shared properties in relation to the total number of properties the concepts have. It should be noted that conceptual proximity is a dynamic rather than static relation, due to the fact that it is computed on the basis of the existing SemNet representation. The well-known semantic distance measure (N.B. ‘semantic distance’ in the SemNet framework differs from the traditional notion of semantic distance. For a detailed discussion see Short) bases the proximity of concepts on the measure of how many links in a semantic network are traversed when going from a starting node to a target node. [Short 1996]. Moreover, the traversal and counting of traversed links is usually performed on a static taxonomy. Conceptual proximity, on the other hand, is based on a degree of conceptual closeness, i.e. the amount of shared properties. These properties are not restricted to attributes (e.g. being a container, having a colour) but computed on the basis of sets of properties. The more information that is taken into account, the better, but any amount will give sound results. This is due to properties of the SemNet representation formalism, namely distributivity and non-linearity. For a more detailed discussion of these, see Short. [Short *et al.* 1996].

Although the current model of conceptual proximity works completely satisfactorily, there is one direction in which it could be developed further: very abstract properties (such as ‘being an object’ or ‘being animate’) get the same weight as more concrete properties (such as ‘being skilled’ or ‘eating fish’). This will not

influence the computation of conceptual proximity if the ratio of abstract and concrete properties is the same for the concepts being compared (i.e. if there is detailed data on the concepts available or concepts on the similar levels of abstraction are compared). However, it does result in inaccuracies if lesser known concepts or concepts situated at different levels of abstractness are compared. Until properties get different weights according to their abstractness, this inaccuracy has to be caught (as is currently the case) by defences outside the conceptual proximity algorithm.

One of these defences currently at work checks the abstractness of the vehicle that is compared to the tenor *before* conceptual proximity is computed. An example of this mechanism can be seen in the debugging output below (Fig. 5.6). For details

```
Jake is something
-----
Topic more detailed/better known than vehicle;
Vehicle either abstract or (remote) super-type of topic;
weak relation.
-----
```

Fig. 5.6: Conceptual proximity debugging output

and examples of the conceptual proximity/salience (im)balance mechanism used in the processing of comparison statements, see Sects. 5.3.7 and Appendix B.3.

5.3 Processable figures

This section discusses examples of the processing of the types of figurative expression presented in Sect. 5.2.2. Before giving details, the general algorithm is outlined.

5.3.1 General algorithm

The algorithm for knowledge-based figures is distinct from the algorithm for comparison statements. The algorithm for handling knowledge-based figures is outlined below, followed by the algorithm for comparison statements.

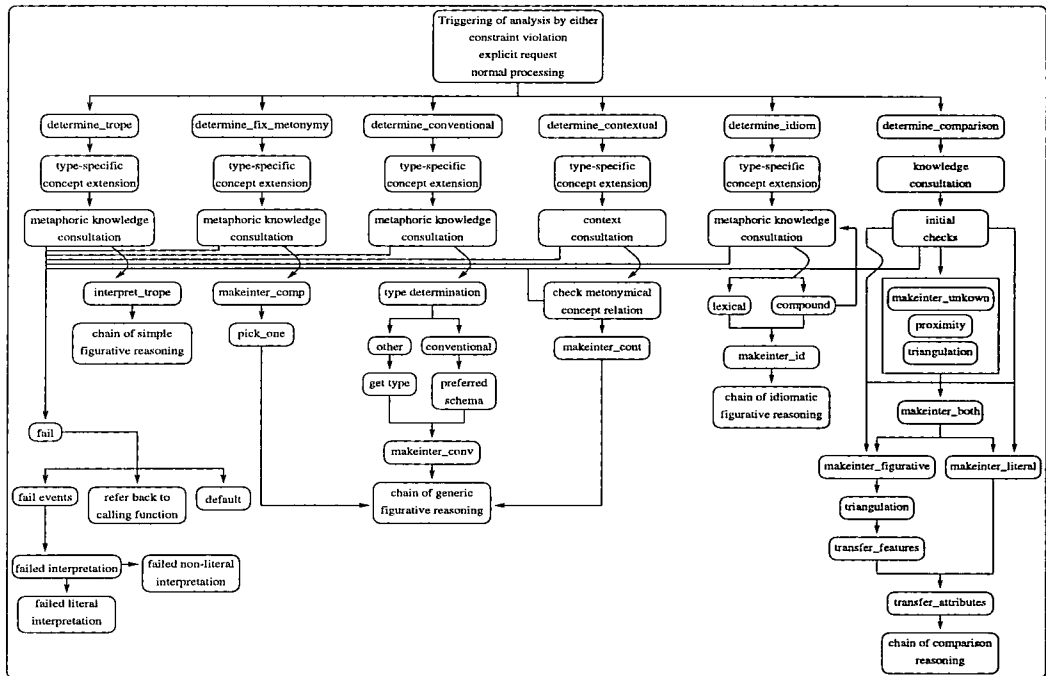


Fig. 5.7: Overview of the algorithmic analysis of figurative language

Knowledge-based figures

The resolution algorithm consists of three main stages, namely, type-dependent concept extension, metaphoric knowledge consultation and initial interpretation. In addition, figurative reasoning documents the system's behaviour and decisions during processing (see also Sect. 5.2.1). A graphic representation of the main functional units of the algorithm is given in Fig. 5.7, showing the stages for the six types of figures (see Sect. 5.2.2) handled in this framework.

The non-literal analysis is triggered either by the detection of a constraint violation or by an explicit request to the figurative resolution module. The problems of relying on a constraint violation are discussed in Sect. 3.2.2, and it has to be noted that, since they severely limit the scope of figurative analysis, constraint violation detection has only been added as a bonus. During normal processing, a literal and figurative interpretation can be sought concurrently and, taking the limitations of the actual, non-parallel implementation into account, the order in which an analysis is performed is irrelevant. If, however, no literal interpretation is found, the 'not believe literally' event is added to the chain of figurative reasoning (c.f. Fig. 5.12 and Appendix B.2.7).

At the first stage, according to the type of trope, the input is extended along conceptual paths which are determined by the type. Concept extension is a lo-

calised way of cutting down on the processing cost associated with brute force methods of handling information from semantic networks (see Sect. 5.2.4). For fixed metonymies (see Sect. 5.3.3 and Fig. 5.5) for example, concept extension concentrates on ontological and meronymic relations, whereas contextual metonymies (c.f. Sect. 5.3.4) require emphasis to be put on contextual information.

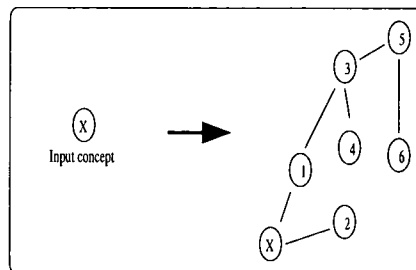


Fig. 5.8: First stage of knowledge-based algorithm: concept extension

The extended concepts are compared to figurative knowledge stored in the Sem-Net (see Fig. 5.9). This approach (data-driven) was preferred over the alternative, an extension of the concepts involved in the figurative knowledge (expectation-driven), as it requires less processing while at the same time it is more successful in finding relevant matches and provides an easier means of ranking matches hierarchically (from specific to general, e.g. PLAYWRIGHT FOR DRAMA, ARTIST FOR ART-FORM, PRODUCER FOR PRODUCT).

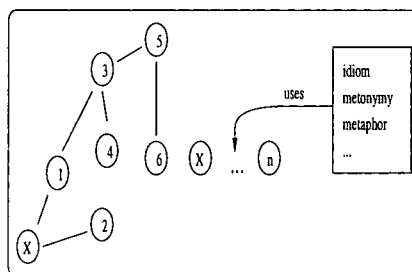


Fig. 5.9: Second stage of knowledge-based algorithm: knowledge consultation

If a match between the (extended) input and stored metaphorical knowledge is found, the recognition stage is successfully completed. It should be noted, however, that conventional figures often require the presence of two domains or concepts (c.f. Carbonell's "X is used to mean Y in context Z" [Carbonell 1982]). To give an example, in the input 'I killed emacs', an instance of the MACHINES ARE PEOPLE metaphor or, more precisely the 'cessation of function is killing' sub-case of the metaphor [Lakoff *et al.* 1995], only one concept appearing in the figurative schema is explicitly present in the input, namely, 'kill'. The results of concept extension

and figurative knowledge lookup for conventional figures are, therefore, filtered through a function preferring ‘double matches’ with metaphoric schemata over ‘single matches’. In this case, ‘emacs’ will, for example, be extended to ‘software’ and ‘process’. Therefore the schema will receive two hits, one from ‘kill’ and one from ‘process’, and thus satisfies the ‘preferred schema’ constraints. However, the ‘preferred schema’ rule is not brittle, but merely a preference. This approach was taken for reasons of coverage and robustness. Consider the example ‘IBM ended the fight with Apple’, where only ‘fight’ is present in the input. If, for one reason or other, the information that IBM and Apple are competitors cannot be retrieved by concept extension and, subsequently, the schema COMPETITION AS PHYSICAL AGGRESSION receives only one hit (from the extension of ‘fight’ to physical aggression), it will in the absence of a preferred schema still be considered as a valid basis for a metaphorical interpretation of the input.

If an acceptable figurative interpretation was found, the ‘is-a trope type’ and ‘believe is-a trope type’ events (c.f. Fig. 5.12) are built in the wake of the figurative reasoning.

Should the interpretation of the input be required, the ‘other’ side of the figurative knowledge (schema) is looked up (see Fig. 5.10). It serves as the initial interpretation of the input. For example, in a AUTHOR FOR WORK metonymy, ‘work’ will be considered to be the initial interpretation of ‘author’ in the context of the metonymy. Depending on the type of trope and the input, this is either the full interpretation or only the starting point for an extended interpretation.

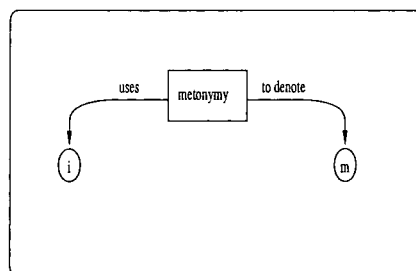


Fig. 5.10: Third stage of knowledge-based algorithm: initial interpretation

The extended interpretation makes use of either the SemNet or the comparison statement engine to elaborate the interpretation. To give an example, in a case where the input ‘I read a lot of Shakespeare’ matches against the AUTHOR FOR WORK schema, the extended interpretation would try to instantiate the initial interpretation (‘work’) in accordance with the input. If knowledge to the effect that Shakespeare was the author of plays and poetry were present in the KNB,

‘work’ could be instantiated to ‘drama’ and ‘poems’, respectively. Currently, an extended interpretation can only be achieved by either keeping detailed figurative knowledge in the SemNet or by working in a well-modelled domain, which, on the other hand, reduces processing time. The fact that the resolution algorithm solely depends on the available knowledge (and its structure) might make it vulnerable (e.g. in domains of low granularity in the KNB), but is deemed to be of great benefit in that it keeps the figurative sub-system in synchronisation with the rest of the system.

The comparison engine is responsible for instantiating conventional figure interpretations. Whereas it is comparatively easy to recognise the input ‘our relationship hit a dead-end street’ as an instance of the LOVE AS JOURNEY metaphor, the recognition alone will not provide enough information on the state of the relationship in question. Therefore, the comparison engine has to provide the information on how concepts relating to journeys can be used to (metaphorically) describe relationships. As the ‘love-as-journey’ example presented in Appendix A.6 shows, it is capable of doing so. The integration of the results of the comparison engine with processing of other types has, however, not been finished yet. This is due to the limited scope of this project and its initial goals, which were namely investigate the feasibility and implement the functionality of a broad range of figurative processing in a robust, real-life, albeit prototypical fashion.

Finally, on the basis of the initial interpretation, the interpretation event or events (if there is more than one possible interpretation for the input) are built and stored in the SemNet.

Comparison statements

The algorithm for comparison statements differs from the algorithm for other types, since knowledge about conventional figures cannot (at least not easily) be utilised for the detection and analysis of comparison statements. For a graphic representation of the major steps in the algorithm for comparison statements see Fig. 5.11.

First, some fundamental properties of the input are checked. If the input does not satisfy the form of a comparison (i.e. if there is no generic is-a relation between the concepts, which can take different surface forms such as ‘X is like Z’, ‘all X are Z’ or ‘my X is a Z in that respect’), it is passed back to the calling function. Likewise, consultation of the SemNet shows whether there is enough information available

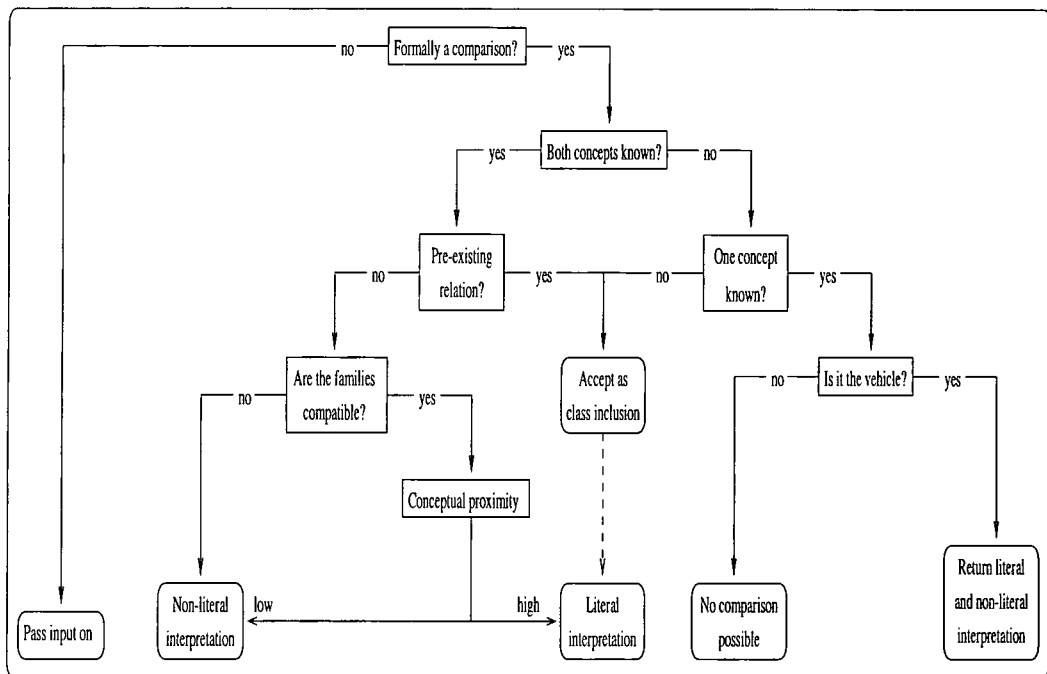


Fig. 5.11: Decision tree for the handling of comparison statements

on the input concepts. If not, processing cannot be guaranteed to be successful. If enough information is available on the vehicle but not the tenor, both a figurative and a literal interpretation are sought, as it is impossible to decide on the relation between the two concepts. The criteria of robustness and integration dictate that the algorithm cannot possibly stop and print an error message. This might be acceptable in a debugging configuration or in an application fully dedicated to non-literal processing, but since the non-literal analysis sub-system is providing services to various other robust parts of the LOLITA system in the present configuration, it must not corrupt the performance and robustness of those other parts.

Clearly, no comparison is possible if the unknown concept is the vehicle. In this case, the input is passed back. Currently, the prototype does not record a failure by building a new event in the SemNet, stating that the system was incapable of analysing the input and giving the reason for this failure, as happens in the course of processing other types of tropes (c.f. Fig. 5.7 and Appendix B.2.7). This is due to the fact that, in our opinion, it is easier to verify input as being a figurative comparison reliably, but it is not possible to reliably say that a given input is neither a figurative nor a literal comparison and thus erroneous, e.g. in the case of comparison statements the *tertium non datur* does not hold. Therefore, underdetermined input is passed back and treated using the system's default mechanisms of acceptance or rejection.

This means in the case of two formerly unknown concepts appearing in the input that the usual mechanisms of, e.g., determining the family value of the concepts (by means of prototypical information and pragmatic rules) is applied, and the input is taken as bona fide literal comparison or class inclusion statement (e.g. ‘computers are machines’).

If the initial checks were passed, if there is information on the concepts being compared available in the SemNet, and if the input satisfies the form of a comparison statement, further properties of the concepts are analysed. If a relation of identity, weak identity (i.e. similarity or conceptual closeness), class inclusion or sub-/super-type already exists between the input concepts, the input is positively taken to be a literal comparison or class inclusion statement and handled as such.

If the concepts do not stand in a previously established relation, their families (i.e. ontological information on them stored in the SemNet) are analysed. Should the concepts involved not belong to the same or similar families (e.g. ‘human’ and ‘inanimate manmade’), it is to be expected that there is a salience imbalance between the concepts, thus conceptual proximity is not computed in order to save resources. The input is then treated as a figurative comparison.

It should be noted that the measurement of family incompatibility is not based on binary, brittle rules. Within the pragmatic analysis of input in LOLITA, there exists a function for the ranking of family distance and thus compatibility, e.g. ‘human’ and ‘animal’ are closer than ‘human’ and ‘cognition’. For reasons of integration (compatible families should be compatible in all sub-systems of LOLITA) and ease of development (re-use of code already written and tested), the comparison algorithm makes use of this function too, thus resulting in a flexible treatment of conceptual distance (in this case based on the taxonomy given by the knowledge engineer).

If both concepts belong to the same or compatible families, conceptual proximity is computed (see Sect. 5.2.4) and a decision is made on this basis as to whether the comparison is literal or figurative. The conceptual proximity measure then determines whether the input is interpreted as literal comparison or as figurative comparison, or whether it is necessary to make both interpretations.

The latter case arises, as mentioned before, if the tenor concept is not known and if the computation of conceptual proximity fails to clarify the relation between the input concepts. This happens only with a very small number of cases, namely,

cases where there is a sufficient and equivalent amount of information on both concepts available but conceptual proximity still fails to safely attribute a type of relation to the concepts. This can be caused by either one of the concepts having a large number of generic, abstract information attached to it (such as high-level ontological information) and little detailed information, by one concept being defined mainly in terms of events (i.e. a concept not already present in the SemNet occurred frequently in previous input without ontological or taxonomic information), by the concepts inhabiting more or less discrete areas of the SemNet, or by faulty data. Except in the case of faulty data, all those configurations could be resolved by additional methods, but since testing showed that this does not occur with a worrying frequency, it was not considered to be essential. Moreover, and more importantly, it is not so much of a problem or shortcoming, since it does not mean that the analysis is impoverished or, indeed, that the system's performance, robustness and usability suffers. On the contrary, since no costly (and presumably sometimes even futile) attempt of deciding in situ on something hard to decide is made, the overall processing is kept at real-time speed. Furthermore, since both interpretations and their justification will be stored in the SemNet, they can be re-evaluated and revised as soon as additional information is available.

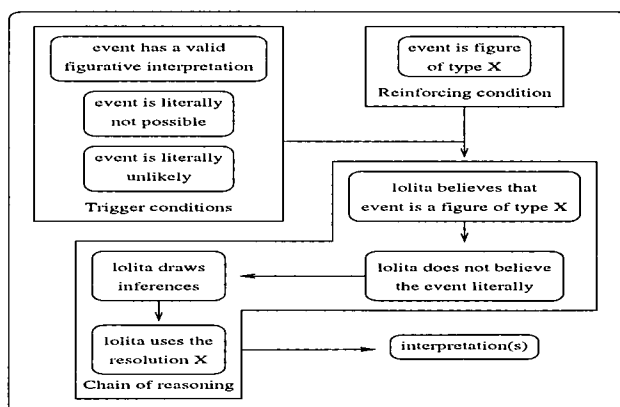


Fig. 5.12: Chain of reasoning in the analysis of figurative language

In the following sections, individual examples of the processing and results for the types of figures presented in Sect. 5.2.2 are given. The NL input that was entered in a query session (see Sect. 2.3), is presented together with the interpretation in the form of NL as generated by the generation module from the portion of SemNet that was constructed in the course of processing the input and other relevant events built during the analysis of the input (for reasons of simplicity again in the form of NL output). Since the generator module is very powerful

and there are many ways of achieving varied NL (for a thorough discussion, see Smith [Smith 1995]), the shortest, most simple and, where appropriate, long NL generation (NLG) of events is given.

Wherever possible, the input was taken from the literature on existing systems for the computational treatment of figurative language and the relevant source for each example is cited. For further examples, see Appendix A.1–A.6.

5.3.2 Simple tropes

The simple tropes are derived from conventional figures such as PEOPLE ARE MACHINES ('he broke down'), MACHINES ARE PEOPLE ('the car is thirsty') and particularly generic schemata such as PART FOR WHOLE. Their ubiquity makes it necessary to process them quickly, and their conventionality makes it possible to arrive at a satisfying depth of interpretation with limited effort (see Sect. 5.2.2). In the current implementation, a constraint violation triggers the processing, although the recognition and interpretation algorithm can also operate independently of the signalling of a constraint violation. Seen in isolation, this configuration can be regarded as similar to Wilks' preference semantics and pseudo-texts [Wilks 1975b, Wilks 1978] and Fass' met* method [Fass 1991]. It should of course be kept in mind that the resolution system as a whole is capable of avoiding shortcomings of the above models (for a discussion, see Sect. 3.2.2), but the choice of using a constraint violation as a trigger (limited to the treatment of simple tropes) was made so that our solution can be compared to the models mentioned.

Concept extension in the case of simple tropes is very limited. Either the information necessary for processing is available directly from the input concept (in the form of 'controls', such as the family type of the concept), which means that no SemNet lookup is needed, or limited extension to, e.g., the super-type(s) is sufficient.

There might, of course, be cases where this fast and shallow mechanism will fail to correctly detect input as figurative. It should be kept in mind, however, that the simple tropes and their resolution algorithm are based on computational needs; a 'shallow' NLP task operating under time constraints requires fast results. Thus this lowest level of resolution will be sufficient and exhibits the best cost-benefit correlation. If no literal interpretation for the input is found and the simple resolution yields no positive results either, the next higher level (involving more

processing) of the resolution algorithm can be consulted. It is important to note that ‘higher’ tiers in the algorithm are capable of analysing the ‘lower’ types, e.g. the fixed metonymy algorithm handles simple tropes and the conventional figure algorithm can resolve simple tropes, fixed metonymies and idioms. Due to their very nature, only comparison statements are not integrated into this hierarchy. One reason why there are discrete solutions is to enable the system to use the method best suited for the task at hand, e.g. to handle figurative input with the highest possible success rate at the lowest possible (processing) cost. Others include the wish to have a workbench for experimenting with different methods of resolution within an NLP system and the possibility of modelling type change (i.e. a frequently used ‘fixed metonymy’ might change its type to ‘idiom’) as well as the software engineering demand for transparent, modular solutions instead of complex, monolithic black boxes.

The way in which NLP tasks and the different levels of the resolution algorithm should interact is, of course, a question of integrating the current solution. The answer should, in our opinion, be based on a corpus analysis. One of the two models used the different levels as a sort of filter. An attempt was made to process input at the lowest level, i.e. the entry point for resolving figurative input was the simple trope part of the figurative resolution algorithm. If processing was not successful, the next higher level was invoked. Thus, the total cost of processing in the case of a failure was higher than would have been the case if the conventional figure resolution had been the starting point. On the other hand, if the input could be handled adequately on a lower level, the total processing cost is lower than would have otherwise been the case. The current configuration does not automatically pass unresolvable input on to the next higher level, but lets the user (or other parts of the system) select resolution methods individually. Only if more than one compatible resolution method is activated (e.g. ‘conventional metaphor’ and ‘simple tropes’) is the built in default to first apply the lower resolution method and then to pass the input on if this fails.

After the stage of concept extension, metaphoric knowledge is consulted to see if the extended input relates to a simple trope. If a match is found, the input is abstracted in accordance with the trope under consideration. For reasons of simplicity and speed, the modified input is not passed back for re-evaluation each time an abstraction is made. Instead, all possible abstractions are made at once, and the appropriate ones collected. The calling function can thus select the best

interpretation (although the built-in default selects the least generic interpretation).

This procedure is comparable to Wilks' preference semantics (see Sect. 3.2.2) in that it uses prototypical information present in the SemNet (c.f. Sect. 5.2.3) to detect at what time the resolved input fits the prototype and to Fass' abstraction hierarchy (Sect. 3.2.2) in that it works by abstracting from the input. While the analogy to Wilks' approach [Wilks 1975b] extends to the prototypes from the SemNet (stating preferences for, e.g., case role fillers), the abstraction is, unlike Fass' abstraction hierarchy [Fass 1991], the abstraction is guided by the metaphoric knowledge, not by the topology of the abstraction hierarchy.

Finally, it should be mentioned that simple tropes mainly deal with 'actions', e.g. figurative use of verbs. Although it is hard to speak of the figurative use of a part of speech without, at the same time, taking up a substitutionist stance (the figurative portion of the utterance is to be replaced by a literal paraphrase), the figurative use of other or more than one part of speech seems, in our experience, to belong to other classes of figures.

The example of a simple trope to be presented here originally comes from Wilks' work on preference semantics [Wilks 1975b] and is cited again by Fass [Fass 1991]. The input to the system was

`my car drinks gasoline.`

In the course of normal processing within the query application, this caused pragmatics to signal an incompatibility between the agent ('car'), an 'inanimate man-made entity', and the prototypical knowledge about the action ('drinking'), which is to be performed by 'animate' agents (see Sect. 5.2.3).

Processing was started at the lowest level of figurative resolution and in the first instance, simple trope concept extension was limited to looking up information directly associated with the concept of 'car'. This produced taxonomic information to the effect that 'car' is a 'machine', 'inanimate' and 'manmade'. In the next step, the metaphoric knowledge contained in the SemNet was consulted to see if there was a schema suitable for the input, namely, some figurative knowledge meeting the requirements of a correlation between 'inanimate', 'manmade' 'machines' and 'animate agents'. This, of course, is the case with personifications and, more precisely, the MACHINES AS PEOPLE metaphor [Lakoff *et al.* 1995].

It should be obvious that no incompatibility of prototypical and actual information (comparable to preference breaking or constraint violation in other models)

would be necessary in order to arrive at a figurative reading of the said input. Due to the fact that a figurative resolution can explicitly be requested and that concept extension will operate on the all input concepts, the same figurative knowledge will yield the same results, regardless of whether or not an incompatibility arises. Nevertheless, it lies in the very nature of the simple tropes and the corresponding algorithm that they are resolved quickly. NLP tasks such as fast parsing do not need a full interpretation of the input, but rather the capability of accepting input that would, according to brittle, ‘literal’ semantics, lie outside the scope of the system’s understanding facilities. Hence it is permissible to let the processing of simple tropes be triggered by a prototype incompatibility. Even more so, as the figures subsumed under the heading of simple tropes can be processed at other levels of the resolution algorithm, such as fixed metonymies and conventional figures, where prototype incompatibility does not feature as a triggering condition.

Since in this case no literal interpretation was found (regardless of whether the triggering condition was a prototype incompatibility), the ‘literally not possible’ event is built, stating that according to present knowledge, the input has no literal interpretation (see also Fig. 5.12). The following is the NLG of the ‘literally not possible’ event.

Short NLG: A car drinking gasoline is literally not possible.

Long NLG: The car that inputs describe and that you control drinking gasoline that it consumes is literally not possible so I believe that it drinks it is a personification. I draw inferences by using a simple resolution because I do not believe literally that it drinks it.

In the long NLG, the causal relationship between the events (giving the figurative reasoning its name) can be seen. Since the information contained in the input is, according to LOLITA’s knowledge, not literally possible and there exists figurative knowledge about the concepts involved, implying that the input is a figure of the type personification, the fact that the input is of the this type is recorded in the ‘is tropetype’ event.

Short NLG: A car drinking gasoline is a personification.

Long NLG: The car that inputs describe and that you control drinking gasoline that it consumes is a personification. I draw inferences by using a simple resolution because I do not believe literally that it drinks it. Since this ‘knowledge’ has to be tied to a source in order to fit into the SemNet structure and since it cannot be taken as definitional, the ‘believe is a tropetype’ event is generated. It states that the system came to the conclusion that the input

is a figure of a particular type on the basis of the triggering conditions (see Fig. 5.12). This separation is necessary for reasons of potential revision and coherence in the SemNet; as there is a difference in stating that condition *a* holds or that condition *a* is believed to hold on the basis of other conditions and rules by which it can be deduced.

Short NLG: I believe that a car drinking gasoline is a personification.

Long NLG: I believe that the car that inputs describe and that you control drinking gasoline that it consumes is a personification because it is literally not possible. It consumes it because I draw inferences by using a simple resolution. I draw them by using a simple resolution because I do not believe literally that it drinks it.

The next event built in the course of resolving a simple trope reflects the fact that the system attempts an interpretation (for want of a better NL representation, this is mirrored in the phrase ‘to draw inferences’). It already appears in the long NLG of the ‘believe is a trope type’ event shown above. The type of reasoning used in this process is given as well, namely, ‘simple resolution’, the algorithm for simple tropes. This information is important if more than one method or level of resolution is used for any given input.

Since the matter conveyed by the input does not conform to the system’s knowledge of the world (in other words since it is not literally possible from the point of view of the semantics, pragmatics and knowledge of LOLITA), the system cannot attribute a high degree of belief to it. This is reflected in the construction of the ‘not believe literally’ event.

Short NLG: I do not believe literally that a car drinks gasoline.

Long NLG: I do not believe literally that the car that inputs describe and that you control drinks gasoline that it consumes because I believe that it is a personification. It consumes it because I draw inferences by using a simple resolution. I draw them by using a simple resolution.

The act of interpreting the input in the light of the applicable figurative knowledge is presented as the drawing of figurative inferences (see above) or ‘draw inferences’ event. Again, the causal relationship between the events is given as well and can be analysed through the long NLG.

Short NLG: I draw inferences by using a simple resolution.

Long NLG: I draw inferences by using a simple resolution because I do not believe literally that the car that inputs describe drinks gasoline that it consumes. It consumes it because I draw them by using a simple

resolution. You control it.

In the long NLG of the ‘draw inferences’ event, the interpretation of the input can already be seen. One possible reading of the long NLG would in this example suggest that ‘the car consumes [gasoline] because [the system draws inferences]’. Of course this ambiguity is due to the fact that the NLG, especially the long form, had to be kept as simple as possible; thus a construction like ‘I believe that the car consumes gasoline and does not drink it, as is said in the input because by reasoning about the input and consulting figurative knowledge and coming to the conclusion that the input is a personification I have no option but to prefer the first reading’ was out of the question. What is meant by the ‘because I draw [inferences]’ is that ‘the input means X because I have reason to believe so’. In addition, the reason for this is also given, namely the type of resolution (in this example ‘using a simple resolution’).

Finally, the interpretation event itself is built. As mentioned in the introduction to the simple tropes, an abstraction from the input (guided by the trope type) is performed and, if no other function or module makes use of the collection of all interpretations, the least generic interpretation is used in the interpretation event. Short NLG: A car consumes gasoline.

Long NLG: The car that inputs describe and that you control consumes gasoline that it drinks because I draw inferences by using a simple resolution. It consumes it because I draw them by using a simple resolution. In the long NLG, the ‘describe as’ event is realised as well. This event belongs to the simple figurative reasoning (see also Appendix B.2.7) and expresses the fact that by using the figure the input concept is described in terms of some other concept, in this example (a personification) in terms of ‘humans’.

Long NLG: Inputs use human features in order to describe the car that you control and that drinks gasoline. It consumes it because I draw inferences by using a simple resolution.

5.3.3 Fixed metonymies

In the case of fixed metonymies, the task of recognition relies on finding a match with stored metaphorical knowledge for the input. Although a constraint violation can be an indication of a figurative meaning, it is neither a sufficient nor a necessary condition for the presence of a figurative expression of the type fixed metonymy.

This becomes obvious if one considers the sentence ‘she is writing a thesis about Shakespeare’, for example, where ‘about Shakespeare’ might refer to the person ‘Shakespeare’ or, in accordance with the well-established figure of either PART FOR WHOLE, PRODUCER FOR PRODUCT or its instance AUTHOR FOR WORK, to the whole or one particular work by ‘Shakespeare’.

As described in Sect. 5.2.4 (see also Fig. 5.5), concept extension for fixed metonymies is more elaborate than for simple tropes. Yet the very nature of fixed metonymies allows one to concentrate on ontologically and taxonomically related concepts and to additionally consider concepts similar to (e.g. synonyms) or standing in a super-type relation to the ones collected in the first step of ontological and taxonomic extension. The input to be presented as an example for the handling of the type fixed metonymy was taken from examples given in Fass’ article on the met* method [Fass 1991]. The system’s input was

Denise drank the bottle.

Concept extension leads from the input ‘bottle’ to the concept ‘container’, which is taxonomically and ontologically closely related to ‘bottle’. Since the system does not find a literal interpretation it can assign a sufficient degree of belief to, the ‘literally not possible event’ is built to record this fact.

Short NLG: Denise drinking a bottle is literally not possible.

Figurative knowledge lookup, on the other hand, results in a match with the CONTAINER FOR CONTENT metonymy. A CONTAINER FOR CONTENT metonymy expresses the content (of a container) through the container itself; this conceptual match between the extended input and figurative knowledge (c.f. Fig. 5.9) is reflected in the long NLG of the interpretation event shown below. Accordingly, the ‘is tropetype’, ‘believe is a tropetype’ and, subsequently, the ‘not believe literally’ events are generated and added to the SemNet:

Short NLG: Denise drinking a bottle is a metonymy.

Short NLG: I believe that Denise drinking a bottle is a metonymy.

Long NLG: I believe that Denise drinking a bottle is a metonymy because she drank it is literally not possible.

Short NLG: I do not believe literally that Denise drank a bottle.

Long NLG: I do not believe literally that Denise drank a bottle because I believe that she drank it is a metonymy.

The interpretation of the input (in the light of the assumption that it is a metonymy) is then made by consulting the ‘other’ side of the figurative schema CONTAINER FOR CONTENT metonymy (c.f. Fig. 5.10). As mentioned above, a CONTAINER FOR

CONTENT metonymy uses a container to denote the content (of this container). Unlike with simple tropes, the exact figurative schema that leads to the interpretation is recorded as well (as can be seen in the long NLG).

Short NLG: Denise drank a content.

Long NLG: Denise drank the content that the container-for-content metonymy that uses a container may describe because I draw inferences .

Of course it would be beneficial if one could determine the exact relationship between the input concept and the concept denoted by the figure ('bottle' and 'content of the bottle' in the example above). Experiments have shown that it is not possible to arrive at a satisfying linking of these concepts. The possible relations vary too much in their nature. Consider the following metonymies: 'the university told him to apply' for 'the post-graduate office told him to apply', 'the Shakespeare we read last term' for 'the play by Shakespeare we read last term' and the 'bottle' example just cited. One solution would have meant that each schema (such as CONTAINER FOR CONTENT) would include the exact nature of the relation between the concept appearing in the surface form of the expression and the denoted concept (e.g. 'the content of the particular container'), because even a generic relation for only one type of metonymy (e.g. PART FOR WHOLE) is not specific enough (c.f. the relation between 'post-graduate office' and 'university' and 'content' and 'bottle'). Moreover, this would, if at all feasible, contravene the very idea of having a hierarchy of generic and specific instances of a schema and a match on the basis of concept extension. Instead, a generic relation is assumed which, because of its approximate nature, is not realised through NLG. Instead, the relation can be retrieved by the 'name' of the figure, 'container-for-content' in the above example, and by consulting the SemNet to find out in which relation containers and contents stand. Thus, the representation of figurative knowledge is kept simple, and no asynchronicity can arise between the portion of the KNB holding the figurative knowledge (in particular the relation between the input and the denoted concept) and the rest of the KNB. If, however, no episodic or world knowledge on the relation of the two concepts is available, the generic relation grounded in the figure is still available as a last resort.

Should there be more than one interpretation, while there is no contextual information that can be used towards disambiguation (c.f. Sect. 5.2.3) all plausible interpretations (e.g. literal and figurative) have to be considered. The figurative sub-system does not rank interpretations; in the whole LOLITA system, accept-

ability of the input or, to be precise, the information conveyed by or meaning behind the input depends on the degree of belief the system can attribute to the input. Thus, in a case where there is, for example, a valid literal and figurative interpretation for the input, the analysis has to consider a higher level of information such as the context or the discourse setting to come to a conclusion about which interpretation to prefer. This task clearly lies outside the scope of this work. Competing figurative interpretations, however, were of interest in the framework of this project, and cases where there is more than one figurative interpretation were examined. A prototypical means of ranking them has been implemented, taking the context of the input into account (c.f. the ‘read Shakespeare’ example in Appendix A.2).

5.3.4 Contextual metonymies

With contextual metonymies, constraint violation can be no more than an indication that there is a portion of figurative input. Only by going through the recognition stage of the algorithm, can anomalous input be distinguished from a contextual metonymy. In other words, if the input violates constraints and the recognition algorithm fails to establish that the input can be classified as a contextual metonymy, we have to assume that the input is anomalous.

Contextual metonymies are related to anaphora and fixed metonymies, but require additional knowledge in the form of valid metonymical relations and detailed knowledge about the ontological and taxonomic status of possible context referents as well as knowledge about concepts which are related to these. Thus the concept extension for contextual metonymies differs from the concept extension for fixed metonymies and simple tropes.

Concept extension is performed mainly along the lines of context referents. A portion of context (i.e. previous input) is checked for occurrences of possible referents of the input concept(s). These referents are not restricted to the same surface form; they can be either the same concept, sub-types of the same concept, super-types of the input concept, synonyms, instances or generalisations of the input, or concepts which are similar to the input. Since previously analysed (new) information is stored in the SemNet in the course of normal processing, even episodic relations between the input concept and context referents can be easily retrieved.

The collection of concepts obtained by this context-based concept extension

is then checked with regard to whether one or more of its members stands in a plausible metonymical relation to the input concept. It should be noted that, in the prototype, this check is performed in situ in the function responsible for recognising contextual metonymies. It currently handles a set of generic metonymical relations such as CONTROLLER FOR CONTROLLED, PART FOR WHOLE or USER FOR USED. Eventually, this functionality should be integrated with the other resolution modules in order to add contextual resolution capabilities to all other types of non-literal analysis. Owing to the scope of this work as well as the main thrust, namely, to provide an analysis for a broad range of figurative expressions rather than in-depth treatment of a few selected examples (see Sect. 4.2), it was considered to be more important to show whether analysis of figurative expressions of the type contextual metonymy (besides others) is possible and how it can be achieved.

If the concept extension and check against metonymical relations is successful, the concepts satisfying the constraints of standing in a contextual as well as a metonymical relationship to the input are passed on to the interpretation function, which will then construct the relevant events, i.e. mirror LOLITA's beliefs about the meaning of the input and its relation to the pre-existing knowledge stored in the SemNet.

The input for the following, slightly overused example of a contextual metonymy was taken from Lakoff and Johnson [Lakoff and Johnson 1980], because of its widespread use in the literature.

A customer entered a restaurant and ordered a sandwich. Now the sandwich is waiting for the bill.

Without the figurative sub-system, input that does not fit in with the system's view of the world is attributed a low degree of belief, but not rejected in a binary fashion. If additional information makes previous input acceptable, a high degree of believe is attributed to it. If a literal interpretation is impossible and the figurative resolution is activated, a further distinction is made between literally acceptable input and input which is not acceptable when understood literally but possibly acceptable when understood figuratively. In our example, since no literal interpretation is found, the input is marked as being 'literally not possible' by building the corresponding event.

Short NLG: A sandwich waiting for a bill is literally not possible.

And the systems beliefs are mirrored in the 'not believe literally' event.

Short NLG: I do not believe literally that a sandwich waits for a bill.

The first step in resolving this contextual metonymy consists in extending the input concepts, similar to the concept extension used for fixed metonymies. The subsequent step is unique to the algorithm for contextual metonymies; the collection of context referents for the initial concept(s) as well as the extended concepts. In our example, the ontological extension of ‘sandwich’ (and ‘bill’) does not lead to any contextual matches as such. Context lookup, on the other hand, shows that the concept ‘customer’ is related to the input concept ‘sandwich’ by means of episodic knowledge, making the ‘customer’ the owner/user of the ‘sandwich’. Thus, later metonymical relation lookup succeeds for the ‘customer’ ‘sandwich’ concept pair. On the basis of the assumption that ‘customer’ and ‘sandwich’ stand in a metonymical relation, the input satisfies the requirements for being classified as metonymy. Consequently, the ‘believe is a trope’ event is added to the SemNet:

Short NLG: I believe that a sandwich waiting for a bill is a metonymy.

Since it has been established that there holds a valid metonymical relation between ‘customer’ and ‘sandwich’, the interpretation of the input consists in recording the fact that the customer is waiting and not the sandwich.

Short NLG: A customer waits for a bill.

Long NLG: The customer who ordered the sandwich that waits for a bill and who waits for a bill waits for a bill because I draw inferences. I draw them because I do not believe literally that it waits for a bill.

It should be kept in mind that the result of the interpretation does not consist in a replacement or literal paraphrase. Although the interpretation event built can be seen as a literal paraphrase, it reflects the system’s beliefs about the input in the light of figurative knowledge. Instead of ascribing a low degree of belief to the input, as would have been the case without figurative resolution, the input proper is accepted and in addition the meaning behind the input is made explicit. Moreover, the connection between input, interpretation and KNB is recorded.

As for the readability of the long NLG of the interpretation event, the problems mentioned in connection with the ‘draw inference’ event (c.f. Sect. 5.3.2) apply here, too. Since all the information contained in the NLG stems from the SemNet and it was our aim not to create unnecessary new nodes in the semantic network or to go out of our way of basing the NLG on the contents of the SemNet (‘canned messages’), the linguistic quality of the long NLG is questionable. In a refined implementation, where long NLG of figurative reasoning would serve not only as debugging aid, measures would have to be taken to ensure that there are

no ambiguities left in the long NLG⁸.

5.3.5 Conventional figures

As is the case with the other knowledge-based figures, a selectional restriction violation is neither a sufficient nor necessary condition for the presence of a conventional figure. Thus, the only feasible way of recognising input as fitting the criteria of this class leads through recognition on the basis of metaphoric knowledge. For the full interpretation of conceptual metaphors, rich and highly interconnected world and encyclopaedic knowledge are required to elaborate the relation between the concepts that are only outlined in the metaphoric knowledge schemata. The following example of a metaphoric knowledge schema can help to illustrate this thought;

LONG-TERM PURPOSEFUL ACTIVITY IS A JOURNEY, sub-schemata LIFE IS A JOURNEY.

All the implications this schema has (the metaphorical mappings that can be made, c.f. Sect. 5.2.1) need background knowledge if one wants to bolster up the interpretation beyond mere recognition. Some of the possible mappings of the LIFE IS A JOURNEY metaphor mentioned above are (the examples are taken from [Lakoff *et al.* 1995]):

The person leading a life is a traveller 'He sails through life' 'He was speeding in the fast lane'

Life states are locations 'He regarded his childhood as a prison'

Long-term purposes are journey destinations 'My goal is to become professor' 'He reached his retirement in good health'

Means for achieving purposes are routes 'I'll try a different approach'

If there is no or only very little knowledge about journeys, knowledge such as the fact that a journey follows a path, has stages, has a starting and an endpoint; that a person can use different means of transportation such as cars, legs, planes; that the path has a spatial dimension and the journey a temporal an spatial dimension; that there are impediments to travel and so forth, the interpretation will be limited to the recognition and expansion of the schemata to whatever knowledge is available.

⁸In our example, something along the lines of 'On the basis of figurative resolution of the input, I believe that it is the customer who ordered the sandwich (that is said to be waiting for a bill), who is waiting for a bill'.

The concept extension stage for conventional figures is a combination of the methods mentioned so far; information on, e.g., the family of the input concepts is gathered as well as ontological and taxonomic information. Moreover, concepts related to the input concepts by means of episodic knowledge (i.e. previous input) are integrated into the collection of concepts gathered during concept extension. Thus it is not necessarily more complex than with other types, but more thorough, since the possibilities of expressing one concept through another concept are not as limited as in the case of a conventional figure.

The algorithm for conventional figures can be used to resolve other types, namely, the other knowledge-based figures (with the obvious exception of contextual metonymies), as mentioned in Sect. 5.3.2. To this end, the additional step of type determination has been integrated into the algorithm (c.f. Fig. 5.7). After a match with figurative knowledge for the extended input has been found, the type of figure the input matches with is determined, so that possibly necessary alterations in the further processing can be taken into account. Details on how the NL representation of figurative knowledge is used in this process can be found in Appendix B.2.4.

The following example of a conventional figure, an instance of the MACHINES AS PEOPLE metaphor [Lakoff *et al.* 1995], was taken from [Martin 1990]:

I killed emacs.

Since the word senses of ‘killing’ known to the system prototypically restrict their agent to an animate entity, no literal interpretation is found, and hence the ‘literally not possible’ and ‘not believe literally’ events are built. Again, it should be kept in mind that there is no anomaly detection at work; a figurative interpretation for a conventional figure is sought if and when the resolution sub-system is activated, not when a constraint violation is detected. The events mentioned are used to record the fact that a literal interpretation has failed. For other events, e.g. events adding the knowledge that a figurative interpretation has failed, see Appendix B.2.7 and the examples of the rejection of anomalous input in Appendix A.1.

Short NLG: You killing emacs is literally not possible.

Long NLG: You killing emacs that you terminated is literally not possible so I believe that you killed it is a metaphor. I draw inferences because I do not believe literally that you killed it.

Short NLG: I do not believe literally that you killed emacs.

Long NLG: I do not believe literally that you killed emacs that you

terminated because I believe that you killed it is a metaphor. I draw inferences because I do not believe literally that you killed it.

Concept extension yields a collection of conceptually related information for the input 'emacs', among them 'process', 'computer program', 'software', 'system', 'computer' and 'machine'. The figurative knowledge that machines can be described in terms of people, especially the CESSATION OF FUNCTION IS KILLING metaphor [Lakoff *et al.* 1995] present in the SemNet, matches with the extended input; thus the system has reason to believe that the input is a metaphor and builds the corresponding 'is tropetype' and 'believe is a tropetype' events:

Short NLG: You killing emacs is a metaphor.

Long NLG: You killing emacs that you terminated is a metaphor. I draw inferences because I do not believe literally that you killed it.

Short NLG: I believe that you killing emacs is a metaphor.

Long NLG: I believe that you killing emacs that you terminated is a metaphor because you killed it is literally not possible. I draw inferences because I do not believe literally that you killed it.

As the name of the CESSATION OF FUNCTION IS KILLING metaphor suggests, 'killing' in the context of this particular metaphor is used to denote that the cessation of function is caused by an active agent. Additionally, the figurative knowledge tells us that 'cessation of function' in the context of software means to 'terminate' the processing. Before the interpretation of the input is sought, the 'preferred schema' constraint is checked (see also under knowledge-based tropes in Sect. 5.3). In order to be classified with absolute certainty as an instance of the MACHINES AS PEOPLE or CESSATION OF FUNCTION IS KILLING metaphor, the input should contain elements from the 'killing' and 'machines' conceptual domain. This constraint is satisfied since 'killing' is explicitly present in the input, and concept extension of 'emacs' also yields a positive result for the 'machines' domain. Thus, the following interpretation of the input is performed by the system:

Short NLG: You terminated emacs.

Long NLG: You terminated emacs that you killed because I draw inferences.

5.3.6 Idioms

While working on the classification of tropes, we found that idioms follow 'normal' semantics more closely than other types of figures, but they appear to be more out-

standing to the average speaker/hearer than other, highly conventionalised figures, such as PART FOR WHOLE metonymies. Often, they consist of ordinary words in ordinary combinations, such as ‘spill the beans’, ‘kick the bucket’ or ‘not his cup of tea’. This means that the criterion of a constraint violation based on semantic or grammatical patterns does not lend itself to the task of recognising idiomatic expressions in the input. Instead, figurative knowledge has to be consulted and attempts have to be made to match the input against it. Of course, pattern-matching methods in the traditional sense will not be of much use; otherwise an idiom could be seen as an additional entry in the lexicon or as a special word sense of a complex phrase. This would effectively reduce the set of recognisable idioms to the ones already listed in the system’s KNB. Moreover, it would completely rule out any possibility of accounting for the principles of ‘lexical flexibility’ and ‘semantic productivity’ observed by Gibbs (see also Sect. 5.2.2 and [Gibbs 1994]), i.e. of recognising, accepting and interpreting idiomatic input if it fails to conform exactly to some pre-defined form. With our conceptual matching on the other hand, where the meaning behind an utterance and its constituents is taken into account, the necessary flexibility and a broader coverage than any ‘dumb’ pattern matching or lookup could provide is achieved. Algorithmically, this conceptual pattern matching manifests itself in the stage of concept extension, where information conceptually related to the input is considered to be of interest, alongside the input proper (i.e. the pure form as used in traditional pattern matching).

Before the figurative resolution sub-system was integrated into LOLITA, the system’s ability to handle idioms was restricted to lexicalised idioms, i.e. expressions that were listed as additional word senses and had been coded into the the system’s KNB. In implementing the solution for figurative resolution, the capabilities were extended to handle idiomatic expressions on the basis of figurative knowledge. It was found that a distinction between two types of idiomatic expressions should be made, namely, lexical idioms and compound idioms. The former type could be regarded as additional word senses of lexical material, but is not resolved on the basis of a pre-coded lexicon, listing its idiomatic sense. Instead, figurative knowledge is consulted, and the method of conceptual extension used to cover lexical flexibility and semantic productivity [Gibbs 1994]. The latter type, compound idioms, is of greater linguistic complexity than a single lexical item, i.e. a phrase, and resembles conventionalised figures such as simple tropes without having to conform to one fixed surface form in our model.

It has to be said that there is not much leeway in terms of flexibility in the prototype as far as the interpretation is concerned. While we can recognise a modified idiom through conceptual pattern matching, we can not yet transfer systematic modifications in the meaning, which correspond to the modification of the form. Unfortunately, we could not find any theory on the correspondences between changes in the form and systematic changes in the meaning of idiomatic expressions. The scope of the work and the non-existent theoretical foundation made it impossible to investigate this line of enquiry any further; the necessity to allow for the variation principles mentioned, however, has been taken into account.

No examples of idiomatic expression were found in the literature, thus the example presented here is a common (compound) idiom that appears in everyday speech. The system's input was

`Jake kicked the bucket.`

With idioms, the context plays an important role. While some contexts suggest an expression is to be understood literally, others suggest the same expression is to be taken as idiomatic. No judgement on the figurativeness of the input can be made in isolation. This is reflected in the figurative reasoning for idioms. Here, the 'literally not possible' event is not based on the degree of belief the system can attribute to (the meaning of) the input but on the likelihood of the particular interpretation in the context. As with the resolution of other types, nothing is replaced or substituted. The input as well as a possible interpretation is kept so that later stages, should they contribute to the context of the input, can decide on whether the literal or the figurative interpretation is to be chosen if there are competing interpretations. Since there are no context referents implying a literal reading in our example, the 'not believe literally' event is built.

Short NLG: `I do not believe literally that Jake kicked a bucket.`

Long NLG: `I do not believe literally that Jake kicked the bucket that he kicked because I believe that he kicked it is an idiom.`

The 'literally not possible' event for this input is as follows:

Short NLG: `Jake kicking a bucket is unlikely literally.`

Long NLG: `Jake kicking the bucket that he kicked is unlikely literally so I believe that he kicked it is an idiom.`

Metaphoric knowledge consultation for idioms is different from the same process for other tropes. Since all of the system's knowledge is stored in the SemNet in the form of concepts and since it was considered to be of importance not to introduce new, task-specific data structures into the system, there is no way of recording

idiomatic meanings in a fashion similar to lexicographic methods, i.e. by listing them. This is especially true for compound idioms. Instead, it is checked whether the (possibly) extended concepts of the input appear individually in figurative knowledge. As with the preferred schema constraint discussed in the context of conventional figures (see above), the input is then examined to make sure that all of the concepts involved in the idiom are present. In other words, not every expression containing ‘kick’, ‘bucket’ or an animate agent is necessarily an idiom, and the processing of the input as idiom is stopped as soon as it has been established that it does not conform to the expected conceptual structure.

In our example, consultation of the metaphoric knowledge succeeds in finding that ‘to kick the bucket’ is an idiomatic expression and that the input concepts and their relations corresponds to the required form, so the ‘is trope-type’ and ‘believe is a trope-type’ events are added to the chain of figurative reasoning.

Short NLG: Jake kicking a bucket is an idiom.

Long NLG: Jake kicking the bucket that he kicked is an idiom.

Short NLG: I believe that Jake kicking a bucket is an idiom.

Long NLG: I believe that Jake kicking the bucket that he kicked is an idiom because he kicked it is unlikely literally.

The interpretation of a compound idiom differs slightly from the interpretation of a lexical idiom in that the syntactic form of the interpretation may change significantly, due to the fact that a (complex) phrase from the input is affected and not simply one (or more) lexical items. Fortunately, this does not cause many problems, since the resolution is performed on the level of concepts (or on a deep meaning level), not on the surface level or some intermediate deep syntactic level. The interpretation of the compound idiom as brought forth by the system was:

Short NLG: Jake died.

Long NLG: Jake died because I draw inferences.

5.3.7 Comparison statements

Comparison statements (see also Sect. 5.2.2) are not processed using the knowledge-based approach of conventional figurative knowledge. Nevertheless, their handling can be described as being knowledge-based to a high degree, too, because all the knowledge available within the system is used for their treatment. Comparing the

domains or semantic vicinity of the input concepts would be a much too costly when taken with a large, highly interconnected semantic network such as SemNet, so instead a form of concept extension takes place (see Sects. 5.3.2 – 5.3.6). This stages collects the information on the concepts that is deemed to be relevant for the comparison, e.g., prototypical acts ('surgeons use scalpels'), attributes ('surgeons are rich'), definitional relations ('surgeons are medics') as well as causal and functional relations ('surgeons treat patients').

This information then forms the basis of the comparison of the two concepts. The individual attributes and relations ('rich', 'treat') are extended once more in order to arrive at conceptual matching and comparisons, not simple surface pattern matching of pre-defined attributes. Pieces of information that satisfy certain requirements, namely, those of the triangulation rule (c.f. Sect. 5.2.4), are then taken to be linked across domains by the comparison or, in other words, to be (metaphorically) compared to one another. For the reasons mentioned above, comparison statements are of course not recognised by using a constraint violation or anomaly detection mechanism. Instead, the form of the input is taken as a first indication that we deal with a comparison statement. Further checks on the concepts (and their relation) establish whether the input is a figurative or a literal comparison (see Sect. 5.2.2 for details).

Before the addition of the new facilities for handling comparison statements, the system took input like 'butchers are surgeons' to be a statement about class inclusion and modified its knowledge into the direction of making butchers a subclass of surgeons. In other words, if no gross contradiction or incompatibility arose from any input of the type comparison statement, the comparison was taken literally, thus establishing a new class relation containing new taxonomic knowledge that had to be added to SemNet. The implementation of the comparison statement algorithm changed this situation. Now a sentence like 'butchers are surgeons' is taken as a statement about butchers and their attributes, and the knowledge the system has about butchers. Although the system used to indicate the occurrence of a taxonomic mismatch (e.g. 'encyclopaedias are gold mines'), it had no choice but to 'trust' its source of information and take the input as a genuine literal comparison statement. Consequently, the only signal that the input was exceptional consisted in the marking of the new information derived from the input as unproved and revisable.

The first example of a (figurative) comparison to be presented was taken from

[Veale 1995]. The system's input in a query session was:

`butchers are surgeons.`

The form of the input, a generic is-a relation, satisfied the first requirements of the recognition stage (for the individual processing steps, see also Fig. 5.11). Consequently, the SemNet was consulted to see if there was pre-existing knowledge on the two concepts being compared. As this was the case, the SemNet was checked for an already established relation between the concept 'surgeon' and the concept 'butcher'. No relation was found and thus the 'families' of the concepts were analysed for compatibility. Both concepts belonged to the same families, e.g. 'human', 'professional'.

Therefore, the conceptual proximity between 'butcher' and 'surgeon' had to be computed in order to establish whether the input was a figurative or literal comparison statement. As mentioned above, the episodic and world-knowledge closely associated with the two concepts as well as taxonomic information formed the basis for this computation. The computation resulted in a low conceptual proximity value for the concept pair and thus a non-literal interpretation was favoured by the system. This step corresponds to the 'proximity/triangulation' stage in Fig. 5.7. It had been established that the input was a figurative comparison, therefore the corresponding events, 'is tropetype' and 'believe is a tropetype' were built:

Short NLG: `Butchers being surgeons is a figurative comparison.`

Short NLG: `I believe that butchers being surgeons is a figurative comparison.`

At the same time, triangulation was performed on the knowledge associated with the two concepts (see Fig. 5.4). In the episodic knowledge about butchers and surgeons, there was a match between the acts performed by the two professions and the implements used.

The concepts satisfying the triangulation rule constraints were passed to the function responsible for performing the figurative interpretation (c.f. 'makeinter_figurative' in Fig. 5.7) where it was established whether they are valid cross-domain correspondences, based on the figurative comparison of 'butchers' and 'surgeons'. Accordingly, the following information was added to the SemNet in the form of new events which embody the system's interpretation of the input.

Short NLG: `A knife is metaphorically a scalpel.`

Long NLG: `A knife is metaphorically a scalpel because I believe that butchers being surgeons is a figurative comparison.`

Short NLG: Slaughter is metaphorically surgery.

Long NLG: Slaughter is metaphorically surgery because I believe that butchers being surgeons is a figurative comparison.

The cross-domain transfer of knowledge that is represented in relations, be it episodic or world knowledge is performed only during a figurative interpretation of a comparison statement (c.f. 'transfer_features' in Fig. 5.7). If a comparison statement is deemed to be literal (class inclusion statements are of this type), transfer of (possibly abstracted) features takes place. Features in this context mean natural dispositions, physical attributes, etc. In the case of a figurative comparison, attributes from the vehicle concept which have an equivalent in the domain of the tenor are also transferred. The relevant events expressing the transfer of attributes (c.f. 'transfer_attributes' in Fig. 5.7) for this example were

Butchers are metaphorically skilled.

Butchers are metaphorically rich.

Butchers are metaphorically surgeons.

It should be noted that in the NLG the fact is expressed that the features and attributes are ascribed to the tenor on the basis of the figurative comparison. In the prototype, this is achieved by adding the adjective 'metaphorically'. As mentioned before, a more elaborate NLG of the events associated with the chain of figurative reasoning would improve readability and style, but was not considered to be of immediate interest in the framework of this project.

Naturally, the depth of the interpretation (i.e. the amount of information that can be metaphorically ascribed to the tenor domain) depends on the information available on both domains or concepts. Without knowledge about the features, relations and attributes of both concepts, triangulation will yield no result. Moreover, the interpretation could be regarded as shallow and impoverished because it does not take all available information (in our case the whole of the SemNet) into account. The points to note are that the algorithm for the resolution of comparison statements was not designed as an evaluation of cognitive or linguistic theories. Instead, the principles of natural language engineering (NLE) and, most importantly, the constraint of integrating the algorithm into an existing, real-time, large-scale system made it unavoidable to concentrate on an acceptable performance of the sub-system.

Like the rest of the figurative processing sub-system, the comparison statement algorithm is performed in real time. There are no noticeable delays in comparison

to processing with figurative analysis switched off. This performance is distinct from other implementations dealing with non-literal analysis. Of three different implementations treating what we call comparison statements, presented by Veale [Veale and Keane 1995], the best performed at 12.5 seconds for the resolution of a comparison statement (5.8 seconds in the incremental case of presenting the same comparison statement to the system again) and claimed to take all the available information into account. This information consisted of “a sample memory network containing 284 concepts nodes and 1597 user-specified inter-concept relation links” [Veale and Keane 1995]. Clearly, the number of concepts and links is not comparable to the number of pieces of information in the SemNet.

In addition, the fact that the algorithm does not perform well if no knowledge is available should not come as any surprise. Since no customised knowledge base or representation is used into which, consciously or unconsciously, a simplified world model and nothing but relevant relations has been encoded, the algorithm has to make use of what is available, with respect to the content and structure of the KNB. Moreover, it seems unlikely that humans would be capable of understanding a literal or figurative comparison if they did not have sufficient knowledge about at least one concept. And finally, the fact that the figurative sub-system, and especially the comparison statement resolution fully depends on the globally available, shared KNB ensures consistent and transparent behaviour of the system, especially since SemNet is not simply consulted during processing but constantly being updated by various components of the LOLITA system (c.f. Sects. 5.2.1 and 5.3).

Finally, let us consider a further example, taken from Ortony [Ortony 1979]. This shows that the ‘interpretation’, i.e. the cross-domain comparison established by the figure, depends on the knowledge available. Although humans would certainly find more cross-domain mappings that can be established on the basis of the comparison (see [Ortony 1979]), LOLITA took the knowledge that was most closely related to the concepts, and since the amount was limited, came up with one cross-domain feature transfer and two attribute transfers for the input, i.e. three interpretations or figurative inferences. The system’s input was **Encyclopaedias are gold mines.**

The input satisfied the formal constraints of a comparison statement (for the individual steps, see Fig. 5.11), both concepts were known, and there was no pre-existing relation between the concepts. The families of the concepts, however, were not compatible and not even related in a way that made a literal comparison seem

likely (see also the discussion of family compatibility in the context of the algorithm for comparison statements in Sect. 5.3.1). Consequently, the initial stages of the comparison statement resolution branched directly to the ‘makeinter_figurative’ (c.f. Fig. 5.7), and the events indicating the system’s position towards the input, namely, ‘is tropetype’ and ‘believe is a tropetype’ were built.

Short NLG: I believe that encyclopaedias being gold mines is a figurative comparison.

Long NLG: I believe that encyclopaedias being gold mines is a figurative comparison so encyclopaedias are metaphorically mines.

Short NLG: Encyclopaedias being gold mines is a figurative comparison.

The outcome of triangulation suggested that ‘information’ and ‘gold’, which both stand in a ‘contained by’ relation to the respective concepts from the input is a candidate for a (metaphorical) cross-domain correspondence. Additionally, the attribute of ‘size’ was transferred from the vehicle to the tenor domain by the ‘transfer_attribute’ stage (c.f. Fig. 5.7) of the resolution algorithm. The interpretation events built are as follows:

Short NLG: An information is metaphorically gold.

Short NLG: Encyclopaedias are metaphorically mines.

Short NLG: Encyclopaedias are metaphorically big.

Long NLG: An information is metaphorically gold because I believe that encyclopaedias being gold mines is a figurative comparison.

Long NLG: Encyclopaedias are metaphorically mines because I believe that encyclopaedias being gold mines is a figurative comparison.

Long NLG: Encyclopaedias are metaphorically big because I believe that encyclopaedias being gold mines is a figurative comparison.

For more examples and the application of the comparison engine to a conventional metaphor LONG-TERM PURPOSEFUL ACTIVITY IS A JOURNEY, sub-schema LOVE IS A JOURNEY, see Appendix A.6.

5.4 Integration and implementation overview

The parent system of the figurative resolution sub-system, LOLITA, is written mainly in the functional programming language Haskell [Hudak *et al.* 1994]. Including the code for non-literal analysis, the system consists (as of March 1996) of 60380 lines of code, 11044 lines of ‘C’ and 49336 lines of Haskell.

It runs on a UNIX⁹ workstation using either SunOS or Solaris¹⁰. System requirements vary with the complexity of the task performed. While a typical query session runs in real-time with 36 MB of main memory, the analysis of complex newspaper articles (e.g. for MUC tasks and content scanning [Garigliano *et al.* 1993, Morgan *et al.* 1995, Costantino *et al.* 1996]) will require around 80 MB of main memory for satisfactory speed of execution.

The figurative resolution system consists of about 2000 lines of code, corresponding to 20000 lines in a declarative language [Turner 1982]. The main part of the code for the prototype resides in one main module. The code for the individual trope types will eventually be moved into separate files. In addition, coding was carried out in other modules, e.g. for interfacing the module to the rest of the system. A separate portion of code was required to realise the settings and options menu and additional work in the existing code dealing with pragmatics secured the integration of the figurative resolution with the established interpretation facilities.

This code is distributed amongst approximately 130 functions containing the resolution functionality. Additional functions were coded to implement the interfacing/integration functionality and the options/settings sub-system.

Full working integration into the LOLITA system has been achieved. However, this integration is only a prototypical linking of the non-literal sub-system's functionality to the rest of the system. Customised interfaces have to be built to allow individual applications (e.g. query) to make the best use of the functionality provided by the non-literal analysis. The design and implementation of such interfaces was, of course, not one of the goals of this project, as it concentrated on a broad coverage of non-literal phenomena. Different applications have different needs with regard to how they treat figurative language. For example, the dialogue application would rather use the interpretation of a metonymy (e.g. input like 'I read a lot of Shakespeare' might result in a response like 'Do you read plays or poetry?') whereas a language-tutoring application may well be content with accepting figurative input. Nevertheless, the sub-system has to provide information in a way that makes this information easy to use for other parts of the system, even more so when considering the NLE requirements of integration, flexibility and usability (see Sect. 2.1). Attention was paid to these requirements and we believe that those objectives have been met.

⁹UNIX is a trademark of X/Open Company Ltd.

¹⁰Solaris and SunOS are trademarks of Sun Microsystems, Inc.

From a system analytical point of view, the functionality of the figurative resolution is located in the area of pragmatic and semantic analysis of LOLITA, interacting with both, and consulting the SemNet to obtain all the available information. From the different angle of interpretational stages, it is located at the exact point where sense selection and creation [Gibbs 1994] is performed (c.f. Fig. 5.1). The non-literal analysis enhances this functionality in two ways. Firstly, if sense selection fails because, for example, a selectional restriction was violated, it will provide further sense creation functionality in addition to the semantics module. Secondly, it provides supplementary sense selection by making previously inaccessible figurative senses available (e.g. ‘her thesis is about Shakespeare’ has the literal reading ‘about the person or writer’ and an additional metonymic reading ‘about the work of Shakespeare’).

The following examples of how integration and open interfaces influenced the design may serve to illustrate the point. As mentioned in the context of the discussion of individual examples (Sects. 5.3.2–5.3.6), the figurative resolution subsystem does not influence the usual interpretation process nor its results as such, because it is in no position to make assumptions about the direction into which a transformation should be performed. This is true of the general purpose figurative processing facilities; any non-literal functionality customised for an application or a figurative analysis workbench should, of course, be extended to enable it to deal with all aspects of processing NL input in situ, clearly without losing rapport with the parent system. Instead of handling its input in an idiosyncratic and opaque fashion, it carries out its work alongside the standard processing, recording the steps taken, problems encountered and results achieved, and provides the means for other components of the system to make use of these resources.

It was assumed that the most important pieces of information resulting from a non-literal analysis in the framework of a general NLP mechanism are the indication that figurative input was detected (recognition), what steps were taken to handle this input (figurative reasoning), and the outcome of the handling (interpretation, fail events). Moreover, the recognition part consists of important sub-parts which have to be covered in terms of recording them, such as the type of trope and context.

All this information created by the non-literal analysis has been interfaced to the rest of the system, i.e. functions were coded to check for the existence of the information and to provide the means to access it. An abstract data type represent-

ing a higher level interface to this information (as opposed to low level interfaces, providing access to information pertaining to one aspect of non-literal analysis, e.g. whether or not a portion of input has been resolved already, which method was used) was designed so that future application and task specific extensions of the analysis have a well-defined, common interface. For details, see Appendix B.5.

The low level interface functions include:

- ‘has-been-analysed’, which provides information on whether a concept relation has been analysed by the non-literal resolution module.
- ‘is-trope’, providing information on whether a concept relation has been classified as a trope and the type of figure.
- ‘interpretations-of-trope’, which, given a concept relation (i.e. piece of SemNet or input) returns the results of any kind of non-literal analysis.

Additionally, it was deemed necessary to allow for the revision and modification of the figurative reasoning and its results, for example, if supplementary contextual information becomes available. Therefore, every single step of the processing is recorded in the globally accessible data structure SemNet. If an application has to revise any of the non-literal sub-system’s results, this is facilitated immensely by the explicitness of the representation. The problematic information (e.g. a classification as a specific type that has to be revised) can be picked up and all relevant information relating to the information to be revised is accessible and modifiable through the well-defined links.

As far as the performance and complexity of the implementation are concerned, the following points are of importance. It is impossible to evaluate a sub-system within a larger framework of other components without a detailed evaluation set-up, factoring out insignificant elements (at least with respect to the evaluation of the sub-system) such as the influence of the rest of the system (see [Smith 1995] for a detailed, exemplary discussion of problems relating to the evaluation of NLP sub-systems). Because of this, it is very difficult to evaluate the figurative sub-system. Not only would the effort needed to design a sound evaluation set-up clearly lie outside the scope of this investigation, but also a simple, individual task whose accomplishment could be easily evaluated is lacking in the figurative resolution model. Moreover, it is interconnected to so many other parts of the LOLITA system and has to rely on their results and performance that the task of

finding a set-up which evaluates nothing but the non-literal analysis seems well nigh impossible. A final difficulty in connection with the evaluation of the sub-system is that it cannot be regarded as a monolithic algorithmic complex. Its functionality is divided into separate parts, and functions performing similar tasks are coded differently, specific to the type of trope they are dealing with.

Nevertheless, an evaluation of some description is needed, especially against the NLE background, and there are some indications as to how the sub-system is performing. All the algorithms perform in real-time. There is no noticeable delay for either the successful or unsuccessful resolution of input, when compared to the processing without non-literal analysis. As is obvious from the description of the algorithms, a more complex concept extension (see Sect. 5.2.4) will involve more processing and result in a higher processing time. Yet due to lazy evaluation, a feature of the programming language Haskell, this does not result in a linear increase in processing time requirements.

However, there is a worst-case scenario which represents an extreme deviation from the average case performance. It is not clear how it can be avoided, since it is related to fundamental approaches and techniques used in the LOLITA system. The worst case occurs when there is an attempted resolution of input containing an ungrammatical or unknown verb, using the conventional resolution (c.f. Sect. 5.3.5). It is an essential design issue of the LOLITA system to accept ungrammatical, erroneous or incomplete input, and on encountering such input, the system tries to find contextual, prototypical or world knowledge information enabling it to, complete, correct and understand the input. In cases where this is not possible immediately, depending on the task, either the system requests more information from the user, or ambiguities and errors are kept until such time when additional information for disambiguation or correction is available. Unfortunately, it is not possible to simply avoid any attempt at resolving input if the verb is found to be ungrammatical or unknown.

This is because incomplete information also occurs during normal processing, since input is processed incrementally and since the non-literal analysis is located at an early enough stage to allow it to play a role in sense selection (as opposed to simply accepting what semantics and pragmatics pass on to the module, an approach that would defy the objective of the whole undertaking). Clearly, it would be beneficial if one could make the distinction between incomplete input that appears in the course of normal processing and incomplete input that resulted

from the use of an ungrammatical or unknown verb. But since the coding was done in a side-effect free, functional programming language, no stack or flag can be kept to signal that the abstract machine entered a certain state. A time-out feature, stopping processing after, e.g., 1.5 seconds is also not a feasible solution either, since it would not enable the system to make a distinction between anomalous, incorrect and possibly figurative input.

Fortunately, the parsing module of LOLITA and its interaction with the SemNet ensures that the worst case will be limited to the sentence or phrase in which the faulty verb occurred. If, for example, input of paragraph length is processed, the rest of the input will be processed in real-time. It was found that it would have been inappropriate to spend a considerable amount of time working on a solution to remedy or by-pass this situation since input of the type under discussion is not encountered frequently enough to call for an instant solution. In this worst case, and depending on the complexity of the input sentence, processing takes between 20 and 35 seconds to recover from the ungrammatical input and to signal that no resolution is possible. Although it is a small consolation, the algorithm has been tested rigorously on input of this kind and, unlike in its initial versions, is now robust enough to recover in all cases.

5.5 Summary

This chapter has presented the solution to the problems discussed in Sect. 4.1 in detail. It has shown how the knowledge-based approach to non-literal analysis of natural language input has been implemented according to NLE principles in the framework of the existing LOLITA system and how full integration has been achieved.

Elements of the solution thus described in this chapter are the tasks and prerequisites connected to the processing of figurative natural language. Among the tasks identified (c.f. Sect. 5.2) and described are the recognition, interpretation and representation of non-literal language and figurative knowledge. The most important prerequisites outlined in this chapter are a suitable classification of tropes (presented in Sect. 5.2.2) and the knowledge structures (see Sect. 5.2.3) needed for a knowledge-based resolution of figurative input.

In addition, it has been shown how auxiliary methods (see Sect. 5.2.4) necessary

to arrive at the knowledge needed for the accomplishment of the resolution tasks can be utilised, and an overview of the algorithm for the treatment of figurative input is given in Sect. 5.3.1. Detailed examples of the algorithms at work, actual in- and output, are presented in Sects. 5.3.2–5.3.6, followed by technical details on the implementation of the functionality and its integration into the parent system LOLITA in Sect. 5.4.

The next chapter presents the evaluation of the undertaking described in this thesis on the basis of the measures and objectives laid out in Sects. 4.1 and 4.3, as well as an assessment of the project as a whole and a concluding outlook on promising areas and directions of research stemming from the analysis of the problem area, but which could not be covered by this project due to its limited scope.

Chapter 6

Evaluation and conclusion

In this chapter, we analyse the extent to which the objectives of the work were met and evaluate the results of this analysis with regard to the methodological approach (c.f. Sect. 4.2), the goal set (see Sect. 4.1), and the problems outlined (c.f. Sect. 4.3).

A short initial discussion of how the primary objectives were met is followed by a review of subordinate results and an analysis of how well the solution presented fits the natural language engineering (NLE) framework and its requirements.

Finally, we assess what has been achieved in the work and provide an outlook of work that can be undertaken to extend the results and of promising directions for research which can be derived from the aims of this project.

6.1 Primary objectives

All the primary objectives (see Chap. 4) have been met. The comprehensive literature survey that takes findings from various disciplines into account as well as a thorough analysis of the state of the art allowed us to decide on the feasibility of the task of processing non-literal language in the LOLITA framework. Moreover, it provided us with a means of distinguishing between promising approaches and less adequate ones as well as showing areas in the field where work remained to be done. This analysis resulted in the acceptance of the knowledge-based view of metaphor (c.f. Sect. 3.2.5.1) as basis for the solution presented in this thesis. In order to arrive at the solution certain sub-tasks were necessary, and these were successfully

carried out; they included establishing a classification of types of figurative language more suited to a computational framework than the traditional ones and finding a suitable representation for figurative knowledge. The representation also touches upon the NLE principle of integration, as discussed below in Sect. 6.3.

Since the solution has been successfully implemented and tested (this being an objective itself), the goal of extending the system's facilities in the direction of figurative language has been achieved: Non-literal input is distinguished from anomalous input, classified according to the system's knowledge about figurative language, accepted as such, and an interpretation of the greatest possible depth is sought. Consequently, information contained in non-literal utterances is made available to the system's other sub-parts, i.e. it can be processed further using existing components.

6.2 Secondary objectives

As discussed in detail in Chap. 4, a traditional comparative analysis cannot be carried out. Nevertheless, it is of interest to see how results from our solution compare with those of other approaches, keeping the differences between the particular intentions and environments in mind.

Wilks' preference semantics (see Sect. 3.2.2) enabled a natural language processing (NLP) system "to accept the abnormal", e.g. non-literal portions in its input [Wilks 1975b]. The augmentation of preference semantics with pseudo-texts [Wilks 1978] extended the NLP system's capabilities to finding a (repair-view-based) 'interpretation' for figurative input of some types of figures. Thus, a very basic detection, acceptance and interpretation mechanism was created for deviant input that can be semantically 'recovered' with the help of pseudo-texts. Our solution is, also capable of accomplishing this. As mentioned before, the established knowledge structure of prototypes (c.f. Sect. 5.2.3) is comparable to pseudo-texts, when inevitably viewed in connection with the rest of the system's knowledge base (KNB). It encodes knowledge about how concepts relate to each other prototypically, and thus enables our algorithm to find an interpretation for figurative input on the basis of the contents of the system's KNB. It has to be pointed out, however, that Wilks used pseudo-texts in a way that abstracted from the actual input on detecting a constraint violation and then consulted the pseudo-text information to find a literal paraphrase for the input. In our solution, a constraint violation

is not a necessary triggering condition, and we do not literally rephrase the input after seeking a fitting abstraction from the input which then results in constraint satisfaction. Although relatively simple bottom-up abstraction (for the sake of efficiency) takes place in the case of simple tropes (see Sect. 5.3.2), the abstraction is not regarded as the repaired, literal interpretation of the figurative input, nor is the abstraction performed on a static, detached ontology but on the contextually influenced and dynamically updated SemNet and, where possible, in a direction indicated by the type of trope.

Fass' met* method (c.f. Sect. 3.2.2.2) provided a basic means of distinguishing between types of tropes such as metaphor and metonymy, and anomalous input. It also used the methods of anomaly detection and abstraction. For finding the interpretation for the input, it relied on a 'sister match' from the static ontology with a concept from the input. Our system is also capable of this but neither does it rely on an anomaly detection nor does it express preferences towards a literal interpretation (see the critique of Fass' met* method in Sect. 3.2.2.2). Of course, it does not prefer figurative interpretations, either. Instead, it tries to find all possible interpretations. Another advantage over the met* method is the ability of our system to arrive at a more complex classification of figures: whereas the met* method applies tests in a fixed order and will classify input according to which test succeeds first (and enables a distinction between anomaly, metonymy and metaphor), our solution correctly classifies input as being of a certain type of figure, independently of the order of application of tests, and does so for more types of figures. When 'sister matches' are used in our model, they are not found in a static, pre-defined ontology but in the common, shared KNB of the SemNet, which means that the knowledge available to the figurative resolution sub-system is always up-to-date and, more importantly, dynamic.

Martin's MIDAS system [Martin 1990] (see also Sect. 3.2.5.1) is comparable in some respects to our solution, too. It was the first computational approach to use explicit figurative knowledge and, in contrast to previous implementations, seeks both literal and figurative interpretations. However, the figurative knowledge is contained in a customised, external representation. Moreover, since the relations between concepts holding in one domain are coded explicitly in great detail to facilitate cross-domain transfer of relations, any extension of this knowledge will be strenuous, if at all feasible. Although figurative knowledge is extensible in MIDAS, this extension is limited to correctly classifying a newly encountered trope

as a sub-type of a previously established one. The resulting hierarchical ordering of tropes is desirable, but not integrated into the rest of the system's knowledge.

The approach taken within the LOLITA framework, on the other hand, keeps the figurative knowledge inside the one and only KNB, the SemNet. Thus, it can be modified, revised and is accessible using the established interfaces. Moreover, no separate mapping structures are introduced. Cross-domain correspondences are computed on the contents of the SemNet, too. Again, this is beneficial, as it bases the figurative interpretation on the whole of the most recent knowledge available. The same holds for the hierarchical ordering of instances of, e.g. metonymies. Since they inherit their ordering through the ordering of their components and the ranking of the components is based on the structure of the SemNet, no discrepancy between the general ontology and the ontology of figures can arise.

Other aspects must also be mentioned, which cannot be compared to existing systems, either because no computational model has dealt with them or because they involve questions specific to the parent system. One such aspect is the extension of LOLITA's facilities to deal with comparison statements in a more refined way. Before the implementation of the non-literal resolution system, comparison statements were either rejected on the basis of a semantic or pragmatic mismatch or taken as class inclusion statements. Now, it is possible to distinguish between literal comparisons of varying degree (e.g. identity and inclusion) and figurative comparison statements, which will both receive a more elaborate interpretation than before. Such an extension of existing abilities has also been achieved for idioms and anaphors. Whereas it was previously not possible to treat idiomatic expressions appropriately if they were not of the lexicalised idiom type, i.e. included as additional word sense in the SemNet, they can now be resolved on the basis of figurative knowledge. This, in addition to the use of concept extension, means that no pattern matching is used in interpreting idioms. Instead, a flexible treatment of idiomatic expressions, or concept matching, allows a wide range of previously unresolvable input to be covered. Likewise, the existing facilities for anaphora resolution have been extended to include figurative anaphoric references, most prominently the contextual metonymies. The context-sensitive interpretation selection which prefers one of several competing interpretations on the basis of how well it fits into the context can be seen as a higher order anaphoric resolution, too.

As mentioned before, the foundation for generating natural language (NL) figures has been laid by integrating the figurative knowledge into the SemNet. Since

NL generation is based on the SemNet, all figurative knowledge contained within it is available for generation purposes. Another secondary achievement that resulted from the well-suited representation is the ability to model type change (see also Sect. 5.2.1). The initial classification of input as belonging to a certain type of trope can be revised on the fly and dynamically. Thus it is possible to model the ageing process of metaphors. An obvious point that should be mentioned in this context is that, since figurative knowledge resides within the system's KNB, it is available to other parts of the system as well. Therefore, meta-linguistic reasoning, for example, can take place. And since the type, the interpretation (or unsuccessful attempt at interpreting) and other additional information is recorded when resolving figurative input, a basis for a (stylistic) corpus analyses has become available.

6.3 Natural language engineering perspective

In this section, a discussion of how well the solution presented conforms to important aspects of NLE (c.f. Sect. 2.1) will help to evaluate if the objective of arriving at an NLE solution has been met. Although the key criteria for an NLE system apply to a system as a whole, it is, of course, a necessity for any sub-component of a natural language engineered system to conform to the requirements derived from these criteria as closely as possible.

- **Feasibility**

As far as the aspect of feasibility is concerned, the figurative sub-system fully adheres to the requirements. Partly due to its good integration, partly due to design features, such as modularisation, it does not make unreasonable demands on resources such as memory allocation and execution time. As mentioned before (see Sect. 5.4), non-literal analysis is performed in real-time, and the requirements on main memory allocation are insignificant in the context of the system.

- **Robustness**

Robustness is a crucial criterion. Any non-robust component can corrupt the system's overall robustness. This is true for both aspects of robustness, namely, (1) the coverage of the system or sub-system and (2) the behaviour in the case of erroneous input, input outside the pre-defined scope of permissible

input and other sources of errors. The coverage of the figurative analysis sub-system is not a priori restricted. It depends on the amount of figurative knowledge that is available as well as on the size and quality of the rest of the KNB. Within these limits, to our knowledge, total robustness has been achieved. During testing, no cases have been encountered where erroneous input resulted in a failure of the non-literal analysis sub-system. In cases where incomplete or incorrect data are passed on to the sub-system (see also the discussion of the worst-case scenario in Sect. 5.4), the average-case behaviour can be adversely affected. However, the sub-system recovers even in this case. It is, of course, very important not to make weak assumptions or to take incorrect decisions at such an early stage of analysis as the one at which the sub-system is located (c.f. Fig. 5.1). Thus, the solution is designed to ensure that, in case of doubt, all possible decisions are taken and revisable ones are kept revisable. The mechanism implementing the chain of figurative reasoning can serve as an example of this (c.f. Fig. 5.12 and Appendix B.2.7). It records all important steps taken by the algorithm during processing and the reason for taking them. Therefore, the processing is open to easy revision. If there is insufficient information available to make a final decision, the requirements of robustness dictate that the decision is postponed until sufficient information is available. By adhering to this rule, overall robustness is increased, because no error is carried over into other modules, and no unjust decisions will induce a negative effect on subsequent stages of processing.

- **Scale**

The NLE criterion of scale does not apply to the sub-component in question directly. Since it is integrated into a larger system and receives most of its scale related parameters from the parent system, e.g. vocabulary size and grammar coverage, it depends on the framework to fulfil these requirements. The coverage of the module, on the other hand, is comparable to one of the two aspects of robustness (see above). However, it is obvious that the implementation and design of a sub-component should ensure that scale is not limited from the very start by design features. This means that extensions and open architecture are of great concern, even for sub-parts.

These requirements have, in our opinion, been met. Since the figurative analysis module makes full use of all the information contained in the SemNet,

its coverage is the same as that of the parent system. Any input that can be processed by other modules can be processed by the non-literal analysis component. Clearly, coverage is also influenced by the amount of figurative knowledge available. In this context, it is important to note that the figurative knowledge can be extended via simple NL interaction.

- **Integration**

The criterion of integration not only makes the obvious demand that any sub-part has to be linked to the system's other modules, but also requires the component's interaction to ensure that no unfounded assumptions are made about other parts (see also the discussion of the worst-case scenario above). Moreover, any solution should be designed as independently as possible to be of general use to an array of other components. This is especially true for solutions that provide basic functionality, such as non-literal analysis, which has to be regarded to some extent as providing semantic and pragmatic functionality.

Not only has the most basic requirement of integration been met, i.e. the non-literal analysis has been fully integrated into the existing LOLITA system, but the other aspects mentioned above are covered, as well. For example, independent guards within the individual algorithms make sure that incomplete input or erroneous data passed on to the figurative module do not result in robustness problems or wrong results. In keeping with the criterion of robustness and feasibility (efficiency), the least possible amount of processing is aimed at and the problem recorded, where possible in such cases. The fact that the solution provides basic functionality is reflected in the openness of the design. Since the project concerned itself with a broad-coverage, prototypical implementation, no refined customised interfaces have been designed. Instead, it was guaranteed that all necessary information is easily accessible. Thus, individual applications can make use of whatever information they require in the context of non-literal analysis. The seamless integration into the existing KNB and the sharing of information between the rest of the system and the figurative module as well as the augmentation of the KNB by the non-literal analysis are another indication of how closely the solution followed the requirements of the NLE criterion of integration.

- **Maintainability**

High internal modularisation, the distribution of functionality on discrete

functions and analogous implementation of comparable functionality (e.g. concept extension and figurative reasoning, see also Fig. 5.7), and the use of pre-existing structures (such as the SemNet and internal data structures used elsewhere in the LOLITA system) ensure, in our opinion, a high maintainability of our solution.

- **Flexibility**

The criterion of flexibility does not play a central role in our solution. A component of such fundamental nature has to be open rather than flexible. By definition, it need not be able to adapt to different tasks and domains but must allow other applications dealing with different domains to operate on its functionality. In this respect, our solution fulfils the requirements of flexibility in that it provides all the functionality needed with regard to non-literal analysis in the framework of the LOLITA system without being dependent on a single task to request this information, nor being restricted to providing this functionality in a limited domain.

- **Usability**

As is the case with the criterion of flexibility, usability is not directly applicable to a sub-part of an NLE system in the traditional sense. However, even a component of a NLE system should be of high usability in that it is useful for other parts of the system. As mentioned above, we believe that this is the case with our solution. It provides the necessary functionality within the resolution of figurative NL input and does so in a way that enables an array of applications to make use of this functionality. Although the non-literal analysis is of little relevance to end-users, its user-friendliness is of importance to academic users. The possibility to extend the figurative knowledge via NL interaction, the ‘intelligent’ traces of the figurative reasoning and the division of the non-literal analysis capability between discrete types of tropes manifest, in our opinion, a high degree of user-friendliness.

6.4 Achievements

In our opinion, the work reported in this thesis represents a successful project. The overall achievement has been the implementation and integration of an operational figurative NLP analysis component, following NLE principles. To this

end, numerous sub-tasks had to be identified and completed. Related theoretical and computational work from various disciplines was thoroughly analysed to determine whether any existing approaches could serve as a basis for a distinctive NLE treatment of the problem of processing figurative language. Moreover, the traditional classification of tropes and the conflicting definitions of figures were re-evaluated to create a foundation on which the solution could be based. A classification of figures based on computational requirements rather than traditional definitions and an overview of types of knowledge needed for the successful processing of various types of figurative expressions on the basis of the preferred knowledge based approach have been given. In this context, the requirements and problems of processing figurative language in a real system as opposed to an evaluation configuration were discussed, as well. The sub-tasks also included the introduction and adaptation of auxiliary methods, such as concept extension, masking, triangulation and conceptual proximity computation. These methods, partly heuristics, are essential to ensure that the solution avoids the problems of feasibility other approaches are faced with. The functionality needed for processing non-literal language has been explained, and a prototypical, robust, domain independent and open implementation that conforms to the NLE principles has been realised and integrated into the parent system.

These implementations and the outline of the requirements in the framework of the knowledge-based approach have, in our opinion, shown that deep techniques are needed. By analysing the requirements of adequate processing of figurative language beyond the scope of ‘toy’ systems or systems whose primary goal it is to evaluate a cognitive or linguistic theory, it became clear that only deep techniques would provide enough power to tackle the problems involved without encountering either computational problems such as intolerable (sometimes factorial) complexity or shattering response time or, at the other end of the spectrum, without arriving at a solution that is severely limited in size or coverage, for example. The basis for treating even seemingly basic figures cannot be provided by any kind of pattern matching or exhaustive ontology, as there cannot be such a thing, especially since context will influence the relation between concepts; c.f. semantic vs. conceptual distance. This is even more so the case, if no built-in safeguarding mechanisms in the form of limited domains or application frameworks are available. Additionally, a broad-coverage solution has to be robust and flexible to the utmost extent. These features can only be achieved if great care is taken during design and implementation. Owing to the adherence to NLE principles in all stages of LOLITA’s

development, the solution presented exhibits those features.

6.5 Future work

The scope and main thrust of this project did not allow all the interesting and important directions of research related to the field of processing figurative language to be pursued with equal intensity. This section presents some promising directions of research that are derived from the project work. The clarification of the practical questions would contribute towards an extension of the solution, the pursual of theoretical objectives would contribute towards the field as a whole while at the same time delivering additional results for the evaluation of the current solution.

Probably the most extensive and promising project concerns figurative corpus work. The automated input of the Master Metaphor List [Lakoff *et al.* 1995], after its conversion to a suitable format would allow for an array of tests and projects. As mentioned in Sect. 6.3, an extension of figurative knowledge is possible via NL interaction. Since the LOLITA system is capable of processing texts in batch mode, arriving at an automated augmentation of the figurative knowledge by reading in the Master Metaphor List and subsequent corpus work (both on the figurative knowledge itself and on text corpora which are processed with the help of this knowledge) would, at the same time, test and improve the coverage of the solution. In addition, it would allow the collection of a range of empirical data, such as stylistic analyses.

Another practical undertaking would entail the (partial) implementation of various approaches to non-literal processing in the framework of the LOLITA system. This would allow an approximation of a comparative analysis of different, computational or non-computational theories of metaphor with LOLITA as a testbed, since LOLITA provides the facilities for the implementation of other theories, such as the analogy approach or sub-symbolic approaches (see Chap. 3).

On a smaller scale, the design and implementation of prototypical ranges, that means the integration of an additional information structure into the SemNet, would allow the common figures hyperbole and understatement to be processed in the framework of the solution presented. Unfortunately, this is not yet possible (for a discussion of the reasons, see Sect. 5.2.2). Although the addition of these ranges would not mean that changes in the structure of SemNet are inevitable,

the work involved is considerable, since the ranges and their applicability can be derived only through a comprehensive corpus analysis.

An idea derived from informal findings during the work on this project would combine automated knowledge acquisition with (not necessarily machine readable) corpus work. It has been claimed by Lakoff [Lakoff 1993] that all metaphors are covered by the conventional metaphor approach. While he argues that ‘novel’ metaphors are reducible to conventional schemata, we found examples where comparison statements are clearly based on conventional metaphors. Therefore, the question of whether the conventional metaphor approach can be helpful in reducing the processing involved when resolving comparison statements is of great interest, even more so in a framework where efficiency is an important factor. This investigation would furthermore serve a theoretical purpose: if a conventional metaphorical schema is at work behind the scenes of comparison statements, which are generally treated as analogy, this would have entailments for the argument regarding the priority of metaphor over analogy.

On a more theoretical basis, an investigation into the implications the latest findings on metaphor or non-literal versus literal meaning have for a semantic theory seems to be a necessary and promising venture. As has been discussed, for example in Chap. 3, traditional truth-conditional semantics have problems in accounting for figurative speech. Semantics, on the other hand, form the bedrock of a complex NLP system like LOLITA. Thus, a thorough analysis of the requirements and implications current (cognitive science) work on metaphor has on theories of meaning and an investigation into which semantic formalisms are capable of accommodating figurative meaning without the shortcomings of the classical ones seem to be a necessity.

6.6 Summary

In this chapter, the results of the project have been evaluated under two different aspects. Firstly, they were, as far as possible, compared to the results of related computational work. This evaluation showed that the achievements equal or excel comparable work, at least when we take into consideration the most important requirement for this undertaking, namely, its real-life applicability in a large-scale NLP system. Secondly, the outcome of this project was analysed in the light of the established criteria for a natural language engineered system. It was shown

that it conforms to the important NLE requirements. In conclusion, the positive evaluation was followed by a discussion of what has been achieved in the project and directions of work stemming from the project which were unfortunately outside its scope.

Appendix A

Examples of figurative processing in LOLITA

This section presents the results of the resolution of examples of the types of tropes defined in Sect. 5.2.2, as produced by the non-literal analysis component of the LOLITA system using the ‘query’ application interface to the LOLITA system. The source of the example is given if the input was taken from the literature on metaphor. The input and the relevant events (the natural language representation of the system’s knowledge) generated by the resolution algorithm are stated. Where appropriate, short and long natural language generation (NLG) is given, short version first. The names of the individual events are described in Sects. 5.3.2–5.3.6.

A.1 Simple tropes

Input: My car drinks gasoline [Wilks 1978].

Interpretation: A car consumes gasoline. The car that inputs describe and that you control consumes gasoline that it drinks because I draw inferences by using a simple resolution. It consumes it because I draw them by using a simple resolution.

Draw inferences: I draw inferences by using a simple resolution. I draw inferences by using a simple resolution because I do not believe literally that the car that inputs describe drinks gasoline that it consumes. It consumes it because I draw them by using a simple resolution. You control it.

Not believe literally: I do not believe literally that a car drinks gasoline. I do not believe literally that the car that inputs describe and that you control drinks gasoline that it consumes because I believe that it is a personification. It consumes it because I draw inferences by using a simple resolution. I draw them by using a simple resolution.

Believe is a trope: I believe that a car drinking gasoline is a personification. I believe that the car that inputs describe and that you control drinking gasoline that it consumes is a personification because it is literally not possible. It consumes it because I draw inferences by using a simple resolution. I draw them by using a simple resolution because I do not believe literally that it drinks it.

Is trope: A car drinking gasoline is a personification. The car that inputs describe and that you control drinking gasoline that it consumes is a personification. I draw inferences by using a simple resolution because I do not believe literally that it drinks it.

Literally not possible: A car drinking gasoline is literally not possible. The car that inputs describe and that you control drinking gasoline that it consumes is literally not possible so I believe that it drinks it is a personification. I draw inferences by using a simple resolution because I do not believe literally that it drinks it.

Input: Jake was owned by the motorbike.

Interpretation: A motorbike controlled Jake. The motorbike that owned Jake and that inputs describe controlled him because I draw inferences by using a simple resolution.

Draw inferences: I draw inferences by using a simple resolution. I draw inferences by using a simple resolution because I do not believe literally that the motorbike that inputs describe owned Jake.

Not believe literally: I do not believe literally that a motorbike owned Jake. I do not believe literally that the motorbike that inputs describe owned Jake because I believe that it owned him is a personification.

Believe is a trope: The motorbike that inputs describe owning Jake is literally not possible so I believe that it owned him is a personification.

Is trope: A motorbike owning Jake is a personification. The motorbike that inputs describe owning Jake is a personification.

Literally not possible: A motorbike owning Jake is literally not possible. The motorbike that inputs describe owning Jake is literally not possible so I believe that it owned him is a personification.

Other events: Inputs use human features in order to describe a motorbike. Inputs use human features in order to describe the motorbike that owned Jake.

Input: My feet are torturing me.

Interpretation: Feet torment you. Feet that inputs describe and that you control torment you because I draw inferences by using a simple resolution. They torment you because I draw them by using a simple resolution. They torture you.

Draw inferences: I draw inferences by using a simple resolution. I draw inferences by using a simple resolution because I do not believe literally that feet that inputs describe torture you. They torment you because I draw them by using a simple resolution. They torment you.

Not believe literally: I do not believe literally that feet torture you. I do not believe literally that feet that inputs describe and that you control torture you because I believe that they torture you is a personification. They torment you because I draw inferences by using a simple resolution. They torment you because I draw them by using a simple resolution.

Believe is a trope: I believe that feet torturing you is a personification.

Is trope: Feet torturing you is a personification. Feet that inputs describe and that you control torturing you is a personification. They torment you because I draw inferences by using a simple resolution. They torment you because I draw them by using a simple resolution.

Literally not possible: Feet torturing you is literally not possible. Feet that inputs describe and that you control torturing you is literally not possible so I believe that they torture you is a personification. They torment you because I draw inferences by using a simple resolution. They torment you because I draw them by using a simple resolution.

Input: The square owns the house.

Resolution method: Simple resolution, automatically passed on to conventional metonymy resolution, since no initial interpretation was found. Neither the latter method nor literal analysis succeeded in finding an interpretation, thus the fail events were generated (see also Fig. 5.7).

Fail event: I do not believe that a square owns a house. I do not believe that a square owns a house because I do not find a literal interpretation and do not find a figurative interpretation with a conventional resolution.

Other events: I do not find a literal interpretation and do not find a figurative interpretation with a conventional resolution. I do not find a literal interpretation and do not find a figurative interpretation with a conventional resolution so I do not believe that a square owns a house. I do not find a literal interpretation. I do not find a figurative interpretation with a conventional resolution.

Input: The circle owns the house.

Interpretation: Organisations own a house. Organisations own a house because I draw inferences by using a simple resolution.

Draw inferences: I draw inferences by using a simple resolution. I draw inferences by using a simple resolution because I do not believe literally that the circle that inputs describe owns organisation's house.

Not believe literally: I do not believe literally that a circle owns organisation's house. I do not believe literally that the circle that inputs describe owns organisation's house because I believe that it owns it is a metonymy.

Believe is a trope: I believe that a circle owning organisation's house is a metonymy. The circle that inputs describe owning organisation's house is literally not possible so I believe that it owns it is a metonymy.

Is trope: A circle owning organisation's house is a metonymy. The circle that inputs describe owning organisation's house is a metonymy.

Literally not possible: A circle owning organisation's house is literally not possible. The circle that inputs describe owning organisation's house is literally not possible so I believe that it owns it is a metonymy.

Other events: Inputs describe a circle. Inputs describe the circle that owns organisation's house. Inputs use human features in order to describe a circle. Inputs use human features in order to describe the circle that owns organisation's house.

Input: Happiness murdered the baron.

Resolution method: Simple resolution, passed on to conventional metonymies. As in the above example, neither a literal nor a figurative interpretation was found for the input. Therefore it was classified as anomalous and attributed a low degree of belief.

Fail event: I do not believe that a happiness murdered a baron. I do not believe that a happiness murdered a baron because I do not find a literal interpretation and do not find a figurative interpretation with a conventional resolution.

Other events: I do not find a literal interpretation. I do not find a figurative interpretation with a conventional resolution.

A.2 Fixed metonymies

Input: Denise drank the bottle [Fass 1991].

Interpretation: Denise drank a content. Denise drank the content that the container-for-content metonymy that uses a container may describe because I draw inferences.

Draw inferences: I draw inferences. I draw inferences because I do not believe literally that Denise drank a bottle.

Not believe literally: I do not believe literally that Denise drank a bottle. I do not believe literally that Denise drank a bottle because I believe that she drank it is a metonymy.

Believe is a trope: I believe that Denise drinking a bottle is a metonymy. I believe that Denise drinking a bottle is a metonymy because she drank it is literally not possible.

Is trope: Denise drinking a bottle is a metonymy.

Literally not possible: Denise drinking a bottle is literally not possible. Denise drinking a bottle is literally not possible so I believe that she drank it is a metonymy.

Input: Ted played Bach [Fass 1991].

Comment: This example illustrates how different levels of abstraction in the metaphoric knowledge result in a varying degree of depth in the interpretation. Since no knowledge about ‘composers producing music’ was present in the SemNet, the information that ‘Bach is a composer’ did not lead to an instantiation of the generic ‘artist-for-art-form’ metonymical schema to ‘composer-for-music’.

Interpretation: Ted played art. Ted played art that the artist-for-art-form metonymy that uses an artist may describe and that he played because I draw inferences. I draw them because I do not believe literally that he played Bach.

Draw inferences: I draw inferences. I draw inferences because I do not believe literally that Ted played Bach.

Not believe literally: I do not believe literally that Ted played Bach. I do not believe literally that Ted played Bach because I believe that he played him or her is a metonymy.

Believe is a trope: I believe that Ted playing Bach is a metonymy. I believe that Ted playing Bach is a metonymy because he played him or her is literally not possible.

Is trope: Ted playing Bach is a metonymy.

Literally not possible: Ted playing Bach is literally not possible. Ted playing Bach is literally not possible so I believe that he played him or her is a metonymy.

Input: I read Shakespeare.

Comment: This example clearly shows the benefit of relying on the SemNet as a shared knowledge base (KNB). The generic ‘author-for-work’ figurative knowledge is, without additional knowledge-engineering instantiated into ‘poet-for-poetry’ and ‘playwright-for-drama’, since ‘Shakespeare’ is, as far as the SemNet is concerned, an instance of both. If the selectional mechanisms is at work, the most appropriate of the interpretations is selected on the basis of contextual information.

Interpretation: You read poetry. You read poetry that the work-for-author metonymy that uses a versifier may describe because I draw inferences.

You read a drama. You read the drama that the work-for-author metonymy that uses a playwright may describe because I draw inferences.

You read a work. You read the work that the work-for-author metonymy that uses an author may describe because I draw inferences.

Draw inferences: I draw inferences by using a conceptual resolution. I draw inferences by using a conceptual resolution because I do not believe literally that you read Shakespeare.

Not believe literally: I do not believe literally that you read Shakespeare. I do not believe literally that you read Shakespeare because I believe that you read him or her is a metonymy.

Believe is a tropetype: I believe that you reading Shakespeare is a metonymy. I believe that you reading Shakespeare is a metonymy because you read him or her is literally not possible.

Is tropetype: You reading Shakespeare is a metonymy.

Literally not possible: You reading Shakespeare is literally not possible. You reading Shakespeare is literally not possible so I believe that you read him or her is a metonymy.

A.3 Contextual metonymies

Input: A customer entered a restaurant and ordered a sandwich. Now the sandwich is waiting for the bill [Lakoff and Johnson 1980].

Comment: The prototypical chain of figurative reasoning produces seemingly contradictory NLG. Of course the sentence ‘The customer who ordered the sandwich that waits for a bill [...]’ should read ‘The customer who ordered the sandwich that is said to be waiting for a bill [...]’. Although these points do not affect the overall quality of the results and could be remedied, the readability of the NLG would suffer, in our opinion. Since the NLG, especially the long form, also serves debugging purposes, no steps have been taken to improve the quality. For further processing only the short form of the NLG, which is always correct natural language is used.

Interpretation: A customer waits for a bill. The customer who ordered the sandwich that waits for a bill and who waits for a bill waits for a bill because I draw inferences. I draw them because I do not believe literally that it waits for a bill.

Draw inferences: I draw inferences.

Not believe literally: I do not believe literally that a sandwich waits for a bill.

Believe is a trope: I believe that a sandwich waiting for a bill is a metonymy.

Is trope: A sandwich waiting for a bill is a metonymy.

Literally not possible: A sandwich waiting for a bill is literally not possible.

Input: Jake uses a Macintosh. The Macintosh finishes his homework first.

Interpretation: Jake finishes homework. Jake finishes homework because I draw inferences.

Draw inferences: I draw inferences by using a contextual resolution. I draw inferences by using a contextual resolution because I do not believe literally that the Macintosh finishes homework.

Not believe literally: I do not believe literally that the Macintosh finishes homework. I do not believe literally that the Macintosh finishes homework because I believe that it finishes them is a metonymy.

Believe is a trope: I believe that the Macintosh finishing homework is a metonymy. I believe that the Macintosh finishing homework is a metonymy because it finishes them is literally not possible.

Is trope: The Macintosh finishing homework is a metonymy.

Literally not possible: The Macintosh finishing homework is literally not possible. The Macintosh finishing homework is literally not possible so I believe that it finishes them is a metonymy.

A.4 Conventional figures

Input: I killed emacs [Martin 1990].

Interpretation: You terminated emacs. You terminated emacs that you killed because I draw inferences.

Draw inferences: I draw inferences. I draw inferences by using a conceptual resolution because I do not believe literally that you killed emacs that you terminated. I draw them because I do not believe literally that you killed it.

Not believe literally: I do not believe literally that you killed emacs. I do not believe literally that you killed emacs that you terminated because I believe that you killed it is a metaphor. I draw inferences because I do not believe literally that you killed it.

Believe is a trope: I believe that you killing emacs is a metaphor. I believe that you killing emacs that you terminated is a metaphor because you killed it is literally not possible. I draw inferences because I do not believe literally that you killed it.

Is trope: You killing emacs is a metaphor. You killing emacs that you terminated is a metaphor. I draw inferences because I do not believe literally that you killed it.

Literally not possible: You killing emacs is literally not possible. You killing emacs that you terminated is literally not possible so I believe that you killed it is a metaphor. I draw inferences because I do not believe literally that you killed it.

Input: Prices are soaring [Carbonell 1982].

Interpretation: Prices increase. Prices increase because I draw inferences.

Draw inferences: I draw inferences by using a conceptual resolution because I do not believe literally that prices soar. I draw inferences by using a conceptual resolution.

Not believe literally: I do not believe literally that prices soar. I do not believe literally that prices soar because I believe that they soar is a metaphor.

Believe is a trope: I believe that prices soaring is a metaphor. I believe that prices soaring is a metaphor because they soar is literally not possible.

Is trope: Prices soaring is a metaphor.

Literally not possible: Prices soaring is literally not possible. Prices soaring is literally not possible so I believe that they soar is a metaphor.

Input: Britain entered the EU [Wilks 1978, Martin 1990].

Interpretation: Britain joined the EU. Britain joined the EU because I draw inferences.

Draw inferences: I draw inferences. I draw inferences because I do not believe literally that Britain entered the EU.

Not believe literally: I do not believe literally that Britain entered the EU. I do not believe literally that Britain entered the EU because I believe that it entered it is a metaphor.

Believe is a trope: I believe that Britain entering the EU is a metaphor. I believe that Britain entering the EU is a metaphor because it entered it is literally not possible.

Is trope: Britain entering the EU is a metaphor.

Literally not possible: Britain entering the EU is literally not possible. Britain entering the EU is literally not possible so I believe that it entered it is a metaphor.

Input: I entered LISP [Martin 1990].

Interpretation: You started LISP. You started LISP because I draw inferences.

Draw inferences: I draw inferences. I draw inferences because I do not believe literally that you entered LISP.

Not believe literally: I do not believe literally that you entered LISP. I do not believe literally that you entered LISP because I believe that you entered it is a metaphor.

Believe is a tropetype: I believe that you entering LISP is a metaphor. I believe that you entering LISP is a metaphor because you entered it is literally not possible.

Is tropetype: You entering LISP is a metaphor.

Literally not possible: You entering LISP is literally not possible. You entering LISP is literally not possible so I believe that you entered it is a metaphor.

A.5 Idioms

Input: Jake is fighting the red tape in the university.

Interpretation: Jake fights a bureaucracy. Jake fights the bureaucracy that the idiom that uses red tape may denote because I draw inferences.

Draw inferences: I draw inferences. I draw inferences because I do not believe literally that Jake fights the red tape that he fights.

Not believe literally: I do not believe literally that Jake fights a red tape. I do not believe literally that Jake fights the red tape that he fights because I believe that he fights it is an idiom.

Believe is a trope: I believe that Jake fighting a red tape is an idiom. I believe that Jake fighting the red tape that he fights is an idiom because he fights it is unlikely literally.

Is trope: Jake fighting a red tape is an idiom. Jake fighting the red tape that he fights is an idiom.

Literally not possible: Jake fighting a red tape is unlikely literally. Jake fighting the red tape that he fights is unlikely literally so I believe that he fights it is an idiom.

Input: Jake kicked the bucket.

Interpretation: Jake died. Jake died because I draw inferences.

Draw inferences: I draw inferences. I draw inferences because I do not believe literally that Jake kicked the bucket that he kicked.

Not believe literally: I do not believe literally that Jake kicked a bucket. I do not believe literally that Jake kicked the bucket that he kicked because I believe that he kicked it is an idiom.

Believe is a trope: I believe that Jake kicking a bucket is an idiom. I believe that Jake kicking the bucket that he kicked is an idiom because he kicked it is unlikely literally.

Is trope: Jake kicking a bucket is an idiom. Jake kicking the bucket that he kicked is an idiom.

Literally not possible: Jake kicking a bucket is unlikely literally. Jake kicking the bucket that he kicked is unlikely literally so I believe that he kicked it is an idiom.

A.6 Comparison statements

Input: Goebbels is a rat [Cooper 1986].

Comment: This example shows how well the non-literal analysis is integrated into the rest of the system. Unlike with other architectures, there is no ‘competition’ between the literal and non-literal components. In this case, semantics correctly but unexpectedly chose the figurative reading of ‘rat’, i.e. ‘informer’ in the course of normal processing. Since the comparison statement mechanism operates independently from the literal analysis, that is, it is not invoked on a failed literal interpretation, the input was checked again. It was found to be figurative and correct and therefore accepted.

Interpretation: Goebbels is metaphorically an informer.

Believe is a tropetype: I believe that Goebbels being a rat is a figurative comparison.

Is tropetype: Goebbels being a rat is a figurative comparison.

Input: Goebbels is a rat [Cooper 1986].

Comment: In this case, the additional figurative word sense of ‘rat’ that was correctly chosen by semantics in the example above had been deleted from the KNB to allow for testing of the bare comparison statement mechanism. Thus this example can be seen as producing the well-established figurative meaning of ‘rat’ in the meaning of ‘informer’, as was planned, on the fly. The brackets were added by hand to increase the readability. The text outside the brackets on its own constitutes the short NLG, the text outside and inside the brackets constitutes the long NLG.

Interpretation: Goebbels is metaphorically poisonous (because I believe that Goebbels being a rat is a figurative comparison). Goebbels is metaphorically unsafe (because I believe that Goebbels being a rat is a figurative comparison). Goebbels is metaphorically vicious (because I believe that Goebbels being a rat is a figurative comparison).

Believe is a trope: I believe that Goebbels being a rat is a figurative comparison.

Is trope: Goebbels being a rat is a figurative comparison.

Input: John is a fox [Carbonell 1982].

Comment: In this example, the good integration and mutual support of the literal and non-literal analysis can be seen, too. Again, the figurative reading of a word, here 'fox' was correctly chosen by the literal analysis. However, this interpretation had no 'side effects' in that it elaborated on what characterises 'John'. The figurative analysis component, however, transferred additional attributes of 'John' on the basis of this figurative comparison, namely being a 'trickster', 'slyboots' and a 'dodger'.

Interpretation: John is metaphorically a trickster. John is metaphorically a trickster because I believe that John being a fox is a figurative comparison.

John is metaphorically a slyboots. John is metaphorically a slyboots because I believe that John being a fox is a figurative comparison.

John is metaphorically a dodger. John is metaphorically a dodger because I believe that John being a fox is a figurative comparison.

Believe is a trope: I believe that John being a fox is a figurative comparison. I believe that John being a fox is a figurative comparison so John is metaphorically a trickster.

Is trope: John being a fox is a figurative comparison.

Input: Love is a journey [Lakoff 1993].

Comment: In this example the comparison statement algorithm was used to (meta-) interpret a conventional metaphor schema, namely LONG-TERM PURPOSEFUL ACTIVITY IS A JOURNEY, sub-schema LOVE IS A JOURNEY. It is important to note that the directionality is preserved. While a lover is metaphorically a traveller, a traveller does not become metaphorically a lover. The interpretation shows which knowledge was available in both domains and could be mapped. It includes the agent (lover, traveller), alteration (the system seems to interpret a relationship as marking a state change, in parallel to the change of location occurring in the context of travel) and impediments (problems, obstacles/delays). This result points into the direction that a computational resolution of comparison statements on the basis of conventional figures might be a feasible undertaking (see also Sect. 6.5).

To avoid repetitious NLG listings, brackets have been added. The text in brackets together with the text outside the brackets constitutes the long NLG, the text outside the brackets on its own constitutes the short NLG.

Interpretation: (I believe that love being a journey is a figurative comparison so) love is metaphorically a change. (I believe that love being a journey is a figurative comparison so) a goal is metaphorically a destination. (I believe that love being a journey is a figurative comparison so) a problem is metaphorically an obstacle. (I believe that love being a journey is a figurative comparison so) a problem is metaphorically a delay. (I believe that love being a journey is a figurative comparison so) a lover is metaphorically a traveller.

A lover is metaphorically a traveller (because I believe that love being a journey is a figurative comparison). A problem is metaphorically a delay (because I believe that love being a journey is a figurative comparison). A problem is metaphorically an obstacle (because I believe that love being a journey is a figurative comparison). A goal is metaphorically a destination. (because I believe that love being a journey is a figurative comparison). Love is metaphorically a change (because I believe that love being a journey is a figurative comparison).

Love is metaphorically a motion (because I believe that love being a journey is a figurative comparison). Love is metaphorically a travelling (because I believe that love being a journey is a figurative comparison).

Believe is a trope type: I believe that love being a journey is a figurative com-

parison.

Is trope: Love being a journey is a figurative comparison.

Other events: The destination that a goal is metaphorically. The traveller that a lover is metaphorically. The delay that a problem is metaphorically. The obstacle that a problem is metaphorically.

Input: Butchers are surgeons [Veale 1995].

Comment: In this example of a comparison statement, a feature and an attribute transfer takes place (c.f. Fig. 5.7). On the basis of the results of triangulation, the features ‘skill’ and ‘wealth’ are transferred. Since triangulation supports the overall comparison, the metaphorical is-a relation between the input concepts is reinforced. The transfer of attributes, again on the basis of triangulation, allows cross-domain correspondences between tools used and acts performed by the input concepts.

The text in brackets, together with the text outside the brackets constitutes the long NLG, the text outside the brackets on its own constitutes the short NLG.

Interpretation: Butchers are metaphorically skilled. Butchers are metaphorically rich. Butchers are metaphorically surgeons.

A knife is metaphorically a scalpel (because I believe that butchers being surgeons is a figurative comparison.) Slaughter is metaphorically surgery (because I believe that butchers being surgeons is a figurative comparison.)

Believe is a trope: I believe that butchers being surgeons is a figurative comparison.

Is trope: Butchers being surgeons is a figurative comparison.

Input: Encyclopaedias are gold mines [Ortony 1979].

Interpretation: An information is metaphorically gold. An information is metaphorically gold because I believe that encyclopaedias being gold mines is a figurative comparison.

Encyclopaedias are metaphorically mines. Encyclopaedias are metaphorically mines because I believe that encyclopaedias being gold mines is a figurative comparison.

Encyclopaedias are metaphorically big. Encyclopaedias are metaphorically big because I believe that encyclopaedias being gold mines is a figurative comparison.

Believe is a trope: I believe that encyclopaedias being gold mines is a figurative comparison. I believe that encyclopaedias being gold mines is a figurative comparison so encyclopaedias are metaphorically mines.

Is trope: Encyclopaedias being gold mines is a figurative comparison.

Other events: The gold that an information is metaphorically.

Appendix B

Algorithmic and representational details

This section describes algorithmic and representational details and considerations, in particular the representation of figurative knowledge in the form of natural language (NL), the choice of using the SemNet as a common data structure and details of the algorithms for the resolution of individual types of figures as well as the important auxiliary methods used.

B.1 Natural language representation of figurative knowledge

The knowledge about the form of figures and the way in which they combine concepts is stored in the SemNet in the form of events (see also Sect. 2.2.1). The figurative knowledge and its organisation have to satisfy certain requirements.

This knowledge should, quite obviously, enable the system to make use of it while resolving input, that is, provide the necessary information for a non-literal analysis on the basis of the knowledge-based approach. Carbonell stated that metaphors use concepts so that “X is used to mean Y in context Z” [Carbonell 1982]. When applied to our question, figurative knowledge thus has to express the concepts X, Y and, where applicable, Z. Moreover, it should represent the organisation of figures, since it has been shown convincingly that a hierarchical organisation of figures not only has a cognitive aspect [Lakoff 1993, Lakoff *et al.* 1995], but also

helps the computational analysis of figurative input (c.f. the discussion of the MIDAS system in Sect. 3.2.5.1). Finally, the non-literal analysis can be more efficient, flexible and general if the figurative knowledge also contains information on the type of figure.

These requirements are all met by the current representation of figurative knowledge. The actual representation consists of events of the generic form ‘A figure of type V, namely figure U, makes use of concept X to express concept Y’. This form, which is remarkably close to Carbonell’s dictum is powerful enough to represent the figurative knowledge needed for handling all types of knowledge-based tropes. The part that expresses the type and ‘name’ of the figure is optional, but greatly helps post-processing analysis and serves as a basis for other tasks, such as stylistic analysis or empirical analysis of the frequency of figures. The fact that the special, figurative concept relation is only valid in a certain context is supplied by the figurative resolution algorithm. Either the lack of a valid literal interpretation, but the existence of an acceptable figurative interpretation, or the analysis of the context (meaning the actual context of the utterance or the conceptual context of the input concepts) supply the information necessary to reliably apply the figurative schema under consideration to the input. The most important part of the figurative knowledge is the ‘X means Y (on the basis of figure U)’ concept relation. It is stored in the SemNet in exactly the same way other knowledge is stored, that means by representing the concepts involved as nodes of the SemNet and their relation as links between the nodes.

This enables the knowledge engineer to input, check, modify and output figurative knowledge using the existing NL interfaces. More importantly, the uniform means of representation and the sharing of information between the figurative and other types of knowledge already contained in the SemNet means that other parts of the system can access figurative knowledge, while the entire knowledge contained in the SemNet can be used by the figurative sub-system. Another extremely important factor that leads to the choice of the existing SemNet structure as the basis for the representation of figurative knowledge is that with a shared knowledge base (KNB), no discrepancies between types of knowledge can arise. This is a plausible risk because the processing of input extends the SemNet continually with new concepts and relations.

Before the concluding remarks, some examples of figurative knowledge are presented. The examples are the input typed in a query session with the LOLITA

system and which added figurative knowledge to the SemNet.

An artist-for-art-form metonymy uses an artist to describe art.

A process-as-living-thing metaphor uses an animal to denote a process.

A work-for-author metonymy uses an author to describe a work.

A capital-for-government metonymy uses a capital to describe the government.

A container-for-content metonymy uses a container to describe the content.

The last piece of figurative knowledge was the input that enabled the non-literal analysis to resolve the ‘Denise drank the bottle’ example from Appendix A.2. The generality and naturality of the figurative knowledge should be noted. What is expressed in this piece of knowledge is the simple linguistic fact that constitutes and describes a particular metonymy. The applicability and instantiation of the knowledge is performed by the non-literal analysis.

It is obvious that the figurative knowledge could be refined with respect to its hierarchical structure. For example, one might want to include ‘a container-for-content metonymy uses a bottle to describe the contents of the bottle’ in order to add to the knowledge about metonymies. While it is possible to do so, it is not necessary. Concept extension is capable of matching input like ‘Denise drank the bottle’ to the generic schema on the basis of the knowledge already contained within the SemNet. When considering the example ‘Ted played Bach’ that was resolved to the interpretation ‘Ted played art’ with the help of the figurative knowledge ‘an artist-for-art-form metonymy uses an artist to describe art.’ the case for including more specific knowledge is strengthened. It is true that figurative knowledge stating that ‘an artist-for-art-form metonymy uses a composer to describe music’ would refine the interpretation. But this refinement could be achieved by other means as well. If there were knowledge in the SemNet that stated that ‘composers produce music’, the method of masking (c.f. Sect. 5.2.4) could be used to derive a refinement of the interpretation. The huge amount of knowledge that is already present in the SemNet and its structure play another important role for the figurative knowledge. They give the figurative knowledge its hierarchical order, as can be seen in the example ‘I read Shakespeare’ from Appendix A.2. While the interpretation ‘I read a work by Shakespeare’ is the most generic, the interpretations ‘I read poetry’ and ‘I read a drama’ are more specific, since the SemNet taxonomy states a sub-type relation between versifier and playwright on the one hand and

author on the other. In order to prepare the ground for stylistic and empirical analyses, the two events ‘a metonymy uses a source to describe a target’ and ‘a metaphor uses a source to describe a target’ are put into the SemNet. Once a metaphor or metonymy has been resolved, the actual concept from the input and the interpretation are marked as source and target, respectively, on the basis of their relation being a figurative one.

A final point concludes the discussion of the representation of figurative knowledge. It would clearly be undesirable if the non-literal analysis were to try and resolve the NL input of figurative knowledge. This is the case when, for example, existing figurative knowledge is refined. To avoid this happening, a function set early into the resolution algorithm for knowledge-based tropes checks whether the input is in fact meta-language, describing figurative knowledge, in which case no further steps into the direction of non-literal analysis are taken.

B.2 Algorithmic details

This section presents details of the algorithm for the individual types of tropes. The steps taken during processing are discussed with reference to Fig. 2.2. The initial stage which is similar for all types, is the triggering of the non-literal analysis. Non-literal analysis of the input is performed either after an explicit request (because, for example, the literal analysis did not yield a satisfying result), after a constraint violation (this is optional and not the main *modus operandi*) or, as default, in the normal course of processing input.

B.2.1 Simple tropes

When input is analysed using the simple resolution method, it is first checked against generic figurative knowledge. The function ‘determine_trope’ is called from within the normal pragmatic analysis process and functions as an interface between the other parts of the system’s analysis and the non-literal analysis. If the input is found not to be a figure, the function returns the appropriate pragmatic analysis result. If, however, the input is a figure, the pragmatic analysis result will be of the type ‘figure’ and an attempt is made to interpret the input as a figure.

To this end, the type-specific concept extension delivers information that al-

allows an examination of aspects of the input concepts, such as their place within the ontology of the system's KNB. With the extended input that constitutes the following step, the metaphoric knowledge consultation checks whether this input satisfies the form of generic figures. If the input is neither anomalous nor literal but figurative, that is if the metaphoric knowledge consultation showed that it is a simple figure, the function 'interpret_trope' is called. At this stage, the figurative analysis could stop if the task were to establish that the input was anomalous or either literal or figurative.

If the input could not be classified as a simple trope, processing branches to the failure handling for simple tropes. This consists, as mentioned above, in returning a pragmatic analysis indicating that the input is not acceptable as such. If the literal analysis of the input is unsuccessful, too, the current default for handling this case comes into action. It consists of attributing a low degree of belief to the input. Should the simple resolution method not be the only figurative resolution method in operation, the input is passed on to the next higher level of resolution instead of the failure handling.

The function 'interpret_trope' takes the extended input and the type of simple trope the input was found to constitute and returns the system's interpretation of the input in the light of it being figurative. For simple figures, the interpretation can be regarded as finding an abstraction from the actual input that yields a plausible relation between the input concepts. In the prototype, single interpretations are not re-evaluated individually. Instead, all possible interpretations are collected at once and the least generic is chosen as the result. Although a more refined interaction between modules might be desirable, this was found to be a fully sufficient, fast and robust approach.

Within the function 'interpret_trope', processing also branches into the figurative reasoning for simple tropes. This means that on the basis of the triggering conditions (c.f. Fig. 5.12) of the input having a valid figurative reading (it may or may not have a literal reading, too) and of being categorised as a certain type of figure, the information necessary for integrating the results of the figurative analysis into the SemNet structure is created (see also Appendix B.2.7).

B.2.2 Fixed metonymies

Processing input using the fixed metonymy method is similar to the simple resolution method. But since no constraint violation detection can guarantee discovery of all metonymies and only of metonymies in the input, processing cannot be triggered by a constraint violation. Instead, input is passed to the function ‘`determine_fix_metonymy`’ from within the normal pragmatic analysis whenever the fixed resolution is switched on.

The type-specific concept extension is considerably more elaborate for fixed metonymies than it is for simple tropes. The input concepts are extended to their ontological relations with the rest of the SemNet, e.g. along the sub- and super-type lines, as well as to similar and conceptually related concepts. The context sensitive and dynamic nature of the SemNet KNB ensures that relevant concepts are taken into consideration, even if they are related to the input solely via previously processed input.

The extended input is then passed to the metaphoric knowledge consultation function, which tries to match it with stored figurative knowledge. A large portion of the applicable code is shared between the simple and the fixed resolution, since simple tropes are a sub-set of fixed metonymies. Thus, simple tropes can be resolved using the fixed method. If input is passed on from the simple resolution to the fixed method, it is, of course, not checked again with respect to simple tropes. If the knowledge consultation shows that the input cannot be classified as a simple trope, but matches with knowledge about fixed metonymies, the particular interpretation function for fixed metonymies, ‘`makeinter_comp`’, is called. It takes the extended input and the metaphoric knowledge that matched it and returns all valid interpretations of the input as a metonymy.

The name ‘`makeinter_comp`’ stems from the early stages of development, when the simple tropes were contrasted with more ‘complex’ ones. A failed simple resolution will temporarily label input that was not acceptable as ‘complex trope’. Pragmatics then decides whether this input is rejected or passed on to a higher resolution level. If it is passed on to the fixed resolution, it will not be processed using the simple resolution, because it is already marked as a (possibly) complex trope. If the fixed metonymy handling does not find a figurative reading, it is finally marked as not being acceptable as figurative input. Should it also fail the literal analysis, it is handled using the default mechanism of being assigned a low

degree of belief.

When a positive match with figurative knowledge is found and the task finds the interpretation of figurative input, ‘makeinter_comp’ returns a list of at least one valid metonymical interpretations for the input. Otherwise, processing can stop because it has been established that the input is acceptable and of the type ‘fixed metonymy’. If there is more than one interpretation, the most appropriate one is selected using the function ‘pick_one’. This function is implemented only in a prototypical fashion and can select interpretations on the basis of simple contextual information. Its default is to select the most specific interpretation available. The others are by no means discarded, selection should rather be seen as ranking the interpretations in the order of their importance. Furthermore, the input concept and its metonymical interpretation are connected in the SemNet by the auxiliary function ‘relate_them’. It is obvious that there is a relation between the two concepts on the basis of the metonymical utterance. The exact nature of this relation, however, cannot be named easily, as there is no possibility of storing it alongside the figurative knowledge. Metonymical relations are too divergent, therefore the relation is a generic one and can only be examined by considering the type of metonymy at the same time.

The generic figurative reasoning is triggered from within the ‘makeinter_comp’ function and explicates the coherence of the events constructed in the course of the fixed resolution (see also Appendix B.2.7).

B.2.3 Contextual metonymies

The algorithm for resolving contextual metonymies is, by and large, similar to the algorithm for fixed metonymies. The main difference is the context consultation which augments the concept extension performed for the fixed metonymies.

After the ‘determine_contextual’ function is called from within the normal pragmatic analysis, concept extension similar to that used for fixed metonymies is performed. The extended input is then checked against contextual information that is compiled in situ by the figurative analysis. Thus no superfluous data are passed around and the contextual information is guaranteed to be the most recent. The context consultation checks the extended input against the context to find possible context referents for concepts (and their extensions) that appear in the input. There are generally a number of contextual references to the collection of

concepts which lie in the extension of the input. These are checked for possible metonymical concept relations with the relevant concepts from the input, whereby the metonymical relations are those defined by the fixed metonymies. It should be noted that in the prototype, the metonymical relations which are checked are generic ones. Eventually, the contextual metonymy branch of the figurative analysis should deal with the context extension and matching, passing on its results to the fixed metonymy resolution. Thus, all the figurative knowledge available to the fixed method would become available to the contextual resolution, too. It is obvious that such an approach is feasible. However, this project took a broad coverage and feasibility position, therefore such an integration issue took no priority over the implementation of the basic functionality.

The function ‘makeinter_cont’ which is similar to the function ‘makeinter_comp’ takes the context referent, the input concept and the metonymical relation in which the two stand and returns the interpretation of the contextual metonymy. From an overly simplistic point of view, this interpretation could be regarded as a sort of anaphora resolution. The point to note is that there is no simple replacement of a concept from the input with another concept that either has already occurred in the text or bears some other relation to the input and is denoted by the relevant input concept. Instead, the possible anaphoric referents (which, incidentally need not appear in previous input but can be traced on the basis of concept extension) are taken from a much wider field, namely, that of conceptually and metonymically related concepts of the input.

If one or more of these is found to stand in a plausible metonymical relation to one of the input concepts, i.e. if the interpretation is performed as opposed to stating that the input is acceptable as a contextual metonymy and not anomalous, the generic chain of figurative reasoning is activated to integrate the information produced by the contextual resolution into the SemNet for future use (see also Appendix B.2.7).

B.2.4 Conventional metaphors

The conventional metaphor resolution allows it to be determined whether input is of the type conventional metaphor (see Sect. 5.2.2) and to interpret it accordingly. Additionally, it allows treatment of lower types of simple and fixed metonymies. The fact that, as Lakoff and Johnson observed, metaphor “is principally a way of

conceiving of one thing in terms of another, and its primary function is understanding”, whereas metonymy “has primarily a referential function, that is, it allows us to use one entity to *stand for* another” [Lakoff and Johnson 1980] indicates that the algorithm for conventional figures has to differ notably from the algorithm for metonymies. Yet again it has to offer a functionality that is comparable to the resolution algorithm for metonymies, in order to be able to resolve them as well.

The most prominent differences to the other algorithms are the necessarily more complex concept extension, the type determination, and the ‘preferred schema’ constraint.

Processing of input is started by activating the conventional resolution. If it is the only figurative analysis method selected, all input will be analysed with respect to its being a conventional figure or metonymy. If lower levels are activated and do not find a figurative interpretation for a portion of input, the input is passed up to the conventional method in the current configuration. The concept extension is not limited to ontological and taxonomic relations. Concepts from the SemNet that stand in a relation to concepts from the input via, e.g., episodic knowledge or in a contextual connection are collected as well as the usual similar, sub- and super-type concepts.

The second stage of metaphoric knowledge consultation, however, does not simply check whether a concept from the extension matches metaphoric knowledge of the class ‘conventional figure’. If any concept from the input extension appears in figurative knowledge about any type of figure stored in the SemNet, the function will record this appearance and pass the input on, so that a resolution according to the type of figure indicated can be performed.

If the input is found to be of a lower type of figure, the type is extracted from the SemNet and passed, together with the input, to the interpretation function ‘makeinter_conv’. If the metaphoric knowledge match occurred with information on a conventional metaphor, the last outstanding feature of this resolution method is activated, namely, the ‘preferred schema’ check. Since metaphors involve two conceptual domains but the input does not necessarily contain apparent concepts from the two domains, it is necessary to establish whether the two domains are present indirectly. For example, it would be absurd to classify input as an instance of the LOVE AS JOURNEY metaphor simply because it contains a concept from the domain of relationships. Even if a metaphorical reading were intended, the existence of one concept is not a sound basis for such a classification. Therefore, a

method similar to concept extension is used to establish whether or not a connection exists between the input and the second, obligatory domain. This is also necessary because a number of conventional metaphorical schemata share the same concepts. Thus the second domain is needed to establish the kind of figurative schema that is applicable to the input. For example, in the sentence ‘I killed emacs’, the ‘killing’ domain is explicitly present. In order to reliably classify the input as an instance of the MACHINES ARE PEOPLE and CESSATION OF FUNCTION IS KILLING metaphor [Lakoff *et al.* 1995], the second domain has to be detected. To this end, the input concepts are extended and this extension checked for compatibility with the second domain. In the example, ‘emacs’, the name of a software, will quickly result in a match with the machines domain and thus the applicability of the MACHINES ARE PEOPLE metaphor is guaranteed.

B.2.5 Idioms

The resolution of idioms cannot reliably and exclusively be triggered by an anomaly detection mechanism. A failed literal interpretation, however, is at least an indication that something out of the ordinary is present in the input. Therefore, it is possible to activate the resolution of idioms on a constraint violation. The default, however, is to pass all input through the ‘determine.idiom’ function, whose task it is to establish whether the input contains a lexical or a compound idiom. Since the conceptually organised structure of the system’s KNB does not easily lend itself to the listing of additional word senses for phrases (whereas lexemes can be given an extra, idiomatic word sense), the concept of compound idioms was introduced. They consist of more than one lexical item and are classified as compound idioms on the basis of the idiom resolution process. Thus, the algorithm for the detection and interpretation of idioms can be seen as effectively augmenting the SemNet with a structure for the registration of complex idiomatic expressions.

The code for concept extension for idioms is comparable to the concept extension that takes place in the case of fixed metonymies. The largest portion of the code is shared.

The metaphoric knowledge consultation differs from the analogous procedure for other types. As mentioned above, no syntactic structures above the level of lexemes can be recorded in the SemNet at the moment. Moreover, the reliable detection of idioms requires that the input conforms to a more or less fixed form; this is not

necessarily a fixed surface form (see also the discussion of lexical flexibility and semantic productivity in Sects. 5.2.2 and 5.3.6), but a fixed conceptual form. This form, however, cannot be covered in one knowledge unit. For example, the phrase ‘to kick the bucket’ cannot be listed in the SemNet as a concept, listing ‘to die’ as its meaning. To solve this problem, the extended concepts are matched individually against metaphoric knowledge, and only if all the required items that make up an individual idiom appear in the input in the right relations will it be classified as an instance of an idiomatic expression. This means that, while a generic event like ‘somebody kicking the bucket’ can be put into the SemNet and described as an idiom, every piece of input has to be analysed in sections to see if it conforms to the requirements of being classifiable as an idiom. This procedure has the added advantage of being efficient, since processing can be stopped if the algorithm fails to find a required item in the input. In this case, the input is evidently not an idiom.

If the metaphoric knowledge consultation yields a positive result, it will also have established whether the input is a lexical or compound idiom. The degree of complexity of the interpretation depends on this criterion, since from a simplistic point of view, lexical idioms can be seen as substitutes for the concept or surface form they denote, whereas the surface form of the interpretation of compound idioms can differ significantly from the original input. Thus, more care has to be taken when building the interpretation event for a compound idiom. Fortunately, so far no problems have occurred because the interpretation is not built using surface level linguistic items but on a deep, conceptual level. The generator then takes care of the correct realisation of the deep structure in natural language.

The chain of idiomatic figurative reasoning prompted by the interpretation function ‘makeinter_id’ differs from the generic figurative reasoning, too. Context plays an important role in the interpretation and detection of idioms, but in most cases, the input could, especially in the absence of contextual information indicating otherwise, be taken literally. Therefore, it is not possible to establish that the idea expressed in the input is ‘literally impossible’. Since we do not rely on a literal analysis to take place before the figurative analysis, we classify possibly idiomatic input as being ‘literally unlikely’. It would help to establish a protocol whereby a failed literal analysis signals the subsequent figurative analysis that the likelihood of the input being either figurative or anomalous is very high. But since the design of this solution was led by the criteria of openness and flexibility, no such rigid

model of information exchange was chosen.

B.2.6 Comparison statements

Comparison statements require the most thorough treatment of all types of figures. While the algorithm for the resolution of the other types is more or less analogous across the type boundaries, the algorithm for comparison statements is distinct from the knowledge-based procedure used elsewhere.

The initial stage consists in knowledge consultation, a form of concept extension. But in addition to an extensive concept extension (parts of the code are shared with the concept extension for metonymies), episodic knowledge and other information related to the input concepts are gathered as well. This proves necessary because it forms not only the basis for the interpretation but also for the recognition of comparison statements.

After the comprehensive knowledge consultation, initial checks are made to establish whether the input is a comparison statement. The initial checks consist of an analysis of the form of the input (e.g. it has to be a generic ‘is-a’ relation, which can take various surface forms), and of determining whether (sufficient) information on both concepts is available in the SemNet (see also Fig. 5.11) and whether a relation between the input concepts has already been established before (e.g. if a sub/super-type relation already exists). This preliminary analysis classifies the input as being either a literal comparison or a figurative comparison, or fails to classify it without further checks. This arrangement enables expensive, thorough computation to be spared if the concept relation can be established more easily. Should the initial checks fail to result in a classification, processing branches neither to the interpretation function for figurative comparisons, ‘makeinter_figurative’, nor to the interpretation function for literal comparisons, namely, ‘makeinter_literal’, but to the function responsible for further elaboration on the input concept’s relation, ‘makeinter_unknown’.

In the latter function, the conceptual proximity of the concepts (or salience imbalance) is computed. This measure helps to establish whether a comparison between the concepts can possibly be literal or not. A high conceptual proximity is taken to indicate a literal comparison, a low conceptual proximity suggests a figurative comparison. In order to bolster the result of the conceptual proximity computation, the domains of the input concepts (in other words, concepts related

to the input concepts) are compared using the triangulation method. If the results of the extended checks yield no result indicating the nature of the comparison, the criterion of robustness makes it necessary to make both the figurative and literal interpretation. Since the figurative reasoning allows revision of steps taken by the figurative analysis, the incorrect analysis can be retracted if additional information invalidating it becomes available.

If the outcome of the analysis has allowed a positive classification of the input, the according interpretation function, ‘makeinter_figurative’ or ‘makeinter_literal’, is called. The simpler case of a literal interpretation treats the input as a class inclusion or expressing a sub/super-type relation. This is the only valid literal reading at that stage, because other types of comparison such as identity have been caught during the initial checks. If attributes of the vehicle concept have been gathered during the knowledge consultation stage, they are transferred to the tenor concept by the ‘transfer_attributes’ function, which is called from within ‘makeinter_literal’.

When a figurative comparison is interpreted, the additional transfer of features from the vehicle to the tenor takes place. In order to find the applicable features, the information collected during the knowledge consultation is again evaluated using the triangulation method. Features that pass the evaluation are transferred and attributed ‘metaphorically’ to the tenor concept. Subsequently, the transfer of attributes takes place. The availability of information clearly influences the performance and reliability of the algorithm. It should be noted that there are cases where a positive classification can be made without knowledge about the concept’s features and attributes, i.e. where the knowledge consultation was not very productive but other criteria allowed a classification. In this case, there will be, of course, no or very little feature and attribute transfer.

After the two branches of literal and figurative interpretation have joined at the shared code of the ‘transfer_features’ function, the type-specific chain of comparison reasoning completes the processing of comparison statements.

B.2.7 Figurative reasoning

This section gives details on the ‘reasoning’ which is part of the resolution of figurative input and internally documents the decisions made by the system and the steps taken in the course of the resolution. For a graphic representation of the

reasoning, see Fig. 5.12.

It should be noted that in the long natural language generation (NLG) of the events pertaining to the resolution of figurative input, the viewpoint is temporally located beyond completion of the resolution. Therefore, the interpretation appears in ‘intermediate’ events, although, of course, it is linked to those events in the SemNet at a latter stage. When using the long NLG as a debugging tool, the temporal succession of the building of events cannot be traced; here the short NLG gives a somewhat clearer picture. Nevertheless, long NLG is extremely helpful in showing how the events are connected and how the information in the SemNet came about.

The chain of reasoning is only prototypical, especially with regard to the surface form of its NLG. It is somewhat similar for all types of tropes and should be refined, not only with respect to the needs of the various types of figures and sub-cases, but also to include more detailed information on how, for example, the figurative analysis was activated.

This refinement clearly was not a goal of this project. The chain of figurative reasoning itself is more of an auxiliary method and by-product of the main objective. The new approach of storing intermediate results in the SemNet, however, was something that was discovered as being very useful in the course of this project and a technical report on this subject is in preparation.

Since knowledge does not exist in isolation in the system’s KNB, it is necessary to integrate the results and intermediate steps of the figurative analysis into the existing SemNet structure. This task is performed by the figurative reasoning. It states the cause and effects of the figurative analysis and ties the interpretation to a source. In doing so, individual steps or the whole analysis can be traced back and, if necessary, be retracted.

Figurative reasoning consists, in the case of a successful resolution, of the trigger condition, the ‘literally not possible’ event, which can take various forms, the reinforcing condition of the algorithm establishing that the input is a figure of a certain type and the actual chain of reasoning, culminating in the interpretation. On the basis of the verification that the input is of a particular type of figure (the ‘is tropetype’ event), this information is tied into the SemNet by generating the ‘believe is a tropetype’ event. This is necessary for two reasons, firstly because there has to be a source for the knowledge that the input is a figure, secondly because

factual information cannot be represented in isolation in the LOLITA KNB. From an overly simplistic point of view, all the information contained in the SemNet can be described as being the world seen through the eyes of LOLITA, the agent using the available information. Thus, there is a need to attribute a degree of belief to all new pieces of knowledge.

In the wake of the ‘believe is a trope’ event, some other manifestations of the belief system with respect to the resolution of the input have to be taken into account. Generally speaking, the system cannot believe the input literally if it assumes that it is a figure. Of course the possible cases arising in this context have to be carefully examined. For example, it is possible that input is plausible both literally and figuratively; instances of the text type ‘humour’ often are based on this ambivalence. Likewise, it should be considered whether a literal analysis took place before the figurative one and whether the outcome was positive or not. But as mentioned before, this is by no means an essential elaboration of the figurative reasoning, merely a refinement based on the results of this project.

The next event constructed in the chain of figurative reasoning is the ‘draw inferences’ event. Since the input is not believed literally, a different interpretation has to be made. This is meant by ‘drawing inferences’, a term that was chosen at the early stage of this project. These figurative inferences are drawn using a specific method, namely, the particular resolution algorithm at hand. This fact is expressed in the ‘draw inferences’ event by simply stating the name of the part of the figurative resolution module that led to the interpretation. The events described are not simply generated and put into the SemNet, they are connected in a fashion that lends itself to partial revision (i.e. the causal structure is represented properly) and conforms to the requirements of the SemNet.

Simple figurative reasoning

The simple reasoning is specific to the resolution algorithm for simple tropes. Although most of the reasoning code is shared between the different methods, each method has particular characteristics. The starting point for the simple chain of reasoning is either the fact that the input is ‘literally not possible’, that is a failed literal interpretation occurred, or the fact that a figurative analysis yielded the result that the input has a valid figurative interpretation, that is the recognition stage was completed successfully. It is expressed in the ‘is trope’ event, whose

source is the LOLITA system itself.

If the algorithm proved the input to be a figure, the information about the resolution process is tied in with the rest of the SemNet by the ‘believe is a trope’ event. It links the algorithmic evidence with LOLITA’s belief system. The belief that the input is a figure is supported by the trigger conditions (see above). In the NLG representation, this support is expressed via a very generic causal relation. Based on the fact that the input is a figure, the ‘not believe literally’ event is put into a causal relation to the ‘believe is a trope’ event. Since the objective of the figurative resolution is to find an interpretation for the input and not to establish whether or not LOLITA believes the input, processing enters the interpretation stage. In the chain of reasoning, this is expressed by a causal relation between the ‘not believe literally’ event and the ‘draw inferences’ event, which expresses that a resolution is taking place and, in the case of simple tropes, is performed using a simple resolution method. If more than one resolution method is used, only the last one is recorded in the ‘draw inferences’ event.

The ‘draw inferences’ event finally is linked to the result of the resolution. This can either be the interpretation, which again is joined to the rest of the reasoning by a generic causal relation or an event stating that the resolution failed. If the recognition stage could not classify the input as a figure, the fail event is of course generated directly, since no further steps are taken. It states the method used and the negative result. Depending on whether a literal analysis has already taken place, the system can then deal with the input accordingly.

The ‘is seen as’ event is generated only in the course of a simple resolution. It expresses the figurative relation between the tenor and the vehicle, a notion derived from the interaction view on metaphor (c.f. Sect. 3.2.4). Although it is very generic and prototypical (e.g. ‘inputs use human features in order to describe a motorbike’ to denote that the ‘motorbike’ is described or seen as ‘human’ in some respect), it is sufficient to convey the basic idea of expressing the domain interaction at a general level.

Generic figurative reasoning

The generic figurative reasoning follows the simple figurative reasoning with the exception that the domain interaction is not represented in the same way as with simple tropes. Although the fixed metonymy method can be used to resolve simple

tropes and the conventional method to resolve fixed metonymies, the relevant NLG capabilities have not been included in the generic reasoning because it is used by three resolution methods. Instead, an event recording the relation between the input concept and the concept denoted by it on the basis of the metonymy is constructed (see also Appendix B.2.2) in the case of the fixed metonymy resolution. Apart from the SemNet representations of the trigger conditions, reinforcing condition and the actual chain of reasoning, no other specific events are built, to allow the use of one type to cover various resolution methods. The type of figure used in the ‘is trope type’, ‘believe is a trope type’ and ‘draw inferences’ events is passed as a parameter to the relevant functions constructing these from the individual parts of the resolution algorithm. Thus, the fixed, conventional and contextual resolution all share the code for generation of these events. Using the conventional resolution, the type of figure is determined in the wake of the knowledge consultation. Therefore it is possible to get results where, for example, a metonymy is resolved using a conventional resolution.

Idiomatic figurative reasoning

The fact that context plays an important role in the interpretation of idiomatic expression and that a large number of idioms have a perfectly plausible literal reading when there is no context and limited world knowledge available made it necessary to modify the status of the trigger conditions for this type of reasoning.

Due to the fact that it is not always possible to access a sufficient amount of knowledge in order to establish a priori whether the input is an idiom or not, the default is to assume that if it has an idiomatic reading, it is more likely to be an idiomatic expression than a literal utterance. In our opinion, problems would arise from this assumption if and only if there were no literal analysis, either before or after the figurative one. Since it is not possible to analyse input only figuratively but not literally at the moment (and it is unlikely that this mode of operation will be implemented), the only weak point of this approach remaining is the ranking of interpretations. A small percentage of cases where there is a valid literal and idiomatic reading constitutes, for example, humorous utterances and puns. Within the figurative resolution, the idiomatic reading will correctly be recorded. The evaluation of the fact that there are two valid interpretations, however, does not lie within the scope of the figurative sub-system. This is reflected in the triggering condition part of the reasoning for idioms, which is distinct from the comparable

events generated elsewhere. It states that the input is ‘literally unlikely’ if and because it is an idiom. Thus, it is possible to ascribe a literal reading to the input as well without arriving at conflicting epistemic information on it.

Comparison reasoning

In order to keep the code simple, the comparison reasoning re-uses a large portion of the generic reasoning code. The particular feature of the comparison statement algorithm is that the type of comparison is verified by the initial checks or the function ‘makeinter.unknown’ (c.f. Appendix B.2.6). Subsequently, the function responsible for generating the chain of reasoning selects the type of comparison on the basis of the interpretation event built beforehand. Of course, the dimension of literal versus figurative does not feature in the same way as it does in the case of metonymies, for example. Once it has been established that the input is a comparison, it remains to be seen whether it is a figurative or literal comparison, but there is no competing, different interpretation. Thus, the trigger condition in the case of comparison reasoning is that the event has a valid interpretation as a comparison. The ‘is tropetype’ event is completed with the appropriate type recursively after it has been established whether the input is a literal or figurative comparison (or indeed both readings are plausible on the basis of the contents of the SemNet). With comparison statements, the fact that the input is a comparison is sufficient for a resolution to take place, since it has to be established which features and attributes are ascribed to the tenor concept and in which mode, literally or metaphorically. Therefore, the ‘believe is a tropetype’ event brought about by the ‘is tropetype’ event ‘causes’ the resolution or ‘draw inferences’ event, which in turn leads to the interpretation.

Fail events

The last type of information added to the SemNet in the course of the figurative analysis of input are the fail events. Although they do not belong to the chain of reasoning as such, they can be seen as the final step of it, being equivalent to the interpretation event(s) built. Whereas an interpretation or at least a classification indicates the successful resolution of input, fail events indicate that the figurative analysis did not succeed.

In order to maintain a high degree of openness and following the criterion of

integration, the non-literal analysis cannot decide on its own how a failed figurative analysis should be treated. There might have already been a literal analysis, it might be still to follow, there may be more figurative methods that are applicable to the input, and all these different configurations have to be taken into account. Not only would it be nearly impossible to describe all possible cases, but future extensions of either part of LOLITA may render the measures obsolete. Therefore, the construction of fail events was deemed to be necessary. They state the situation of the non-literal analysis and on this basis, all other parts of the system and subsequent figurative modules can determine the course of action to be taken.

As mentioned before, attributing a low degree of belief is LOLITA's default for handling input whose analysis is not compatible with the knowledge in the SemNet. Clearly, stating that what is expressed in the input cannot be believed is only a first step towards 'intelligent' results. The fail events of the non-literal analysis can be seen as an extension of the basic default mechanism. Although they are implemented only prototypically, they give a more elaborate picture of the system's internal processes, stating the reason for the rejection of the input and the methods of analysis used. Analogous to the figurative reasoning recording single steps of the system in processing input, the fail events are used to record intermediate results in a way that is suitable for further handling by the system itself. Instead of giving error messages, the system registers its own actions and can retrieve information on its own behaviour. Moreover, the information recorded can serve as basis for subsequent steps of processing.

The actual fail events (see also Appendix A.1) consist of information giving the analysis method used, that is, literal or figurative. If a figurative method failed on the input, its type is given, as well, e.g. simple resolution or contextual resolution.

B.3 Conceptual proximity

This section presents details on the computation of conceptual proximity. Conceptual proximity is used in the resolution of comparison statements, and can be regarded as an implementation of the salience imbalance principle proposed by Ortony [Ortony 1979].

This method looks for similarities in the two domains of the tenor and vehicle concept and is comparable to finding analogies between the two domains. Since it

is a computationally expensive undertaking to compare many concepts, triangulation is not performed at random, but only on concepts from the two domains which are closely related to the input concepts and it is only performed when absolutely necessary. This is the case, for example, when no previous method could establish whether a comparison is figurative or literal. The proximity measure is computed using the following algorithm: recursively collect all concepts standing in a ‘universal’ and ‘generalisation’ relation to the tenor and vehicle concepts, respectively, and add them to the input. Add the recursive collection of all concepts standing in the ‘similar’ relation to the concepts from the set of concepts collected in the last step. To this list, add the recursive collection of all concepts standing in the ‘generalisation’ relation to the concepts from the result. The result is the basis for the final step, in which information that is conceptually related¹ to the members of the list of concepts obtained by processing so far is added to the result. The collection of concepts obtained by this algorithm for both the tenor and the vehicle form the basis for the formula $proximity = (100/concepts-not-shared)*concepts-shared$.

It should be noted that there is, of course, a defence for the case where there are no non-shared concepts, since the input is initially included in the collection of concepts. But more importantly, concept relations where there might be a number of shared concepts but no non-shared concepts never reach the stage of processing conceptual proximity. Such cases (identity, weak identity and class inclusion) are caught at an earlier stage in the algorithm for comparison statements, prior to the computation of conceptual proximity (c.f. Sect. 5.3.7 and Appendix B.2.6 for details). This ‘split functionality’ is permissible since the computation of conceptual proximity is by no means a general function, but only part of the whole algorithm for comparison statements.

The following debugging output presents a few examples of comparison statements, their conceptual proximity value, and the resultant classification with respect to the status of the comparison. The first two examples were not passed to the conceptual proximity computation because the concept relation between tenor and vehicle could be established during the ‘initial check’ phase (c.f. Appendix B.2.6 and Fig. 5.11). In these examples, the first line represents the status and classification, the second line the input, followed by the trace, beginning with the header indicating debugging output, namely, ‘RE:’ and the name of the function

¹Conceptually related information in this case means, e.g., the inclusion of ‘knife’ in the sense of ‘cutlery’, if the input was ‘knife’ in the meaning of ‘weapon’.

producing the output, here, 'determine_comparison'. The last line of each example is debugging output giving the tenor and vehicle concepts along the corresponding Noderefs (c.f. Sect. 2.2.1).

(Weak) Identity

```
something is something
RE:determine_comparison
Tautology/Identity/Close relation
Topic: something: 19714 Vehicle: something: 19714
```

Tenor is sub-type of vehicle, literal comparison

```
gerbils are gnawers
RE:determine_comparison
Direct class inclusion
Topic: gerbil: 6431 Vehicle: gnawer: 44340
```

```
killers are murderers
RE:determine_comparison
Indirect class inclusion
Topic: killer: 48231 Vehicle: murderer: 23982
```

The following are examples of debugging output from the function responsible for the computation of conceptual proximity. The first line of each example presents the input, followed by fancy traces, stating the system's classification of the comparison, the conceptual proximity value ('closeness') and the number of features available on each of the concepts in the SemNet².

Tenor is closely related to vehicle, literal comparison

```
lamps are furnishings
-----
Comparison; very close, presumably class inclusion?
Conceptual measures
Closeness : 70.0000000
Topic features : 3
Vehicle features : 3
-----
```

Shakespeare is an author

```
-----
Comparison; very close, presumably class inclusion?
Conceptual measures
```

²It should be noted that identity counts as a feature, thus one feature is similar to having no information on the concept in question.

Closeness : 63.4146347
Topic features : 3
Vehicle features : 12

Lolita is a girl

Comparison; very close, literal.
Conceptual measures
Closeness : 51.6483536
Topic features : 5
Vehicle features : 43

guitarists are musicians

Comparison; very close, literal.
Conceptual measures
Closeness : 51.4285736
Topic features : 4
Vehicle features : 13

gerbils are mice

Comparison; very close.
Conceptual measures
Closeness : 40.0000000
Topic features : 6
Vehicle features : 6

killers are like criminals

Comparison; related concepts.
Conceptual measures
Closeness : 38.8888855
Topic features : 3
Vehicle features : 19

murderers are criminals

Comparison; weak relation.
Conceptual measures
Closeness : 17.3913040
Topic features : 14
Vehicle features : 24

lecturers are academics

Comparison; odd relation. Data missing?

Conceptual measures

Closeness : 4.16666651

Topic features : 20

Vehicle features : 26

Politicians are killers

Comparison; no relation, predicate introduction

Conceptual measures

Closeness : 0.00000000

Topic features : 27

Vehicle features : 28

Politicians are glynskis³

No information on the vehicle available;
no relation

Conceptual measures

Closeness : 0.00000000

Topic features : 27

Vehicle features : 1

Glynskis are computers⁴

Comparison; no relation, predicate introduction

Conceptual measures

Closeness : 0.00000000

Topic features : 1

Vehicle features : 16

B.4 Invariance hierarchy

In this section, details on a modified invariance hierarchy are given. Invariance hierarchies are used to rank and evaluate the cross-domain relations which occur in

³Since there is no information on the vehicle, no comparison is possible and therefore the relation between tenor and vehicle is labelled as 'no relation' (see also Fig. 5.11).

⁴In this case, there is no information on the tenor. Therefore both a literal and figurative interpretation is sought. In both cases, information from the vehicle domain is transferred into the tenor domain. This is reflected in the label 'predicate introduction' comparison.

comparison statements (see also Sect. 3.2.3.2). Our invariance hierarchy is based loosely on the invariance hierarchy put forward by Carbonell [Carbonell 1982]. While his model uses invariance hierarchies to find ‘relevant matches’ between the two domains of the tenor and vehicle, this solution uses the preliminary invariance hierarchy as an additional filter after the cross-domain correspondences have been found. This task is important when the most salient interpretation of a comparison is needed, for example, in the dialogue application of the LOLITA system. Since the prototypical implementation of the figurative analysis has not yet been polished, there is no practical use of the invariance hierarchy as such. The basic functionality is trying to find all plausible interpretations, the ranking and further processing is the responsibility of customised interfaces to the figurative analysis or the individual applications.

Nevertheless, a casual analysis of examples has proven that the filtering of interpretations according to the concept relation they express can act as a further refinement of the resolution process. Carbonell’s invariance hierarchy consists of the concept relations goals, planning strategies, causal structures, functional attributes, temporal orderings, natural tendencies, social roles, structural relations, descriptive properties and object identity.

While Carbonell suggests that a high concept relation, for example, causal structures, effectively blocks a lower cross-domain correspondence, for example, the correspondence of descriptive properties between the tenor and vehicle, we suggest that a higher relation takes priority and is preferred over a lower correspondence, but does not halt the process of evaluating all correspondences. This approach is feasible, because in our model the invariance hierarchy filtering takes place after a number of cross-domain correspondences have been established, as opposed to Carbonell’s theoretical model, where the invariance hierarchy consultation forms part of the domain comparison stage. Moreover, we do not use all of Carbonell’s relations. Since the relations are not explicitly present in the SemNet, for example, as labelled arcs between concepts, the only way of inspecting the SemNet for possibly matching concept relations lies in the application of the masking method (c.f. Sect. 5.2.4). The cost-benefit ratio for the additional processing is, judged by the meagre results of achieving a non-essential means of ranking interpretations, not acceptable. Therefore a modified and simplified invariance hierarchy is used. It prefers causal relations over functional relations, which in turn are preferred over natural attributes and tendencies. By means of concept extension, or simple abstraction

from the actual input, relations like ‘use’ and ‘make’ can be matched with functional and causal relations, respectively. Natural attributes and tendencies, such as size, colour or skill are found on the assumption that they are represented in definitional knowledge in the SemNet. Thus the processing cost for the application of the invariance hierarchy is kept as low as possible, while the benefit of being able to rank interpretations is maximised. Again, it is important to note that the sole basis for this is the information available in the SemNet. Therefore neither external knowledge sources nor inconsistencies can lead to unexpected results.

Work remains to be done with respect to the invariance hierarchy mechanism. A thorough corpus analysis of comparison statements must yield empirical data on which a final invariance hierarchy can be based. Moreover, the mechanism has to be implemented properly as part of a customised figurative resolution interface, e.g. for the dialogue application of LOLITA. This application obviously would benefit from the (contextually) most appropriate interpretation of a comparison statement to enable it to conduct a natural, interactive dialogue. As it stands now, the interpretations which are kept in a data structure of the type list are compared one by one to the entries of the invariance hierarchy, starting with the topmost, e.g. causal relations. If a relation matches, it is copied into the result list and removed from the input list. After the list of interpretations has been processed, the same procedure starts over again if there are lower entries in the invariance hierarchy. Otherwise, the resultant list is the ranking of the interpretations according to the invariance hierarchy. This very basic implementation was used during the experimental stage of the design of the solution and proved the feasibility and usefulness. However, it must be refined when used within a final application interface.

B.5 Abstract data type metaphor

As mentioned before, the interfacing of the non-literal analysis to the rest of the complex LOLITA system has been implemented in a basic way, sufficient to show the capabilities and to allow the parent system to make full use of the figurative resolution’s results. The NLE criteria such as integration, flexibility and usability required the solution to take interfacing needs into account from the very early design stage on. Thus it was ensured that an open, flexible and powerful interface that delivered all the necessary information avoided any kind of bottleneck between the components of the system.

In our opinion, the information that is derived from the analysis of figurative input and required by other parts of the system consist of an indication that figurative input was detected, the steps that were taken during the analysis and the results of the analysis.

In order to enable a clear interface to be built, this information has to be bundled into an abstract data type that hides away details from the rest of the system. Interfacing is then only possible via pre-defined functions that access this data type and its building blocks in a uniform fashion. Thus any changes in the actual implementation have no negative effect on the interfaces and on other parts of the system's code. The abstract data type proposed for the general use by other parts of the LOLITA system is the data type 'metaphor'.

$$\text{metaphor} = (\text{base } [[\text{ctx}, m, i]_n, \dots, [\text{ctx}, m, i]_x])$$

Where a 'metaphor', in the sense of an entity to be represented in the LOLITA system's inter-module information exchange, consists of

- a *base*, that is the input of the figurative resolution algorithm. This is an utterance which has been pre-processed by the system. Pre-processing includes syntactic, semantic, and optionally also partial pragmatic analysis. Only input above the complexity level of lexemes will be processed fully, since 'words' cannot be used metaphorically as such. The next part of the abstract data type is
- a *list of triples* $[\text{ctx}, m, i]$, where
 - *ctx* denotes the context of the utterance or concept. For the time being, zero contexts and contexts of the type conforming to the established abstract data type 'context' within the LOLITA system are permissible. The second entry in the triplet represents
 - *m*, the method applied to base in *ctx* by the non-literal analysis, which yields
 - *i*, the interpretation of base in *ctx* according to *m*. This can be the actual interpretation, or, in the case of an unsuccessful resolution, the appropriate fail event (see Appendix B.2.7).

The reason for having ordered triples is that the same base can result in different interpretations, depending on the context, when the same method is applied to

it. Likewise, one base can yield different interpretations in the same context when different methods, a conventional resolution and comparison resolution, for example, are applied to it. Also, the same context, method and base can yield distinct interpretations, varying in the degree of generality. It should be noted that, due to the well-defined structure of the figurative reasoning (c.f. Appendix B.2.7) all pieces of information created during the non-literal analysis are accessible through the interpretation part of the data type.

Since the implementation of interfaces based on the abstract data type presented was not one of the objectives of this undertaking, no such interface exists. An implementation would have meant that a considerable amount of work in various parts of the system's code would have had to take place, improving other parts with no immediate benefit to the figurative sub-system apart from the further proof of its feasibility and functionality. All the information that will be bundled into the abstract data type 'metaphor' already exists and can be accessed by other means (see also Sect. 6.3), yet the design of the abstract data type 'metaphor' anticipated the requirements of polished and refined applications with respect to the information handled by the non-literal analysis.

Bibliography

- [Austin 1975] J. Austin, *How to do things with words.*, Oxford University Press, London, second edition, 1975.
- [Black 1962] M. Black, *Models and metaphors*, Cornell University Press, Ithaca, NY, 1962.
- [Black 1979] M. Black, *More about metaphor*, pages 19–43, Cambridge University Press, first edition, 1979.
- [Bokma and Garigliano 1992] A. Bokma and R. Garigliano, “Uncertainty management through source control: A heuristic approach”, in *Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Mallorca, 1992.
- [Brachman and Schmolze 1985] R. Brachman and J. Schmolze, “An overview of the KL-ONE knowledge representation system”, *Cognitive Science*, 9:171–216, 1985.
- [Bußman 1990] H. Bußman, *Lexikon der Sprachwissenschaft*, Kröner, Stuttgart, second edition, 1990.
- [Carbonell and Minton 1985] J. Carbonell and S. Minton, *Metaphor and common-sense reasoning*, pages 405–426, Ablex, Norwood, NJ, 1985.
- [Carbonell 1982] J. Carbonell, *Metaphor: A Key to Extensible Semantic Analysis*, pages 415–434, Erlbaum, Hillsdale, NJ, 1982.
- [Carnap 1956] R. Carnap, *Meaning and necessity*, University of Chicago Press, Chicago, second edition, 1956.
- [Charniak 1981] E. Charniak, “The case-slot identity theory”, *Cognitive Science*, 5:285–292, 1981.

- [Cicero 1942] Cicero, *De oratore lib. III*, Heinemann, London, 1942.
- [Cooper 1986] D. Cooper, *Metaphor*, Basil Blackwell, Oxford, 1986.
- [Costantino *et al.* 1996] M. Costantino, R. Morgan, and R. Collingham, "Natural language processing in finance: Automatic extraction of information from financial news articles", *Artificial Intelligence in Finance*, 2(4), 1996.
- [Davidson 1979] D. Davidson, *What metaphors mean*, University of Chicago Press, Chicago, 1979.
- [Ellis *et al.* 1993] N. Ellis, R. Garigliano, and R. Morgan, "A new transformation into deterministically parsable form for natural language grammars", in *Proceedings of the 3rd International Workshop on Parsing Technologies*, Tilburg, 1993.
- [Emery 1994] S. K. Emery, "An investigation into the characterisation of style in everyday text", Master's thesis, University of Sunderland, Sunderland, 1994.
- [Falkenhainer *et al.* 1989] B. Falkenhainer, K. Forbus, and D. Gentner, "The structure-mapping engine: algorithm and examples", *Artificial Intelligence*, pages 1-63, 1989.
- [Fass 1988] D. Fass, *Colative semantics: A semantics for natural language*, PhD thesis, New Mexico State University, Las Cruces, 1988.
- [Fass 1991] D. Fass, "Met*: A method for discriminating metonymy and metaphor by computer", *Computational linguistics*, 17(1):49-90, 1991.
- [Frege 1952] G. Frege, "On sense and reference", in P. Geach and M. Black, editors, *Philosophical writings of Gottlob Frege*, Blackwell, Oxford, 1952.
- [Galliers and Jones 1993] J. Galliers and K. S. Jones, "Evaluating Natural Language Systems", Technical Report 291, Cambridge, 1993.
- [Garigliano *et al.* 1993] R. Garigliano, R. Morgan, and M. Smith, "The LOLITA system as a contents scanning tool", in *Proceedings of the 13th International Conference on Artificial Intelligence, Expert Systems and Natural Language Processing*, Avignon, 1993.
- [Garigliano *et al.* 1995] R. Garigliano, B. Boguraev, and J. Tait, "Editorial", *Journal of natural language engineering*, 1:1-7, 1995.

- [Gentner *et al.* 1989] D. Gentner, B. Falkenhainer, and J. Skorstad, *Viewing metaphor as analogy: the good, the bad, and the ugly*, pages 171–177, Erlbaum, Hillsdale, NJ, 1989.
- [Gentner 1983] D. Gentner, “Structure-Mapping: a theoretical framework for analogy”, *Cognitive Science*, 7:155–170, 1983.
- [Gentner 1989] D. Gentner, *The mechanism of analogical learning*, chapter 7, Cambridge, 1989.
- [Gibbs 1993] R. Gibbs, Jr., *Process and products in making sense of tropes*, chapter 12, Cambridge University Press, second edition, 1993.
- [Gibbs 1994] R. Gibbs, Jr., *The poetics of mind: Figurative thought, language and understanding*, Cambridge University Press, Cambridge, 1994.
- [Görz 1993] G. Görz, editor, *Einführung in die künstliche Intelligenz*, Addison Wesley, Bonn, Paris, Reading MA, 1993.
- [Grice 1975] H. P. Grice, “Logic and conversation”, in J. M. P. Cole, editor, *Syntax and Semantics*, volume 3, pages 41–58, Academic Press, New York, 1975.
- [Grice 1978] H. P. Grice, “Further notes on logic and conversation”, in P. Cole, editor, *Syntax and Semantics*, volume 9, pages 113–127, Academic Press, New York, 1978.
- [Holyoak and Thagard 1989a] K. Holyoak and P. Thagard, “Analogical mapping by constraint satisfaction”, *Cognitive Science*, 13:295–355, 1989.
- [Holyoak and Thagard 1989b] K. Holyoak and P. Thagard, *A computational model of analogical problem solving*, chapter 8, Cambridge, 1989.
- [Hudak *et al.* 1994] P. Hudak, S. Jones, and P. Wadler, “Report on the functional programming language Haskell, Version 1.2”, 1994, ACM SIGPLAN Notices 27.
- [Johnson 1987] M. Johnson, *The body in the mind: The bodily basis of reason and imagination*, University of Chicago Press, Chicago, 1987.
- [Katz and Fodor 1963] J. J. Katz and J. Fodor, “The structure of semantic theory”, *Language*, (39), 1963.
- [Katz 1966] J. J. Katz, *The philosophy of language*, New York, 1966.

- [Katz 1967] J. J. Katz, "Recent issues in semantic theory", *Foundations of Language*, 3:124–194, 1967.
- [Kittay 1987] E. Kittay, *Metaphor: its cognitive force and linguistic structure*, Oxford University Press, 1987.
- [Lakoff and Johnson 1980] G. Lakoff and M. Johnson, *Metaphors we live by*, The University of Chicago Press, 1980.
- [Lakoff et al. 1995] G. Lakoff, J. Espenson, and A. Schwartz, editors, *Master metaphor list*, Berkeley, California, 1995, <http://cogsci.berkeley.edu/>.
- [Lakoff 1993] G. Lakoff, "The contemporary theory of metaphor", in A. Ortony, editor, *Metaphor and Thought*, Cambridge University Press, second edition, 1993.
- [Levin 1977] S. Levin, *The semantics of metaphor*, Johns Hopkins University Press, Baltimore, 1977.
- [LNLE 1995] LNLE,
"LNLE Homepage, URL <http://www.dur.ac.uk/~dcs0rjc/lnle/lnlehome.html>",
1995, University of Durham.
- [Long and Garigliano 1994] D. Long and R. Garigliano, *Reasoning by analogy and causality: a model and application*, Ellis Horwood, 1994.
- [Lyons 1977] J. Lyons, *Semantics I*, Cambridge, 1977.
- [Martin 1990] J. Martin, *A computational model of metaphor interpretation*, Academic Press, San Diego, 1990.
- [Martin 1991a] J. Martin, *MetaBank: A knowledge-base of metaphoric language conventions*, University of Colorado, Boulder, 1991.
- [Martin 1991b] J. Martin, *Representing and acquiring metaphor-based polysemy*, Erlbaum, Hillsdale, 1991.
- [Martin 1992] J. Martin, "Conventional metaphor and the lexicon", in J. Pustejovsky and S. Bergler, editors, *Proceedings of the First SIGLEX Workshop on Lexical Semantics and Knowledge Representation*, pages 61–73, Berlin, Heidelberg, 1992, Springer-Verlag.
- [Meier 1963] H. Meier, *Die Metapher: Versuch einer zusammenfassenden Betrachtung ihrer linguistischen Merkmale*, Winterthur, 1963.

- [Miller 1990] G. Miller, "WordNet: An on-line lexical database", *International Journal of Lexicography*, 3(4):1-7, 1990.
- [Morgan *et al.* 1995] R. Morgan, R. Garigliano, P. Callaghan, S. Poria, M. Smith, A. Urbanowicz, R. Collingham, M. Costantino, C. Cooper, and the LOLITA group, "University of Durham: Description of the LOLITA system used in MUC-6", in *Proceedings of the Message Understanding Conference (MUC-6)*, 1995.
- [Nettleton and Garigliano 1995] D. J. Nettleton and R. Garigliano, *Evolutionary Algorithms and Dialogue*, volume 2, CRC Press, 1995.
- [Nietzsche 1974] F. Nietzsche, *On truth and falsity in their ultramoral sense*, Gordon Press, New York, 1974.
- [Noble 1988] H. Noble, *Natural Language Processing*, Blackwell Scientific Publications, Oxford, London, Edinburgh, 1988.
- [Ortony 1979] A. Ortony, *Similarity in similes and metaphors*, pages 186-201, Cambridge University Press, first edition, 1979.
- [Plantinga 1989] E. Plantinga, *Mental Models and Metaphor*, Erlbaum, Hillsdale, NJ, 1989.
- [Quintilian 1922] Quintilian, *De institutione oratoria lib VIII*, pages 176-345, Heinemann, London, 1922.
- [Reddy 1979] R. Reddy, "The conduit metaphor", in A. Ortony, editor, *Metaphor and thought*, pages 284-324, Cambridge University Press, first edition, 1979.
- [Richards 1936] I. A. Richards, *The philosophy of rhetoric*, Oxford University Press, New York, 1936.
- [Searle 1969] J. Searle, *Speech acts: an essay in the philosophy of language*, Cambridge University Press, London, 1969.
- [Searle 1975] J. R. Searle, "Indirect speech acts", *Syntax and Semantics*, 3, 1975.
- [Searle 1979] J. Searle, *Expression and meaning: studies in the theory of speech acts*, Cambridge University Press, Cambridge, 1979.
- [Searle 1980] J. Searle, *Speech act theory and pragmatics*, Reidel, Dordrecht, 1980.

- [Shank and Abelson 1977] R. C. Shank and R. P. Abelson, *Scripts, plans, goals and understanding*, Erlbaum, Hillsdale, 1977.
- [Short *et al.* 1996] S. Short, S. Shiu, and R. Garigliano, "The distributivity and non-linearity of LOLITA's semantic net", 1996, Submitted to COLING 96.
- [Short 1996] S. Short, *SemNet: the knowledge representation of LOLITA*, PhD thesis, University of Durham, Durham, 1996.
- [Smith *et al.* 1994] M. H. Smith, R. Garigliano, R. Morgan, S. Shiu, and S. Jarvis, "Lolita: An engineering approach to natural language processing", Technical report, University of Durham, 1994.
- [Smith 1995] M. H. Smith, *Natural language generation in the LOLITA system: an engineering approach*, PhD thesis, University of Durham, Durham, 1995.
- [Sowa 1984] J. Sowa, *Conceptual structures - information processing in man and machine*, Addison Wesley, Reading MA, 1984.
- [Steinhart 1993] E. Steinhart, *NETMET: a program for analogy and metaphor*, Department of Philosophy, University at Stony Brook, Stony Brook, 1993, [gopher://cougar.cc.oxy.edu:70/59/Software for Philosophers/Metaphor and Analogy Software/netmet.zip](mailto:gopher://cougar.cc.oxy.edu:70/59/Software%20for%20Philosophers/Metaphor%20and%20Analogy%20Software/netmet.zip).
- [Sun 1995] R. Sun, "A microfeature based approach towards metaphor interpretation", 1995, University of Alabama, Tuscaloosa.
- [Tourangeau and Sternberg 1982] R. Tourangeau and R. Sternberg, "Aptness in metaphor", *Cognitive Science*, 11:27-55, 1982.
- [Turner 1982] D. Turner, *Recursion equations as a programming language*, Cambridge University Press, 1982.
- [Turner 1987] M. Turner, *Death is the mother of beauty*, University of Chicago Press, Chicago, 1987.
- [Veale and Keane 1992a] T. Veale and M. Keane, "Conceptual scaffolding: a spatially founded meaning representation for metaphor comprehension", *Computational Intelligence*, 8(3), 1992.
- [Veale and Keane 1992b] T. Veale and M. Keane, "Conceptual scaffolding: using metaphors to build knowledge structures", Vienna, August 1992, Presented at the European Conference on Artificial Intelligence 1992.

- [Veale and Keane 1993] T. Veale and M. Keane, "A connectionist model of semantic memory for metaphor interpretation", U.S.C. California, October 1993, Presented at Neural Architectures and Distributed AI Workshop 1993.
- [Veale and Keane 1994] T. Veale and M. Keane, "Belief modelling, intentionality and perlocution in metaphor comprehension", Atlanta, Georgia, 1994, Presented at Cognitive Science 1994.
- [Veale and Keane 1995] T. Veale and M. Keane, "Epistemological pitfalls in metaphor comprehension: a comparison of three models and a new theory of metaphor", Albuquerque, July 1995, Presented at the International Cognitive Linguistics Conference 1995.
- [Veale 1995] T. Veale, *Metaphor, memory and meaning: symbolic and connectionist principles in metaphor comprehension*, PhD thesis, Trinity College, University of Dublin, Dublin, 1995.
- [Wang and Garigliano 1992] Y. Wang and R. Garigliano, "An Intelligent Tutoring System for Handling Errors caused by Transfer", in *Lecture Notes in Artificial Intelligence 608*, Springer-Verlag, Heidelberg, 1992.
- [Way 1991] E. C. Way, *Knowledge Representation and Metaphor*, Kluwer, Boston, 1991.
- [Wilks 1975a] Y. Wilks, "An intelligent analyzer and understander of English", *Communications of the ACM*, 18(5):264-274, 1975.
- [Wilks 1975b] Y. Wilks, *Preference semantics*, pages 329-348, Cambridge University Press, Cambridge, 1975.
- [Wilks 1978] Y. Wilks, "Making preferences more active", *Artificial Intelligence*, 11, 1978.

