

Dissertation

Mining Semantics from Text: A Connectionist Approach to Ontology Enhancement for a Tourism Information System

ausgeführt zum Zwecke der Erlangung
des akademischen Grades eines Doktors
der technischen Wissenschaften

unter der Leitung von
ao. Univ. Prof. Dr. Dieter Merkl
E188 Institut für Softwaretechnik
und interaktive Systeme

eingereicht an der Technischen Universität Wien
Fakultät für Technische Naturwissenschaften und Informatik

von

DI Michael Dittenbach
9525201
Schenkendorfgasse 14–16/2/12
A–1210 Wien

Wien, im Mai 2003

Unterschrift

Kurzfassung der Dissertation

Mit der stetig wachsenden Menge an Information die im Internet verfügbar ist, ist es eine der herausforderndsten Aufgaben, einfache und intuitive Suchfunktionen zu entwickeln, die wirklich das finden was die Benutzerin/der Benutzer suchen und bei denen die man keine bestimmte Syntax lernen muss. Wir präsentieren eine Suchschnittstelle fuer ein Informationssystem in der Tourismusdomäne, welches die Intuitivität der natürlichen Sprache ausnützt. Besonders im Tourismus ist dies eine attraktive Alternative, weil solche Informationssysteme von einer sehr inhomogenen Benutzergruppe frequentiert wird was ihre Computerkenntnisse betrifft. Deshalb haben wir eine natürlichsprachliche Schnittstelle für eines der größten europäischen Tourismusportale, *Tiscover*, entwickelt. Diese Schnittstelle ermöglicht die Formulierung von Suchanfragen in einem deutschen oder englischen Satz, um nach Unterkünften in ganz Österreich zu suchen, die durch eine Vielzahl von Attributen beschrieben werden. Die Sprache der Anfrage wird automatisch vom System erkannt und das Suchergebnis in der jeweiligen Sprache zurückgeliefert. Eine einfache Verarbeitung natürlicher Sprache wandelt die ursprüngliche Anfrage mit Hilfe einer Ontologie, in der domänenspezifische Konzepte gespeichert sind, in eine strukturierte Datenbankabfrage um.

Um jedoch die steigende Komplexität einer solchen Ontologie während ihres Wachstums und ihrer Weiterentwicklung zu beherrschen, präsentieren wir eine auf einer Methode basierend auf *Self-Organizing Maps*, Worte aufgrund semantischer Ähnlichkeiten zu clustern. Die *Self-Organizing Map* ist ein neuronales Netz, das Daten von einem hochdimensionalen Raum auf eine 2-dimensionale Karte abbildet wobei Ähnlichkeitsbeziehungen zwischen den Daten erhalten bleiben. Wir nutzen textuelle Beschreibungen der Unterkünfte, um eine numerische Darstellung der Worte zu erzeugen, die auf dem gemeinsamen Auftreten von Wörtern in Wortfenstern bestimmter Größe basieren. Weiters präsentieren wir einen Ansatz, um Wörter bezüglich ihrer Wichtigkeit für gewisse geographische Regionen zu gewichten. Durch diese vorgestellten Methoden wird die Bildung bzw. der Ausbau solcher Ontologien erleichtert. Einerseits wird das Vokabular der Domäne übersichtlich dargestellt, und andererseits werden Worte die potentiell interessant für die Integration in die Ontologie wären, hervorgehoben.

Abstract

With the increasing amount of information available on the Internet one of the most challenging tasks is to provide search interfaces that are easy to use without having to learn a specific syntax and let people find what they really want. We present a query interface for an information system in the tourism domain exploiting the intuitiveness of natural language. This is especially appealing, because web-based tourism information systems are faced with a highly inhomogeneous mix of potential consumers regarding their computer literacy. Hence, we have developed a natural language query interface for one of the largest European web-based tourism information systems *Tiscover*. This interface allows for posing queries in either German or English to search for accommodations that are described by a variety of characteristics throughout Austria. The language of the query is automatically detected and the search result is presented accordingly. Shallow language processing transforms the original query into a structured database request by means of an ontology where domain-relevant concepts are stored in order to retrieve information about matching accommodations.

However, to cope with the increasing complexity with growing size of such ontologies, we present a clustering approach based on the self-organizing map to visualize semantic relation between words that occur in domain-relevant free-form text documents. The self-organizing map is a well-known neural network model that maps high-dimensional data onto a two-dimensional map preserving similarities between data items. In particular, we use textual descriptions of the accommodations and create numerical representations of the words based on the local co-occurrence of semantically similar words appearing in the vicinity of a term regarding the position in the text. Furthermore, we propose a term-weighting technique to highlight words that are distinct for certain geographical regions. Hence, the construction and enhancement of the ontology is facilitated, first, by displaying the vocabulary of the domain in a convenient way, and second, by weighting terms according to their potential value for being integrated into the ontology.

Contents

1	Introduction	1
2	Ad.M.In: A Natural Language Information Retrieval System	8
2.1	Introduction	8
2.2	Tourism Data	10
2.3	Original System	12
2.3.1	System Design	12
2.3.2	Knowledge Base	14
2.3.3	Query Processing	15
2.4	Enhanced System	20
2.4.1	Associative Networks and Spreading Activation	21
2.4.2	Redesigned Knowledge Base	22
2.5	Discussion	26
3	The User's Perspective	29
3.1	Introduction	29
3.2	Design Considerations for the Web-Based User Interface	30
3.3	Field Trial	33
3.3.1	Prerequisites	33
3.3.2	Results	35
3.3.3	Lessons Learned	42
3.4	Usability Evaluation	45
3.4.1	Test Setup	45
3.4.2	Results	48
3.4.3	Discussion	49

4	Ontology Enhancement	51
4.1	Introduction	51
4.2	Domain-Related Text Descriptions	56
4.3	Visualization of Semantic Relations	59
4.3.1	The Self-Organizing Map	59
4.3.2	Document/Term Matrix	63
4.3.3	Encoding the Semantic Contexts	70
4.4	Finding Geographical Peculiarities	82
4.5	Discussion	86
5	Conclusions	87

List of Figures

2.1	Overview of the system architecture.	13
2.2	Architecture of the redesigned ontology.	23
2.3	Results of the spreading activation-based system.	26
3.1	Natural language query interface.	31
3.2	Standard <i>Tiscover</i> search interface.	32
3.3	Advanced search page of the original <i>Tiscover</i> interface presenting all facilities and services that can be used as search criteria (the screenshot is split into three parts).	33
3.4	Result page with matching accommodations and feedback form.	34
4.1	Two different accommodation descriptions.	58
4.2	The units of a <i>SOM</i> can be arranged in different types of lattices. An n -dimensional weight vector is assigned to each unit, depicted by an array of shaded boxes representing the different values of the weight vector components.	60
4.3	Adaptation of weight vectors during SOM training. The adaptation strength of the individual units is indicated by different shades of gray.	63
4.4	Frequency distribution of the vocabulary of the accommodation descriptions. Please note the logarithmic scale of the y-axis.	65
4.5	Two descriptions of farms containing the names of farm animals such as <i>Hasen</i> (bunnies) and <i>Schafe</i> (sheep).	68
4.6	Semantic map of the original experiments by Ritter and Kohonen (1989). The manually drawn cluster boundaries separate the syntactic word classes.	71

4.7	Distribution of pairwise inner products of the random vectors with n dimensions (reproduced from Honkela (1997)).	73
4.8	A sample description of a holiday flat in a suburb of Vienna. On the left-hand side, the original description is shown, and on the right-hand side the remaining words after removing all words not starting with a capital letter are presented.	77
4.9	English translation of the terms shown on the right-hand side in Figure 4.8.	78
4.10	A self-organizing semantic map of terms in the tourism domain with labels denoting general semantic clusters. The cluster boundaries have been drawn manually.	79
4.11	An enlargement of the cluster covering room types, furnitures and fixtures located in the lower left corner of the map.	81
4.12	A map of Austria and its nine federal states.	83

List of Tables

2.1	List of accommodation-specific properties in the database.	11
2.2	List of city-specific location and activity features.	12
2.3	Top ten tri-gram occurrences of German and English text (under-scores represent blanks).	16
3.1	Origin of queries (derived from the top-level domain of the accessing host).	35
3.2	Manual analysis of language identification accuracy.	36
3.3	Distribution of query lengths.	37
3.4	Number of concepts per query (counted by manual inspection). . .	39
3.5	Concepts that have been identified or not identified by the natural language processing module of our interface.	40
3.6	Usage of modifiers <i>and</i> , <i>or</i> , <i>not</i> and <i>near</i>	42
3.7	Combined usage of modifiers.	43
4.1	Term/Document Matrix. N documents are described by n terms.	66
4.2	Document/Term Matrix with some sample vector elements showing the co-occurrence of farm animal names in the descriptions. .	67
4.3	List of terms occurring exclusively in descriptions of Viennese accommodations.	84
4.4	Sample terms denoting or related to regions that are crossing the borders of federal states.	85

Acknowledgments

acknowledgment, *n.*

VARIANT FORMS: **acknowledgement**

NOUN: **1.** The act of admitting or owning to something. **2.** Recognition of another's existence, validity, authority, or right. **3.** An answer or response in return for something done. **4.** An expression of thanks or a token of appreciation. **5.** A formal declaration made to authoritative witnesses to ensure legal validity.

Taken from the American Heritage[©] Dictionary
of the English Language: Fourth Edition. 2000.

Since semantic relationships are not the most important ones, the meanings number one to three, but especially number four, definitely go to my parents, my girlfriend Iris, Helmut, Dieter, Werner, the colleagues at EC3 ... and all the other nice people I forgot to mention ...

Within a computer natural language is unnatural

Alan J. Perlis' Epigrams in Programming, No. 114

Chapter 1

Introduction

The development of efficient and appropriate search functions is still a challenge in the field of database and information systems. In particular, we should mention interfaces of the kind that are used by *non-specialist* people. Hardly any computer scientist or technically adept person has problems understanding the Boolean logic underlying most conventional web search engines. Unfortunately, a growing majority of people using such search engines has, or even worse, doesn't know about the mechanisms that would be at hand to – sometimes dramatically – improve the quality of the search results. Just to mention an example consider searching for the soliloquy of Hamlet by using the rather famous beginning ‘*To be, or not to be*’, from Shakespeare’s *The Tragedy of Hamlet, Prince of Denmark*. Using *Google*¹, the query *to be or not to be shakespeare* yields 1,490,000 pages generally dealing with Shakespeare, because the words *to*, *be*, *or* and *not* were ignored. Without evident reason, capitalizing all letters of the query (*TO BE OR NOT TO BE SHAKESPEARE*) provides a result set of 3,940,000 pages where the top-ranked ones are quite different from those of the previous search. By enclosing the phrase with double quotes, “*to be or not to be*” *shakespeare*, about 17,100 pages are found where the top-ranked can be considered relevant regarding the intention of the search.

An analysis of query logs of the search engine *Excite* has shown that, in practice, only 9% of the queries contain Boolean operators or the modifiers ‘+’ and ‘-’ (Jansen et al., 1998). The latter two require that a query term must or

¹<http://www.google.com>

must not be present in the matching pages. Even if a description on using the search interface is given, which is the case for many search engines, users can easily be discouraged from reading lengthy descriptions of the search functionality. This can be a crucial point in a commercial environment where every distraction from the originally intended task increases the risk of a potential customer to cancel a shopping tour. To deliver a practical example, consider a large and information intensive website like that of Microsoft providing tons of product information, support documents and much more. The webpage providing help on the use of their search engine² covers about four screens of explanations and advice, i.e. about 1700 words just to describe how one should perform the search task.

Furthermore, an important issue is that, although large web search engines like *Google*, *AltaVista*³ and of course thousands of smaller site-specific search facilities have a similar superficial appearance, they tend to interpret queries with subtle differences that can lead to searches not meeting the user's intention. Without reading further information, one cannot be sure if a query is treated case sensitive or not, or how the keywords are connected logically, i.e. if all or any of the terms have to apply (Shneiderman et al., 1998). Another pitfall is hiding functions behind obscure syntactic details. *Google*, just to mention an example, removes an *or* in a query because it is treated as a stop word, but *OR* logically connects the adjacent terms regarding the position in the query string.

To take away the fear of this rather technically oriented way of searching for information, natural language should present a convenient form of interaction with such systems from the user's point of view. In particular, we foresee the following benefits for the user. She or he is relieved from the burden of having to learn and to use either strictly logical or highly structured query languages. The user could interact naturally with the system, using her or his style of describing the needed information. However, using natural language as a means of query formulation introduces new problems that have to be dealt with, or more generally, the complexity is being moved away from the user towards the engineer developing such a system. If the structure of the query is determined by a predefined grammar such as is SQL, errors can easily be detected by the system and

²http://search.microsoft.com/us/search_help.asp

³<http://www.altavista.com>

reported quite accurately to the user. Natural language, on the other hand, usually allows a specific intention to be expressed in multiple ways. Additionally, it creates room for ambiguities when the system covers a large conceptual domain.

Androutsopoulos et al. (1995) report several disadvantages of natural language interfaces that need further consideration. Because natural language interfaces only operate on a limited subset of natural language, it is not obvious to the user which kind of questions can be answered, i.e. how sophisticated the linguistic capabilities of the system are. Two patterns of this type of interaction problems can be distinguished. The *false positive expectation* occurs when a user expects a certain functionality she or he has noticed in a previous query to work again in similarly structured but semantically different queries in the future. Contrarily, expecting a specific erroneous behavior to happen again due to a previously posed query the system was not able to process is called *false negative expectation*, if it actually would work. The first pattern could cause some frustration to the user whereas the latter one potentially inhibits queries to be posed due to negative experience.

Another uncertainty lies in the type of failures of the natural language processing. From the user's perspective, it can often not be identified whether a question that led to an error was outside the linguistic or the conceptual coverage. Hence, it is not always clear whether rephrasing the query is useful or not. On the one hand, if the query contains concepts the system has no knowledge about, reformulating the question usually is of no help. On the other hand, if a query expressed in a certain form can not be handled by the language processing but the concepts are covered by the knowledge base, restating the query would be sensible in order to retrieve the relevant information.

Another technical aspect is the intricate configuration of natural language interfaces. It is usually more convenient to use systems with built-in indexing tools and formal query languages that already provide user interfaces. One point that can be hardly tackled by means of technology is that users assume intelligence when being confronted with a system that suggests to *understand* natural language. Often, the ability of reasoning and deduction goes along with this assumed intelligence. This misleading conception can only diminish in the course of time, when natural language interfaces are more in common use than they are now.

Lastly, natural language is claimed to be too verbose and ambiguous regarding human-computer interaction, for example, in the case of natural language search queries. Contrarily, form-based or graphical interfaces offer clearly defined, unambiguous elements for structured data entry. Nevertheless, it has to be noted that design failures and functional flaws of form-based interfaces, many of them leading to ambiguities or unclarities, fill whole books (see, e.g. Johnson, 2000). Disregarding functional deficiencies, information systems covering very large, if not unrestricted, application domains such as web search engines, are especially prone to ambiguities regarding the words occurring in the relatively short queries that are posed by users.

Consequently, we restrict our appeal to use natural language as a means for querying information to search engines operating on limited and well-defined domains. In combination with some interface design considerations, many of the problematic arguments mentioned above can be eliminated or at least weakened. Primarily, the potential ambiguity of words can be largely alleviated since meanings of polysemous words such as the often-cited *bank* (e.g. a bank of a river or financial institute) are less likely to occur in a restricted domain. Additionally, the conceptual coverage of a natural language search interface operating on a specialized domain is better defined. Establishing clarity regarding the linguistic coverage lies in the responsibility of the interface design to provide sufficient clues about the system's capabilities. For the task of natural language query analysis we followed the assumption that shallow natural language processing is sufficient in restricted and well-defined domains (Nielsen, 1993).

Limiting the domain bears another advantage, namely the feasibility of creating a domain ontology that can be used to improve search results by incorporating knowledge about semantic relations between concepts into the system. Ontologies go hand in hand with information retrieval systems based on natural language, because they can be used to model linguistic relations as well. Furthermore, using semantic knowledge to support the information retrieval task can address some of the assumptions regarding the intelligence of such a system as mentioned above. Taking additional information into account, which is present in the ontology but not provided by the query, can be interpreted as intelligence from the user's point of view, when this information is presented accordingly.

Consider, for example, the context of tourism information systems where intuitive search functionality plays a crucial role for the economic success. Querying an information system in natural language is especially appealing in the tourism domain because users usually have very different backgrounds regarding computer literacy. This inhomogeneous mix of users mainly forms due to the circumstance that, pragmatically speaking, almost anybody is a tourist sometimes. Figures provided by the largest Austrian web-based tourism platform *Tiscover*⁴ (Pröll et al., 1998) show that the number of reservation requests and bookings has increased from 242,953 in the year 1999 to 868,203 in 2002. To give a more general view, estimates presented by Schuster (1998) say that in 10 years time, about 30% of all tourism business will be conducted via Internet. This indicates the potential of using natural language as a means for searching for tourism information.

Hence, we have developed a natural language interface for *Tiscover*, which is a well-known tourism information system and booking service in Europe that already covers more than 50,000 accommodations in Austria, Germany, Liechtenstein and Switzerland. It integrates a variety of additional services like live weather reports, event booking, special holiday package offers, route planning or a jobmarket. More specifically, our natural language interface allows users to search for accommodations throughout Austria by formulating the query in a natural language sentence either in German or English with the language of the query being automatically detected and the result presented in the respective language. A preliminary version of the system has first been reported in Berger et al. (2001) and described thoroughly in Dittenbach et al. (2003a) and Berger et al. (2003b).

During 10 days of March 2002, we tested the assumptions behind the natural language interface in a field trial where the interface was accessible via a hyperlink from the *Tiscover* homepage (Dittenbach et al., 2002a,b). Furthermore, we conducted a usability study to find out how such a system is approached by users and to get feedback in order to be able to derive valuable information that we can use for further research and development. As a result of both, the field trial and the usability study, a follow-up system has been developed founding on a rather different semantic knowledge representation and query matching approach

⁴<http://www.tiscover.com>

(Berger et al., 2003a). Dealing with a more theoretical aspect of the system regarding the ontology and its construction, a neural network-based approach has been employed to visualize semantic relations in the vocabulary extracted from domain-related texts (Dittenbach et al., 2003b). Moreover, we have adapted a term weighting technique and used a term distribution model to highlight words that are specific for a certain geographical region and that might be considered for inclusion in the ontology.

The remainder of this thesis is structured as follows. Chapter 2 provides information about the data of our specific application domain of tourism, to make clear and delimit its complexity and size. Then, we describe the architecture of the first version of our natural language interface that has been used for the field trial and usability study detailed in Chapter 3. The insights gained from the real-world queries obtained during the field trial stimulated the development of a second version based on associative networks and spreading activation. This approach differs from the first system primarily in the way the ontology is structured and how a query is processed to retrieve the relevant information from the database.

The first part of Chapter 3 deals with the field trial we have conducted by making the search interface publicly available and promoting it on the *Tiscover* homepage. We present a detailed analysis of the queries that were posed during the time of the field trial by examining some statistical properties of the data, e.g. average query length, number of domain-relevant concepts or the number of conjunctions and adverbial phrases connecting these concepts. The second part of the chapter describes a supervised usability study in which our natural language interface is compared with the original *Tiscover* accommodation search interface.

In Chapter 4 we discuss methods, how information in free-form text documents can be used to enrich domain ontologies. In particular, we present an approach based on neural networks to display a map showing the vocabulary extracted from such texts, which is automatically organized according to semantic similarities between words. This map can serve as a supporting tool for ontology engineers. We illustrate this method from the tourism domain where we semantically cluster words that occur in textual descriptions of accommodations.

Furthermore we present an approach derived from a common term weighting technique in information retrieval that is especially suitable for the tourism domain to detect terms that are characteristic for particular geographical regions.

Finally, Chapter 5 recapitulates the findings that emerged during the various stages of the research presented in this thesis as well as provides an outlook to future research fields related to this subject matter.

Chapter 2

Ad.M.In: A Natural Language Information Retrieval System

2.1 Introduction

Natural language interfaces have a long tradition in computer science regarding the young age of the discipline. Over 30 years ago, the first prototypes were developed predominantly as a means for accessing database systems by experts. At that time, end-users were not in the scope of the developers of these systems. In the course of time, quite a number of natural language interfaces have been developed but never found broad acceptance. Androutsopoulos et al. (1995) give a nice and concise overview of the history of natural language interfaces to databases.

grobe einteilung: pattern matching, syntax-based, semantic grammar, intermediate lang. vielleicht (Lewis and Spärck Jones, 1996)

In this chapter, we present **Ad.M.In**, providing a multilingual natural language interface for accommodation search. The system can be characterized as a combination of a pattern matching and a semantic grammar approach where some rules are used to identify relevant concepts in the query together with an ontology defining the semantics of the concepts. Although, we present the system applied to the domain of tourism, domain independence was one of the preeminent goals of the system design. Consequently, the knowledge about the domain, in form of an ontology and rules for mapping concepts to, is separated from the processing

logic to allow for easy portability to other application domains.

Because of the many different languages spoken in Europe and the intrinsic internationality of the domain, providing multilingual access to information is especially appealing in the field of tourism. Therefore, we used a technique to automatically determine the language of the query that can be posed in any of the languages supported by the system, i.e. German and English at the current state of the prototype. Depending on the language of the query, different chains of processing units are activated that transform the natural language query into the structured query language (SQL). We rely on the sufficiency of shallow language processing when operating on a well-defined and limited domain. Finally, the query results are presented according to the needs of the client device used for posing the query.

As a consequence of the field trial and the usability study described in the next chapter, a second system has been developed based on associative networks and spreading activations. This approach uses a different approach for modeling the semantic relations of concepts present in the domain knowledge that facilitates the definition of more general concepts.

GETESS, described by (Staab et al., 1999), is a system that gathers information from multiple sources on the Internet, extracts semantic relations, and provides a uniform natural language interface to query this information. The system also operates on restricted domains and one of the sample application domains is tourism.

The MIETTA project focuses on multilingual access to tourism information provided by different sources in different languages (Xu et al., 2000). In particular, after translating the documents to the xxx the relevant information is extracted from the webpages by using templates that are filled by processing natural language documents using shallow parsing.

Mädche and Staab (2002) propose the application of semantic web technologies for tourism information systems to bridge the gap between isolated sources of information and provide the means for a semantics-supported approach for searching tourism information that is spread across the Internet.

The remainder of this chapter is organized as follows. First, in Section 2.2 we present the tourism data that our search interface operates on. Then, the

Ad.M.In system is detailed in Section 2.3 followed by a description of an alternative information retrieval approach based on associative networks in Section 2.4. The chapter concludes with some discussions presented in Section 2.5.

2.2 Tourism Data

To gain a better understanding of the complexity and size of the particular application domain presented herein, we first provide a description of the tourism data. The database that can be queried via our natural language information retrieval system consists of a part of the *Tiscover* database, which, as of October 2001, provided access to information about 13,117 accommodations. To avoid any confusion with the current state of the *Tiscover* system, it has to be noted that in the remainder of this thesis we will talk about our specific snapshot of the data taken in October 2001 if not stated otherwise. Since *Tiscover* is one of the largest European web-based tourism information portals and booking services, it provides much more information than we are using for demonstrating the potential of our search interface. *Tiscover* already covers more than 50,000 accommodations in Austria, Germany, Liechtenstein, Switzerland and Italy and integrates a variety of additional services like live weather reports, event booking, special holiday package offers, route planning or a job market.

However, the accommodations are described by 82 features including various facilities, services, accommodation type, location, suitability, the types of dining and catering. These features are shown in Table 2.1. Furthermore, they have certain numbers of various room types and are categorized from zero to five stars, or zero, two, three and four flowers ("*Blumen*") in the case of farms. The numerical attributes are not included in the table.

The accommodations are located in 1,923 towns and cities that are again described by various features, mainly information about possible sports activities, e.g. mountain biking or skiing. These 50 city attributes are presented in Table 2.2. Additionally, details like the number of inhabitants or the sea level are stored in the database. The federal states are the higher-level geographic units. In between the city and the federal state level, our system lacked information about regions. The field trial described in Section 3.3 will show that this would

Facilities	Services	Recreation	Location
hotel bar	office services	steam bath	outskirts of village
beauty farm	internet access	bicycle rental	close to a lake
discotheque	skier's shuttle	workout program	close to a river
floor service	railroad station shuttle	golf	next to thermal bath
TV room	airport shuttle service	indoor swimming pool	center
hairstresser	english spoken	bowling alley	ski-in/ski-out
garden/private	french spoken	massages	
pets welcome	dutch spoken	horseback riding	Suitable for
air condition	italian spoken	ski lift	groups
reading lounge	spanish spoken	tanning beds	seniors
indoor car park	telephone service	dance	phys. handicapped persons
car park	babysitter	indoor tennis facilities	children
restaurant		tennis court	
playground	Type	table tennis	Dining
toys	hotel	health club	spa cuisine/health foods
meeting rooms	pension	miniature golf	fresh farm produce
washing machines	farm	squash	international
guest lounge	private accommodation	jacuzzi	specialties of the region
ice maker	holiday flat	dry heat sauna	organic foods
safe deposit	guest house	swimming pool	austrian specialties
elevator	youth hostel		vegetarian
playroom		Catering	
		no catering	
		breakfast only	
		half board	
		full board	

Table 2.1: List of accommodation-specific properties in the database.

have been valuable information, because, except for large cities, users searched for accommodations in tourist-relevant regions rather than in specific towns. These geographic regions are independent of the borders of the federal states and have sometimes naturally grown out of history and have sometimes been defined by the machinery of tourist marketing. Consider as an example the *Salzkammergut*, which is mainly located in Upper Austria but also overlaps into Salzburg and Styria. For a part of the data, more precisely, for most of Tyrol, we have integrated the geographic coordinates of the cities and towns to additionally provide rough information about the distance between places. Therefore, the system can be queried for accommodations close to a certain place as will be shown later in Subsection 2.3.3.

Location	beach volleyball	paragliding	rafting	squash
close to mountains	archery	golf	horseback riding	diving
lakeside	canyoning	roller skating	toboggan	tennis
riverside	curling	hunting	rowing	table tennis
am skigebiet	hang gliding	kayaking	snowshoeing	skeet shooting
	ice skating	bowling	swimming	trekking
Activity	curling	climbing	soaring	hiking
alpine skiing	cycling	miniature golf	sailing	water skiing
fishing	parachuting	motocross	cross country skiing	sailboarding
badminton	fencing	aquatic motor sports	snow boarding	
ballooning	fitness training	mountain biking	summer toboggan	

Table 2.2: List of city-specific location and activity features.

2.3 Original System

2.3.1 System Design

multilinguality,
no elliptical
sentences

One of the primary design goals of the system was domain independence, i.e. it should also be possible to apply the natural language interface to different domains other than tourism. To achieve this, it was necessary to strictly separate domain knowledge from program logic. Furthermore, we implemented a pattern quite common in natural language processing, where the natural language query is being processed by different modules one after another. In software engineering this architectural pattern is known as *Pipes and Filters* (Buschmann, 1996). The query is passed through *pipes* from one processing module (*filter*) operating on the data to the next. Since the system can handle queries posed in more than one language, defining separate processing pipelines is useful, because different languages have different requirements regarding their analysis. Because of a variety of good stemming algorithms available for the English language, e.g. the well known Porter stemming algorithm (Porter, 1980), it would make sense to have a stemming module to remove common morphological and inflectional endings from words, for example, reducing *facilities* and *facility* to their stem *facilit*. Stemming has the advantage, that a person creating the ontology does not have to consider the various word forms, but only the stems. Although modified versions of various English stemming algorithms exist for the German language, the morphological complexity seems to be a pitfall for a stemmer, thus hindering the achievement of a reasonable accuracy. Hence, the pipeline for the German language does not contain such a module.

An additional criterion for pipeline selection is the input medium, e.g. whether a web-based interface or speech was used to pose the query. In the latter case a spell checking module would not be necessary, because a speech recognition engine is usually not prone to typographical errors. As can be seen in Figure 2.1, in the first step the language is identified. Depending on the query language and the medium used for input, the according pipeline is activated.

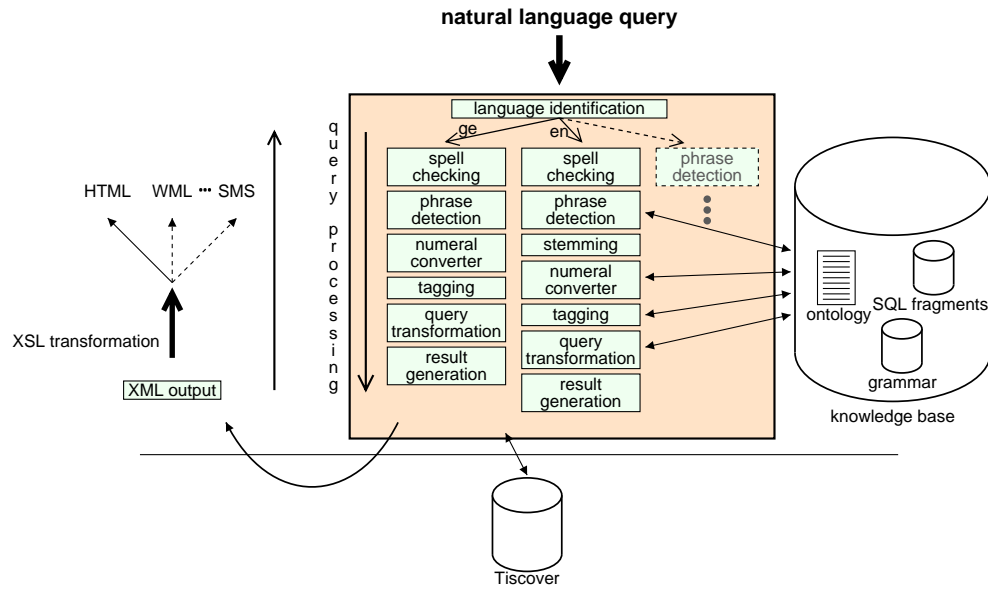


Figure 2.1: Overview of the system architecture.

In our current implementation the various modules check the spelling, detect phrases¹, convert numerals to numbers, tag elements of the query, transform the query into SQL and generate the appropriate output that is presented to the user. Some of these modules interact with various parts of the knowledge base as described in the next subsection. The single processing steps will be detailed thereafter.

¹For readers proficient in linguistics it has to be noted that we use the term *phrase* not in its strict sense, but rather use it for describing names or nouns consisting of multiple words.

2.3.2 Knowledge Base

With the term *knowledge base*, we refer to the sum of the domain-specific knowledge stored in our system. It comprises an *ontology*, i.e. a specification of the domain-relevant concepts and their relations (Gruber, 1993), information about proper names occurring in the domain, a simple grammar and the definition of parameterized SQL fragments. The ontology and the grammar are stored separately for each language supported by the system.

structure of
onto/kb,
erkläerung:
concepts

Ontology

To describe the ontology in a simple and straightforward manner, we have defined an XML document type and abandoned using feature-rich markup languages such as DAML+OIL (Horrocks, 2002) or OXML². Our ontology defines the domain-relevant concepts, numerals, conjunctions, prepositions and adjectival and adverbial phrases that can act as modifiers of concepts. These linguistic components are represented by *classes*, the basic units in an ontology. With each of the classes, the respective synonyms and syntactic cases are also defined. For example, the class *information* explains that ‘*alpine skiing*’ is an activity that can be carried out in a city or that ‘*organic food*’ is a dining option associated with an accommodation.

more examples
(mit syns),
complexity
issues,

Grammar

The grammar defines which prepositions, adjectival or adverbial phrases can modify the meaning of which concept and furthermore links the ontology with the parameterized SQL fragments.

way too short

SQL Fragments

The parameterized SQL fragments define parts of the final SQL statement with certain operators or operands being variables that are assigned values depending on the matching information determined by the grammar.

²OXML is an ontology representation language for OntoEdit[©] (<http://www.ontoprise.de>)

To illustrate this part of the knowledge base with an example, consider the following SQL fragment regarding city selection as an example:

```
SELECT entity."EID" FROM entity WHERE
  entity."CID" = city."CID" AND
  city."Name" @OP1 '@PARAM1'
```

Depending on modifying terms found in the query as specified in the grammar, the SQL fragment is selected and the parameters are substituted with the appropriate values. The query for an accommodation in Innsbruck produces the following SQL fragment.

```
SELECT entity."EID" FROM entity WHERE
  entity."CID" = city."CID" AND
  city."Name" ~* '^(.*)?Innsbruck(.*)?$',
```

Searching for an accommodation *in* Innsbruck led to the substitution of operator @OP1 with '~*' denoting a regular expression match, a '*not in*' in front of Innsbruck would have resulted in '!~*'. It has to be noted that we sometimes use non-standard SQL statements and functions we are using *PostgreSQL*-specific features. To give an example, we use regular expression matching functionality where !~* denotes a case-insensitive regular expression mismatch.³ In the case of looking for an accommodation *close to* Innsbruck, an entirely different SQL fragment would have been chosen by the grammar, since the statement has to perform more complex operations. An example will be given in the next subsection.

2.3.3 Query Processing

Language Identification

To identify the language of a query, we use an *n-gram*-based text classification approach (Cavnar and Trenkle, 1994) where each language is represented by a class. An *n-gram* is a character slice of length *n* of a longer character string.

³Since the technical details of the database system are not important for the research presented here, we omit them in this thesis. However, we refer the reader interested in these syntactic details to the documentation at <http://www.postgresql.org>.

As an example, for $n = 3$, the *tri-grams* of the string ‘*language*’ are: {*_la*, *lan*, *ang*, *ngu*, *gua*, *uag*, *age*, *ge_*}. Dealing with multiple words in a string, whitespace characters are usually replaced by an underscore ‘*_*’ and are also taken into account for the construction of an n -gram document representation.

This language classification approach using n -grams requires a sample text for each language to build statistical models of the languages, i.e. n -gram frequency profiles. We used various tourism-related texts, e.g. hotel and holiday package descriptions, as well as news articles both in English and German. The n -grams, with n ranging from 1 to 5, of these sample texts were analyzed and sorted in descending order according to their frequency, separately for each language. These sorted histograms are the n -gram frequency profiles for a given language.

As an example, the top ten tri-gram occurrences in the German and English language texts are shown in Table 2.3. In the English texts, it can be seen that ‘*the*’, ‘*and*’, ‘*_of*’, ‘*_in*’ and the ending ‘*ion*’ are the most frequent tri-grams. Contrarily, in the German texts, the most frequent tri-grams are, for example, ‘*der*’, ‘*ich*’ or ‘*ein*’ as well as word endings like ‘*en_*’, ‘*er_*’, ‘*ie_*’ or ‘*ch_*’.

German		English	
en_	1786	_th	1333
er_	1570	the	1142
de	949	he	928
der	880	_of	592
ie_	779	of_	575
ich	763	_an	439
ein	730	nd_	407
sch	681	_in	389
ch_	642	ion	385
che	599	and	385

Table 2.3: Top ten tri-gram occurrences of German and English text (underscores represent blanks).

To determine the language of a query, the n -gram profile, $n = 1 \dots 5$, of the query string is built as described above. The distance between two n -gram profiles is computed by a simple rank-order statistics. For each n -gram occurring in the query, the difference between the rank of the n -gram in the query profile and the rank in a language profile is calculated. For example, the tri-gram ‘*the*’ might be at rank five in a hypothetical query but is at rank two in the English

language profile. Hence, the difference in this example is three. These differences are computed analogously for every available language.

The sum of these differences is the distance between the query and the language in question. Such a distance is computed for all languages, and the language with the profile having the smallest distance to the query is selected as the identified language, in other words, the language of the query. If the smallest distance is still above a certain threshold, it can be assumed that the language of the query is not identifiable with a sufficient accuracy. In such a case the user will be asked to rephrase her or his query. Obviously, displaying a list of supported languages to choose from would be an alternative. Generally, it can be said that this approach works reasonably well as long as the query is not all too short, because then the query profile is not significant enough from the statistics point of view.

Error Correction

To improve the retrieval performance, potential orthographic mistakes have to be considered in our web-based interface. After identifying the language as described above we use a spell checking module to determine the correctness of the query terms. The efficiency of the spell checking process improves during the runtime of the system by learning from previously posed queries. The spell checker uses the *metaphone* algorithm (Philips, 1990) to transform the words into their soundalikes. Because this algorithm has originally been developed for the English language, the rule set defining the mapping of words to the phonetic code has to be adapted for other languages. In addition to the base dictionary of the spell checker, domain-dependent words and proper names like names of cities, regions or states have to be added to the dictionary. These words and names are extracted from the ontology and the corresponding fields in the database and added to the dictionary.

For every misspelled term of the query, a list of potentially correct words is returned. First, the misspelled word is mapped onto its metaphone equivalent, then the words in the dictionary, whose metaphone translations have at most an edit distance of two, are added to the list of suggested words. The suggestions are ranked according to the mean of

- the edit distance between the misspelled word and the suggested word, and
- the edit distance between the misspelled word's metaphone and the suggested word's metaphone.

The edit distance is the minimum number of insertions, deletions and substitutions required to transform one string into another (Levenshtein, 1966). The smaller this value is for a suggestion, the more likely it is to be the correct substitution from the orthographic or phonetic point of view. However, this ranking does not take domain-specific knowledge into account.

Because of this deficiency, correctly spelled words in queries are stored and their respective number of occurrences are counted. The words in the suggestion list for a misspelled query term are looked up in this repository and the suggested word having the highest number of occurrences is chosen as the replacement of the erroneous original query term. In case of two or more words having the same number of occurrences the word that is ranked first is selected. If the query term is not present in the repository up to this moment, it is replaced by the first suggestion, i.e. the word being orthographically or phonetically closest. Therefore, suggested words that are very similar to the misspelled word, yet make no sense in the context of the application domain, might be rejected as replacements. Consequently, the word correction process described above is improved by dynamic adaptation to past knowledge.

Detection of Phrases and Proper Names

An important issue in interpreting natural language queries is to detect names and concepts consisting of multiple words. Proper names like '*St. Johann im Pongau*' or substantives like '*steam bath*' have to be treated as one element of the query. We use regular expressions to identify such cases and to deal with different possible spellings of parts of names like, e.g. *St.* or *Sankt*.

Another inevitable issue in the field of tourism are proper names since this class of linguistic entities are, first, important search criteria, and second, representing more than 20% of the total occurring words (Velardi et al., 2001). Hence, the correct identification is crucial for retrieving relevant search results. Regarding multilinguality it is rather trivial for German and English, because only some

place or city names have to be held in a simple translation table, e.g. *Wien*, *Vienna* or *Niederösterreich*, *Lower Austria*, to name but a few. A more complex task would be incorporating a language into the system in which proper names are effected by a rich inflectional morphology such as Czech. Consider the name of the city of Vienna as an example. On its own, the Czech equivalent would be *Vídeň*. If you are going *to* Vienna the translation is ‘*do Vídně*’, and if you are looking for an accommodation *in* Vienna the correct form is ‘*ve Vídni*’. Despite a few regularities in building the forms of place names, most of these cases have to be considered separately.

Tagging and SQL Mapping

With the underlying relational database system, the natural language query has to be transformed into a SQL statement to retrieve the requested information. The query terms are tagged with class information, i.e. the relevant concepts of the domain (e.g. *hotel* as a type of accommodation or *sauna* as a facility provided by a hotel), numerals or modifying terms like *not*, *at least*, *close to* or *in*. If none of the classes specified in the ontology can be applied, the database tables containing proper names have to be searched. If a name is found in one of these tables, it is tagged with the respective table’s name, such that *Vorarlberg* will be marked as a federal state.

In the next step, this class information is used by the grammar to select the appropriate SQL fragments. Finally, the SQL fragments have to be combined to a single SQL statement reflecting the natural language query of the user. The operators combining the SQL fragments are again chosen according to the definitions in the grammar.

Examples

As our first example consider the following English query: “*I am looking for a hotl in St. Abton am Arlberg with sauna and a swiming pool. The hotel should furthermore be suitable for children and pets should be allowed*”. As can be seen, the query contains several misspellings such as ‘*hotl*’, ‘*Abton*’ and ‘*swiming pool*’. In the case of ‘*Abton*’, our improved spell checking mechanism does not choose the

word ‘*Baton*’, which is ranked first in the list of suggested corrections, but instead chooses ‘*Anton*’. This selection is performed because of a previously posed query, where ‘*St. Anton am Arlberg*’ has been spelled correctly.

For our second example we use the following German query: “*Ich brauche ein Einzelzimmer mit Frühstück in einer Pensoin in der Nähe von Innsbruck aber nicht in Innsbruck selbst*”. This query, again with misspellings, shows the effect of different prepositions modifying a noun. The query states that a pension with breakfast close to Innsbruck but not in the city of Innsbruck is searched for. The first occurrence of *Innsbruck* is preceded with *close to* and therefore the following SQL fragment is constructed:

```
SELECT entity."EID" FROM entity WHERE
  entity."CID" IN (SELECT b."CID" FROM city AS a, city AS b WHERE
    a."DEC_Lat" != 0 AND a."Name" ~* '^(* )?Innsbruck( .*)?$' AND
    (|/(((a."DEC_Lat" - b."DEC_Lat")^2) +
      ((a."DEC_Long" - b."DEC_Long")^2)) <= 0.13489734))
```

This statement is based on the assumption that *close to* means within radius of approximately 15 kilometers beeline. This range can be adapted by the user to her or his particular needs. Regarding the second occurrence of *Innsbruck*, the identification of *in* before the city name leads to the following SQL fragment:

```
SELECT entity."EID" FROM entity WHERE
  entity."CID" = city."CID" AND
  city."Name" !~* '^(* )?Innsbruck( .*)?$'
```

The negation *nicht (not)*, preceding ‘*in Innsbruck*’, determines the operator that will be applied to merge the two sets of accommodations retrieved by the above SQL fragments. In this particular case, the pensions in the result set will be situated close to Innsbruck except those located in Innsbruck directly.

2.4 Enhanced System

The analysis of real-world queries received during a field trial, which is described in the next chapter, has shown some deficiencies regarding the ontology of the

original system. First, modeling fine-grained similarity relations between concepts is rather difficult with the ontology of the original system since no degree of similarity between two concepts could be defined. Additionally, with the rather fixed structure defining how the concepts are organized, some of the concepts the users asked for could be implemented either, only in a cumbersome way or not at all. Especially highly subjective search criteria such as *romantic* posed a difficult task. Hence, we have developed a more flexible way of representing domain knowledge that suits the needs of our natural language information retrieval system.

2.4.1 Associative Networks and Spreading Activation

Semantic networks as introduced by Quillian (1968) have played an important role in the field of knowledge representation. The basic elements that constitute a semantic network, which is in fact a directed graph, are concepts, their attributes and hierarchical sub-superclass relations between concepts. The concepts are linked via *is-a* or *instance-of* relations. The higher a concept is located in the hierarchy, the more abstract it is regarding its meaning. Attributes that are assigned to a higher-level node are inherited to sub-concepts.

Semantic networks used in information retrieval are usually referred to as *associative networks* because of a slightly different structure. An associative network is a generic network consisting of pieces of information represented by nodes that are connected with either unlabeled or labeled links that can also be weighted to express a certain strength of the relations. Associative networks have quite a tradition in information retrieval and were first used to model relations between terms and terms, between terms and documents and between documents and documents.

A processing framework for associative networks is *spreading activation*, which emerged from the field of cognitive sciences. The basic idea is to distribute activation potentials expressed by numerical values along the directed, weighted links connecting the nodes of the network. This is usually an iterative process, where during one iteration a *pulse* is triggered and a termination criterion is checked. In other words, the activation is transferred from activated nodes along

the directed connections to the immediately adjacent nodes until, e.g., a certain number of iterations is reached.

In information retrieval, spreading activation has been used to process associative networks in order to retrieve a ranking of relevant information with regard to a search request (Cohen and Kjeldsen, 1987; Salton and Buckley, 1988a; Crestani, 1997). Pragmatically speaking, the initial activations in the network are assigned to nodes that represent the terms in the query. Then, for a certain number of iterations the activation is spread across the network and the documents are ranked according to the final activation potentials.

In practice, the spreading process is more complex than just sending activations via weighted links. Without implementing constraints that control the activation flow, the network ends up with all nodes being uncontrollably activated. Hence, several spreading constraints that are commonly used in such an information retrieval systems have to be mentioned. First, the *distance constraint* gradually reduces the activation that is passed on to nodes being farther away from the initially activated node. This also complies with the decreasing semantic similarity between concepts that are only connected via one or more intermediate concepts. Second, to avoid the disproportionate activation of neighbors of highly connected nodes, the *fan-out constraint* is used. Third, *path constraints* can be imposed to control the flow of activation depending on the type of links according to application-specific requirements. Fourth, a threshold defining a certain amount of activation necessary for further propagation is called *activation constraint*. For a more detailed discussion about constrained spreading activation, see the works of Preece (1981) and Crestani and Lee (2000).

2.4.2 Redesigned Knowledge Base

To provide an illustrative example, a very small part of the redesigned knowledge base is depicted in Figure 2.2. The associative network can primarily be divided into three layers. The *conceptual layer* contains concepts that are actually covered in the database, i.e. the properties of accommodations and cities as shown in Table 2.1 and Table 2.2. The links connecting the concepts are unlabeled and have been weighted manually according to the strength of the relation. Furthermore,

the geographic information originally stored in the database is also incorporated into the conceptual layer. Hence, the concept *Austria* is connected to the concepts representing the nine federal state and those, in turn, have connections to the respective cities. Additionally, the rather arbitrary definition of *near* or *close* meaning *within 15 kilometers*, has been resolved by creating connections between neighboring cities. Due to the nature of the constrained spreading activation algorithm, neighboring cities will be activated higher than those being farther away, i.e. being connected via intermediate nodes (cities). This geographical relations have been modeled for a part of Tyrol where we had the geographical data of the cities. Each of the concepts in the conceptual layer is associated with a list of synonyms and, due to the lack of a stemmer, also the morphological variants thereof. This organization of concepts and the actual terms representing them also facilitates the support of multiple languages, because only the list of words has to be created separately for each language. There might exist some cultural issues where differently organized associations would be necessary for different languages, but we will disregard this possibility in the scope of this work.

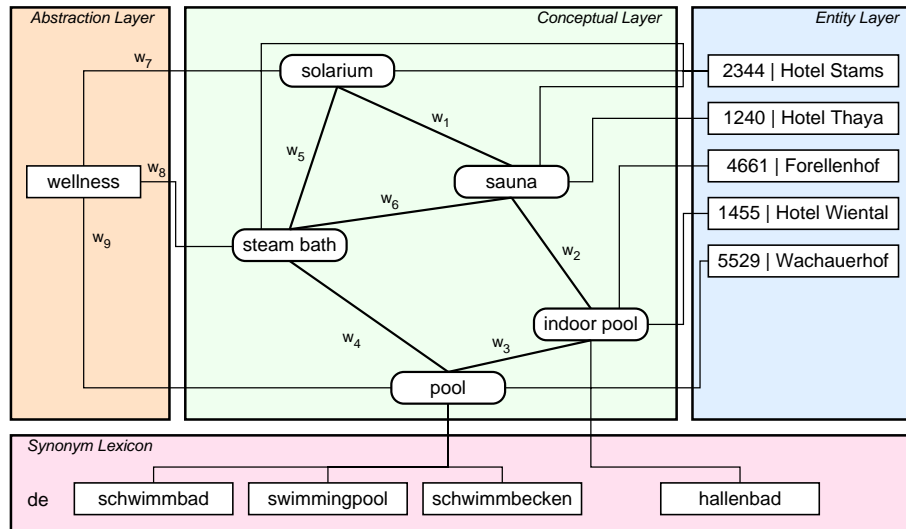


Figure 2.2: Architecture of the redesigned ontology.

The *abstraction layer* allows the definition of more general concepts we further refer to as *abstract*. In Figure 2.2, the abstract concept *wellness* is used as

example. *Wellness* does not have an attribute modeled in the database as a direct counterpart, but several facilities and services are representing it, e.g. *solarium*, *steam bath* or *pool* in the illustration. Consequently the abstract concepts are connected with nodes in the conceptual layer.

The third layer of this architecture is the *entity layer* where the actual accommodations are stored in. The accommodations are represented by nodes and are connected to the respective concepts they are related to. These can either be facilities or services the hotel offers, but also the city they are located in. The figure shows, for example, that the *Hotel Stams* offers a sauna, a solarium and a steam bath, whereas the *Wachauerhof* only provides a pool.

Due to the flexible pipeline structure of the original system, replacing the according query processing modules by the spreading activation modules required only little effort. Other modules such as the spell checker, phrase identifier or the numeral converter were reused without altering them. Hence, after processing the query as described in the previous section, our implementation of the spreading activation process is used to rank the accommodations according to their relevance. In the initialization phase of the spreading activation process the following steps have to be performed. First, with all activations being zero, a certain activation level has to be added to the concepts that have been identified in the query and that are present in the conceptual layer. Second, abstract concepts occurring in the query have to be resolved by way of following the weighted links to concepts in the conceptual layer. Again, a certain value is added to the associated concepts. Hence, it is possible that the activation level of a concept is increased multiple times, e.g. due to the explicit occurrence of a certain word in the query and the occurrence of an abstract concept relating to it. In both steps mentioned above, negating modifiers such as *not*, *without* or *no*, intended for excluding certain concepts in the query, have to be considered. We multiply the activation with a negative value to create an inhibitory activation potential.

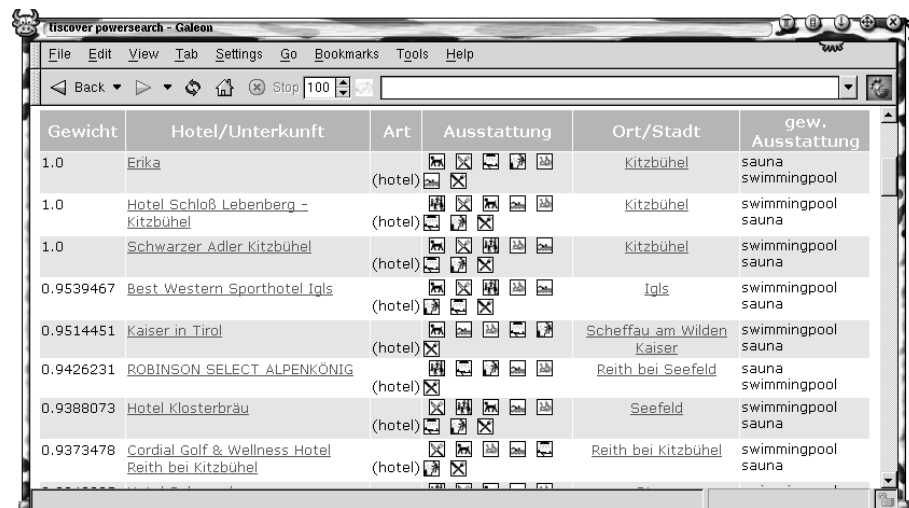
Then, the spreading activation takes place by using a propagation rule defined in Equation 2.1 implementing several of the spreading constraints mentioned above.

$$O_i(p) = \begin{cases} 0 & \text{if } I_i(p) < \tau, \\ \frac{F_i}{p+1} \cdot I_i(p) & \text{otherwise, with } F_i = (1 - \frac{C_i}{C_T}) \end{cases} \quad (2.1)$$

The output $O_i(p)$ of a unit i at iteration (pulse) p is defined as the fraction of a fan-out value F_i and $p + 1$ denoting a distance constraint, multiplied by the current activation level $I_i(p)$, if $I_i(p)$ is above a certain threshold τ . This threshold acts as an activation constraint. The fan-out constraint F_i for a certain unit is determined by the fraction of the number of concepts C_i that are connected to the unit and the total number of concepts C_T in the network. This minimizes the effect of concepts having a broad semantic meaning, i.e. which are connected to a large number of other concepts, to activate the whole network. The denominator $p + 1$ of the fraction has the effect that the activation diminishes with further progress of the spreading pulses. In other words, units that are located farther away from the initially activated units are activated to a lesser extent.

With every pulse, the activation levels $I_i(p)$ of all units in the concept layer are added to an internal relevance variable of the respective concepts in the entity layer they are associated with. When a certain number of iterations is reached, the accommodations are ranked according to their relevance value. Consequently, the higher the level of agreement between the query and the accommodation, the higher it is ranked.

Consider the search results presented in Figure 2.3 as an example. The query translates to “*I am looking for a wellness hotel in Kitzbühel with sauna and swimming pool.*” The concepts in the conceptual layer that were directly activated were, *Kitzbühel*, *sauna* and *swimming pool*. The the abstract concept *wellness hotel* additionally activated the concept *hotel* and several other wellness-related facilities. The spreading activation process has ranked three hotels first, which are directly located in Kitzbühel. The following accommodations offer the same facilities as well but are ranked according to their location and additional wellness-related feature they provide. For instance, the hotel *Best Western Sporthotel Igls* is ranked higher than the hotel *Cordial Golf & Wellness Hotel Reith bei Kitzbühel* even though the latter is located closer to Kitzbühel. This happened, because the first offers more wellness-related facilities and services that outweigh the smaller distance between Kitzbühel and the city, the second hotel is located in.



Gewicht	Hotel/Unterkunft	Art	Ausstattung	Ort/Stadt	gew. Ausstattung
1.0	Erika	(hotel)		Kitzbühel	sauna swimmingpool
1.0	Hotel Schloß Lehenberg - Kitzbühel	(hotel)		Kitzbühel	swimmingpool sauna
1.0	Schwarzer Adler Kitzbühel	(hotel)		Kitzbühel	swimmingpool sauna
0.9539467	Best Western Sporthotel Igls	(hotel)		Igls	swimmingpool sauna
0.9514451	Kaiser in Tirol	(hotel)		Scheffau am Wilden Kaiser	swimmingpool sauna
0.9426231	ROBINSON SELECT ALPENKÖNIG	(hotel)		Reith bei Seefeld	sauna swimmingpool
0.9388073	Hotel Klosterbräu	(hotel)		Seefeld	swimmingpool sauna
0.9373478	Cordial Golf & Wellness Hotel Reith bei Kitzbühel	(hotel)		Reith bei Kitzbühel	swimmingpool sauna

Figure 2.3: Results of the spreading activation-based system.

This shows, on the one hand, the influence of the weights on the ranking and the potential difficulties in defining how strong a concept x is related to a concept y , but on the other hand, this approach also offers an interesting potential for personalization of the system by biasing weights according to some personal profile.

2.5 Discussion

In this chapter we have detailed the multilingual natural language information retrieval system Ad.M.In that, in this particular application domain, allows users to search for accommodations throughout Austria via natural language queries. The queries can be posed either in German or English and the search result is automatically presented in the appropriate language. The system is open for integration of other languages without the need for changing the program logic. We have described the generic architecture of the system that clearly separates domain knowledge from application logic in order to be able to apply the system to other domains.

The focus of our research so far was directed towards a web-based interface where the user types the query into a text box. First experiments with speech-

based I/O have shown that the modular architecture of the Ad.M.In system facilitates the integration in such an environment, but the crucial point for the quality of the search results is the speech recognition engine.

The second version of the natural language interface based on associative networks arose, on the one hand, from some difficulties imposed by the structure of the original ontology design, and on the other hand, from some findings of a field trial that will be described in the next chapter. The formal description of rules when to apply which constraint on the structured query has been exchanged with a network of concepts that are associated with each other via weighted links. The query defines initial energy potentials at specific nodes, i.e. the relevant query components, that are then spread across the network until a certain stopping criterion is reached. The final activations of the concepts determine the activations of the entities (i.e. accommodations) that offer these concepts. In other words, the accommodation that fulfills most of the requirements stated in the query and additionally offers semantically related features or is located in a city nearby, is ranked first, because it has the highest activation. Consequently, the exact-match search strategy has implicitly changed to a best-match retrieval of accommodations.

An advantage of the associative network over the original SQL mapping approach is that certain difficulties regarding the natural language processing are solved implicitly. For example, using the associative network approach relieves us from the burden of discriminating between the conjunctions *and* and *or*. If someone is searching for an accommodation offering a certain facility *a* *and* service *b*, can be treated the same as if someone is querying for facility *a* *or* service *b*, because the according accommodations offering both will be ranked higher either way. The same is true for enumerations of potentially interesting cities the accommodation should be located in. A query of the type “*I am looking for a pension suitable for children either in Feldkirch, Rankweil or Hohenems having a parking garage.*” will be answered with a list where accommodations are ranked first, which fulfill the requirements mentioned and which will be located in either city. In the original system, selecting the correct logical operators connecting the SQL fragments requires a more complex mechanism. Furthermore, modeling the geographical relations by way of the associative network is especially suitable for

the tourism domain.

Chapter 3

The User's Perspective

3.1 Introduction

This chapter deals with the users point of view, describing a field trial we have conducted to test the acceptance of such an interface in the field of tourism. Furthermore, we present a usability study that compares the conventional accommodation search interface of *Tiscover* with our interface.

These first of these two studies was conducted in order to be able to measure whether our assumptions about the required effort that has to be put into natural language processing in a limited domain withstand real world queries. We have used the popularity of *Tiscover* to direct people that are possibly looking for accommodations to our natural language interface by promoting it on the *Tiscover* homepage. Despite testing the functionality of the system, we also wanted to gather a broad range of queries regarding accommodation search, because a large number of people was likely to produce a large number of different search requests. This variety of queries should point out issues and problems that we were not aware of during the construction of the prototype.

Second, a usability study was conducted to get feedback on how the natural language interface compares to the conventional, form-based *Tiscover* interface. This study should provide information about how people approach this alternative possibility to search for information. Furthermore, it should provide data for a qualitative analysis of the natural language-based approach.

The remainder of this chapter begins with details about the design of the user

interface in Section 3.2. Then, we present the field trial in Section 3.3 and the usability study in Section 3.4.

3.2 Design Considerations for the Web-Based User Interface

Our major design goal at the outset of the project was to provide a simple and easy-to-use interface. Hence, the interface is dominated by a text box where the user can enter her or his query and a button for submission, the latter labeled with ‘ask’ (cf. Figure 3.1). We have also implemented the look and feel of the *Tiscover* design in order to avoid distraction from the user’s task. Additionally, we have provided short textual descriptions in both German and English in form of a sample query and a short teaser to stimulate the user’s curiosity. If more languages should be supported, the layout might require reconsideration due to the additional space that would be required. The sample query “*I am looking for a double room in the center of Salzburg with indoor pool.*” is the only hint on what can be searched with the interface. In other words, the capabilities of our information retrieval system were hidden with the major intention not to bias the user’s imagination when formulating a query.

In comparison, Figure 3.2 shows the conventional interface of *Tiscover* for searching accommodations. The area (federal state, region, city) can be chosen either by typing the name directly into the text field or by selecting it by way of a separate page. On this separate page, first, a federal state or region can be selected in a drop-down box (containing a rather exhausting list of 160-odd names), and a filter can be set optionally, defining the type of the place to be searched, i.e. whether it should be a country, province, region or a city. By following a link, misleadingly labeled ‘*show regions*’, a list of matching places is presented as search result that does not contain only regions but usually a lot more cities. In this case, the notion of region as a well-defined entity in the *Tiscover* system has been mixed up with the more general meaning of a geographical area that can be of any size. Then, another option can be applied to filter the list and finally the selected item can be chosen to be the place in the

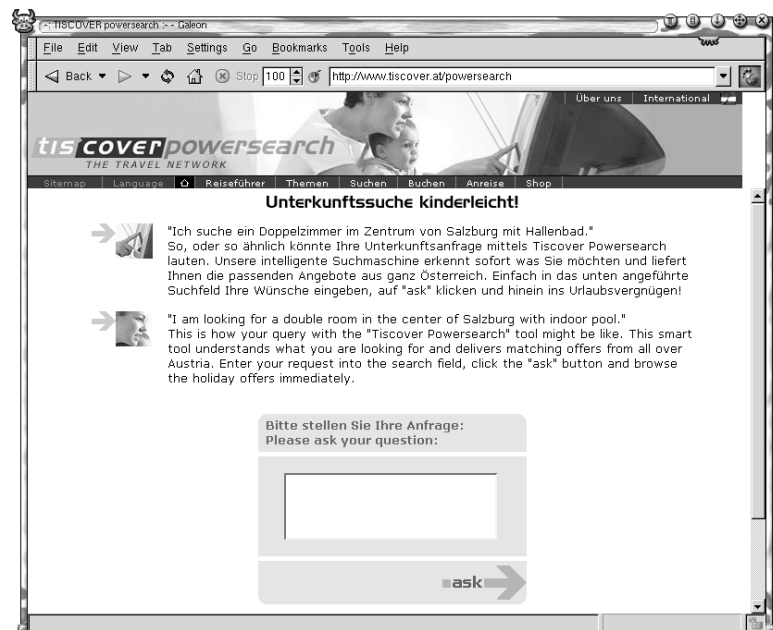
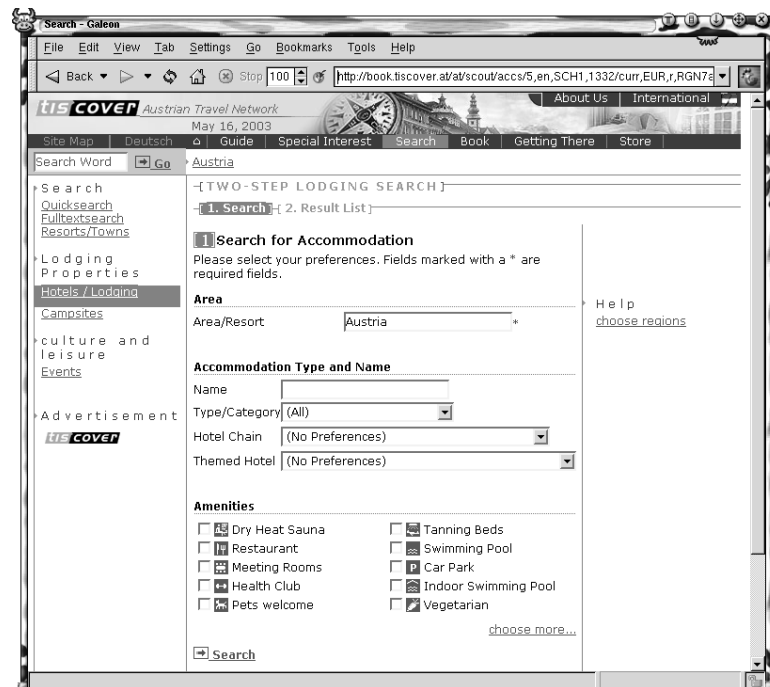


Figure 3.1: Natural language query interface.

original accommodation search form.

Further search criteria are the name of the accommodation, the chain it belongs to (e.g. *Best Western*) and a particular *theme*, e.g. family hotel, as well as several amenities the accommodation should provide. It has to be noted that this list of facilities and services contains only ten items, e.g. dry heat sauna, restaurant, car park or vegetarian, to keep the interface concise. These are the ten characteristics that are most searched for, as an analysis of the weblogs by *Tiscover* has shown. Only recently, the whole list of amenities has again made accessible due to many users' requests who have felt being too limited in their possibilities of selecting search criteria. Hence, the link labeled 'choose more' at the bottom of the page depicted in Figure 3.2 was not present at the time the usability study described later was conducted. The complete list comprising 159 items is shown on a separate page (cf. Figure 3.3) and is approximately three screens long and underlines the importance of providing an alternative search interface that is easier to handle.

Regarding our natural language interface, on the page showing the search results (see Figure 3.4), the original query as well as the concepts identified by

Figure 3.2: Standard *Tiscover* search interface.

the natural language processing are presented to provide the user with feedback regarding the quality of natural language analysis. Below the list of accommodations matching the criteria, a feedback form is provided where users can enter a comment and rate the quality of the result. At the bottom of the page, the input field prefilled with the recent query is presented to allow for convenient query reformulation or refinement.

The presentation of HTML pages is only one form of output supported by the system. We use XML and XSLT technology to create the output according to the needs of the client device. Static content such as the start page is stored in, and dynamic pages such as the result page are generated in XML format and are then transformed into the appropriate output format. Currently, we provide two versions of HTML representation, one for conventional web browsers and one for web browsers running on PDAs with only a very limited display size (240x320 pixels in the case of an iPAQ). Another possible text-based output format would be, for instance, WML. For speech-based I/O, the system, has to generate natural language sentences emitting them via a text-to-speech system.

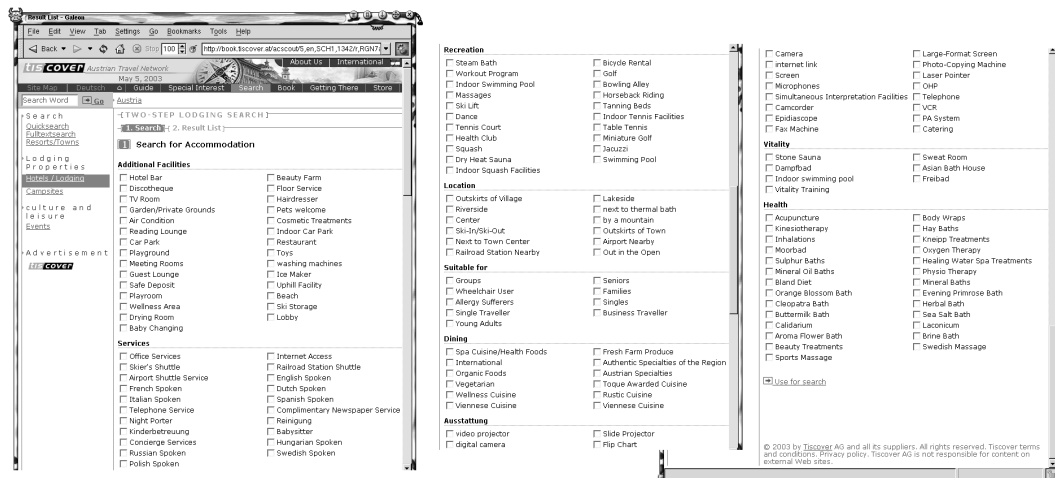


Figure 3.3: Advanced search page of the original *Tiscover* interface presenting all facilities and services that can be used as search criteria (the screenshot is split into three parts).

3.3 Field Trial

3.3.1 Prerequisites

The field trial was carried out using the original Ad.M.In system as described in Section 2.3 from March 15 to March 25, 2002. The time for the trial was chosen deliberately, because close to vacation periods such as the Easter week in our case, the traffic at a web-based tourism information system is usually higher than during other times. During this time our natural language interface was promoted on and linked from the main *Tiscover* page to attract the attention and address the curiosity of potential users.

One of the major objectives of this field trial was to find out whether or not users actually type whole sentences to express their wishes. Considering the relatively low average number of terms users type in conventional search engines (cf. Silverstein et al. (1998)), this seemed rather unlikely. Another important issue was to capture a wide variety of queries regarding accommodation search. As already mentioned in the previous section, only a small hint was provided on the capabilities of the search interface to avoid limiting the imagination of the user. This, admittedly, bore the risk of disappointing the user when no or just

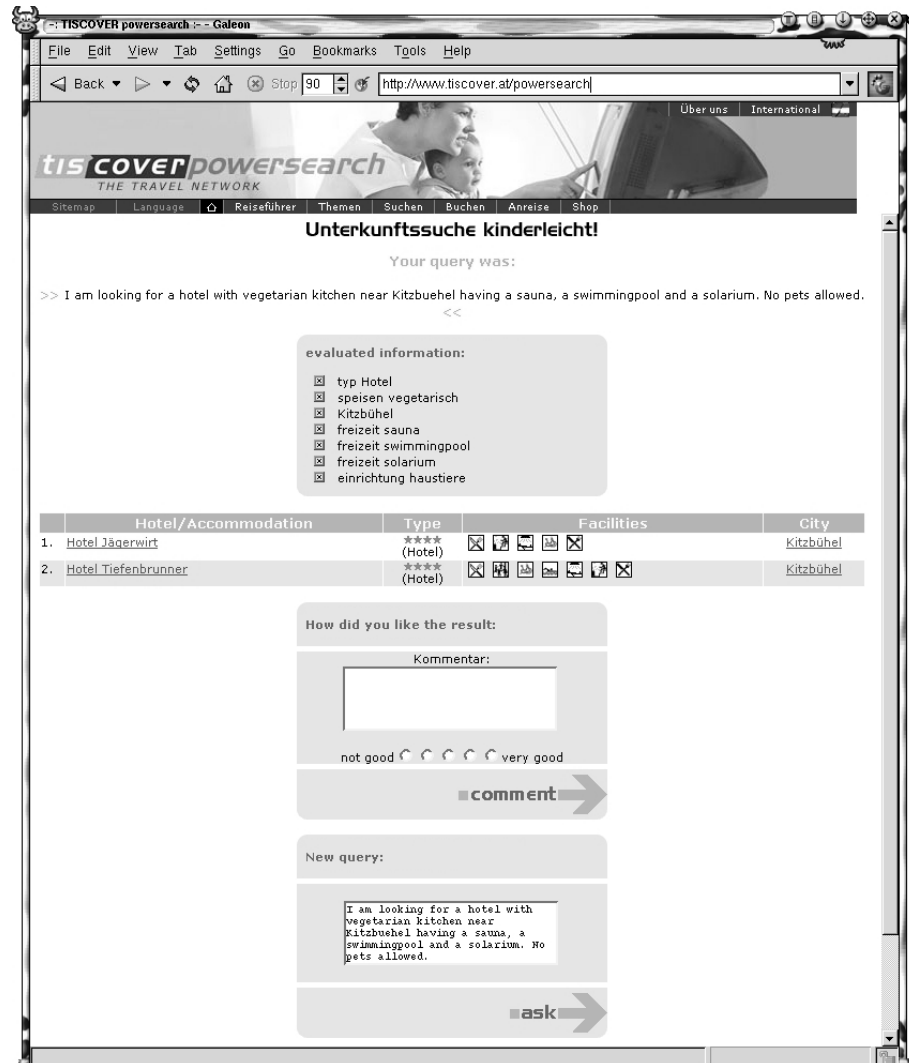


Figure 3.4: Result page with matching accommodations and feedback form.

inappropriate results were found.

3.3.2 Results

We obtained 1,425 unique queries through our interface, i.e. equal queries from the same client host have been reduced to one entry in the query log to eliminate a possible bias for our evaluation of the query complexity. In Table 3.1, a list of countries and the respective numbers of queries is shown. Naturally, most of the queries (39.73%) came from Austrian hosts, followed by hosts from the *.net* top-level domain, most of which have been identified as German Internet service providers by manual inspection. After the 13.13% of queries from the US commercial domain several European countries can be found. A country could not be assigned to 20.42% of the queries because of a non-resolvable domain name.

# of queries (%)	country	# of queries (%)	country
566 (39.73%)	Austria	6 (0.42%)	Luxembourg
229 (16.07%)	.net (mostly German ISPs)	5 (0.35%)	Hungary
187 (13.13%)	US commercial	4 (0.28%)	Belgium
70 (4.91%)	Germany	2 (0.14%)	South Africa
22 (1.54%)	Switzerland	2 (0.14%)	Australia
17 (1.19%)	Italy	1 (0.07%)	US military
14 (0.98%)	Netherlands	1 (0.07%)	France
8 (0.56%)	UK	291 (20.42%)	unknown (not resolved)

Table 3.1: Origin of queries (derived from the top-level domain of the accessing host).

Of those 1,425 unique queries, 1,213 (85.12%) were identified as German, 120 (8.42%) were identified as English and 92 (6.46%) were not identifiable, e.g. non-sentence queries like *“hotel salzburg”* that are possible in either language, or just nonsense like *“xzcvkjjz”*. Based on the 1,333 identified queries we found 85 queries that were not in the scope of our natural language interface. Among these were, for example, questions about car rentals and, of course, sex. Obviously, having any kind of publicly available service like this, not all of the people are using it for the intended purpose. However, this number is rather low assuming the

rather short description we displayed on the start page to give an idea what kind of information can be queried.

To assess the overall quality of the language identification we manually inspected the submitted natural language queries. As explained in Section 2.3, for each query the system assigns either the most probable language or considers the language of the query as being ambiguous. For example, a German query can either be identified correctly, as English or as ambiguous. In Table 3.2, we provide the actual figures resulting from the manual inspection. Thus, of the 1,213 queries identified as German, 1,210 were correctly identified. However, three in fact English queries have been misclassified as being German. In the third row of the table, we can see that of the 92 queries identified as ambiguous, 74 were actually German. This classification error can be explained by the peculiarities of the language identification algorithm based on n-grams. Especially short queries lead to n-gram distributions that do not allow to distinguish between English and German with the required accuracy. In total, of the submitted queries 1,306 were in fact German of which 1,210 were correctly identified. This yields an identification accuracy of 92.6% for the German language. The respective result for the English language is 95.1%.

	manual analysis			
	german	english	ambiguous	
german	1,210	3	0	1,213
english	22	98	0	120
ambiguous	74	2	16	92
totals	1,306	103		
identification accuracy	92.6%	95.1%		

Table 3.2: Manual analysis of language identification accuracy.

To provide some technical information, for the 1,333 processed queries, the mean processing time was 2.63 seconds with a standard deviation of 1.42 seconds. The median of 2.27 seconds shows that there were only a few outliers with longer processing times. Given these figures, we can safely say that our system is usable regarding its response time. Even with adding a few seconds for data transmission time over the Internet, the response time still lies below the magic number of ten seconds as suggested by Nielsen (2000). In usability studies, these ten seconds

have been measured as the approximate maximum attention span of users when waiting for a webpage to be loaded before canceling the request.

We will compare the results of two studies analyzing query log files of the large and popular search engines *AltaVista* and *Excite* with the results of our analysis, since only few research papers dealing with user behavior in web searches exist. Jansen et al. (1998) and Silverstein et al. (1998) have shown that the average number of words per query is very small, namely 2.35, interestingly the same in both studies. This indicates that most of the people searching for information on the Internet could improve the quality of the results by specifying more query terms. Our field trial revealed the encouraging result of an average query length of 8.90 words for German queries, and of 6.53 for the English queries. Due to compound words occurring frequently in the German language, the average number of words should have been higher for the English query. The smaller number of terms occurring in English queries might be explained by their insufficient and non-representative number compared to the number of German queries. Most of the English queries are also likely to have been posed by German-speaking users just to test the interface. Table 3.3 shows the distribution of the number of terms in the queries.

# of words/query	# of queries (%)	# of words/query	# of queries (%)
1	76 (5.33%)	18	12 (0.84%)
2	92 (6.46%)	19	14 (0.98%)
3	117 (8.21%)	20	6 (0.42%)
4	82 (5.74%)	21	9 (0.63%)
5	109 (7.65%)	22	5 (0.35%)
6	147 (10.32%)	23	8 (0.56%)
7	87 (6.11%)	24	2 (0.14%)
8	105 (7.37%)	25	7 (0.49%)
9	98 (6.88%)	26	2 (0.14%)
10	101 (7.08%)	27	3 (0.21%)
11	66 (4.63%)	28	3 (0.21%)
12	75 (5.25%)	29	2 (0.14%)
13	55 (3.86%)	32	1 (0.07%)
14	53 (3.90%)	35	1 (0.07%)
15	30 (2.11%)	37	3 (0.21%)
16	22 (1.54%)	66	1 (0.07%)
17	28 (1.96%)	76	1 (0.07%)

Table 3.3: Distribution of query lengths.

In more than a half (57.05%) of the 1,425 queries, users formulated complete, grammatically correct sentences whereas only 21.69% used our interface like a keyword-based search engine. The remaining set of queries (21.26%) were partial sentences like *“double room for 2 nights in Vienna”*. Several of the queries consisted of more than one natural language sentence, e.g. *“We look for a house at one of the lakes in Austria from July 22 until July 28, 2002. We are a family with 2 children of 8 and 11 years and have a dog. We are searching for a house with lake-side entrance.”* This approves our assumption that users accept the natural language interface and are willing to type more than just a few keywords to search for information. More than this, a substantial portion of the users typed complete sentences to express their information needs. Furthermore, the average number of relevant concepts occurring in the German queries is 3.41 with a standard deviation of 1.96, which is still one word per query more than found in the surveys mentioned above. It can be assumed, that, by formulating a query in natural language, users are more specific than compared to keyword-based searches.

To inspect the complexity of the queries, we considered the number of concepts and the usage of modifiers like *and*, *or*, *not*, *near* and some combinations of those as quantitative measures. Table 3.4 shows the distribution of the numbers of concepts per query. For example, consider row four of this table. The entries in this row show the number of queries with three concepts. In particular, we have 310 German and 28 English queries. Note that these figures were derived by manual inspection of the users' original natural language queries. The majority of German queries contains one to five concepts relevant to the tourism domain with a few outliers of more than 10 concepts. The latter can be explained by people asking for an accommodation in a specific region by enumerating potentially interesting cities and villages.

In analogy to Table 3.4, the Tables 3.5(a) and 3.5(b) give an indication regarding the quality of the natural language query analysis. In particular, Table 3.5(a) provides the numbers of identified concepts per query, whereas Table 3.5(b) that of not identified concepts. Again, the figures given in Table 3.5(b) were derived by manual inspection. We shall note that most of the concepts that were not identified by the system, originated from queries falling into the categories of region

concepts	query language		
	german	english	totals
0	47	5	52
1	77	28	105
2	272	38	310
3	310	28	338
4	245	12	257
5	137	5	142
6	49	2	51
7	38	1	39
8	18	1	19
9	11	0	11
10	4	0	4
11	1	0	1
17	3	0	3
21	1	0	1
totals	1,213	120	1,333

Table 3.4: Number of concepts per query (counted by manual inspection).

names, pricing information, room availability and arrival and departure dates. This information was not disclose by *Tiscover* and. thus, was not contained in the database of our natural language system.

Another aspect of the complexity of natural language queries are words connecting concepts logically or modifying their meaning. These modifiers can be compared to operators like “AND”, “OR”, “+” or “−” of web search engines. In Table 3.6(a) we can see that the distribution of occurrences of the modifier *and* corresponds to the number of concepts. In 320 queries the modifier *and* was used twice which relates to the occurrence of three concepts per query (cf. Table 3.4). The occurrence statistic includes all implicitly used modifiers *and*, i.e. those *ands* that are included because of the resulting SQL statement, as well as those explicitly defined, i.e. those *ands* that are provided with the natural language query. For instance, the query “*I am looking for a hotel with sauna, solarium and whirlpool in Tyrol*” includes one explicitly used *and*, and three implicit *and* modifiers. Due to the assumption that the underlying semantics of combining concepts is based on the intention to provide facilities somebody wants to have, we defined the *and* modifier to be the default logic for combining concepts if no explicitly defined modifier is present. This assumption is made to provide a convenient technique to map the concepts used in a query onto the underlying program logic. However,

concepts	query language		
	german	english	totals
0	71	14	85
1	104	27	131
2	326	39	365
3	312	24	336
4	201	10	211
5	106	2	108
6	50	2	52
7	19	2	21
8	13	0	13
9	6	0	6
10	1	0	1
16	3	0	3
20	1	0	1
totals	1,213	120	1,333

(a) Concepts identified by the natural language processing

concepts	query language		
	german	english	totals
0	817	88	905
1	348	29	377
2	45	3	48
3	3	0	3
totals	1,213	120	1,333

(b) Concepts not identified by the natural language processing

Table 3.5: Concepts that have been identified or not identified by the natural language processing module of our interface.

in contrast to the assumption that the default operator of combining concepts is *and*, the modifier *or* must be used to combine geographical concepts that are enumerated in queries with only a comma used as separator. Consider, for example, the German query *“Ich suche ein Appartement für 5 Personen (3 Kinder, 10,3,2 Jahre) mit Frühstück in einer familienfreundlichen Gemeinde in Oberösterreich, Bayern, Südtirol oder Salzburg”*. Here, assuming the modifier *and* to connect the geographical locations would yield an empty result set since the user is searching for an apartment in either *Upper Austria, Bavaria, South Tyrol* or *Salzburg*.

The modifier *or* is used far less often than *and*, as shown in Table 3.6(b). In particular, *or* is used in 103 queries only. *Or* is mostly used to explicitly separate the elements of a set of locations or types of accommodations of interest, e.g. *“I am looking for a farm or an apartment in Tyrol or Salzburg”*.

An interesting fact is, that the *not* modifier is used in a very small subset of queries (cf. Table 3.6(c)). The modifier *not* occurs in only 19 German and 3 English queries. This implies, that the vast majority of users formulate their intentions without the need of excluding concepts. In most of the cases where

a *not* is used to exclude a specific property of a region or an accommodation, users wanted to avoid places where pets are allowed as well as accommodations that are *not* particularly well-suited for children, the latter, perhaps, to stress the desire to find a quiet place. Another common use of *not* is to exclude one or more cities from a query where an accommodation in a federal state or region was wanted, e.g. “*I am looking for a hotel in Tyrol, but not in Innsbruck and not in Zillertal.*”

Table 3.6(d) shows the number of occurrences of the modifier *near* which has been expressed by terms like *around*, *close to* or *near* itself. Generally, geographical concepts or relations are essential to provide a high-quality tourism information service. Comparing the modifier usage statistics a remarkable detail is noticeable. In 122 out of 1,425 queries (8.6%) the modifier *near* is used. This circumstance makes *near* to the modifier second-most frequently used, in the queries collected during the field trial. A common way to use *near* is to find accommodations in the surroundings of popular sites, cities or facilities, e.g. “*I am looking for a hotel with sauna and pool in St. Anton near the Galzig-Seilbahn.*”

Table 3.7(a) illustrates the combined usage of the modifiers *and* and *or*. Most commonly used is a combination of one *or* and several *and* modifiers, e.g. two *and* and one *or* are used in 17 German queries. As shown in Table 3.7(b), the usage of *near* corresponds with the presence of an *and* modifier.

We can say that the overall sentence complexity, i.e. the frequency of concept combination, is relatively low. In general, queries are formulated on the basis of combining concepts in a simple manner, e.g. “*I am looking for a room with sauna and steam bath in Kirchberg.*” Only a small subset of queries consist of complex sentence constructs that would require a more sophisticated sentence evaluation process. This is the case if, for instance, the scope or type of the modifier cannot be determined correctly. As an example, consider the query “*I am looking for an accommodation in Serfaus, Fiss or Ladis.*” For the reader who is not familiar with the geography of Austria, in particular the Tyrol in this case, we shall note that *Serfaus*, *Fiss*, and *Ladis* are names of towns, and collectively they refer to an attractive skiing resort. An example having a similar structure but different semantics is “*I am looking for an accommodation with en suite bathroom, bar and sauna.*” As already mentioned above, the type of the

	query language		
and	german	english	totals
1	281	38	319
2	320	29	349
3	246	11	257
4	140	6	146
5	41	1	42
6	33	1	34
7	16	0	16
8	4	0	4
9	2	0	2
10	1	0	1
totals	1,084	86	1,170

(a) Usage of modifier *and*

	query language		
or	german	english	totals
1	67	4	71
2	18	1	19
3	6	1	7
6	1	0	1
8	1	0	1
12	3	0	3
16	1	0	1
totals	97	6	103

(b) Usage of modifier *or*

	query language		
not	german	english	totals
1	12	3	15
2	7	0	7
totals	19	3	22

(c) Usage of modifier *not*

	query language		
near	german	english	totals
1	112	9	121
2	0	1	1
totals	112	10	122

(d) Usage of modifier *near*Table 3.6: Usage of modifiers *and*, *or*, *not* and *near*.

elements (e.g. cities, facilities) that are enumerated has to be considered when transforming the natural language query into a formal database query. In both cases there are concepts that are separated by commas, but in the database query they have to be combined with an *or* operator and an *and* operator respectively.

3.3.3 Lessons Learned

In this section we have discussed the findings of a ten day field trial where we collected about 1,400 queries, most of which were posed in German language. Most importantly, it can be said that the users willingly typed natural language queries to express their information needs. In more than a half of the queries, users formulated complete, grammatically correct sentences, about one fifth were partial sentences and the remaining set were keyword-type queries. Several of the queries consisted of more than one sentence. This observation is approved by

		query language		
and	or	german	english	totals
1	1	9	1	10
	2	3	0	3
2	1	17	2	19
	2	3	0	3
3	1	16	0	16
	2	5	0	5
	3	2	1	3
	6	1	0	1
4	1	12	1	13
	2	3	0	3
	12	3	0	3
	16	1	0	1
5	1	8	0	8
	2	1	1	2
	3	2	0	2
6	1	2	0	2
	2	2	0	2
	3	2	0	2
7	1	2	0	2
8	1	1	0	1
	2	1	0	1
totals		96	6	102

(a) Combined usage of modifiers *and* and *or*

		query language		
and	near	german	english	totals
1	1	18	1	19
2	1	32	2	34
3	1	26	1	27
4	1	21	4	25
5	1	7	0	7
	2	0	1	1
6	1	2	1	3
7	1	2	0	2
8	1	1	0	1
totals		109	10	119

(b) Combined usage of modifiers *and* and *near*

Table 3.7: Combined usage of modifiers.

a comparison with web-search engines, where the average number of words per query is substantially smaller than the average number of words per query posed during our field trial.

Although the complexity of the queries is higher than with standard web-search engines, the analysis of combinations of conjunctions and modifiers has shown that deep language understanding is not necessary to achieve an adequate retrieval performance. The fact that the level of sentence complexity is not very high suggests, that shallow text parsing should be sufficient to analyze the queries emerging in a limited domain like tourism. This is also backed by the fact that most of the query concepts, which had their counterpart in the knowledge base, were successfully extracted from the natural language query.

By way of this field trial allowing natural language descriptions of information needs as opposed to the strictly limited variability of form-based information entry, we have got an impression of what the customers actually look for. Among the most important issues we just mention geographic information as when you describe the location of your preferred accommodation relative to some geographical landmarks. An interesting example emphasizing the importance of geographical knowledge in tourism information systems is, for instance, the query *“Ich suche 2 Hotelzimmer für 2 Erwachsene und 2 Kinder mit Hund nahe der Schweizer Grenze im Juli 2002 für 14 tage”*. Here, somebody is searching for a hotel room close to the Swiss border. A structurally complex example, pointing out a more mobile aspect of the system is the query *“Zimmer auf der Strecke Innsbruck – Söll, Ankunft ist heute um Mitternacht Autobahn Raststätte A12”* that translates roughly to *“room along the route Innsbruck – Söll, arrival is today at midnight autobahn A12 roadhouse”*. In order to answer this query in a satisfying way, the system should have information about roads and routes as well as their relation to accommodations. The issue of geography related information will be addressed in Section 4.4.

We also noticed that users' queries contained vague or highly subjective criteria like *romantic*, *cheap* or *within walking distance to*. Even *wellness*, a term broadly used in tourism nowadays, is far from being exactly defined. These concepts are difficult to model in the ontology of the original system and this issue was a motivation for the development of an alternative approach such as the one

based on associative networks as presented in Section 2.4.

It furthermore turned out that a deficiency of our ontology was the lack of diversity of the terminology. To provide better quality search results, it is necessary to enrich the ontology with additional synonyms. Besides the structured information stored in our database about the accommodations, the webpages describing the accommodations offer a lot more information in form of natural language descriptions. Hence, the words occurring in these texts constitute a very specialized vocabulary for this domain and are predestined to be used for ontology enrichment. The next obvious step was to exploit this information to enhance the domain ontology of the information retrieval system. Due to the size of this vocabulary, some intelligent form of representation is necessary to express semantic relations between the words. A method for visualizing semantic relations between words mined from free-form texts based on a neural network model, namely the *self-organizing map*, will be presented in Section 4.3.

Regarding the quality of the search results as seen by the users themselves, it turned out that only 3.37% of the queries have either been annotated or rated where the number of positive and negative comments were nearly equal. Due to the unsupervised nature of the test without any reward for the users, this figure is not surprising because of the additional time it takes to assess the quality of the result and then comment on it. This leads us directly to the next section describing a supervised usability study comparing the conventional *Tiscover* interface with our natural language approach.

3.4 Usability Evaluation

3.4.1 Test Setup

The usability test was designed, to compare the original *Tiscover* accommodation search interface with our natural language search interface with several aspects in mind. First, and most importantly, we wanted to test the acceptance of such a natural language interface to justify our research. Second, we wanted to get feedback from the users regarding their contentment with the search results, which was not possible in the unsupervised field trial. Third, we were interested

in how people express their search in natural language when confronted with specific search tasks they have to carry out. Finally, as a result of addressing these points we expected to be able to infer important information for the future development of the natural language interface.

For the usability study, we have set up five scenarios defining different situations and search tasks the test users had to solve with both the *Tiscover* accommodation search interface as well as our natural language interface. An important point in defining the tasks was to ensure that all of the tasks could be solved with both interfaces. Hence, to eliminate such problems we have conducted a preliminary study with three persons to check the feasibility of the scenarios for the main test. One scenario had to be changed because the search task could not be performed with the conventional interface. It has to be noted again that, at the time the study was conducted, the extended search that allows to select all available accommodation features was not available (see Section 3.2). Therefore, we were limited to defining scenarios that covered the few features selectable with the conventional interface.

The test included an orientation script to explain the purpose of the test and to introduce the participants to the nature of the study. In our case, the script presented the two interfaces and their differences and made clear what we mean by a *natural language query* in order to eliminate potential unclarities. We additionally asked for demographic characteristics of the test persons as well as their background regarding their computer literacy and frequency of Internet use.

The five scenarios had to be performed in alternating order, i.e. the first task had to be solved with the conventional interface first, the second task with the natural language interface first to avoid any bias towards either interface. Just as an example, we show the first and the second scenario of the study. Each scenario usually describes a living situation that already gives some hints on what has to be considered when searching for an accommodation. Furthermore, the destination may be given and what kind of activities should be possible at the destination.

Scenario 1 Thermal Bath/Spa: *Imagine being a parent of a family consisting of four people and you intend to go on holiday to a thermal bath. You and your*

family have no distinct plans regarding the location of the destination. Your 82-year old grandmother is physically challenged and is bound to a wheelchair. You want to take your grandmother with you. Your husband/wife enjoys attending solariums and does not want to miss it in this holiday. Your children are one and four years old respectively.

Scenario 2 Golf Holiday: *You want to play golf with some friends on a weekend in Tyrol. You don't mind if this holiday costs a little bit more than usual and want to live in an upmarket hotel. In the evening you want to go to the sauna and do some swimming afterwards. Since you have problems with your back, the possibility of having a massage should be offered.*

The other three scenarios where the test persons should search for a certain type of pension, a hotel suitable for a conference and a farm for spending their holiday respectively, are similarly structured and not described any further in this thesis. A detailed report on the usability study including all scenarios and the questionnaires can be found in Pribernig (2003). After performing the tasks, we asked the following questions:

1. Did you think that the scenarios were easy and intuitive?
2. How easy/hard was it for you to imagine being the persons described in the scenarios?
3. How easy/hard was it for you to express your query? (answer separately for each interface)
4. Did you think that the search results were satisfying? (answer separately for each interface)
5. If you had the choice, would you prefer a natural language interface over a conventional (form-based) interface? (answer separately for tourism-related information and for general information)
6. How do you judge the usability of the interfaces? (answer separately for each interface)

7. Do you have any general requests for or comments on the interfaces? (answer separately for each interface)

3.4.2 Results

The main usability test was performed by 17 people (13 males, 4 females). According to Nielsen and Landauer (1993), this number of test persons is sufficient to detect most usability problems of user interfaces. It has to be noted though, that finding usability flaws of the interfaces is not the sole objective. In general, the test persons can be characterized as mid-twenties with either high school or university degree who are more or less daily Internet users. Despite one person, they are going on holidays more than once a year. 16 of the 17 persons already did holiday-related research online but only 6 have already booked a holiday via Internet.

Regarding the layout of both interfaces, a clear preference towards one or another can not be detected. Most of the people had no problems understanding the scenarios and were able to imagine being the persons described by the scenarios. The answers to Question 3 concerning the ease of expressing the intended query showed a rather balanced distribution with a slight tendency towards hard for the conventional interface, whereas the natural language interface has generally been rated as easy. Furthermore, 14 persons felt comfortable being able to express their queries in natural language.

The satisfaction regarding the quality of the search results has been rated slightly better when using the natural language interface, but this should not be overrated since this is a highly subjective perception. The usability rating of the interfaces (see Question 6) turned out to be better for the natural language version where 13 test persons rated it positively in contrast to only six persons who did so for the conventional interface.

General remarks on the conventional interface were that the region selection could be more comfortable, that the list of *themes* is not clearly arranged, that the list of hotel chains is superfluous and that the elements of the page could be better arranged. Another important point of criticism was that the number of features that can be selected was too small.

There were also some comments and suggestions for improvements for the natural language interface. Some compound words as well as geographic regions were not recognized. The first was mainly caused by erroneous interaction between the spell checking and the phrase recognition module, whereas the second problem arose, because information about regions was missing in our database. Since it is not as clear as in the form-based interface which features have been selected (i.e. recognized by the language processing), the test persons demanded a better feedback on the parts of the query that were *understood*. Another notable comment was that a certain time was necessary for familiarization with the interface although the method of entering the query in form of a sentence was considered comfortable. Due to the unfamiliar kind of interaction the test persons expressed the need for guidance on how the interface works and therefore criticized the missing – or rather too short – documentation. This was in fact our fault to a certain extent, because we left the interface, especially the short descriptive piece of text, unchanged since the field trial where the intention was to gather lots of diverse queries. It seems that the information provided by the orientation script was forgotten as soon as the test persons were about to actually work with the interface.

3.4.3 Discussion

The most important findings derived from this study can be divided into two major blocks, i.e. subjective versus functionality-related issues. First, the test users were clearly accustomed to search interfaces of the form-based type like the conventional *Discover* interface. Nevertheless, in the course of the test, a shift of sympathy from the conventional towards the natural language interface could be noticed. This supports our assumption that natural language interfaces, even when users have to type the questions, could become an accepted alternative for searching information. However, the duration of the process of familiarization to this rather different search paradigm will depend on the willingness of large and well-known information providers to employ these semantics-supported natural language search facilities on their sites.

Second, concerning the functionality, both interfaces definitely require refur-

bishment. The original form-based interface shows what problems do arise when the amount of information provided by an information system like *Tiscover* becomes too large. On the one hand, users want to be able to select more features than those presented on the start page, but on the other hand, demand a more concise interface. Therefore, we think that expressing the search request in natural language can facilitate providing a clear and concise interface for a search engine that would otherwise require a complex form-based search interface consisting of radio buttons, check buttons, drop-down lists and more. The deficiency of the natural language interface regarding compound words that was remarked by the test persons, showed that some of the language processing modules left space for improvement. Finally, feedback is especially important when a large part of the functionality is hidden from the users. In our case, users felt quite uncertain which concepts have been identified and were used as search criteria even though we provided a list of recognized concepts. Hence, a more sophisticated kind of feedback has to be provided in the future.

Chapter 4

Ontology Enhancement

4.1 Introduction

Ontologies gained increasing importance in many fields of computer science where they are used to specify a particular part of the world by describing concepts and their relations in a domain of discourse (Gruber, 1993). Especially for information retrieval systems, ontologies can be a valuable means of representing and modeling domain knowledge to deliver higher quality search results. Semantic relations can improve the quality of search results by automatically considering additional knowledge about the application domain that is not explicitly stated in the query. To suit our application domain of tourism, consider, for example, a search request where someone is looking for a quiet hotel with sauna and swimming pool. Supporting the information retrieval task with semantics can mean, on the one hand, to relate the term *quiet*, which is not a property of an accommodation in our case, to concepts that are present in the database and characterize the absence of hubbub, e.g. there are no facilities for children or no pets allowed. On the other hand, knowing that *sauna* and *swimming pool* have something to do with recreation and sports, hotels with similar but different facilities and services would be ranked higher in the list of retrieved results. This can be seen as a technique similar to *query expansion* (e.g., see Mitra et al., 1998), where additional query terms are taken either from thesauri or from documents a users has judged relevant for the query.

However, a problem crucial for the applicability and usability of ontologies

in real-world applications, namely the increasing complexity with growing size of the application domain, remains. Even for the comparatively small domain that our natural language interface covers, the construction of the according ontology required a lot of effort. Historically, the construction of dictionaries and thesauri, which are of course by orders of magnitude larger, was always associated with reading through long concordances (Church and Hanks, 1990). Regarding dictionaries, the first and also most humongous act was performed by James Murray when creating the first edition of the *Oxford English Dictionary*, which took him, some collaborators and a large number of volunteers about 44 years (Murray, 2001).

Since ontologies became more frequently used in computer science, the number of approaches and algorithms for automatically or semi-automatically constructing and enriching ontologies has also increased. In particular, the topic of mining concepts and relations from text corpora in the context of thesaurus and dictionary generation has been given a fresh impulse. We outline a few publications dealing with the topic of extracting knowledge from texts for thesaurus and ontology engineering.

Church and Hanks (1990) propose a measure to estimate word association norms based on mutual information. This *association ratio* can be used to estimate the level of associativity between two words based on the probability of occurring jointly in a fix-sized window of words and the probability of independent occurrence. This method of measurement can be used in lexicography to support the analysis of concordances for complex words.

A framework for maintaining domain-specific ontologies based on reasoning and hypothesis generation is reported in Hahn and Schnattinger (1997, 1998) The goal of this approach is to integrate new knowledge items occurring in a text into an already existing concept hierarchy. Based on a linguistic analysis of unknown lexical items, several hypotheses are generated and then ranked according to a plausibility measure derived from existing domain knowledge.

Grefenstette (1992) presents a system that syntactically analyzes texts to extract contexts for calculating a similarity measure between two terms. The user can specify a set of context relations that are extracted by the system according to the information gained by a linguistic analysis consisting of several processing

steps including morphological analysis, grammatical disambiguation, noun and verb phrase detection and relation extraction. Then, the relations are measured using the Jaccard measure to compare terms relative to the contexts they share.

Byrd and Ravin (1999) report a rule-based extraction approach for proper names and concepts with regular expression-like rules defining grammatical structures. Furthermore, relations between such concepts and proper names are extracted by detecting patterns such as “PERSON, ... of ORGANIZATION” in the texts.

The objective of the work described by Sanderson and Croft (1999) is to automatically derive concept hierarchies from document collections without prior knowledge. A subsumption condition, where a word x subsumes a word y if the documents y occurs in are a subset of the documents x occurs in, was used to create the concept hierarchy.

Velardi et al. (2001) and Missikoff et al. (2002a,b) describe a text mining tool for term extraction to support the ontology construction process. The system uses a rule set to detect named entities and extracts candidate concepts using shallow language processing techniques. Terms are then assessed according to some plausibility measure based on mutual information in order to rank them. Finally, the extracted terms are organized into a concept hierarchy using WordNet (Fellbaum, 1998) and SemCor (Miller et al., 1993), a semantic concordance package where texts have been manually tagged with WordNet meanings.

A series of publications by Mädche and Staab (2000a,b,c) deals with the topic of mining ontologies from text in the context of the *Semantic Net* (Berners-Lee et al., 2001). Contrary to some of the work mentioned above, emphasis is put on extraction of non-taxonomic relations, i.e. relations that do not describe hierarchical *is-a* relations but rather relations such as *part-of* or *is-located-in*, to give an example. An algorithm for discovering generalized association rules is used to detect relations between concepts and assign them confidence values. The authors present an example from the tourism domain where relations such as *area – accommodation* or *room – furnishing* are detected.

Another ontology enrichment approach is reported in (Agirre et al., 2000) where documents from the Internet, which are related to a specific topic, are retrieved and analyzed to build hierarchical clusters of terms.

Hearst (1998, 1992) presents a method for automatic extraction of hyponym relations from text corpora. Hyponyms, i.e. *is-a* relations, are detected through lexicosyntactic patterns defining the position of noun phrases and certain keywords such as ‘*including*’, ‘*especially*’ or ‘*such as*’. A whole set of patterns has been defined that are typically representing *is-a* relations and yield good results in detecting them. Hearst argues that the WordNet lexical database lacks many relations and proper nouns that would be useful for certain applications but do not suit a general purpose lexical database. Therefore, it is often necessary to enhance the database for specific purposes which is a costly task as already mentioned. Consequently, the major goal of this work is to automatically suggest relevant hyponymic term relations from domain-relevant documents to be integrated into WordNet.

For further reading we suggest the reviews by Ding and Foo (2002a,b) who provide an extensive overview of ontology research and development covering ontology generation as well as mapping and evolving.

The motivation for the work presented in this chapter was, first, that creating and refining the ontologies of both systems presented in Chapter 2 from scratch, turned out to be more complex than anticipated. Second, the results of the field trial have shown that we have overlooked quite a number of synonyms for concepts that were therefore not detected by the natural language processing of our system. As already mentioned, regional characteristics, subjective and fuzzy criteria also played an important role in the users’ queries. Third, comparing the number of attributes of accommodations at the time we have developed the first system (October 2001) with the current number, it can be seen that an ontology describing such a dynamic area like tourism is exposed to constant change, in this particular case, growth and reorganization. At the time of writing, accommodations can be queried by 159 features compared to 82 at the end of the year 2001. Some features like *massage* or *steam bath* that were then seen as part of *recreation*, are now subsumed under separate topics, i.e. *health* and *vitality*, with a number of related features. In particular, *health* includes 27 facilities and services like *hay baths*, *kinesiotherapy*, *herbal bath* or *Swedish massage*, to name but a few. Despite this increased number of wellness-related characteristics, also the variety of facilities offered for conference hosting, e.g. *microphones*, *video*

projector or *laser pointer*, has grown to 20 items.

Hence, we present an approach based on a neural network model, namely the *self-organizing map* (*SOM*), to assist domain engineers in creating or enhancing ontologies for information retrieval systems. Contrary to some of the work described above, we do not aim at the automatic construction of concept taxonomies but rather at providing an informal tool for convenient exploration of the semantic space spanned by the domain vocabulary. Sticking to our application in the tourism domain, we show how the semantic relations between words occurring in free-form text descriptions of the accommodations can be visualized to support the construction and the enrichment of the domain ontology of our natural language information system. We exploit information inherent in the textual descriptions that are accessible but separated from the structured information the search engine operates on. The vector representations of the terms are created by generating statistics about local contexts of the words occurring in natural language descriptions of the accommodations.

These descriptions have in common that words belonging together with respect to their semantics, are found spatially close together regarding their position in the text, even though the descriptions are written by different authors, i.e. the accommodation providers themselves in case of our application. We believe that the approach presented herein can be applied to a variety of domains, since, for instance, product descriptions generally have similarly structured content. Consider for example, typical computer hardware descriptions where information about, say, storage devices are normally grouped together, rather than being intertwined with input and display devices.

More specifically, we use the *self-organizing map* to cluster terms relevant to the application domain to provide an intuitive representation of their semantic relations. With this kind of two-dimensional map representation at hand, finding synonyms, adding new relations between concepts or detecting new concepts that would be important to be added to the ontology, is facilitated. More traditional clustering techniques are used in the DARE system (Frakes et al., 1998) as methods supporting combined top-down and bottom-up ontology engineering (Prieto-Díaz, 2002).

Furthermore, due to the obviously large vocabulary constituting a certain

number of free-form text documents, we propose a variant of a term weighting approach common in information retrieval to describe and detect geography-dependent terms regarding local attractions, specialties or landmarks, which is especially attractive in the tourism domain.

The remainder of this chapter is structured as follows. First, we will discuss the characteristics of the the text documents in Section 4.2. In Section 4.3, we first present the unsupervised learning algorithm of the *self-organizing map*, a neural network model we use for visualizing semantic relations among terms in the domain vocabulary. Then follow the descriptions of two term encoding strategies where the first is based on a document/term matrix and the latter is based on local contexts. Methods for mining terms from texts, which are potentially relevant to the domain ontology, are presented in Section 4.4. Finally, we conclude this chapter in Section 4.5 with a discussion of the presented algorithms.

4.2 Domain-Related Text Descriptions

The data provided by *Tiscover* consist, on the one hand, of structured information as described in Subsection 2.2, and, on the other hand, of free-form texts describing the accommodations on the according webpages. Here, the accommodation providers have the possibility to describe and promote their lodgings in their own words. This allows for much more detailed descriptions than would be possible with the fixed set of properties provided by the database structure. Because the data can be supplied by the accommodation providers themselves, the descriptions vary dramatically in length and style and are not uniform or even quality controlled regarding spelling.

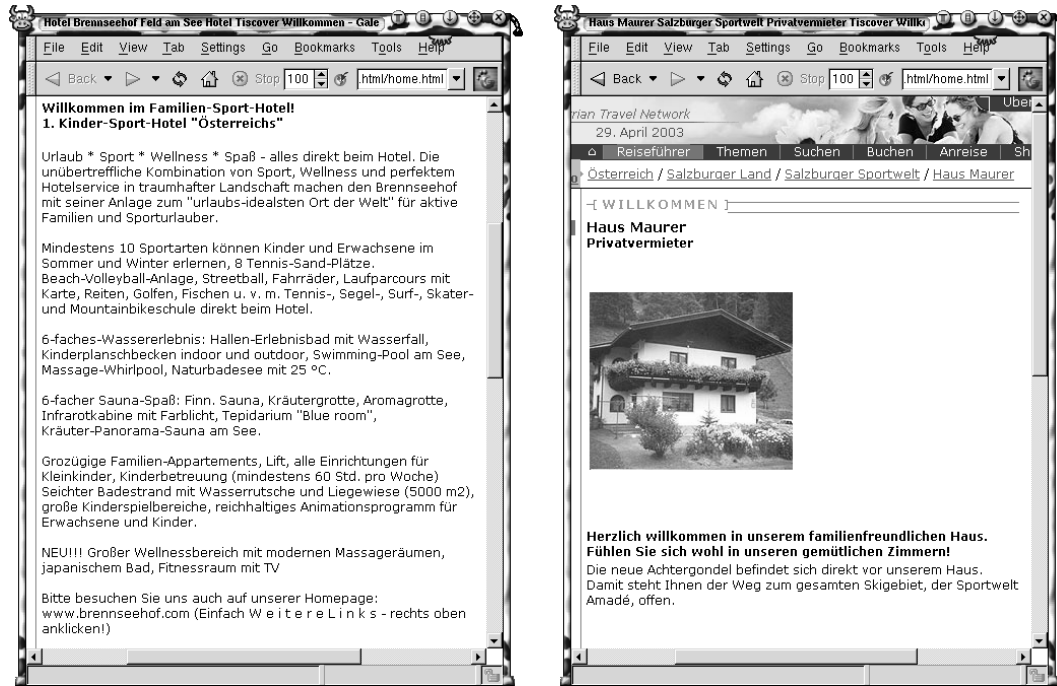
HTML tags, which are allowed to format the descriptions, had to be removed to have plain-text files for further processing. For the experiments presented in this chapter, we used the German descriptions of the accommodations since they are more comprehensive than the English ones. Especially small and medium-sized accommodations provide only a very rudimentary English description, many being far from correctly spelled. The document collection we used for the experiments presented hereafter consists of 12,471 descriptions. For the curious reader we shall note that not all of the 13,117 accommodations in our snapshot of the

Tiscover database provide a textual description.

Before showing some examples, a few problematic issues regarding the quality of the text documents should be mentioned. One difficulty was to clearly define word boundaries due to the inconsistent spelling of words. We decided to treat character strings that are directly connected by a dash without blank characters inbetween as one word, accepting the consequence that in one case, *green-fee* is treated as one word, and in the other case, *green* and *fee* are regarded separately. In very few cases, this rule resulted in extremely long words, because the accommodation provider connected a whole list of facilities and services with dashes without using blank characters as delimiters. Another problem was the capitalization of whole paragraphs in the descriptions to emphasize specific parts of the description in order to attract the attention of the potential customers reading it. Consequently, we used case-insensitive string comparison when indexing the documents. Abbreviations pose another difficulty if written differently from customary standards. Yet another problem arises due to the possibility of omitting parts of compound words in enumerations, with one part being the same for all words, e.g. '*Spazier- und Wanderwege*'. This is a shorter and less redundant form of writing '*Spazierwege und Wanderwege*'. Again, a little bit of semantic information is lost, but assuming a large enough number of texts, these deficiencies should be alleviated by a statistically significant number of correct samples. Classic typographical errors, e.g. swapping or omitting characters, form only a minority of the errors in the texts has been observed by analyzing the extracted vocabulary. It has to be emphasized again that an essential type of error is the different spelling of words including deliberately created syntactical errors such as *Geh-Minute* even though the correct German word would be the compound *Gehminute*.

To give a practical example, consider Figure 4.1 showing two sample descriptions of a hotel and a private lodging provider respectively. On the left-hand side in Figure 4.1(b) we can see a comparatively extensive description of a sports hotel. It contains enumerations of sporting activities that can be undertaken on the hotel grounds, e.g. tennis, beach volleyball, cycling, horseback riding and many more. Furthermore, the description underlines the possibilities to relax by describing the different swimming facilities, different types of saunas and other

wellness-related facilities and services. Please note the semantic grouping of the characteristics regarding the position in the text. The private accommodation in Figure 4.1(b) on the other hand, just mentions the cozy and family-friendly atmosphere and the location close to a ski lift tying the accommodation to a very large skiing area that can be conveniently reached with.



(a) An opulently equipped hotel praising its sports-related facilities and the possibilities of relaxation.

(b) A rather short description of a private accommodation simply stating the cozy atmosphere and the proximity to the ski lift.

Figure 4.1: Two different accommodation descriptions.

What makes these textual descriptions interesting is the rich vocabulary that can be used to improve the domain ontology by adding semantic information in order to provide better search results. In total, the document collection comprises 1,003,017 words, yielding a mean of about 84 words per accommodation with a standard deviation of 40 words. 49,294 unique terms form the vocabulary of the documents. This figure shows the necessity of developing appropriate tools

supporting the ontology engineering process when mining concepts from text.

4.3 Visualization of Semantic Relations

4.3.1 The Self-Organizing Map

The *self-organizing map* (*SOM*), as proposed in (Kohonen, 1982) and described thoroughly in (Kohonen, 1989, 1995) is a well known representative of unsupervised artificial neural networks especially in the fields of clustering, data classification and data visualization. It performs a non-linear projection of high-dimensional data onto a usually two-dimensional map preserving the topology of the input space as faithfully as possible, i.e. similar input patterns will be mapped onto spatially close regions in the output space. As a consequence, the relationship between input data is mirrored in terms of the distance of the respective representatives in the output space. Thus, the *SOM* is a convenient tool for the visualization and the exploration of high-dimensional data.

Architecture

The data are represented by n -dimensional vectors $x \in \Re^n$, $x = (x_1, x_2, \dots, x_n)^T$, i.e. they are described by n features in the input space. The *self-organizing map* consists of an input layer which propagates the input data in parallel to a number of neurons (units) in the output layer which may be organized in a hexagonal (Figure 4.2(a)), rectangular (Figure 4.2(b)) or even irregular, usually two-dimensional, lattice. A rectangular layout will be assumed for the rest of this description. Every unit i has an associated weight vector $m_i \in \Re^n$ of the same dimensionality n as the input vectors. In Figure 4.2, the weight vectors are represented by arrays of boxes in different shades of gray according to the respective values of the weight vector components. These weight vectors may either be initialized randomly, with random samples from the input data set or by more sophisticated methods such as, for example, Principle Component Analysis.

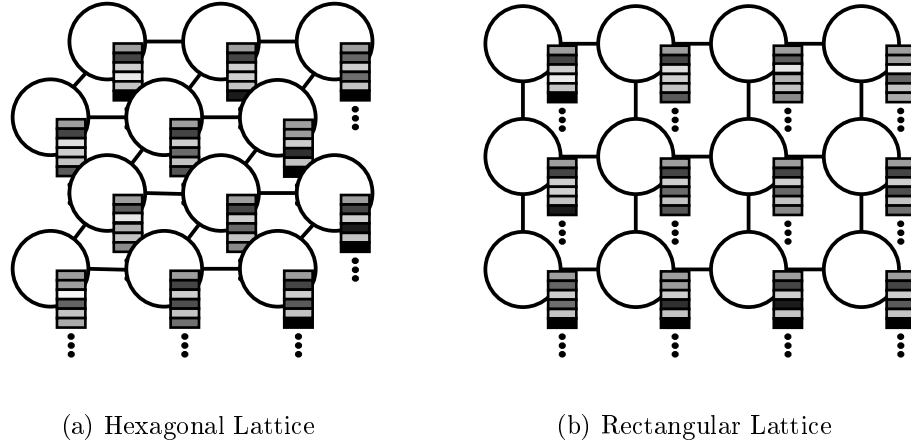


Figure 4.2: The units of a *SOM* can be arranged in different types of lattices. An n -dimensional weight vector is assigned to each unit, depicted by an array of shaded boxes representing the different values of the weight vector components.

Training Algorithm

In the following equations, we make use of a discrete time notation, with t denoting the current training iteration. The training starts by random selection of an input vector $x(t)$. Then, the unit c with the smallest distance between its assigned weight vector m_c and input $x(t)$ in the Euclidean space is selected as the *best-matching unit* (hereafter referred to as *winner*) according to Equation 4.1. The Euclidean distance is denoted as $\|\cdot\|$.

$$c = \arg \min_i (\|x(t) - m_i\|) \quad (4.1)$$

In other words, the input $x(t)$ is represented best by unit c . To increase the probability for this unit to be chosen as *winner* if the same input is selected in subsequent training iterations, the difference between the unit's weight vector m_c and input $x(t)$ will be decreased. This gradual adaptation of the weight vector is controlled by the *learning rate* parameter $\alpha(t) \in [0, 1]$. The value of parameter is defined by a time-decreasing function with $\lim_{t \rightarrow \infty} \alpha(t) = 0$. Hence, weight vectors will be adapted stronger at the beginning of the training process. A rather low value of $\alpha(t)$ at the end of the training process leads to a fine-tuning

phase.

To achieve a topology preserving mapping, i.e. preserving similarity relations between input data on the output space, not only the weight vector of the *winner* c will be adapted, but also the weight vectors of units in its vicinity. Thereby, input data similar to $x(t)$ are more likely to be represented in the region of the *SOM* where the *winner* is located. The adaptation strength $h_{ci}(t)$ of neighboring units is determined by their distance from unit c on the *self-organizing map*. This is a time-decreasing function as given in Equation 4.2, where a Gaussian function is used.

$$h_{ci}(t) = \exp\left(-\frac{\|r_c - r_i\|^2}{2 \cdot \sigma^2(t)}\right) \quad (4.2)$$

$r_c \in \mathbb{R}^2$ denotes the location vector of the winning unit c on the grid and $r_i \in \mathbb{R}^2$ the location of a neighboring unit i . Parameter $\sigma(t)$ is the time-dependent factor. It can be seen from Equation 4.2 that units closer to the winner are adapted stronger than units which are farther away. A high value of h_{ci} at the beginning of the training process leads to a global organization of the units' weight vectors, i.e. neighboring units have similar weight vectors. By decreasing the neighborhood function successionaly in the course of time, the adaptations become more local.

A computationally cheaper neighborhood function is, to define a set of units $N_c(t)$ (*neighborhood kernel*) around *winner* c at time t , whereby the adaptation strength h_{ci} of the neighboring units is determined as follows:

$$h_{ci}(t) = \begin{cases} \exp\left(-\frac{\|r_c - r_i\|^2}{2 \cdot \sigma^2(t)}\right) & \text{if } i \in N_c(t), \\ 0 & \text{if } i \notin N_c(t). \end{cases} \quad (4.3)$$

Hence, only weight vectors of units within the *neighborhood kernel* are adapted. The computational load during training a large map can benefit from this approach, because only a subset of the units require weight vector adaptation. Whereas, with the previously described function (see Equation 4.2), every unit's weight vector is adapted at every training iteration. Again, a rather large neighborhood kernel of, say, half the diameter of the network, at the beginning of the training process is required to reach a globally ordered map.

Having defined the learning rate $\alpha(t)$ and the neighborhood function $h_{ci}(t)$, the weight vector $m_i(t+1)$ of a unit i is adapted by adding a portion $\alpha(t) \cdot h_{ci}(t)$ of the vector difference $[x(t) - m_i(t)]$ to $m_i(t)$ according to Equation 4.4. $x(t)$ denotes the current input vector at time t of the set of input vectors.

$$m_i(t+1) = m_i(t) + \alpha(t) \cdot h_{ci}(t) \cdot [x(t) - m_i(t)] \quad (4.4)$$

As a consequence of Equation 4.4, the weight vector of the winner and the weight vectors of the units in its vicinity are moved towards the input vector. Hence, its more likely that this and similar input vectors are mapped into this very region of the map in successional learning iterations.

Figure 4.3 illustrates a *SOM* along with a graphical representation of the Gaussian neighborhood function. On the left-hand side, the input space \Re^n is depicted. We find the location of the *winner's* weight vector $m_c(t)$ at time t and of the current input vector $x(t)$ in this input space. The weight vector of the *winner* $m_c(t+1)$ after the adaptation at time $t+1$ can be found closer to the input vector. The movement is represented by the solid arrow.

On the right-hand side of Figure 4.3, the *self-organizing map* is depicted by an array of circles with different shades of gray. A dotted arrow shows the relation between the winning unit c and its weight vector in the input space. The different shades of the units represent the different adaptation strengths according to the distance from the *winner*. The darker the unit, i.e. the closer it is to the *winner*, the stronger its weight vector is adapted.

In short, one iteration of the *SOM* training algorithm can be summarized as follows:

1. random selection of an input vector x
2. search for best-matching unit (Eq. 4.1)
3. weight vector adaptation of the winner and its neighbors (Eq. 4.4)
4. modification of learning rate and neighborhood range

Now, that one iteration is finished, the training process proceeds with the selection of the next input vector and continues until a predefined number of

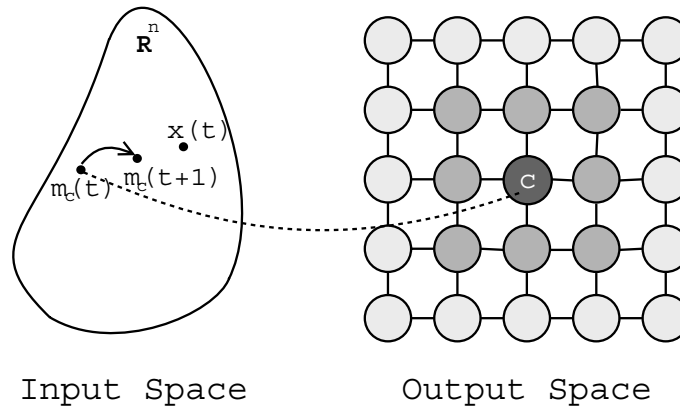


Figure 4.3: Adaptation of weight vectors during SOM training. The adaptation strength of the individual units is indicated by different shades of gray.

training steps or another stopping criterion is met. For example, a certain mathematical quality criterion could serve as a condition for terminating the training process. Alternatively, the training could be terminated, if a stable organization of the input vectors within the two-dimensional lattice is reached.

4.3.2 Document/Term Matrix

Term Encoding

As described in the previous section, in order to be able to use the *self-organizing map* for clustering, the data has to be encoded by n -dimensional numerical vectors. Since our goal is to cluster words according to their semantic similarity, the crucial point is the feature selection and weighting. A preliminary experiment for encoding the words is based on the vector space model in information retrieval (Salton et al., 1975). In a first step, the documents have to be indexed, i.e. in the case of using single-term full-text indexing, a list of all words occurring in the document collection is created yielding an n -dimensional vector with n being the size of the vocabulary. This indexing process can be preceded by using a stemmer to eliminate the inflectional endings of the words in order to increase the quality of content representation. Consider, for example, the removal of the trailing ‘-s’ of ‘-ed’ of English words usually marking plural and past tense respectively. This

would have the effect that, e.g. *hotel* and *hotels*, were treated as the same as opposed to regarding them as separate terms. As already mentioned in Section 2.3, due to the lack of a mature stemming algorithm for the German language, we omitted this preprocessing step. Another method for reducing the noise in the data is using stop-word lists to remove terms such as articles, determiners or conjunctions to name but a few.

The straight-forward process of indexing all words occurring in a document collection of this size can lead to the so-called *curse of dimensionality*, i.e. a huge and therefore computationally costly dimensionality of the vector space. To antagonize this effect, we decided to omit terms from the vocabulary that occur in less than or that occur in more than a certain number of documents. It can be argued that, on the one hand, words that are too infrequently occurring in the collection, e.g. in only one or two documents, are not important enough to be used to describe the documents. This, of course, depends on the desired granularity of document representation. For example, if terms appearing in less than, say, five documents are omitted, topics covered by only four or less documents are not distinguishable from each other. Hence, a trade-off between dimensionality and desired topical granularity of distinction has to be found.

On the other end of the word frequency axis, the most frequent words occurring in nearly all documents do not contribute to the distinction between documents. Examples in our accommodation descriptions are the German equivalents of *and*, *in*, *with*, *for*, *our*, but also *welcome*, *prices*, *location* or *vacation*. Hence, we abstained from using a stop-word list but rather manually defined a certain upper limit for the document frequency to remove such words. For this specific experiment presented hereafter, we also omitted terms that occurred in less than three documents resulting in a vector dimensionality of 9,624.

The distribution of term frequencies in our document collection is depicted in Figure 4.4. As can be seen, the lower limit for term omission has to be chosen with care due to the large numbers of words in this region of the distribution. Note that 29,692 terms, i.e. about 60% of the total number of words, have a document frequency of only one. Contrarily, there is a comparatively small number of terms occurring very frequently, with *and* being the word occurring in 10,869 of 12,471 documents.

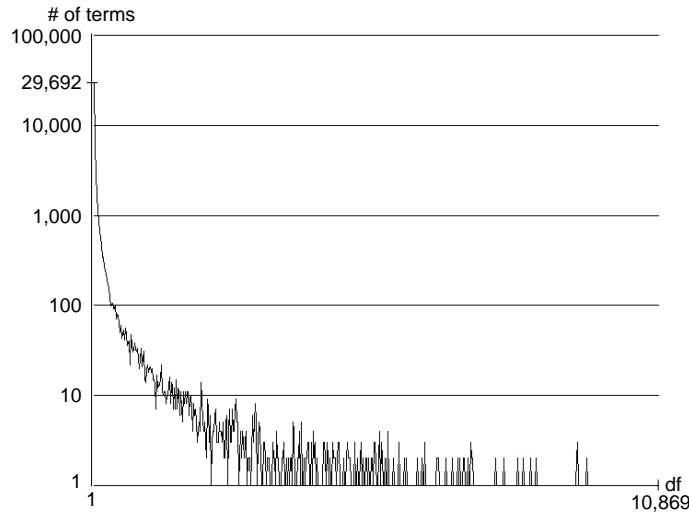


Figure 4.4: Frequency distribution of the vocabulary of the accommodation descriptions. Please note the logarithmic scale of the y-axis.

In the next step of the indexing process, for each document i , a vector $d_i \in \mathbb{R}^n$ is created where the values of the components are determined by the number of occurrences of the respective terms. In other words, the frequencies of word occurrences in the respective documents are used for describing them. Hence, the vector components are usually weighted, because not all words occurring in a document are content-bearing words, i.e. terms that discriminate a certain document from others. Salton and McGill (1983) and Salton and Buckley (1988b) provide a rather extensive discussion on term-weighting techniques. Consider terms such as *accommodation*, *tourism* or *room* as examples. Obviously, they are not a distinctive feature when dealing with a corpus of accommodation descriptions having a rather high frequency in most of the documents. Hence, the importance of a word for a specific document should be determined by a more sophisticated method.

A well-known and frequently used term-weighting approach in information retrieval is called $tf \times idf$, i.e. term frequency times inverse document frequency. The most commonly used $tf \times idf$ variant to calculate the weight w_{ik} of a word i for document k is

$$w_{ik} = tf_{ik} \times \log \frac{N}{df_i} \quad (4.5)$$

with the term frequency tf_{ik} denoting the number of times a term i occurs in a specific document k . The total number of documents is denoted by N and the document frequency df_i is the number of documents in the collection a term occurs in. This weighting has the effect that a word is assigned a high weight, if it occurs frequently in a certain document but is rather rare in the whole collection. Hence, it is important for describing this document. Contrarily, a term in a specific document is considered unimportant, i.e. weighted low, if it occurs in many other documents as well. The set of document vectors for all documents in the collection can be written in form of a term/document matrix as shown in Table 4.1. The documents $d_1 \dots d_N$ are described by numerical components $w_{11} \dots w_{nN}$ representing the importance of the words for the respective documents. Finally, the document vectors are usually normalized to unit length to avoid any bias towards longer vectors.

Terms	Documents					
	d_1	d_2	d_3	d_4	\dots	d_N
$term_1$	w_{11}	w_{12}	w_{13}	w_{14}	\dots	w_{1N}
$term_2$	w_{21}	w_{22}	w_{23}	w_{24}	\dots	w_{2N}
\vdots	\vdots	\vdots	\vdots	\vdots	\ddots	\vdots
$term_n$	w_{n1}	w_{n2}	w_{n3}	w_{n4}	\dots	w_{nN}

Table 4.1: Term/Document Matrix. N documents are described by n terms.

By transposing this matrix, we get a description of the terms on the basis of the documents they appear in, i.e. a document/term matrix. Hence, we have a numerical description of co-occurrences of words, i.e. the row vectors of the matrix that can be used for analyzing semantic relations between them.

Again, our document collection contains 12,471 documents and the full-text indexing yielded a vector dimensionality of 9,624 when omitting terms with a document frequency of less than three. After transposing the matrix, we have 9,624 terms that are described by their occurrence in 12,471 documents. The vectors representing the words are very sparse regarding the ratio of these two

figures. To give an example from our document collection, consider the words *Hasen* (bunnies) and *Schafe* (sheep), obviously animals to be found on farms. Both vectors coincide in 11 elements and have only very few other non-zero elements. In Table 4.2, we have provided a very small part of the document/term matrix, showing the co-occurrence of these terms in several documents. Additionally, we have added the third column showing the according weights of another farm animal, i.e. *Schweine* (pigs). To illustrate that not all vector components in this example are congruent, we have also included the weights for document d_{4801} where the term *Schweine* (pigs) does not appear. Excerpts of two sample documents describing farms, both containing the words *Hasen* and *Schafe*, are shown in Figure 4.5.

Documents	Terms		
	<i>Hasen</i>	<i>Schafe</i>	<i>Schweine</i>
d_1	0	0	0
d_2	0	0	0
\vdots	\vdots	\vdots	\vdots
d_{1260}	0.01	0.0294	0.02
\vdots	\vdots	\vdots	\vdots
d_{2318}	0.01	0.0294	0.02
\vdots	\vdots	\vdots	\vdots
d_{3348}	0.01	0.0294	0.02
d_{3349}	0.01	0.0294	0.02
\vdots	\vdots	\vdots	\vdots
d_{4801}	0.01	0.0294	0
\vdots	\vdots	\vdots	\vdots
d_{12471}	0	0	0

Table 4.2: Document/Term Matrix with some sample vector elements showing the co-occurrence of farm animal names in the descriptions.

An alternative to the vector space model are term distribution models that can be used to characterize the importance of terms based on statistical distributions. Hence, we have also conducted experiments using the Poisson distribution to model the importance of terms (Manning and Schütze, 2000, Chap. 15). In particular, the Poisson distribution can be used to estimate the probability that

Willkommen am [...] Grafenberg- und Grieskareckbahnen. [...] Sommer: Besonders Kinder fühlen sich bei uns wohl. Wir haben: Räder, Scooter, Miniauto, Dreirad, Kindertraktor, Bälle, verschiedene Spiele und Bücher, Sandkasten, Schaukel und Rutsch, Kletterbaum, Basketballnetz, Tischtennis, Dart-Spiel, Nagelstock, Brunnen mit Trinkwasser. Verschiedene Tiere: Hasen, Hühner mit Hahn, Enten, kleiner Hund, Schafe, Rinder und Kälber, Forellenteich, auch zum fischen. Man kann unsere Tiere beobachten, streicheln und auch füttern [...] Rodeln zum verleihen.

Urlaub am Biobauernhof Am Mallhof finden Sie neben komfortablen Gästezimmern das Erlebnis eines voll bewirtschafteten Bauernhofs: Kühe, Schweine, Schafe und Hasen sind besonders für die kleinen Gäste ein Abenteuer. Alle Erzeugnisse unseres biologisch bewirtschafteten Hofes werden selbst verarbeitet und finden sich als Brot, Joghurt, Wurst und Käse am Frühstücksbuffet wieder. Die ‘‘Sonnwiesenbahn’’ liegt direkt vor der Haustür und eröffnet im Winter den Zugang zur Skischaukel. In nur 5 Gehminuten sind Sie auch schon im Zentrum mit der Therme St.Kathrein.

Figure 4.5: Two descriptions of farms containing the names of farm animals such as *Hasen* (bunnies) and *Schafe* (sheep).

a term i occurs exactly k times in a document, given as follows.

$$p(k; \lambda_i) = e^{-\lambda_i} \frac{\lambda_i^k}{k!} \quad (4.6)$$

The parameter λ_i denotes both the mean and the variance of the distribution and is in our case calculated as the average number of occurrences of word i per document, i.e. $\lambda_i = \frac{cf_i}{N}$ with cf_i being the collection frequency of a term and N the number of documents. The collection frequency is the total number of occurrences of a term in the whole collection as opposed to the document frequency, where only the number of documents a term appears in is counted. We can use this value to estimate the document frequency $\widehat{df_i}$ of a term i by multiplying the number of documents predicted to have at least one occurrence by the number of documents N .

$$\widehat{df_i} = N(1 - p(0; \lambda_i)) \quad (4.7)$$

Because one of the basic assumptions of the Poisson distribution is the in-

dependence of term occurrences, this estimate works quite well for non-content words that do not convey information about the document when viewed without the corresponding context. Contrarily, the predicted document frequency for content-bearing words is overestimated due to their nature of being repeatedly used when they are important for a certain document.

We intended to use this property of overestimation as an alternative approach to weigh domain-relevant terms. The analysis of the overestimation factors of the terms has shown that our document collection was too small to provide statistically sufficient data. The only terms that have been overestimated were primarily accommodation names occurring multiple times in a document. Hence, in our particular application, this approach can not serve to provide a useful indicator for important terms. Consequently, we did not use this data for the following experiments.

Experiments

For the clustering experiment, we have trained several *SOMs* with different sizes and training parameters but we will concentrate on one particular map consisting of 35×35 units in order to obtain a representation granularity of about eight terms per unit on average. Generally, it can be said that the results are not satisfying. The organization of the terms does not resemble semantic similarities as they would have been detected by a human. Despite a few units containing a somewhat reasonable set of related terms, the map is unusable for the purpose of assisting ontology engineers.

To point out the low quality of term representation, consider the terms related to *Sauna* as an example. These words are quite a good indication that this term encoding strategy is not useful for our purpose, because the topic of *wellness* is well-represented by the documents in our collection. Therefore, this particular example is not based on insufficient data. The set of 24 words comprising *Sauna* itself and compound words containing *sauna* as a part is scattered over 19 units, which are themselves located in different areas of the map. Some sample words denominating sauna-related objects and activities are *Biosauna* (low temperature sauna), *Kräutersauna* (herbal sauna), *Infrarotsauna* (infrared sauna), *Saunawelt* (sauna world), *Saunabesuch* (sauna visit) or *Saunagang* (one period of heating

up and cooling down during a sauna visit).

To mention a positive example we take a closer look at the terms denoting farm animals described above. One unit of the map represents a remarkably homogeneous set of terms, namely *Kälber* (calves), *Rinder* (cattle), *Hasen* (bunnies), *Hühner* (chicken), *Enten* (ducks), *Schafe* (sheep), *Hund* (dog), *lebendige* (alive), *Ziegen* (goats), *Schweine* (pigs) including only three outliers, i.e. *Kinderfahrräder* (children’s bicycles), *Räder* (bicycles) and *Hinteregger* (proper name). Even the outliers are not too far away from the farm animals, because many descriptions of farms include the possibility to rent bikes. In some other areas of the map one can find accumulations of words representing major topics present in the collection such as skiing or terms related to a specific region.

Nevertheless, it requires quite an intellectual effort to interpret this map as semantically ordered. Obviously, the context used for describing the terms, i.e. a complete document, is too large to provide distinctive features that can be used to reflect semantic relations. As we have shown, this method works only for a few particular terms that co-occur in a limited set of documents but nearly nowhere else. The majority of terms appear in sets of documents whose contents are not homogeneous enough. Consequently, we reduced the context describing the terms by defining a window of fixed size around the word as will be detailed in the next section.

4.3.3 Encoding the Semantic Contexts

Term Encoding

Early experiments by Ritter and Kohonen (1989) using the *self-organizing map* have shown that it is possible to cluster terms according to their syntactic category as well as their semantics by encoding the local contexts of words. For their experiments, an artificially created data set was used comprising three-word sentences that consist of nouns, verbs and adverbs, such as, e.g. “*Jim speaks often*”, “*Bob sells beer*” or “*Mary buys meat*”. The terms were numerically encoded by 7-dimensional *random vectors* normalized to unit length. Then, the *context vector* for each word was created by concatenating the two mean vectors derived from the random vector representations of the surrounding words regarding the

word order in the sentence. In this particular experiment, the direct predecessor/successor pair was considered as context. Additionally, the 7-dimensional vectors representing the words themselves were also attached to the context vectors yielding a 21-dimensional vector description of each word. The respective vector elements representing the symbol part of the context vector were weighted with a value of 0.2 to reduce its influence on the context vector.

The resulting *word category map* depicted in Figure 4.6 clearly showed three main clusters corresponding to the three word classes used in the sentences. It should furthermore be noted that within each cluster, the words of a class were arranged according to their semantic relation. This was imposed by the structure of the sentence patterns that determined the structure of the data set used for the experiments. For example, the adverbs *poorly* and *well*, located on the left border of the map, were located closer together than *poorly* and *much*, the latter was located spatially close to *little*. Examples from a different cluster – namely that of verbs covering the bottom half of the map – would be *likes/hates*, *buys/sells* or *visits/phones*.

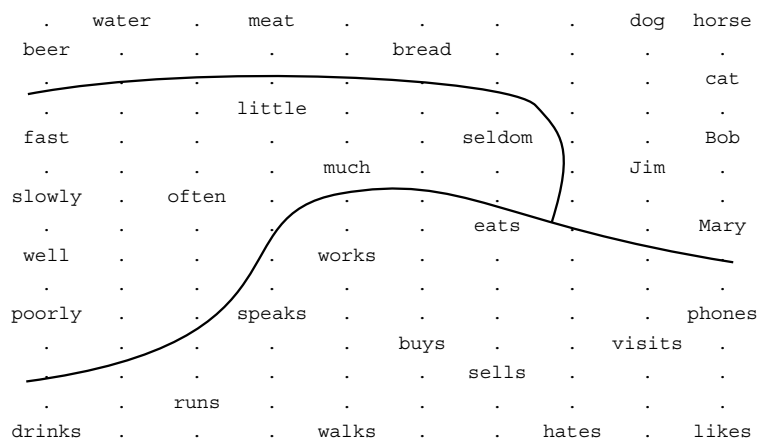


Figure 4.6: Semantic map of the original experiments by Ritter and Kohonen (1989). The manually drawn cluster boundaries separate the syntactic word classes.

Later experiments using a collection of 200 fairy tales by the Grimm Brothers have shown that this method works well with real-world text documents (Honkela et al., 1995). The terms on the *SOM* were primarily divided into three clusters,

namely nouns, verbs and all other word classes occurring in the texts. Within the cluster containing nouns, again, groupings of semantically related word such as *daughter/father/mother* or *night/day* can be detected. This semantically induced ordering can also be observed in the verb cluster. The third group consisting of the remaining word classes such as pronouns, adjectives, adverbs, determiners or numerals is predominantly organized accordingly.

The results of these experiments have been elaborated later to reduce the vector dimensionality for document clustering in the WEBSOM project (Kaski et al., 1998). As has been outlined in the previous section, indexing all terms occurring in large document collections for creating vector representations results in a large dimensionality of the document space. Hence, the characteristic of the word category map to map semantically similar terms onto similar units can be used to reduce the dimensionality. First, the vector representations of the words are created as will be described later. Next, a word category map is trained with the terms occurring in the document collection to subsume words with similar context under one semantic category, i.e. unit. The number of units on this map determines the dimensionality of the reduced vector space. Finally, for each document, a histogram is created by counting the number of times the words are represented by the various units on the word category map. This histogram is normalized and used as the new, *compressed* document vector. As a result, semantically similar words are subsumed under one category and therefore represented by only one component of the new vector. Because new methods of dimensionality reduction have been developed, the word category map has been dropped for this particular purpose as reported by Kohonen et al. (2000).

Nevertheless, since our objective is to disclose semantic relations between words, we decided to use word category maps. First, to serve as a representation of the terms occurring in the document collection, an n -dimensional numerical random vector is created for each term. The random values of the vector components are drawn from a uniform probability distribution, thus, being statistically independent. Creating a truly independent vector representation with one dimension for each term, i.e. the inner product of all vector pairs equaling zero, would be computationally expensive regarding the large number of terms in the vocabulary of a text collection like ours. Honkela (1997) has shown that, in case

of n being large enough, the random vectors are quasi-orthogonal and therefore sufficiently independent of each other. Thus, unwanted geometrical dependence of the word representation is avoided. This is a necessary condition, because otherwise the clustering result could be dominated by random effects overriding the semantic similarity of words.

In Figure 4.7, reproduced from Honkela (1997), the distributions of pairwise inner products of the random vectors for different dimensionalities are depicted. It can be seen that the vectors are not perfectly independent but bear enough statistical independence to be used for our experiments even with a dimensionality as low as 90. We have tested the quasi-orthogonality of the random vectors created for our experiments and came to the same results. The distribution of the pairwise inner products of 90-dimensional random vectors showed a standard deviation of 0.139, about the same as can be derived from the corresponding distribution in Figure 4.7.

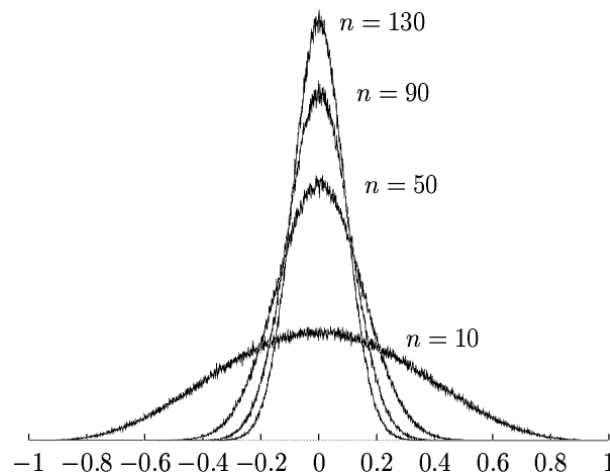


Figure 4.7: Distribution of pairwise inner products of the random vectors with n dimensions (reproduced from Honkela (1997)).

The assumption these experiments are based on is, that in textual descriptions dominated by enumerations, semantic similarity is captured by contextual closeness within the description. For example, when arguing about the attractions offered for children, things like a *playground*, a *sandbox* or the availability of a *baby-sitter* will be located close together in the text. Using the illustrative exam-

ple of the farm animals presented in the previous chapter (cf. Figure 4.5), those were also mentioned one after another rather than randomly mixed with, say, toys. Even within the set of children’s toys a grouping according to types of toys can be noticed, i.e. *Räder* (bikes), *Scooter*, *Miniauto* (small cars), *Dreirad* (tricycle) and *Kindertraktor* (children’s tractor) are enumerated sequentially. Then follow several other types of toys. In the second description, animals and the goods produced on the farm are well-separated. Analogously, the semantic grouping of words is true for recreational facilities like a *sauna*, a *steam bath* or an *infrared cabin* in descriptions of spa-like hotels.

To capture this contextual closeness, we use a context window where a particular word i is described by the set of words that appear up to a fixed number of words before and after word i in the textual description. Given that every word is represented by an n -dimensional random vector, the context vector of a word i is built as the concatenation of the average vectors of all words preceding as well as succeeding word i at the respective positions. In other words, the vectors representing the words that occur at a certain position relative to the term in question throughout the whole document collection are averaged. This average vector represents the context for a term at a specific displacement.

Technically speaking, a context vector x_i representing word i is a concatenation of n -dimensional vectors $x_i^{(d_j)}$ that are the mean vectors of terms occurring at the set of displacements $\{d_1, \dots, d_N\}$ of the term as given in Equation 4.8. Consequently, the dimensionality of x_i is $n \times N$. Finally, the context vectors are normalized to unit length. This kind of representation has the effect that words appearing in similar contexts are represented by similar vectors in a high-dimensional space.

$$x_i = \begin{bmatrix} x_i^{(d_1)} \\ \vdots \\ x_i^{(d_N)} \end{bmatrix} \quad (4.8)$$

The vector $x_i^{(d_1)}$ computes as the mean of all vectors representing the words that occur at displacement d_1 from the term x_i in the whole document collection. Consider, for example, the term *Skifahren* (skiing). The set of words occurring directly before the term at displacement -1 consists of words like *Langlaufen* (cross

country skiing), *Rodeln* (toboggan), *Pulverschnee* (powder snow) or *Winter* to name but a few. By averaging the respective vectors of these terms, a statistical model of word contexts is created. To illustrate the context vector generation with an example, consider x_i in Equation 4.9 as the vector describing a term i by the average vectors of its immediate neighbors. In other words, the average contexts of words at displacements -1 and $+1$ constitute the contextual description.

$$x_i = \begin{bmatrix} x_i^{(-1)} \\ x_i^{(1)} \end{bmatrix} \quad (4.9)$$

As opposed to the original method of incorporating the random vector of the term itself into the context vector, we omitted the symbolic part since we want terms that have the same context to be mapped onto the same unit in any case. To tackle the problem of words occurring at the beginning or at the end of a text, i.e. possibly not having a context at a certain displacement, we assigned random values to the missing vector components to maintain statistical independence. Otherwise, terms that appear, for example, at the beginning of the documents would have been characterized to have something in common, if the missing values would have been assigned a fixed value.

Preprocessing the Data

As already mentioned, it has been shown with a text collection consisting of fairy tales that the word classes clearly dominate the cluster structure of such a map when using free-form text documents. To create semantic maps primarily reflecting the semantic similarity of words rather than categorizing word classes, we removed words other than a certain word class. In particular, we discarded words other than nouns and proper names from the accommodation descriptions for the experiments presented hereafter. Nouns, traditionally regarded as names of persons, places or things, represent the most important word class due to the character of our application domain, i.e. most of the features that can be searched for are things and places. Of course, if the goal of an ontology engineer is to model certain subjective qualities that were also missing in our field test such as *cheap*, *upmarket*, *romantic*, *cozy* or *quiet*, to name but a few, adjectives have to be extracted.

Focusing on the extraction of terms denoting things and places, we used the characteristic, unique to the German language, of nouns starting with a capital letter to filter the nouns and proper names occurring in the texts. Obviously, using this method, some other words like adjectives, verbs or adverbs at the beginning of sentences or in improperly written documents are also filtered. Contrarily, some nouns can be missed when being incorrectly written without an upper case letter at the beginning. A different method of determining nouns or other relevant word classes, especially for languages other than German, would be part-of-speech (POS) taggers that try to determine the word classes in a text. But even state-of-the-art POS taggers do not reach an accuracy of 100% due to unresolvable ambiguities (Manning and Schütze, 2000).

For the rest of this section, the numbers and figures presented, refer to the already preprocessed documents, if not stated otherwise. The collection consisting of 12,471 documents contains a total number of 481,580 capitalized words, i.e. on average, a description contains about 39 words very high probability of being nouns or proper names. The vocabulary of the document collection comprises 35,873 unique terms, but for the sake of readability of the maps we reduced the number of terms by excluding those occurring less than ten times in the whole collection. Consequently, we used 3,662 terms for creating the semantic maps.

In Figure 4.8, a natural language description of a holiday flat in Vienna is shown. Beginning with the location of the flat, the accessibility by public transport is mentioned, followed by some terms describing the dining and living room together with enumerations of the respective furniture and fixtures. Other parts of the flat are the sleeping room, a single bedroom and the bathroom. In this particular example, the only words not being nouns or proper names are the determiner *Die* and the preposition *In* at the beginning of sentences. For the sake of convenience, we have provided an English translation in Figure 4.9.

Experiments

For encoding the terms we have chosen 90-dimensional random vectors. The vectors used for training the semantic map depicted in Figure 4.10 were created by using a context window of length four, i.e. two words before and two words after a term. But instead of treating all four sets of context terms separately, we

Die 118 m2 große Ferienwohnung befindet sich in absolut ruhiger Lage am Stadtrand von Wien (23. Bezirk-Mauer), ist jedoch sehr verkehrsgünstig gelegen: In 10 Gehminuten erreicht man die Schnellbahn, nach 15 Fahrminuten ist man in Wien Mitte (Innere Stadt). Die Wohnung besteht aus einem 42 m2 großen Wohn-Eßraum mit großem offenen Kamin, SAT-TV, einer komplett eingerichteten Küche mit Geschirrspüler zwei Schlafzimmer mit mit je zwei Einzelbetten, 1 Einbettzimmer einem Badezimmer mit Wanne, Doppelwaschbecken und Dusche, einem Extra-WC sowie einer großen gartenseitig gelegenen Terrasse mit Sitzgarnitur und Ruhebetten. Die großzügig angelegte Ferienwohnung ist komplett neu eingerichtet und läßt für einen erholsamen und gemütlichen Aufenthalt keine Wünsche offen.	Die ___ __ ____ Ferienwohnung ____ _ __ ____ Lage __ Stadtrand __ Wien __ Bezirk_ Mauer_ __ ____ ____ In __ Gehminuten ____ __ Schnellbahn_ ____ Fahrminuten __ __ _ Wien Mitte _Innere Stadt_ Die Wohnung ____ _ __ __ ____ Wohn_Eßraum __ ____ Kamin_ SAT-TV_ ____ ____ Küche __ Geschirrspüler __ Schlafzimmer __ __ __ Einzelbetten_ _ Einbettzimmer ____ Badezimmer __ Wanne_ Doppelwaschbecken __ Dusche_ ____ Extra_WC ____ ____ ____ Terrasse __ Sitzgarnitur __ Ruhebetten_ Die ____ Ferienwohnung __ ____ ____ ____ Aufenthalt ____ Wünsche ____
---	---

Figure 4.8: A sample description of a holiday flat in a suburb of Vienna. On the left-hand side, the original description is shown, and on the right-hand side the remaining words after removing all words not starting with a capital letter are presented.

have put terms at displacements -2 and -1 as well as those at displacements $+1$ and $+2$ together. Then the average vectors of both sets were calculated and finally concatenated to create the 180-dimensional context vectors. Further experiments have shown that this setting yielded the best result.

For example, using a context window of length four but considering all displacements separately, i.e. the final context vector dimensionality is 360, has led to a map where the clusters were not as coherent as on the map shown below. Since the vocabulary constituting the documents comprises nearly 36,000 unique words, the chance of similar patterns of five-word sequences happening a sufficient number of times are rather low, especially because the collection has a total number of nearly 500,000 words. Hence, the context vector representations are

<p>the_(fem.), holiday flat, location, outskirts, Vienna, district, Mauer, in, minutes to walk, urban railway, minutes to drive, Wien Mitte (station name), inner, city, the_(fem.), flat, living, dining room, fireplace, satellite tv, kitchen, dishwasher, sleeping room, single beds, single-bed room, bathroom, bathtub, double washbasin, shower, separate toilet, terrace, chairs and table, couches, the_(fem.), holiday flat, stay, wishes</p>

Figure 4.9: English translation of the terms shown on the right-hand side in Figure 4.8.

likely to bein the second more different from each other with this setting. Due to the nature of the vocabulary size of a specific document collection to converge on a certain number with linear increase of the collection size, it can be assumed that a larger document collection could alleviate this deficiency. This is especially true for domcuments covering a limited domain. The convergence of vocabulary size with much larger text collections has been analyzed by Zobel et al. (1995).

On the other hand, a smaller context window of length two, taking only the surrounding words at displacements -1 and $+1$ into account, had a similar effect, yet not as strong. This indicates that the amount of text available for creating such a statistical model is crucial for the quality of the resulting map. By subsuming the context words at displacements before as well as after the word, the disadvantage of having an insufficient amount of text can be alleviated, because having twice the number of contexts with displacements -1 and $+1$ is simulated. Due to the enumerative nature of the accommodation descriptions, the exact position of the context terms can be disregarded.

The *self-organizing map* depicted in Figure 4.10 consists of 20×20 illustrated by circles. The semantic clusters that are shaded gray have been determined by manual inspection of the map. These clusters consist of very homogeneous sets of terms related to distinct aspects of the tourism domain. The parts in the right half of the map that have not been shaded, predominantly contain proper names of places, lakes, mountains, cities or accommodations. However, it shall be noted, that e.g. names of lakes or mountains have been homogeneously grouped into separate clusters.

In the upper left corner, mostly verbs, adverbs, adjectives or conjunctions are

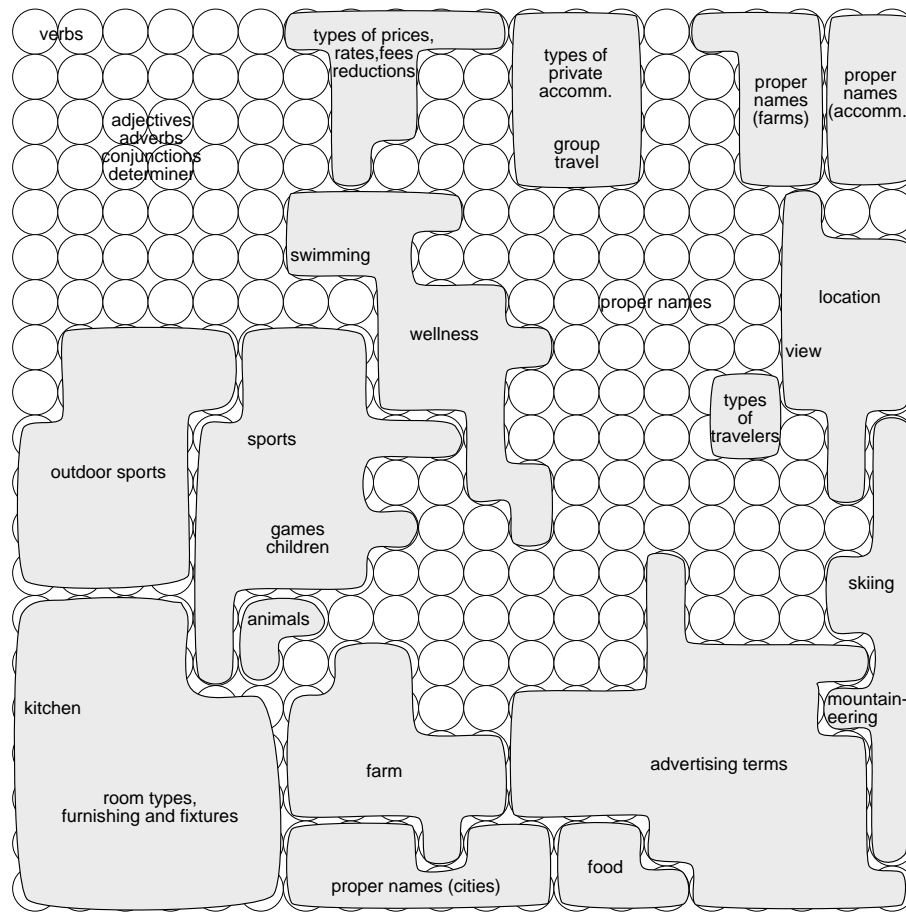


Figure 4.10: A self-organizing semantic map of terms in the tourism domain with labels denoting general semantic clusters. The cluster boundaries have been drawn manually.

located. The unit in the top left corner of the map represents 48 terms that are mostly verbs such as choose, call, escape, feel, have, are, experience, enjoy, make, explore and the like. These are terms that have been inadvertently included in the set of nouns and proper names as described in the previous subsection. Again, the property of dominantly clustering word classes can be noticed, because these words are rather strictly separated from the rest of the vocabulary.

In the upper part of the map, a cluster containing terms related to pricing, fees and reductions can be found containing terms such as *Kinderermäßigung* (reduction for children), *Vollpension* (full board), *Zustellbett* (additional bed), *Zim-*

merpreise (room charges), *Zusatzkosten* (additional costs), *Zuschlag* (surcharge), *Kurzaufenthaltszuschlag* (surcharge for short-term stays) or *Einzelzimmerpreis* (single room charge). Other clusters in this area predominantly deal with words describing accommodation types and contain terms related to group travel. In the top-right corner two strong clusters containing farm and accommodation names can be found. Especially on the top-right unit we can find almost exclusively first names and surnames that are often part of private accommodations or hotels in skiing areas, e.g. *Michael*, *Erika*, *Jäger* or *Wechselberger*.

On the right-hand border of the map, terms denoting geographical locations such as *central*, *outskirts*, or *close to a forest* as well as various views, e.g. *lake view*, *panorama view* or *valley view*, have been mapped. This happened due to the usually combined description of the location of an accommodation and what can be seen from it. Skiing and mountaineering-related words are located adjacent to the lower boundary of this cluster, taking up about 14 units stretching to the lower right corner of the map.

The cluster labeled *advertising terms*, covering a large area in the bottom-right part of the map, predominately contains words that are found at the beginning of the documents where the pleasures awaiting the potential customer are described. These are words like *atmosphere*, *recreation*, *beauties*, *magic* or *dream vacation*, but also negative things one can escape when going on holidays, e.g. *stress* or *everyday life*.

A very dominant cluster containing words that describe room types, furnishing and fixtures can be found in the lower left corner of the map covering roughly 6×7 units. Here, we can find nearly everything from electrical appliances over washing machines to bath tubs and showers, of course being ordered according to similarity. Figure 4.11 presents four units in the lower left corner of the map in more detail. The left-most unit contains words relating to heating and sanitary facilities whereas the next unit mainly covers kitchen and dining-related terms. The third unit contains similar terms with a focus on room types whereas terms relating to living and sleeping rooms have been mapped onto the right-most unit.

Starting from the bottom left corner of the map moving upwards, a transition from facilities located inside rooms towards outdoor facilities that are usually located around accommodations such as *slide*, *playground* and furthermore towards

toilette suedbalkon wohnbereich diele elektroheizung garderobe doppelwaschbecken wohnkuechen waschbecken wc bidet	kuechenblock essecke wohnkueche couch schlafgelegenheit ausziehcouch vorraum stockbetten kuechenzeile wohnzimmer essplatz esszimmer doppelcouch wohnraum flur	kochenische wanne sofa badewanne waschraum doppelbett schlafmoeglichkeiten hotelzimmer essraum kochecke duschen kinderzimmer schlafraum wohnschlafzimmer badezimmer wohnstube	bad stockbett doppelzimmern doppelbettzimmer dusche schlafraeume zimmerausstattung dreibettzimmer wohnschlafraum schlafzimmer zimmer fliesswasser einbettzimmer komfortzimmer doppelschlafzimmer schlafraeumen gaestezimmer
---	---	--	---

Figure 4.11: An enlargement of the cluster covering room types, furnitures and fixtures located in the lower left corner of the map.

sports facilities like *volleyball field*, *soccer field* or *basketball court* is noticeable. A cluster containing names of action sports, e.g. *paragliding*, *white-water rafting*, located close to the outdoor sports activities continues this topical thread.

Interesting inter-cluster relations showing the semantic ordering of the terms can be found in the bottom part of the map. The cluster labeled *farm* contains terms describing, amongst other things, typical goods produced on farms like, *organic products*, *jam*, *grape juice* or *schnaps*. In the upper left corner of the cluster, names of farm animals (e.g. *pig*, *cow*, *chicken*) as well as animals usually found in a petting zoo (e.g. *donkey*, *dwarf goats*, *cats*, *calves*) are located. This cluster describing animals adjoins a cluster primarily containing terms related to children, toys and games. Some terms are *playroom*, *tabletop soccer*, *sandbox* and *volleyball*, to name but a few.

Close to the center of the map, a cluster covering wellness-related terms is located. Here, we can find all types of saunas, pools or massages. To provide an example for showing the rich vocabulary, we found a wealth of terms describing sauna-like recreational facilities having in common that the vacationer sojourns in a closed room with well-tempered atmosphere, e.g. *sauna*, *tepidarium*, *bio sauna*, *herbal sauna*, *Finnish sauna*, *steam sauna*, *thermarium* or *infrared cabin*. When creating the ontology for our first prototype, we did not have such a variety of terms describing nearly the same concept in mind.

This representation of a domain vocabulary supports the construction and enrichment of domain ontologies by making relevant concepts and their relations

evident. Using this visualization approach, on the one hand, major semantic categories identified by inspecting and evaluating the semantic map can be used as a basis for a top-down ontology engineering approach. On the other hand, the clustered terms, extracted from domain-relevant documents, can be used for bottom-up engineering of an existing ontology.

4.4 Finding Geographical Peculiarities

As described in Section 3.3.3, the field trial has shown that geographical features are important search criteria in the tourism domain. Such geographical features are landmarks, local attractions, of even food specialties specific to a certain region. In order to facilitate the extraction of such features from the vocabulary we have modified the $tf \times idf$ term-weighting approach explained in Section 4.3.2.

In particular, we have changed the notion of term frequency tf and document frequency df . Instead of counting the number of occurrences of a certain word within the scope of single documents we count it with respect to the geographical area the document belongs to. To remove any bias towards geographical regions with more accommodations than others, the numbers of occurrences are normalized by the number of documents in the respective region. Consequently, we get the relative number of occurrences for the words for each federal state. Finally, these numbers are transformed to a uniform scale by dividing the relative term frequencies by the sum of all relative term frequencies for each word. More formally, the importance w_{ik} of a term i for region k is given as

$$w_{ik} = \frac{rf_{ik}}{N_k} \times \frac{1}{\sum_k \frac{rf_i}{N_k}} \quad (4.10)$$

with N_k being the number of documents in region k and rf_{ik} the *region frequency* of a term i in region k , i.e. the number of documents related to a specific region the terms appears in.

Terms that are exclusively occurring in documents related to a certain region can be regarded to express something that characterizes this area with a rather high probability. For the sake of convenience, we have included a map of Austria showing the borders of the federal states in Figure 4.12. It has to be noted that

we have the information about a specific text description of an accommodation belonging to a federal state. Furthermore, we are able to assign a document to an even smaller geographical area based on the zip code of the city in which the accommodation is located.

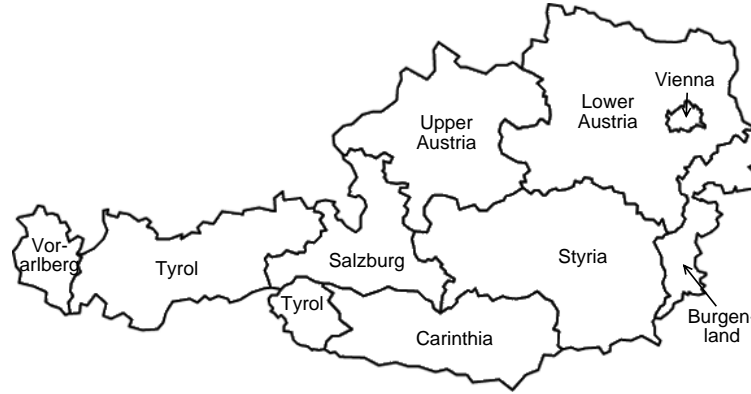


Figure 4.12: A map of Austria and its nine federal states.

Consider, for example, terms that appear only in descriptions of accommodations that are located in Vienna but not in any other federal state. We have ranked the 687 remaining words according to their document frequency and obtained a nice enumeration of sights and attractions that could be included in the ontology of the **Ad.M.In** system. Table 4.3 shows the 30 top-ranked terms that occur only in descriptions of accommodations located in Vienna. Because of Vienna being a federal state and a city at the same time, the top-ranked terms predominantly denote buildings, streets, squares or parks that are well-known to tourists. There are only a few terms that do not belong into these categories such as *air*, which is most probably a fragment of *air condition*.

The terms occurring in the larger, more rural federal states are quite different in nature. Regarding Lower Austria, among the top-ranked terms are the names of three of the four quarters it is divided into, i.e. *Waldviertel*, *Mostviertel* and *Weinviertel*. Further terms are names of smaller region like *Wachau* or *Kamptal*, names of castles (e.g. *Schallaburg*) and other places of interest for tourists. States in the west of Austria that are more mountainous such as Tyrol or Vorarlberg are characterized by names of mountains, mountain ranges, glaciers, lakes or valleys. Sample terms are *Zillertal* (valley), *Brixental* (valley), *Pitztal* (valley), *Zugspitze*

rank	term	rank	term	rank	term
1	stephansdom	11	mariahilferstraße	21	biedermeierstil
2	ringstraße	12	einkaufsstraßen	22	westbahnhofes
3	staatsoper	13	burgtheater	23	walzer
4	stephansplatz	14	air	24	vollklimatisierten
5	mariahilfer	15	u-bahnstation	25	uno
6	westbahnhof	16	riesenrad	26	spittelberg
7	schönbrunn	17	raimundtheater	27	parlament
8	ringstrasse	18	kärntnerstraße	28	opernkarten
9	prater	19	donauinsel	29	altwiener
10	wien-aufenthalt	20	museumsquartier	30	wienerberg

Table 4.3: List of terms occurring exclusively in descriptions of Viennese accommodations.

(mountain), *Wetterstein* (mountain), *Kaunertaler* (glacier) or *Achensee* (lake) in Tyrol and *Rheintal* (valley), *Montafon* (region), *Hochjoch* (mountain) or *Klostertal* (valley) in Vorarlberg. Term occurring in Burgenland, the most easterly state, are coined by the extensive and well-known viticulture. Hence, the list of terms contains, for example, *Blaufränkischland* (specific wine region), *Weinprobe* (wine tasting), *Prädikatswein* (specific quality of wine) or *Weinlehrpfad* (wine trail).

We have also used the zip codes to create sets of documents relating to smaller areas than federal states. The results are very different depending on the area. Consider the district of Bregenz, which is located in Vorarlberg, as an example. Because this district includes a popular holiday region, the amount of text available is sufficient to have interesting words included. Especially references to dairies producing renowned cheese and some smaller geographical highlights yield terms that can be considered as potential candidates for enhancing the ontology. Generally, it can be said that for most of the smaller regions the number of documents seems to be too small and the terms are therefore too specifically determined by the wording used by single persons. Hence, they do not reflect a kind of statistical profile where the document frequencies indicate a general agreement on the use of certain terms.

So far, we have presented lists of terms that are exclusively appearing in documents related to only one region. Terms denoting special features of regions crossing borders between federal states, e.g. those presented in Table 4.4, are

excluded in these lists because they usually occur in more than just one state. In order to include these terms into the list, the restriction of a term occurring in one specific state but not in any other can be relaxed by defining thresholds to allow the occurrence of a term also in other states to a certain extent. As shown in the table, the occurrences of *Salzkammergut* reflect its geographical location quite well, most of it being located in Upper Austria and rather small parts in Salzburg and Styria. The same accounts for *Arlberg*, a rather famous skiing area shared by Vorarlberg and Tyrol and *Thermenland*, a region in Burgenland and Styria where many thermal baths are situated.

Terms	Federal States								
	Vie	Low. A	Upp. A	St	Bgl	Sbg	Car	Tyr	Vbg
Salzkammergut	0	0	0.8	0.14	0	0.06	0	0	0
Salzkammergutes	0	0	0.76	0.11	0	0.13	0	0	0
Salzkammergutseen	0	0	0.89	0	0	0.11	0	0	0
Arlberg	0	0	0	0	0	0	0	0.11	0.89
Arlberger	0	0	0	0	0	0	0	0.15	0.85
Arlbergs	0	0	0	0	0	0	0	0.02	0.98
Thermenland	0	0	0	0.88	0.12	0	0	0	0
Thermenregion	0	0.13	0.16	0.35	0.36	0	0	0	0
Thermenhotel	0	0	0.2	0.62	0.18	0	0	0	0

Table 4.4: Sample terms denoting or related to regions that are crossing the borders of federal states.

For these terms occurring in more than one region, the standard deviation of the geographical distribution can be used as indicator of its type. Usually, stop words (if not removed) like *in*, *and* but also frequently occurring domain-dependent words like *atmosphere*, *deliciousness* or *guest* are evenly distributed and have therefore a small standard deviation. Words that are discriminating the various regions or federal states by having rather different weight values show a higher standard deviation. Terms that occur in only one region or state obviously have the highest standard deviation.

4.5 Discussion

In this chapter, we have presented a method, based on the *self-organizing map*, to support the construction and enrichment of domain ontologies. The words occurring in free-form text documents from the application domain are clustered according to their semantic similarity based on statistical context analysis. More precisely, we have shown that when a word is described by words that appear within a fix-sized context window, semantic relations of words unfold in the *self-organizing map*. Thus, words that refer to similar objects can be found in neighboring parts of the map. The two-dimensional map representation provides an intuitive interface for browsing through the vocabulary to discover new concepts or relations between concepts that are still missing in the ontology. We illustrated this approach with an example from the tourism domain. The clustering results revealed a number of relevant tourism-related terms that can now be integrated into the ontology to provide better retrieval results when searching for accommodations.

Furthermore, we have presented a method to detect words that are descriptive for specific geographical areas. The analysis of word occurrences with regard to the location of the described accommodation yields valuable information for extending the ontology. The measure of importance derived from the standard deviation of the geographical distribution of the terms can also be used for highlighting the terms on the semantic map. We achieved this form of visualization by analysis of self-descriptions written by accommodation providers, thus assisting substantially the costly and time-consuming process of ontology engineering.

Chapter 5

Conclusions

In this thesis we have presented Ad.M.In, a natural language interface for searching accommodations throughout Austria. Our objectives were, first, to provide a more convenient interface compared to a common form-based interface inhibiting users from being able to express their actual intention, and second, to test whether this method of searching for information will be accepted by users.

Hence, we have developed a natural language interface as described in Chapter 2 that allows for posing queries in a natural language question in either German or English. The system uses a domain ontology covering concepts and semantic relations between these concepts that are relevant to a particular application. The query is processed by a chain of modules extracting the essential terms, associating the terms with their meaning and transforming them to a structured database query according to domain-dependent rules. Generally, the system has been designed to be application independent, i.e. the domain knowledge has been separated from the processing logic to permit the application of the system to other domains. Based on a prototype implementation in the tourism domain, we wanted to test our assumption that shallow natural language processing is sufficient in a well-defined, limited domain and an interface of this kind will be accepted even though it is necessary to type complete sentences as queries.

Consequently, we conducted a field trial where we promoted the possibility of searching for accommodations via natural language on the *Tiscover* website and monitored the queries that were entered. Chapter 3 presents an analysis of the queries posed during ten days in March 2001, showing that, first, the com-

plexity of the queries was to a great extent within the linguistic coverage of our shallow natural language processing. Nevertheless, we have been adverted to some linguistic details that are necessary to be dealt with in order to provide better quality search results. Second, more than half of the users entered complete, grammatically correct sentences which is quite surprising when comparing this result to studies investigating the queries of conventional keyword-based search engines where the average number of query terms is rather low. Third, we found out that the ontology needed improvement regarding the vocabulary, because the conceptual coverage of the system was partly insufficient. Especially synonyms for existing concepts were missing as well as rather subjective criteria people have searched for, e.g. ‘*upscale hotel*’ or ‘*romantic weekend*’. Another application-specific issue was the missing information about regions and geographic landmarks.

Due to the lack of feedback by the users during the field trial, we additionally conducted a usability study. A questionnaire was created defining several scenarios including descriptions of specific search tasks the test persons had to carry out using both the conventional *Discover* interface as well as our natural language interface. One of the major conclusions that can be drawn from this study is that, when being confronted with such a natural language interface for the first time, it requires a certain time to get accustomed to this type of searching for information. After this phase, most of the test persons favored the natural language interface and found that it was a more convenient way for solving the tasks. Again, the users were not deterred from searching by the effort of having to phrase the query in complete natural language sentences involving more typing than with the conventional interface. An important point that has been criticized was the insufficient feedback of the system about the recognized concepts. This deficiency has to be addressed, because otherwise the search results might not be considered trustworthy.

The results of both the field trial and the usability study have led to further research into (a) an alternative approach method of knowledge representation and query processing, and (b) a supporting tool for aiding ontology engineers in creating and enhancing domain ontologies. Although the findings of the field trial were promising regarding the quality of the search results, the rule and grammar-

based approach of the first prototype seemed to be too restricted for the future regarding our experiences during the creation of the system and the analysis of users queries. Hence, a second system outlined in Section 2.4 has been developed combining some of the language processing methods already in use, but with a different form of knowledge representation based on associative networks and an alternative approach regarding query processing based on spreading activation. This approach enables a more convenient definition of non-taxonomic relations that can represent similarities across the network of concepts and it also facilitates the definition of vague concepts like *upscale* or *romantic* as mentioned above. Furthermore, the exact-match strategy where only accommodations are returned as search result that match the criteria defined by the user has changed to a best-match strategy where all accommodations are ranked according to the similarity of the users' requirements taking into account the semantic knowledge defined by the ontology.

The second direction of research has led to the development of a visualization technique for representing semantic similarities between words in domain-related documents as discussed in Chapter 4. We use a neural networks-based approach to cluster terms extracted from free-form text descriptions of the accommodations. Separated from the structured information about the accommodations, these documents contain loads of information that can be used to provide search results of better quality. Due to the large vocabulary constituting these documents, a sensible representation had to be found reflecting the similarity of words rather than providing the ontology engineer with long lists of words. Hence, we used the characteristic of the *self-organizing map*, i.e. a neural network model with unsupervised learning function, of providing a topology-preserving mapping from a high-dimensional input space onto a two-dimensional output space. In other words, data that are similar in the input space will be located spatially close on the map.

To describe the terms we first tried an approach based on the vector space model used in information retrieval where documents are described by numeric vectors containing information about the words appearing in the documents. By transposing this representation and describing words by the documents they occur in, we have clustered them with the *SOM* to gain insight into the semantic

relations between the words. Despite a few positive examples where homogeneous groups of similar terms were found, the overall quality of the map representation can be regarded as being insufficient. Apparently, the assumed context of whole documents was too large to provide a semantic description of the words. Consequently, we reduced the context that describes the words to a window of a specific size including the terms before and after the word regarding the sequential position in the text. Using this approach was induced by the observation that the accommodation descriptions are dominated by grouped enumerations of semantically related concepts such as facilities, services or attractions. The semantic vector representations were created by concatenations of average vectors derived from the words occurring in the context window. Then, we have presented a map that clearly showed large clusters of semantically related concepts such as types of sports or facilities of accommodations to name but a few. These large clusters were themselves structured according to subcategories and relations to other clusters.

A further issue that has arisen from the field trial was the extraction of names of geographical landmarks or attractions. We have introduced a term-weighting scheme to weigh terms according to their importance for specific regions and used the standard deviation of the weight values as indicators for the importance of a term.

The **Ad.M.In** natural language interface has shown to have the potential to be an alternative way of searching for information in certain domains such as tourism where people are often characterized by having rather unstructured imagination of their information need (O'Brien, 2001). Nevertheless, some issues are still open for further research.

One point that might need further consideration is the integration of more sophisticated natural language processing tools such as chunking parsers and the like to further improve the recognition of relevant concepts. It has been noted during the field trial that some multi-word terms were not recognized as expected and have therefore led to unsatisfactory results. However, the level of *shallow* natural language processing should not be left to avoid unnecessary complexity of the system. Another important point is further usability research to explore the way users interact with and expect from such systems, because even the best

natural language processing is useless if nobody wants to use such an interface due to feeling uncomfortable.

Regarding the proposed tool for supporting ontology engineering, it has to be noted that visualizing all the terms of the vocabulary in an appropriate resolution of display requires a *SOM* of according size. The problem with large *SOMs* is that they tend to be rather impractical to survey, of course depending on the graphical implementation of the map interface. Nevertheless, it would be sensible to use variants like the *growing hierarchical self-organizing map (GHSOM)* that consists of a hierarchy of independent *SOMs* (Dittenbach et al., 2000, 2002c). The *SOMs* and the layers of the hierarchy grow according to structure of the data, i.e. if the representation of a part of the data is not sufficient enough, more map space is acquired and new branches in the hierarchy are created. With this neural network model it is possible to view the data in various levels of granularity. The top layer of the hierarchy provides the coarsest view and with each step deeper into the hierarchy the respective subset of the data is presented in more detail. This architecture bears the advantage that hierarchical relations inherent in the data are mirrored and that the single maps are smaller and easier to survey.

We hope that the research presented in this thesis, located somewhere in between information retrieval, ontology engineering, linguistics, associative networks, usability and text mining, has contributed to the research and development towards more intuitive, less ambiguous and therefore more user-oriented search interfaces.

Bibliography

- Agirre, E., Ansa, O., Hovy, E., and Martínez, D. Enriching very large ontologies using the WWW. In Staab, S., Mädche, Nédellec, C., and Wiemer-Hastings, P., editors, *Proceedings of the 1st Workshop on Ontology Learning (OL 2000)*, Berlin, Germany, August 2000.
- Androutsopoulos, I., Ritchie, G. D., and Thanisch, P. Natural language interfaces to databases – An introduction. *Journal of Language Engineering*, 1(1):29–81, 1995.
- Berger, H., Dittenbach, M., and Merkl, D. Activation on the move. In *Proceedings of the 14th International Conference on Database and Expert Systems Applications (DEXA 2003)*, Prague, Czech Republic, September 3–5 2003a. accepted for publication.
- Berger, H., Dittenbach, M., and Merkl, D. Querying tourism information systems in natural language. In *Proceedings of the 2nd International Conference on Information System Technology and its Applications (ISTA 2003)*, Kharkiv, Ukraine, June 19–21 2003b. accepted for publication.
- Berger, H., Dittenbach, M., Merkl, D., and Winiwarter, W. Providing multilingual natural language access to tourism data. In Winiwarter, W., Bressan, St., and Ibrahim, I. K., editors, *Proceedings of the 3rd International Conference on Information Integration and Web-based Applications and Services (IIWAS 2001)*, pages 269–276, Linz, Austria, September 10–12 2001. Austrian Computer Society.
- Berners-Lee, T., Hendler, J., and Lassila, O. The semantic web. *Scientific American*, May 2001.

- Buschmann, F. *A System of Patterns. Pattern-Oriented Software Architecture*. John Wiley & Sons, 1996.
- Byrd, R. J. and Ravin, Y. Identifying and extracting relations in text. In Friedl, G. and Mayr, H. C., editors, *Proceedings of 4th International Conference on Applications of Natural Language to Information Systems (NLDB 1999*, Klagenfurt, Austria, 1999. Austrian Computer Society.
- Cavnar, W. B. and Trenkle, J. M. N-gram-based text categorization. In *Proceedings of the 3rd International Symposium on Document Analysis and Information Retrieval (SDAIR 1994)*, pages 161–175, Las Vegas, NV, 1994.
- Church, K. W. and Hanks, P. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29, March 1990.
- Cohen, P. R. and Kjeldsen, R. Information retrieval by constrained spreading activation in semantic networks. *Information Processing and Management*, 23(4):255–268, 1987.
- Crestani, F. Application of spreading activation techniques in information retrieval. *Artificial Intelligence Review*, 11(6):453–582, 1997.
- Crestani, F. and Lee, P. L. Searching the web by constrained spreading activation. *Information Processing and Management*, 36(4):585–605, 2000.
- Ding, Y. and Foo, S. Ontology research and development. Part 1: A review of ontology generation. *Journal of Information Science*, 28(2):123–136, 2002a.
- Ding, Y. and Foo, S. Ontology research and development. Part 2: A review of ontology mapping and evolving. *Journal of Information Science*, 28(2):375–388, 2002b.
- Dittenbach, M., Merkl, D., and Berger, H. Free speech for tourists. In *Proc. of the 13th Australasian Conference on Information Systems (ACIS 2002)*, Melbourne, Australia, December 4–6 2002a.
- Dittenbach, M., Merkl, D., and Berger, H. What customers really want from tourism information systems but never dared to ask. In *Proceedings of the*

- 5th International Conference on Electronic Commerce Research (ICECR-5)*, Montreal, Canada, October 23–27 2002b.
- Dittenbach, M., Merkl, D., and Berger, H. A natural language query interface for tourism information. In Frew, A. J., Hitz, M., and O'Connor, P., editors, *Proceedings of the 10th International Conference on Information Technologies in Tourism (ENTER 2003)*, pages 152–162, Helsinki, Finland, January 29–31 2003a. Springer-Verlag.
- Dittenbach, M., Merkl, D., and Berger, H. Using a connectionist approach for enhancing domain ontologies: Self-organizing word category maps revisited. In *Proceedings of the 5th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2003)*, Prague, Czech Republic, September 3–5 2003b. accepted for publication.
- Dittenbach, M., Merkl, D., and Rauber, A. The growing hierarchical self-organizing map. In Amari, S.-I., Giles, C. L., Gori, M., and Puri, V., editors, *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2000)*, volume 6, pages 15–19, Como, Italy, July 24–27 2000. IEEE Computer Society.
- Dittenbach, M., Rauber, A., and Merkl, D. Uncovering the hierarchical structure in data using the growing hierarchical self-organizing map. *Neurocomputing, Elsevier Science*, 48(1–4):199–216, October 2002c.
- Fellbaum, C., editor. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.
- Frakes, W., Prieto-Díaz, R., and Fox, C. DARE: Domain analysis and reuse environment. *Annals of Software Engineering, Kluwer*, 5:125–141, 1998.
- Grefenstette, G. Sextant: Exploring unexplored contexts for semantic extraction from syntactic analysis. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics (ACL 1992)*, pages 324–326, Newark, DE, June/July 1992.

- Gruber, T. R. A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199–220, 1993.
- Hahn, U. and Schnattinger, K. Knowledge mining from textual sources. In Golshani, F. and Makki, K., editors, *Proceedings of the 6th International Conference on Information and Knowledge Management (CIKM 1997)*, pages 83–90, Las Vegas, NV, November 1997. ACM Press.
- Hahn, U. and Schnattinger, K. Towards text knowledge engineering. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI 1998)*, pages 524–531, Madison, WI, July 1998.
- Hearst, M. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING 1992)*, Nantes, France, July 1992.
- Hearst, M. Automated discovery of wordnet relations. In Fellbaum, C., editor, *WordNet: An Electronic Lexical Database*. MIT Press, 1998.
- Honkela, T. *Self-Organizing Maps in Natural Language Processing*. PhD thesis, Helsinki University of Technology, Helsinki, Finland, 1997.
- Honkela, T., Pulkki, V., and Kohonen, T. Contextual relations of words in grimm tales, analyzed by self-organizing map. In Fogelman-Soulie, F. and Gallinari, P., editors, *Proceedings of the International Conference on Artificial Neural Networks (ICANN 1995)*, pages 3–7, Paris, France, 1995. EC2 et Cie.
- Horrocks, I. DAML+OIL: A description logic for the semantic web. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 25(1): 4–9, March 2002.
- Jansen, B. J., Spink, A., Bateman, J., and Saracevic, T. Real life information retrieval: A study of user queries on the web. *SIGIR Forum*, 32(1):5–17, 1998.
- Johnson, J. *GUI Bloopers. Don'ts and Do's for Software Developers and Web Designers*. Morgan Kaufmann, 2000.

- Kaski, S., Honkela, T., Lagus, K., and Kohonen, T. WEBSOM – Self-organizing maps of document collections. *Neurocomputing, Elsevier*, 21:101–117, November 1998.
- Kohonen, T. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43, 1982.
- Kohonen, T. *Self-Organization and Associative Memory*. Springer-Verlag, Berlin, 3rd edition, 1989.
- Kohonen, T. *Self-organizing maps*. Springer-Verlag, Berlin, 1995.
- Kohonen, T., Kaski, S., Lagus, K., Salojärvi, J., Honkela, J., Paatero, V., and Saarela, A. Self organization of a massive document collection. *IEEE Transactions on Neural Networks*, 11(3):574–585, May 2000.
- Levenshtein, V. I. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966.
- Lewis, D. D. and Spärck Jones, K. Natural language processing for information retrieval. *Communications of the ACM*, 39(1):92–101, January 1996.
- Mädche, A. and Staab, S. Discovering conceptual relations from text. In Horn, W., editor, *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI 2000)*, Berlin, Germany, August 2000a. IOS Press.
- Mädche, A. and Staab, S. Mining ontologies from text. In Dieng, R. and Corby, O., editors, *Proceedings of the 12th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2000)*, number 1937 in LNAI, pages 189–202, Juan-les-Pins, France, October 2000b. Springer-Verlag.
- Mädche, A. and Staab, S. Semi-automatic engineering of ontologies from text. In *Proceedings of the 12th International Conference on Software Engineering and Knowledge Engineering (SEKE 2000)*, Chicago, IL, July 2000c.
- Mädche, A. and Staab, S. Applying semantic web technologies to tourism information systems. In Wöber, K., Frew, A., and Hitz, M., editors, *Proceedings of the 9th International Conference on Information and Communication*

- Technologies in Tourism (ENTER 2002)*, Innsbruck, Austria, January 2002. Springer-Verlag.
- Manning, C. and Schütze, H. *Foundations of statistical natural language processing*. MIT Press, 2000.
- Miller, G. A., Leacock, C., Teng, R., and Bunker, R. T. A semantic concordance. In *Proceedings of the 3rd DARPA Workshop on Human Language Technology*, pages 303–308, Plainsboro, NJ, 1993. Morgan Kaufmann.
- Missikoff, M., Navigli, R., and Velardi, P. Integrated approach to web ontology learning and engineering. *IEEE Computer*, pages 60–63, November 2002a.
- Missikoff, M., Navigli, R., and Velardi, P. The usable ontology: An environment for building and assessing a domain ontology. In *Proceedings of the 1st International Semantic Web Conference (ISWC 2002)*, number 2342 in LNCS, pages 39–53, Chia, Italy, 2002b. Springer-Verlag.
- Mitra, M., Singhal, A., and Buckley, C. Improving automatic query expansion. In *Proceedings of the 21st International Conference on Research and Development in Information Retrieval (SIGIR 1998)*, pages 206–214, Melbourne, Australia, August 1998. ACM Press.
- Murray, K. M. E. *Caught in the Web of Words: James Murray and the Oxford English Dictionary*. Yale University Press, 2001.
- Nielsen, J. Noncommand user interfaces. *Communications of the ACM*, 36(4): 83–99, April 1993.
- Nielsen, J. *Designing Web Usability: The Practice of Simplicity*. New Riders Publishing, 2000.
- Nielsen, J. and Landauer, T. K. A mathematical model of the finding of usability problems. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 206–213, Amsterdam, The Netherlands, 1993.
- O'Brien, P. Dynamic travel itinerary management: The ubiquitous travel agent. In *Proc. of the 12th Australasian Conference on Information Systems*, Coffs Harbour, Australia, 2001.

- Philips, L. Hanging on the metaphone. *Computer Language Magazine*, 7(12), 1990.
- Porter, M. F. An algorithm for suffix stripping. *Program*, 14(3):130–137, July 1980.
- Preece, S. *A spreading activation model for Information Retrieval*. PhD thesis, University of Illinois, Urbana-Champaign, USA, 1981.
- Pribernik, M. Usability-Studie für ein Suchmaschineninterface. Master's thesis, Vienna University of Technology, Vienna, Austria, February 2003. (in German).
- Prieto-Díaz, R. A faceted approach to building ontologies. In Spaccapietra, S., March, S. T., and Kambayashi, Y., editors, *Proceedings of the 21st International Conference on Conceptual Modeling (ER 2002)*, number 2503 in LNCS, Tampere, Finland, 2002. Springer-Verlag.
- Pröll, B., Retschitzegger, W., Wagner, R. R., and Ebner, A. Beyond traditional tourism information systems – TIScover. *Information Technology and Tourism*, 1, 1998.
- Quillian, M. R. Semantic memory. In Minsky, M., editor, *Semantic Information Processing*, pages 227–270. MIT Press, 1968.
- Ritter, H. and Kohonen, T. Self-organizing semantic maps. *Biological Cybernetics*, 61(4):241–254, 1989.
- Salton, G. and Buckley, C. On the use of spreading activation methods in automatic information retrieval. In *Proceedings of the 11th International Conference on Research and Development in Information Retrieval (SIGIR 1988)*, pages 147–160, Grenoble, France, June 1988a.
- Salton, G. and Buckley, C. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988b.
- Salton, G. and McGill, M. J. *Introduction to modern information retrieval*. McGraw-Hill, 1983.

- Salton, G., Wang, A., and Yang, C. A vector space model for information retrieval. *Journal of the American Society for Information Science*, 18:613–620, 1975.
- Sanderson, M. and Croft, B. Deriving concept hierarchies from text. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1999)*, pages 206–213, Berkeley, CA, August 1999. ACM Press.
- Schuster, A. G. A delphi survey on electronic distribution channels for intermediaries in the tourism industry: The situation in german speaking countries. In *Proceedings of the International Conference on Information and Communication Technologies in Tourism (ENTER 1998)*, pages 224–234, Innsbruck, Austria, 1998. Springer-Verlag.
- Shneiderman, B., Byrd, D., and Croft, W. B. Sorting out searching. *Communications of the ACM*, 41(4):95–98, April 1998.
- Silverstein, C., Henzinger, M., Marais, H., and Moricz, M. Analysis of a very large AltaVista query log. Technical Report 1998-014, digital Systems Research Center, Palo Alto, CA, 1998.
- Staab, S., Braun, C., Düsterhöft, A., Heuer, A., Klettke, M., Neumann, G., Prager, B., Pretzel, J., Schnurr, H.-P., Struder, R., Uszkoreit, H., and Wrenger, B. GETESS—searching the web exploiting german texts. In Klusch, M., Shehory, O. M., and Weiss, G., editors, *Proceedings of the 3rd Workshop on Cooperative Information Agents (CIA 1999)*, number 1652 in LNCS, pages 113–124, Uppsala, Sweden, July/August 1999. Springer-Verlag.
- Velardi, P., Fabriani, P., and Missikoff, M. Using text processing techniques to automatically enrich a domain ontology. In *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS 2001)*, pages 270–284, Ogunquit, ME, 2001. ACM Press.
- Xu, F., Netter, K., and Stenzhorn, H. MIETTA – A framework for uniform and multilingual access to structured database and web information. In *Proceed-*

- ings of the 5th International Workshop on Information Retrieval with Asian Languages (IRAL 2000)*, Hong Kong, 2000.
- Zobel, J., Moffat, A., Wilkinson, R., and Sacks-Davis, R. Efficient retrieval of partial documents. *Information preocessing and Management*, 31(3):361–377, May/June 1995.

Curriculum Vitae

Persönliche Daten

Name: Michael Dittenbach
Anschrift: Schenkendorfgasse 14-16/2/12
A-1210 Wien
geboren am: 21.2.1976
Familienstand: ledig

Aktuelle Position

seit 2000 Forschungsassistent beim E-Commerce Competence Center – EC3
seit 1999 freier Mitarbeiter am Institut für Softwaretechnik
und interaktive Systeme, TU Wien

Universitätsausbildung

2001 – 2003 Doktoratsstudium Informatik an der TU Wien
1995 – 2001 Studium der Informatik an der TU Wien (mit Auszeichnung)

Schulbildung

1988 – 1994 Bundesrealgymnasium Feldkirch
1986 – 1988 Bundesrealgymnasium Laa/Thaya
1982 – 1986 Volksschule Mistelbach

Auszeichnungen

2001 Leistungsstipendium der Fakultät für technische Naturwissenschaften und Informatik (Diplomarbeit)

Publikationen

Papers in Journalen

Rauber, A., Merkl, D., and Dittenbach, M. The growing hierarchical self-organizing map: Exploratory analysis of high-dimensional data. *IEEE Transactions on Neural Networks, IEEE Press*, 13(6):1331–1341, November 2002.

Dittenbach, M., Rauber, A., and Merkl, D. Uncovering the hierarchical structure in data using the growing hierarchical self-organizing map. *Neurocomputing, Elsevier Science*, 48(1–4):199–216, October 2002.

Papers in referierten Konferenzbänden

Berger, H., Dittenbach, M., and Merkl, D. Activation on the move. In *Proceedings of the 14th International Conference on Database and Expert Systems Applications (DEXA 2003)*, Prague, Czech Republic, September 3–5 2003. accepted for publication.

Dittenbach, M., Merkl, D., and Berger, H. Using a connectionist approach for enhancing domain ontologies: Self-organizing word category maps revisited. In *Proceedings of the 5th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2003)*, Prague, Czech Republic, September 3–5 2003. accepted for publication.

Berger, H., Dittenbach, M., and Merkl, D. Querying tourism information systems in natural language. In *Proceedings of the 2nd International Conference on Information System Technology and its Applications (ISTA 2003)*, Kharkiv, Ukraine, June 19–21 2003. accepted for publication.

Dittenbach, M., Merkl, D., and Berger, H. A natural language query interface for tourism information. In Frew, A. J., Hitz, M., and O'Connor, P., editors,

Proceedings of the 10th International Conference on Information Technologies in Tourism (ENTER 2003), pages 152–162, Helsinki, Finland, January 29–31 2003. Springer-Verlag.

Dittenbach, M., Merkl, D., and Berger, H. Free speech for tourists. In Wenn, A., McGrath, M., and Burstein, F., editors, *Proceedings of the 13th Australasian Conference on Information Systems (ACIS 2002)*, volume 3, pages 1145–1154, Melbourne, Australia, December 4–6 2002.

Dittenbach, M., Merkl, D., and Rauber, A. Organizing and exploring high-dimensional data with the growing hierarchical self-organizing map. In Wang, L., Halgamuge, S., and Yao, X., editors, *Proceedings of the 1st International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2002)*, volume 2, pages 626–630, Singapore, November 18–22 2002.

Dittenbach, M., Merkl, D., and Berger, H. What customers really want from tourism information systems but never dared to ask. In *Proceedings of the 5th International Conference on Electronic Commerce Research (ICECR-5)*, Montreal, Canada, October 23–27 2002.

Schweighofer, E., Haneder, G., Rauber, A., and Dittenbach, M. Improvement of vector representations of legal documents with legal ontologies. In *Proceedings of the 5th International Conference on Business Information Systems (BIS 2002)*, Poznan, Poland, April 23–25 2002.

Berger, H., Dittenbach, M., Merkl, D., and Winiwarter, W. Providing multilingual natural language access to tourism data. In Winiwarter, W., Bressan, St., and Ibrahim, I. K., editors, *Proceedings of the 3rd Third International Conference on Information Integration and Web-based Applications and Services (IIWAS 2001)*, pages 269–276, Linz, Austria, September 10–12 2001. Austrian Computer Society.

Rauber, A., Dittenbach, M., and Merkl, D. Towards automatic content-based organization of multilingual digital libraries: An english, french, and german view of the russian information agency novosti news. In *Proceedings of the 3rd All-Russian Scientific Conference “Digital Libraries: Advanced Methods And Technologies, Digital Collections” (RCDL 2001)*, pages 88–95, Petrozavodsk, Russia,

September 11–13 2001.

Schweighofer, E., Rauber, A., and Dittenbach, M. Improving the quality of labels for self-organising maps using fine-tuning. In Tjoa, A. M. and Wagner, R. R., editors, *Proceedings of the 12th International Workshop on Database and Expert Systems Applications (DEXA 2001)*, pages 804–808, Munich, Germany, September 3–7 2001. IEEE Computer Society.

Dittenbach, M., Rauber, A., and Merkl, D. Business, culture, politics, and sports – how to find your way through a bulk of news? on content-based hierarchical structuring and organization of large document archives. In Mayr, H. C., Lazanski, G., Quirchmayr, G., and Vogel, P., editors, *Proceedings of the 12th International Conference on Database and Expert Systems Applications (DEXA 2001)*, number 2113 in LNCS, pages 200–210, Munich, Germany, 2001. Springer-Verlag.

Dittenbach, M., Merkl, D., and Rauber, A. Hierarchical clustering of document archives with the growing hierarchical self-organizing map. In Dorffner, G., Bischof, H., and Hornik, K., editors, *Proceedings of the International Conference on Artificial Neural Networks (ICANN 2001)*, number 2130 in LNCS, pages 500–505, Vienna, Austria, August 21–25 2001. Springer-Verlag.

Dittenbach, M., Rauber, A., and Merkl, D. Recent advances with the growing hierarchical self-organizing map. In Allison, N., Yin, H., Allison, L., and Slack, J., editors, *Advances in Self-Organising Maps*, pages 140–145, Lincoln, UK, June 13–15 2001. Springer-Verlag.

Schweighofer, E., Rauber, A., and Dittenbach, M. Automatic text representation, classification and labeling in european law. In *Proceedings of the 8th International Conference on Artificial Intelligence and Law (ICAAIL 2001)*, St. Louis, MO, USA, May 21–25 2001. ACM Press.

Rauber, A., Dittenbach, M., and Merkl, D. Automatically detecting and organizing documents into topic hierarchies: A neural-network based approach to bookshelf creation and arrangement. In Borbinha, J. and Baker, T., editors, *Proceedings of the 4th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2000)*, number 1923 in LNCS, pages 348–351, Lisbon, Portugal, September 18–20 2000. Springer-Verlag.

Dittenbach, M., Merkl, D., and Rauber, A. The growing hierarchical self-organizing map. In Amari, S.-I., Giles, C. L., Gori, M., and Puri, V., editors, *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2000)*, volume 6, pages 15–19, Como, Italy, July 24–27 2000. IEEE Computer Society.

Dittenbach, M., Merkl, D., and Rauber, A. Using growing hierarchical self-organizing maps for document classification. In *Proceedings of the 8th European Symposium on Artificial Neural Networks (ESANN 2000)*, pages 7–12, Bruges, Belgium, April 26–28 2000. D-Facto Publications.

Diplomarbeit

Dittenbach, M. The growing hierarchical self-organizing map: Uncovering hierarchical structure in data. Master's thesis, Vienna University of Technology, Vienna, Austria, December 2000.