Querying Tourism Information Systems in Natural Language

Helmut Berger[1], Michael Dittenbach[1], Dieter Merkl[1,2]

[1] E-Commerce Competence Center – EC3
Donau-City-Straße 1, A–1220 Wien, Austria
{helmut.berger, michael.dittenbach, dieter.merkl}@ec3.at

[2] Institut für Softwaretechnik
Technische Universität Wien
Favoritenstraße 9–11/188, A–1040 Wien, Austria
dieter@ifs.tuwien.ac.at

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 provides a means for finding what they really want. In this paper we present a query interface for an information system in the tourism domain exploiting the intuitiveness of natural language. Furthermore, we will describe the results and our insights from analyzing the natural language queries collected during a field trial in which the interface was publicly accessible. This analysis shows how users formulate queries when their imagination is not limited by conventional search interfaces with structured forms consisting of check boxes, radio buttons and special-purpose text fields.

1 Introduction

The development and availability of efficient and appropriate search functions are still a challenge in the field of database and information systems. Consider, for example, the context of tourism information systems where intuitive search functionality plays a crucial role for economic success. Querying an information system in natural language is especially appealing in the tourism domain because users usually have very different backgrounds regarding their computer literacy. Hardly any computer scientist or technically interested person has problems understanding the Boolean logic underlying conventional web search engines. Unfortunately, a growing majority of people using search engines has.

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 $-$ [JSBS98]. The latter two require that a query term must or must not be present in the searched pages. Although large web search engines like *Google*, *AltaVista* and of course thousands of smaller

site-specific search facilities have the same superficial appearance, they tend to interpret queries with subtle differences that can lead to searches not meeting the user's intention. Without 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 [SBC98].

To take away the fear of this rather technical way of searching for information, natural language should present a convenient form of interaction with such systems. 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 description of the needed information. Obviously, this should be expected to be of special importance in the tourism sector where people are often characterized by having rather unstructured imagination of their information need [O'B01].

Hence, we have developed a natural language interface for the largest Austrian web-based tourism platform *Tiscover* (*http://www.tiscover.com*) [PRWE98]. *Tiscover* 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 and a job market.

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 [BDMW01]. The language of the query is automatically detected and the result is presented in the respective language. 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 [Nie93]. In particular, our approach relies on the selection of query concepts followed by syntactic and semantic analysis of the portion of the natural language query where the concepts appear.

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. The time for the trial was chosen deliberately because close to vacation periods, as the Easter week in our case, the traffic at a web-based tourism information system is higher than during other times. The major objectives for the field trial were, first, to verify whether or not users accept natural language interaction. That means, we are interested if the users actually type natural language sentences to describe their information needs. Second, we hoped for a broad spectrum of natural language requests for tourism information, now that the users are no longer biased by available tick-boxes, radio buttons or selection lists. Finally, we were interested in the practical performance of the natural language interface given a real-world setting.

The remainder of the paper is structured as follows. In Section 2, we detail the natural language processing steps of our system. The field trial and its results are presented in Section 3. Finally, we present some conclusions in Section 4.

2 Natural Language Query Processing

This section contains a brief description of the various steps performed during natural language query processing of the demonstrator system. The software architecture is designed according to a flexible pipeline structure as shown on the right-hand side of Figure 1. Successively activated pipeline elements apply transformations on natural language queries that are posed via arbitrary client devices, such as, for instance, web browsers, PDAs or mobile phones. The generated output is based on a generic XML structure that is fine-tuned for the respective client device by using XSL transformations as shown on the left-hand side of the picture.

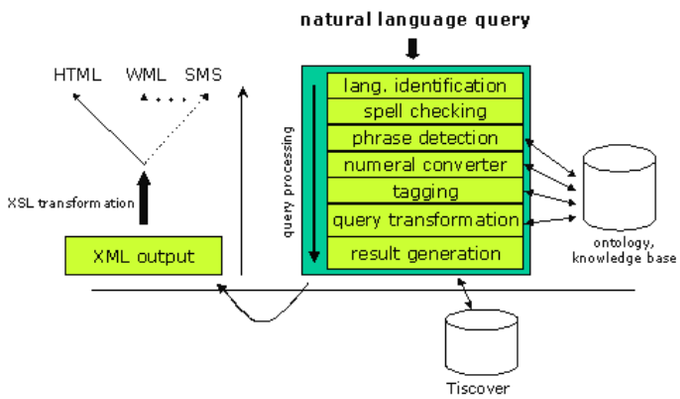

Figure 1: Software Architecture

2.1 Language Identification

To identify the language of a query, we use an *n–gram*-based text classification approach [CT94] where each language is represented by a class. An *n–gram* is an n-character slice of a longer character string. 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, the blank character is usually replaced by an underscore '␣' and is 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, i.e. *n–gram* frequency profiles, of the languages. We used various tourism-related texts, e.g. hotel descriptions 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.

In the English text, it can be seen that $\{$*the, and, of, in*$\}$ and the ending $\{$*ion*$\}$ are the most frequent *tri–grams*. Contrarily, in the German text, the most frequent *tri–grams* are endings like $\{$*en, er, ie, ch*$\}$ and *tri-grams* like $\{$*der, ich, ein*$\}$.

To determine the language of a query, the *n–gram* profile, $n = 1 \ldots 5$, of the query string is built as described above. The distance between two *n–gram* profiles is computed by a simple rank-order statistic. 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 sufficient accuracy. In such a case the user will be asked to rephrase her or his query.

2.2 Error Correction

To improve the retrieval performance, potential orthographic errors and misspellings have to be considered in our web-based interface. After identifying the language 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 previous queries.

The spell checker uses the metaphone algorithm [Phi90] 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. For every misspelled term of the query, a list of potentially correct words is returned. First, the misspelled word is mapped to its metaphone equivalent, then the words in the dictionary, whose metaphone translations have at most an edit distance [Lev66] 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 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 is counted. The words in the suggestion list for a misspelled query term are looked up in this repos-

itory 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 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 phonetically or orthographically 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. Another important issue in interpreting the natural language query is to detect terms consisting of multiple words. Proper names like *St. Anton am Arlberg* or substantives like *swimming pool* have to be treated as one element of the query. Regular expressions are used to identify such cases.

2.3 SQL Mapping

With the underlying relational database management system PostgreSQL, the natural language query has to be transformed into a SQL statement to retrieve the requested information. The knowledge base of the domain is split into three parts. First, we have an ontology specifying the concepts that are relevant in the application domain and describing linguistic relationships like synonymy. Second, a lightweight grammar describes how certain concepts may be modified by prepositions, adverbial or adjectival structures that are also specified in the ontology. Finally, the third part of the knowledge base describes parameterized SQL fragments that are used to build a single SQL statement representing the natural language query. 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 substantive is found in one of these tables, it is tagged with the respective table's name, such that *Tyrol* 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.

2.4 Design Considerations for the Web-Based Interface

In Figure 2, a screen shot of our interface is depicted. A simple and easy to use interface was our major design goal. The sample query "*I am looking for a double room in the center of Salzburg with indoor pool.*" is the only hint on the capabilities of the interface. The intention was to cover a broad range of accommodation requests and to find out what the user really wants. We wanted to avoid to narrow the user's imagination when formulating a query. Hence, we provided only short textual descriptions in both German and English,

Figure 2: Natural language query interface

a text area in which the user can enter the query and the submit button. This, admittedly, with the risk of disappointing the user when no or inappropriate results were found.

With 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 via clicking through the hierarchy of place names. Further criteria are the name of the accommodation, the chain it belongs to and, perhaps, a particular *theme*, e.g. family hotel, as well as several amenities the accommodation should provide. Note, this list of amenities is rather small compared to the complete information contained in the Tiscover database to keep the interface concise.

2.5 Data and Examples

For the examples described below, we demonstrate the application of our natural language interface to search for accommodations throughout Austria. In particular, we use a part of the database of the largest Austrian web-based tourism platform Tiscover [PRWE98], which provides access to information about 13,117 accommodations. These are described by a large number of properties including the respective numbers of various room types, different facilities and services provided in the accommodation, or even the type of food.

These 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, but also the number of inhabitants or the sea level. The federal states are the higher-level geographical units. For a part of the data, we integrated the geographical coordinates of the cities and towns to additionally provide information about the distance between places. Therefore, the system can be queried for accommodations close to a certain place as will be shown in the second example.

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 Insbruck aber nicht in Innsbruck selbst"*. This query, again including misspellings, shows the effect of different prepositions modifying a substantive. The query states that we are looking for a pension including breakfast close to Innsbruck but not in the city of Innsbruck. This is based on the assumption that *close to* means within approximately 15 kilometers. This range can be adapted by the user to her or his particular needs. In this particular case, the pensions in the result set will be those close to Innsbruck except those located in Innsbruck directly.

3 Field Trial

The field trial was carried out from March 15 to March 25, 2002. During this time our natural language interface was promoted on and linked from the main *Tiscover* page. 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.

Of those 1,425 unique queries, 1,213 (85.12%) were German, 120 (8.42%) were English and 92 (6.46%) were not identifiable, e.g. non-sentence queries like *"hotel salzburg"* that are possible in both languages or just nonsense like *mvbcvbn*. 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 used cars or sex among other topics that could not be answered by the system. Obviously, in any kind of publicly available service like this, not 100% 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 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 say that our system is usable regarding its response time.

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. [SHMM98] and [JSBS98] 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 amazing result of an average query length of 8.90 words for German queries, and of 6.53 for the English queries. 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"*. This approves our assumption that users accept and are willing to type more than just a few keywords to search for information. 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 1 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 1, the Tables 2(a) and 2(b) give an indication regarding the quality of the natural language query analysis. In particular, Table 2(a) provides the numbers of identified concepts per query, whereas Table 2(b) that of not identified concepts. Again, the figures given in Table 2(b) were derived by manual inspection. We shall note that most of the concepts not identified, originated from queries falling into the categories of region names, pricing information, room availability and arrival and departure dates. These informations were not contained in the part of the database used for 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(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 two times which relates to the occurrence of three concepts per query (cf. Table 1). The occurrence statistic includes all implicitly used modifiers *and* as well as those explicitly defined. 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.

	query language		
concepts	**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 1: Total concepts per query (counted by manual inspection of the queries)

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. The modifier *or* is used far less than *and*, as shown in Table 3(b). *Or* is mostly used to provide 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(c)). The modifier *not* occurs in only 19 German 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. For instance, users wanted to avoid places where pets are allowed as well as quiet accommodations without children. Another common use of *not* was 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 Innbruck and not in Zillertal*."

Table 3(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 1425 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"*. Furthermore, we can see, that only a very small number of queries consists of concepts combined with *or* (103 out of 1,425), and only 22 queries contain the modifier *not*.

We can say that the sentence complexity, i.e. the frequency of concept combination, is

	query language		
concepts	**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

	query language		
concepts	**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 2: Concepts that have been identified or not identified by the natural language processing module of our interface

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 require a more sophisticated sentence evaluation process. For instance, if 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"*. In contrast to the assumption that the default operator of combining concepts is *and*, the modifier *or* must be used to combine the geographical concepts in this sample query.

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. Nevertheless, we found out that regions or local attractions have been addressed in a large number of queries and thus, have to be integrated in such systems. We also noticed that users' queries contained vague or highly subjective criteria like *romantic*, *wellness*, *cheap* or *within walking distance to*. These concepts are difficult to model in the knowledge base of information systems and pose a challenge for the future.

4 Conclusions

Web-based tourism information systems are faced with a highly inhomogeneous mix of potential consumers. The reason, obviously, has to do with the tourism domain because, pragmatically speaking, almost everybody is a tourist sometimes. Hence, people with

	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: Usage of modifiers *and*, *or*, *not* and *near*

highly different backgrounds regarding their language, their preciseness in the description of information needs, or their computer literacy, to name but a few, are the customers of a web-based tourism information system. To cope with this situation we designed a multilingual natural language query interface for *Tiscover*, the largest Austrian tourism platform. By way of this interface, the user can search for more than 13,000 accommodations in about 2,000 towns throughout Austria.

In this paper we have discussed the findings after a 10 day field test where we collected about 1,400 queries, most of which were written in German language. Most importantly, the users willingly type natural language queries to express their information needs. This observation is approved by a comparison with web-search engines, where the average number of words per query is substantially smaller than with our tourism information system. Second, the complexity of these queries is higher than with standard web-search engines. We have shown the distribution of various modifier combinations extracted from the queries. Third, our expectation that shallow language processing is sufficient given a limited application domain is 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. Fourth, by way of this field trial allowing natural language descriptions of information needs as opposed to the strictly limited variability of tabular-based information entry, we have got an impression of what the customers actually look for. Among the

most important things we just mention geographic information as when you describe the location of your preferred accommodation relative to some geographical landmarks. This gives enough room for interesting future research to improve the knowledge base of the system and thus to better serve the customers.

References

[BDMW01] H. Berger, M. Dittenbach, D. Merkl, and W. Winiwarter. Providing Multilingual Natural Language Access to Tourism Data. In *Proc. of the 3rd International Conference on Information Integration and Web-based Applications and Services (IIWAS 2001)*, Linz, Austria, September 10-12 2001.

[CT94] W. B. Cavnar and J. M. Trenkle. N-gram-based text categorization. In *Proc. of the 3rd Int'l Symposium on Document Analysis and Information Retrieval (SDAIR'94)*, pages 161–175, Las Vegas, NV, 1994.

[JSBS98] B. J. Jansen, A. Spink, J. Bateman, and T. Saracevic. Real life information retrieval: A study of user queries on the Web. *SIGIR Forum*, 32(1):5–17, 1998.

[Lev66] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707 – 710, 1966.

[Nie93] J. Nielsen. Noncommand User Interfaces. *Communications of the ACM*, 36(4):83–99, April 1993.

[O'B01] P. O'Brien. Dynamic Travel Itinerary Management: The Ubiquitous Travel Agent. In *Proc. of the 12th Australasian Conference on Information Systems*, Coffs Harbour, Australia, 2001.

[Phi90] L. Philips. Hanging on the Metaphone. *Computer Language Magazine*, 7(12), December 1990.

[PRWE98] B. Pröll, W. Retschitzegger, R. R. Wagner, and A. Ebner. Beyond traditional tourism information systems – TIScover. *Information Technology and Tourism*, 1, 1998.

[SBC98] B. Shneiderman, D. Byrd, and W. B. Croft. Sorting out Searching. *Communications of the ACM*, 41(4):95 – 98, April 1998.

[SHMM98] C. Silverstein, M. Henzinger, H. Marais, and M. Moricz. Analysis of a Very Large AltaVista Query Log. Technical Report 1998-014, digital Systems Research Center, Palo Alto, CA, 1998.