Conceptual Views for Entity-Centric Search: Turning Data into Meaningful Concepts

Joachim Selke, Silviu Homoceanu, and Wolf-Tilo Balke

Institut für Informationssysteme
Technische Universität Braunschweig
{selke, silviu, balke}@ifis.cs.tu-bs.de

Abstract: The storage, management, and retrieval of entity data has always been among the core applications of database systems. However, since nowadays many people access entity collections over the Web (e.g., when searching for products, people, or events), there is a growing need for integrating unconventional types of data into these systems, most notably entity descriptions in unstructured textual form. Prime examples are product reviews, user ratings, tags, and images. While the storage of this data is well-supported by modern database technology, the means for querying it in semantically meaningful ways remain very limited. Consequently, in entity-centric search suffers from a growing semantic gap between the users' intended queries and the database's schema. In this paper, we introduce the notion of conceptual views, an innovative extension of traditional database views, which aim to uncover those query-relevant concepts that are primarily reflected by unstructured data. We focus on concepts that are vague in nature and cannot be easily extracted by existing technology (e.g., *business phone* and *romantic movie*). After discussing different types of concepts and conceptual queries, we present two case studies, which illustrate how meaningful conceptual information can automatically be extracted from existing data, thus enabling the effective handling of vague real-world query concepts.

1 Introduction

With the widespread use of the Web as primary information source, entity-centric search has become a common task for many people, with product search arguably being most prominent. In this context, typical entity types are mobile phones, movies, and books, but could of course also be people, news items or events. Although the handling of entity data traditionally falls into the domain of database systems [Che76], database methodology alone is becoming less and less adequate to master this task. Entities are no longer characterized by structured data alone but to a large extent also by semi-structured and unstructured information. For example, besides technical specifications, a typical e-shopping website features detailed textual product descriptions, expert reviews, and a large variety of user-generated content such as ratings, tags, and opinions. While modern database systems offer extensive technical capabilities for storing a large variety of data types (e.g., text documents, XML documents, and even multimedia content), the means for querying this data remain very limited [Wei07].

Therefore, recent research has been more and more focused on integrating information retrieval capabilities into database systems, in particular by structuring unstructured data for use by structured queries [WKRS09, CRS+07, MS06]. Most of this ongoing research focuses on extracting *precise* facts from textual data using methods from the area of information extraction [Moe06]. While preliminary results are promising, still many problems remain to be solved.

But to make things even more complicated, an analysis about product search we performed on the AOL search query log revealed the following: When searching for mobile phones, people very often include *vague* concepts (e.g. *business phone*, *portability*, or *for kids*) in their queries, about as often as they refer to precise technical product details (e.g. *weight*, *display diagonal size*, or *talk time*). Figure 1 illustrates the different types and respective frequencies of queries related to mobile phones we identified in the AOL search query log.

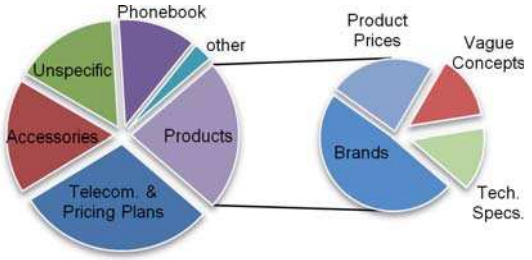

Figure 1: Different types of queries related to mobile phones in the AOL search query log

We also investigated what information about mobile phones is provided by current online shops, price comparison services, and media news sites such as CNET.com. We found that while almost all sites collect and allow searching for a broad range of technical specifications, the coverage of vague product features is fragmentary at best. Typically, information about concepts such as *business phone* is only available through manually-created top ten lists, which have been published as ordinary pages. User-defined top lists are particular popular and even market leaders such as Amazon.com have recognized them as important means for providing conceptual information about products. None of the web sites we studied offered structured search functionality for vague concepts.

Our analyses indicate that there exists a significant mismatch between the users' intended queries and the database's schema. A very similar issue has been identified in the field of multimedia databases, where it is usually referred to as *semantic gap* [HLES06]. As argued above, in case of entity-centric search, the semantic gap mainly exists because users' information needs often are based on natural but typically vague concepts, which information providers usually do not model explicitly in their databases. However, previous studies also indicate that information about many query-relevant concepts is already contained in those parts of entity databases that are currently not used for answering the users' queries [SB10, KT09]. This mainly refers to unstructured information (e.g. textual product reviews), but may also include structured information (e.g. user ratings, which currently are mostly used to compute average product ratings, thus ignoring the users' hidden preferential structures).

In this paper, we present our approach to bridging the sematic gap in entity-centric search. As a key element, we introduce the notion of *conceptual views*, an innovative extension of traditional database views, which aims at systematically providing the means for making implicit conceptual information explicit to database applications. In particular, using case studies from the domains of mobile phones and movies, we demonstrate how conceptual views can be constructed automatically from the existing data and provide a rough classification of typical query types and matching extraction techniques.

The rest of the paper is structured as follows: First, we introduce and discuss the notion of conceptual views as well their use in modern database systems in Section 2. In the following sections we present our case studies. We continue by reviewing related work in Section 5, and conclude by highlighting some important findings from cognitive psychology in Section 6, which are strongly aligned to our approach and will guide our further work. We conclude by summarizing the results of our current research efforts and discuss open problems in Section 7.

2 Designing a View Mechanism for Answering Conceptual Queries

In this section we will discuss the basic mechanism for answering conceptual queries. Since database entities usually represent entities of the real world, the key idea is to understand concepts as special database attributes in a structured form. The attribute's value for every entity is obviously determined by the "degree of applicability" of the concept, which can be defined in a variety of ways as we will discuss later. In any case, this specific way of mediating between some user's or application's information need and the logical design of a database or information system is generally provided by the view mechanism. In the following we will briefly discuss how concepts are prepared for retrieval purposes using conceptual views.

2.1 Detecting Concepts in Queries

As we have seen, many queries address a rather conceptual understanding of database items (or entities) and therefore cannot be answered directly. But, how can such queries be handled in an effective yet easy-to-use way? The first step of course is to detect some new concept within queries, and thus a new information need. Whereas this is easy to do in SQL-style declarative query languages, where a mapping of previously unknown attributes to actually existing attributes in the underlying source(s) can be derived (see for instance the work on malleable schemas [ZGBN07]), the recognition of new concepts in simple keyword queries is somewhat harder. Of course, it is impossible to mine all individual concepts from a vast number of query terms put together in millions of queries regarding some topic. But preparing the underlying database to answer at least the most often occurring keyword queries, and thus providing for predominant groups of users, can be a strategic advantage, especially for e-commerce portals.

Following our running example, we therefore determined typical characteristics of predominant conceptual queries, i.e., what concepts with respect to mobile phones are there and how often do they occur? To answer this question we inspected the online advertising platform Google AdWords[1] and related the monthly number of general queries on our example domain of mobile phones (and all common spelling variants like "mobile phones" and "cell phones") to the number of queries reflecting often used concept terms as derived from the AOL query log (again expanded with common spelling variants, but not related queries, e.g., the query "business phone" was not expanded by related terms like "calendar" or "organizer"). Since Google AdWords only allows for monthly averages, the results shown in Table 1 can only be seen as an intuition about the demand for individual concepts. For the month of September 2010, Google AdWords reported a total of 22,110,560 general purpose queries on mobile phones. Considering the top-5 concepts from the AOL query log we find that the relative monthly amount of queries ranges between 0.2% and 1.4%. Still, the result clearly shows that individual concepts will occur in significant numbers of queries and thus are easily detectable in a query log. Thus, periodically inspecting query logs for often co-occurring combinations of keywords can be expected to lead to the detection of currently relevant query concepts.

	Frequency	
Concept	Absolute	Relative
Cheap phone	313,400	1.4%
Business phone	87,520	0.4%
TV cell phone	62,700	0.3%
Music cell phone	49,300	0.2%
Cell phone for kids	33,260	0.2%

Table 1: Relative frequencies of concepts related to queries about mobile phones.

2.2 Building Conceptual Views

Since long relational databases have provided a mechanism for supporting queries that do not directly address attributes predefined in the logical design: views. Whereas views were often understood as a security feature regulating access to database tables and even providing some statistical data security by pre-aggregating several attributes, after the introduction of materialized views the performance implications became paramount. Especially for expensive aggregations a pre-computation and materialization of view attributes is essential. This perfectly fits to the complex nature of concepts and their problematic deduction from entity information.

The basic idea for building conceptual views is to derive each entity's score with respect to some concept and offer it to query processing engines under the name of the concept (basically the used query term, for complex mappings of different queries to abstract concepts please refer to the large body of work on schema mapping). The score assigned

[1] http://adwords.google.com

to each entity with respect to a concept can be interpreted in several ways. Of course the easiest way is to employ expert judgments simply rating all items. However, relying on editors (like e.g., the allmusic portal[2]) is an expensive and cumbersome method, which can only be employed on small collections, where trust in the scoring process is vital. On the other hand, given the variety of information about entities collected in today's databases and information systems, such as Amazon.com's shopping portal or the IMDb movie database[3], conceptual information can be derived with adequate extractors. Before discussing these extractors, we will provide a brief overview of how concept scores are typically interpreted:

1. *Possibility that an entity represents a concept given structured information.* A good example is the concept *portability* for mobile phones or laptops. Here, the degree of membership (score) can be assumed to be a simple weighted aggregation of the weight and size attributes that will be part of the structured technical specifications.

2. *Possibility that an entity represents a concept, also considering unstructured information.* This is essential for concepts that cannot directly be derived from structured data, such as the concept of *business phone* in the domain of mobile phones. Usually, such concepts are to some degree based on opinions or user expectations, which are just supported by structured information.

3. *Probability that an arbitrary user would rate an entity as matching the concept.* The way of scoring is often modeled as degree of belief. A typical example is the notion of *beauty,* which again is sometimes supported by structured information, but in the end relies on (probably differing) opinions.

4. *Average user judgments.* User judgments already form a significant type of data in most information portals. Users are invited to express a personal opinion, and the scoring of each entity can then be derived by suitable aggregations of such ratings.

Of course, the major feat for the successful generation of conceptual views lies in the respective extraction algorithms for the concept scoring. Indeed, for the four interpretations above there are some typical extraction techniques (which we will described in more detail and tied to conceptual query types in a later section). Generally speaking all extraction algorithms have to rely on a set of sample entities exhibiting the concept in question. Of course, such typical entities can always be provided by users in a query-by-example fashion (e.g., the *iPhone* as a typical smart phone or *Hugh Grant movies* as typical romantic comedies), but also a simple keyword search in unstructured data associated with some entities in our experiments proved to yield sufficiently accurate examples. The basic structure of extraction algorithms for the above interpretations can be roughly classified as follows:

1. *Extractors working only on structured data are usually of a purely statistical nature trying to find correlations between different attributes for the sample entities.* Generally attributes allowing for a good clustering of the sample, while showing a

[2] `http://www.allmusic.com`
[3] `http://www.imdb.com`

different overall distribution, can be expected to have some meaning with respect to the query concept. Typical algorithms like association rule mining for categorical data and Bayesian classification, or clustering algorithms for numerical data are well understood and already often used [WKQ+08].

2. *Extractors integrating structured and unstructured data usually involve some natural language processing techniques and are generally a mixture between statistical methods and techniques from information retrieval.* Since they form a currently very active and complex research topic, we will revisit a typical representative of these algorithms as use case in the Section 3 and discuss the result quality.

3. *Extractors for degrees of belief are usually relying on user relevance feedback in some form and thus tend to be interactive algorithms.* Generally speaking, all methods need some time to derive meaningful scorings, but following the wisdom-of-the-crowds principle [Sur04] eventually result in scorings of good quality. Due to their unobtrusiveness, recently the combined evaluation of query logs together with the results users clicked on has been a prime candidate for establishing degree of belief values [BHJ+10].

4. *Extractors for exploiting rating information often use an abstract semantic space for entity representations and then derive scorings from evaluating similarities in this space.* A typical representative of such algorithms is Latent Semantic Analysis [DDF+90]. Here, the feature space is rotated into the direction of prominent eigenvectors representing predominant topics that can be used to distinguish between sets of entities. We will also revisit this kind of extraction as a use case.

Having built a new attribute in the conceptual view for each relevant target concept using an adequate extractor depending on the type of concept, the view can be queried. Obviously, the extraction algorithms tend to be rather complex and time-consuming such that a materialized version of the view has to be maintained. This immediately raises questions about possibilities to update such views, which in turn reflects on the extraction algorithms used. However a detailed discussion of this problem is beyond the scope of this paper.

2.3 Answering Conceptual Queries

For answering conceptual queries by means of conceptual views, we must be aware of the dichotomy between precise concepts (which usually are already modeled explicitly in the database) and vague concepts (which are provided by conceptual views). The former typically will be used to specify hard logical constraints within the query (e.g., retrieve only Nokia phones or phones being cheaper than 300 Euros), while the latter are the primary focus of queries involving vague concepts (e.g., retrieve all *business phones* that are *mid-priced* and *iPhone-like*). Since those queries cannot be formulated and processed in a semantically meaningful manner using precise query languages such as SQL, a different approach is needed.

Since vague concepts almost always go hand-in-hand with the notion of degree of membership, a purely set-oriented retrieval approach seems inappropriate for conceptual attributes; ranking-based methods are more appropriate here. Fortunately, there already exists a large body of research dealing with exactly this type of query formulation and processing. As soon as all relevant concepts have been made explicit in structured form, a whole bunch of existing methods for supporting vagueness in queries and data can be applied, thus enabling a concept-enriched and more intuitive entity-centric search. Notable approaches are fuzzy databases [GUP06], the VAGUE system [Mot88], top-k retrieval [IBS08] and typicality queries [HPF+09], just to name a few. To integrate both types of query concepts, preference-based database retrieval [Kie02, Cho03] offers a large variety of options.

Figure 2 summarizes our vision of conceptual views and embeds this notion into the context of existing database systems. Conceptual views can actively and automatically be maintained by analyzing user query logs. As soon as relevant query concepts have been identified that currently cannot be handled using structured data, a suitable extractor is chosen to extend the conceptual view accordingly. Thus, conceptual views provide a systematic and unifying perspective on all available data, regardless of its type. Since all relevant concepts have been made explicit in structured form, existing methods for concept-based query processing can be applied to satisfy a broad range of information needs, which could not be handled using the previously existing structured data alone.

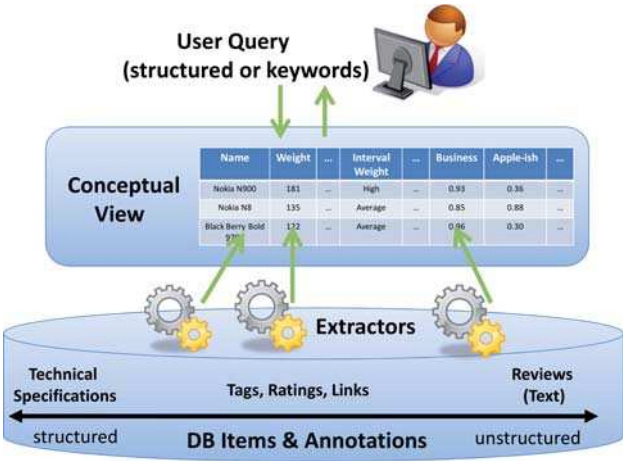

Figure 2: Conceptual views within a database system.

3 Case Study: Mobile Phones

Our first case study concerns the domain of mobile phones, which already has been discussed briefly. Here, in addition to providing a number of structured technical specifications for each phone, a typical product database also contains a textual description of the phone along with a (possibly large) collection of detailed reviews written by expert users or

journalists. The relevant query concepts vary from those that are primarily defined in terms of structured data (e.g., *portability*) to those that have almost no connection to the technical specifications (e.g. *well-designed*). In between, there are concepts being defined by both types of data (e.g. *business phone*). In this case study, we present a method that jointly analyzes both structured and textual data to extract a meaningful score value for some target concept, which in the following will be *business phone*.

In order to store the degree of membership for each entity towards the target concept, one first needs to build a model-based representation of this concept. Such a model comprises a feature collection together with the corresponding strengths, which we will refer to as *model vocabulary (MV)* in the following. Moreover, we will need an *entity representation function*, used for calculating the degree of membership of each entity in the database towards the target concept.

3.1 Method: Feature Analysis

The approach to be presented in the following is based on *conceptual features*, which are either "real" product features extracted from the technical specifications or nouns (and noun phrases) contained in some textual description or review of a product [LHC05]. We first extract all conceptual features from the available structured and unstructured data, and then try to find meaningful relationships between them (e.g., business phones tend to have advanced calendar functionalities). Our algorithm is based on a self-supervised learning technique that uses two types of training data for each product: a concept-related product review provided by some professional editor and the product's technical specifications in structured form.

The method works as follows: We start by automatically splitting the training data (and thus also the entities) with classical information retrieval techniques (such as keyword search) into the explicitly concept-relevant data, further referred to as R, and the remaining data (for which the relevance towards the concept is unknown), further referred to as U. The sets R and U are disjoint. Of course, U will typically not only contain irrelevant entities, but also some entities for which the concept is only visible in terms of related features. In order to ensure a model of high quality, we have to split the training set by performing the concept keyword search both in the structured and unstructured data of each entity. We also have trained the model by using only editor product reviews which are extensive by nature, explicitly covering a broad spectrum of features and concepts.

Adapting procedures from document classification, we extract those product features that tend to discriminate entities in the set R from those in U (assuming that most entities in U will be irrelevant to the target concept). For this purpose, we assign a numerical strength to each feature, which measure the feature's importance with respect to the given concept. We consider only the strongest ones for our MV. The strength of a feature f_i is defined as follows:

$$\text{strength}(f_i) = \frac{n_R(f_i) - \min_j\big(n_R(f_j)\big)}{\max_j\big(n_R(f_j)\big) - \min_j\big(n_R(f_j)\big)} - \frac{n_U(f_i) - \min_j\big(n_U(f_j)\big)}{\max_j\big(n_U(f_j)\big) - \min_j\big(n_U(f_j)\big)},$$

where $n_R(f_i)$ is the number of entities in R containing the feature f_i. The first summand calculates the normalized feature strength relative to entities belonging to R, while the second summand calculates the normalized strength relative to U.

Our method considers only those features having a reasonably high strength, namely three times the standard deviation above the average strength found in the entire population. An example of the resulting MV for the concept *business phone* is shown in Figure 3. The technical features and specification labels are extracted from the structured data, along with part of the corresponding values. Other features are extracted from unstructured data. Together with their corresponding strengths calculated with the above formula, all these features describe our target concept.

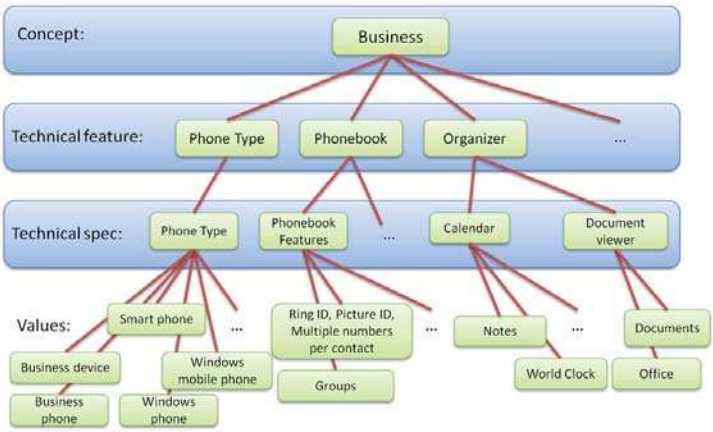

Figure 3: Model vocabulary for concept *business phone*.

In order to be able to evaluate the degree of membership of an entity E towards the target concept, we have used the entity representation function

$$\sum_{f_i \in \mathrm{MV} \cap E} \mathrm{strength}(f_i),$$

which states that an entity is as relevant to the concept as the sum the strengths of those features belonging to both the model as well as the entity.

As an example, consider that we want to compute the degree of membership towards the *business phone* concept for an entity which is described by the following text: "Powerful, but incredibly cumbersome. Pros: Has Microsoft Office, full featured calendar, support for multiple email accounts, internet connectivity, Wi-Fi, and a good battery life. Cons: It's incredibly cumbersome and has a cluttered Windows 3.1 UI." After evaluating the strength of this text only, by using our weighting function on the features from the text which also belong to the model (see Table 2), the entity gains a strength of 1.115, increasing its relevance towards the concept. Knowing that the total strength of the model is 57.082, the previous text provides for an increase in relevance by about 2%.

Feature	Strength
calendar	0.346
Wi-Fi	0.260
Windows	0.255
Office	0.155
battery life	0.099

Table 2: Features and associated strengths for concept *business phone*.

3.2 Experimental Setup

To evaluate our approach, we collected a training data set from PhoneArena.com, a major customer portal in the area of mobile phones. Our data set consists of expert reviews and technical specifications for 500 different phones. This data set has been used to build a model for the concept *business phone* as described above. Of course, this concept is not explicitly mentioned in PhoneArena.com's structured data.

To test the predictive power of this model, we downloaded 200 user-provided reviews of the latest mobile phones from CNET.com. These reviews then have been manually labeled by experienced mobile phone users, either as being relevant or not relevant with respect to the concept *business phone*. We then compared the entity scores derived by our model to these manually created assessments. As evaluation metric, we used a precision–recall curve, which the dominant methodology for evaluating information retrieval systems. We compared our approach to two different baselines: document ranking by TF–IDF and Latent Semantic Indexing.

3.3 Results

The results of our evaluation are displayed in Figure 4. As we can see, our method is close to the two baseline approaches for high-precision scenarios, and outperforms them in high-recall settings. This clearly shows that integrating the available structured information into the retrieval process allows us to create a much more accurate model of the target concept than it is possible using previous methods. However, there is still room for improvement, which will be our primary goal in future work. To conclude, these results indicate that our method is well-suited for the task of constructing conceptual views from structured data and textual information.

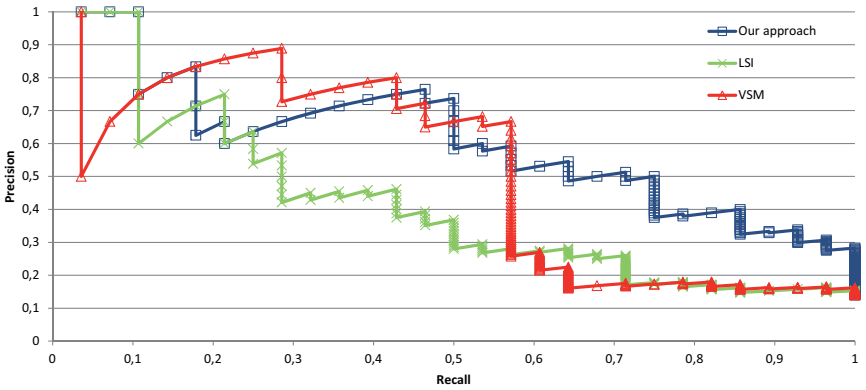

Figure 4: Precision–recall graph for the concept *business phone*.

4 Case Study: Movies

In our second case study, we are considering a database of movies. There are many popular examples on the web, e.g. IMDb, Netflix[4], and Rotten Tomatoes[5]. Typically, those services offer their customers a broad spectrum of structured information about each movie, such as its title, release date, director, cast, genre, running time, and a short plot description. Also, they often allow people to contribute by providing their personal opinions in form of textual user reviews or ratings on a fixed numerical scale (e.g. one to five stars). The former usually are published on the respective service's web site, the latter are used to compute a mean rating (which then is published) or to generate personalized movie recommendations for each user.

In contrast to our first case study, taste in movies is extremely complex and individual, and can only approximated very coarsely by the usual ways of cataloging movies. Therefore, the structured data available often is of very limited use for finding movies matching a user's current mood or taste. To counter this problem, some providers have started to manually classify each movie along a wide range of semantically more meaningful concepts (e.g., *complexity* or *character depth*). This method is sometimes referred to as the Movie Genome approach and adopted by Clerkdogs[6] and Jinni[7], amongst others. However, since movie databases tend to be very large (ranging from around 10,000 movies in smaller systems to almost 1.7 million in larger ones such as IMDb), manually evaluating each movie with respect to many different vaguely defined concepts seems to be a challenging, if not impossible, task.

In the following, we demonstrate how such movie concepts can be made explicit by a conceptual view that extracts all necessary conceptual information from a large number of user ratings (where each user just assigns a number to each rated movie but does not

[4] http://www.netflix.com
[5] http://www.rottentomatoes.com
[6] http://www.clerkdogs.com
[7] http://www.jinni.com

provide any additional details). Each concept included in the conceptual view is defined by providing a small number of exemplary movie–score pairs. In a sense, this setting is similar to the machine learning task of semi-supervised learning [ZG09]. The approach to be presented in the following is based on but significantly extends previous work, which has been published recently [SB10].

4.1 Method: Semantic Spaces

In contrast to the feature-based approach presented in the previous section, we now purely rely on similarities and differences in users' perception of movies, which are modeled by embedding the movies into an artificial high-dimensional coordinate space ("semantic space"). The individual dimensions of this space do not necessarily correspond to conceptual features of movies as recognized by humans.

We start by giving a formal definition of the problem to be solved. In the following, we use the variable m to identify movies, whereas u denote users. We are given a set of n_{M} movies and n_{U} users, where each user may rate each movie on some predefined numerical scale (e.g., the set of integers from one to ten). The provided ratings thus can be represented as a rating matrix $R = (r_{m,u}) \in \{\mathbb{R} \cup \emptyset\}^{n_{\mathrm{M}} \times n_{\mathrm{U}}}$, where each entry corresponds to a possible rating and $r_{m,u} = \emptyset$ indicates that movie m has not been rated (yet) by user u. Typically, the total number of ratings provided is very small compared to the number of possible ratings $I \cdot U$, often lying in the range of 1–2%. We are also given a small set of n movie–score pairs $C = \big\{(m_1, s_1), \ldots, (m_n, s_n)\big\}$, which correspond to a human evaluation of the target concept for a random selection of movies. Our task is to estimate the score of all remaining movies.

In line with methodology that recently has been successfully applied in the area of collaborative recommender systems [KBV09], we first perform a factorization of the rating matrix R into two smaller matrices $A = (a_{m,i}) \in \mathbb{R}^{n_{\mathrm{M}} \times d}$ and $B = (b_{i,u}) \in \mathbb{R}^{d \times m_{\mathrm{U}}}$ such that the product $A \cdot B$ closely approximates R on all entries that are different from $\emptyset$; the constant d is chosen in advance and typically ranges between 50 and 200. The idea is similar to the Latent Semantic Indexing (LSI) approach used in information retrieval [DDF+90]: Reduce the n_{U}-dimensional movie space (each movie is described by a vector of user ratings) and the n_{M}-dimensional user space (each user is described by a vector of movie ratings) to its most significant d-dimensional subspace.

Formally, the matrices A and B can be defined as the solution of the following optimization problem:

$$\min_{A,B} \quad \mathrm{SSE}\big(R, A \cdot B\big) + \lambda \sum_{(m,u)\,|\,r_{m,u} \in \mathbb{R}} \; \sum_{i=1}^{d} \big(a_{m,i}^2 + b_{i,u}^2\big)\,,$$

where the SSE (sum of squared errors) function is defined as

$$\mathrm{SSE}\big(R, \hat{R}\big) = \sum_{(m,u)\,|\,r_{m,u} \in \mathbb{R}} \big(r_{m,u} - \hat{r}_{m,u}\big)^2$$

and $\lambda \geq 0$ is a regularization constant used to avoid overfitting. Besides this specific formulation of the matrix decomposition problem, many other versions have been proposed [KBV09]. However, the one just presented is the most fundamental.

To arrive at a canonicalized solution exhibiting some desirable properties (orthogonal axes, axes weighted by importance), we apply a singular value decomposition to represent the product $A \cdot B$ in the form $U \cdot S \cdot V$, where $U \in \mathbb{R}^{n_M \times d}$ is a column-orthonormal matrix, $S \in \mathbb{R}^{d \times d}$ is a diagonal matrix, and $V \in \mathbb{R}^{d \times n_U}$ is a row-orthonormal matrix. By reordering rows and columns, S can be chosen such that its diagonal elements (the singular values) are ordered by increasing magnitude. To arrive at a data representation that distinguishes only between movies and users, we integrate the weight matrix S to equal parts into U and V. Therefore, we define $U' = US^{\frac{1}{2}}$ and $V' = S^{\frac{1}{2}}V$ to be our final coordinate representation.

Now, each row of U' corresponds to a movie, and each column of V' corresponds to a user, both being represented as points in some d-dimensional space. This representation of movies provides the basis for learning the target concept from the examples in the set C. Taken together, the n_M movie points can be interpreted as a *semantic space*, which captures the fundamental properties of each movie [SB10]. Now, the target concept can be learned using specialized algorithms from the fields of statistics and machine learning. For this case study, we decided to use kernel-based support vector regression [SS04].

4.2 Experimental Setup

This case study uses the MovieLens 10M data set [8], which consists of about 10 million ratings collected by the online movie recommender service MovieLens[9]. After post-processing the original data (removing one non-existing movie and merging several duplicate movie entries), our new data set consists of 9,999,960 ratings of 10,674 movies provided by 69,878 users (thus, about 1.3% of all possible ratings have been observed). The ratings use a 10-point scale from 0.5 (worst) to 5 (best). Each user contributed at least 20 ratings. Movie coordinates have been extracted using a method based on gradient descent [RIK07]. The regularization parameter λ has been chosen by cross-validation such that the SSE is minimized on a randomly chosen test set. We arrived at a value of $\lambda = 0.04$.

As target concepts to be learned from examples, we decided to use the collection of 37 concepts that have been manually created by movie experts from Clerkdogs. For each movie, an expert selected a subset of the available concepts (probably the most relevant ones) and scored the movie with respect to each of these concepts on a 12-point scale (0 to 11). We retrieved a total number of 137,521 scores for the 13,287 movie entries in their database (thus, each movie has been evaluated with respect to 10.4 concepts on average). After mapping these movies to the MovieLens 10M data set (and removing 9 movie entries which have been duplicates), we identified 7,813 movies that are covered by both data sets. Since the extracted coordinates of movies with only a small number of ratings by

[8] http://www.grouplens.org/node/73
[9] http://www.movielens.org

MovieLens users tend to be unreliable, we restricted our experiments to those movies that received at least 100 ratings. Finally, we ended up with a collection of 5,283 movies.

To learn each of the 37 target concepts from a set of examples, we randomly selected a subset of all the movies that have been scored with respect to the respective concept and applied kernel-based support vector regression to estimate the scores of all remaining movies. We then compared the estimated scores to the correct ones and measured both Pearson correlation and Spearman rank correlation to measure the estimates' accuracy. All experiments have been performed using MATLAB in combination with the SVM$^{\text{light}}$ package[10] for kernel regression. After some initial experiments we found the Gaussian radial basis function kernel to be most useful. We chose the learning parameters $C = 10$, $\gamma = 0.1$, and $\varepsilon = 0.1$ as they seemed to generate results of high quality. We did not yet perform a systematic tuning of these parameters.

We tried training sets of different sizes, ranging from 1% of the scored movies up to 90%. Although our scenario clearly focused on small training sets (to enable an easy definition of concepts within the conceptual view), we also included larger training sets to see the effect of the number of training examples on overall performance. For each combination of target concept and training size, we performed 20 experimental runs on randomly selected training sets. All numbers reported in the following section are averages over these 20 runs. Depending on the training size, the whole learning and estimation process took between 0.2 and 202 seconds on a notebook computer with a 2.6 GHz Intel Core Duo CPU (we used only a single core) and 4 GB of RAM.

4.3 Results

The results of our experiments are listed in Table 3. Since there have been large differences in performances among the different concepts, we abstained from aggregating the results into a single performance score over all target concepts. The table only reports the Pearson correlation between our estimations and the correct scores, as we found Pearson correlation and Spearman rank correlation to be extremely similar in most cases.

The most notable result is that all Pearson correlations are positive, that is, it was always possible to learn the target concept correctly at least to a certain degree. While for some concepts we have been able to achieve a quite high accuracy (e.g., *character depth* and *suspense*), there also have been concepts which proved to be hard to learn (e.g., *slow pace* and *revenge*); we can only speculate that these concepts do not significantly influence human movie preferences and thus are not reflected in the user ratings, but leave this question open for further research. We can also observe that for most concepts we can obtain a correlation between 0.2 and 0.3, even for a very small number of training examples. It is also interesting to see that even with small training sizes we are able to come close to the performance achieved on extremely large sizes.

Given that the correlation coefficient of a perfect estimation is 1 and that of a naive baseline (estimating each score by the average score in the training set) is 0, the estimated scores

[10] http://www.cs.cornell.edu/People/tj/svm_light

Concept	#movies	Size of the training set 1%	2%	5%	10%	20%	50%	90%
Action	2480	0.35	0.43	0.50	0.53	0.55	0.59	0.62
Bad Taste	470	0.09	0.18	0.29	0.38	0.44	0.49	0.49
Black Humor	1102	0.09	0.15	0.25	0.28	0.30	0.36	0.36
Blood & Gore	787	0.21	0.27	0.35	0.43	0.47	0.52	0.54
Cerebral	468	0.22	0.25	0.40	0.43	0.45	0.47	0.47
Character Depth	5198	0.68	0.70	0.72	0.73	0.75	0.76	0.76
Cinematography	3019	0.35	0.41	0.46	0.49	0.51	0.54	0.56
Complexity	2611	0.43	0.48	0.52	0.54	0.56	0.59	0.61
Crude Humor	677	0.23	0.36	0.48	0.53	0.57	0.60	0.62
Disturbing	1804	0.30	0.37	0.45	0.49	0.51	0.54	0.56
Downbeat	2831	0.28	0.32	0.38	0.40	0.43	0.46	0.47
Dry Humor	1656	0.15	0.18	0.26	0.29	0.32	0.37	0.40
Fantasy	604	0.10	0.13	0.20	0.27	0.32	0.41	0.48
Fast Pace	2988	0.12	0.15	0.19	0.22	0.23	0.25	0.29
Geek Factor	636	0.15	0.24	0.33	0.40	0.46	0.51	0.59
Hollywood Feel	2885	0.16	0.22	0.28	0.32	0.34	0.38	0.41
Humor	800	0.11	0.20	0.28	0.41	0.48	0.55	0.56
Informative	168	0.09	0.06	0.14	0.18	0.24	0.26	0.26
Offbeat	2414	0.33	0.37	0.40	0.43	0.45	0.47	0.51
Parental Appeal	521	0.20	0.37	0.45	0.48	0.52	0.54	0.58
Political	400	0.15	0.15	0.15	0.19	0.25	0.31	0.32
Revenge	526	0.08	0.09	0.14	0.20	0.25	0.29	0.29
Romance	2728	0.22	0.28	0.37	0.41	0.44	0.48	0.51
Screwball Humor	1133	0.13	0.18	0.23	0.27	0.29	0.34	0.34
Sex	2122	0.23	0.30	0.38	0.43	0.48	0.52	0.53
Slapstick Humor	1098	0.14	0.18	0.26	0.34	0.35	0.40	0.44
Slow Pace	2111	0.09	0.14	0.20	0.23	0.22	0.25	0.27
So Bad It's Good	154	0.02	0.04	0.03	0.09	0.15	0.22	0.27
Soundtrack	1932	0.19	0.24	0.31	0.35	0.39	0.44	0.47
Special Effects	655	0.20	0.24	0.30	0.36	0.42	0.46	0.50
Suspense	2693	0.36	0.44	0.51	0.54	0.57	0.60	0.62
Tearjerker	520	0.01	0.03	0.10	0.14	0.20	0.23	0.25
Terror	678	0.15	0.22	0.35	0.42	0.50	0.54	0.52
Truthfulness	177	0.14	0.29	0.35	0.42	0.47	0.49	0.49
Upbeat	2448	0.18	0.26	0.33	0.38	0.41	0.44	0.44
Violence	2914	0.33	0.42	0.49	0.52	0.55	0.60	0.63

Table 3: Pearson correlations for different target concepts and numbers of training examples.

do not seem to be very accurate. But this assessment take too narrow a view, as it relies on the assumption that the scores provided by Clerkdogs' experts are indeed objectively correct. In our analysis of Clerkdogs' data we found nine movies that occur twice in the movie database, and thus have also been evaluated more than once by the experts, most probably without being aware of it. In total we located 63 movie–concept combinations which have been assessed twice, and used this data to estimate the inter-expert consistency. We found that the Pearson correlation between the rating pairs is only 0.60, which is an surprisingly low value. Therefore, the quality for most of our own results lies somewhere in the middle between a naive approach and a human expert assessment, which makes the results of this study a promising starting point for further research. Since our estimates are based on a broad range of user opinions, there is hope that at least for some concepts the wisdom of the crowd can outperform the experts [Sur04].

To conclude this case study, we have been able to show that a meaningful conceptual view can be created from rating data and a few examples of each target concept. Due to the short computation times, new concepts can be integrated easily.

5 Related Work

Our general approach and methods are related to two other prominent areas of research, namely, multimedia databases and recommender systems.

5.1 Multimedia Databases

The notion of sematic gap as used in this paper originates from research in content-based image and video retrieval, where the mismatch between the users' information needs and the available descriptive information has very early been identified as one of the foremost problems to be solved [HLES06]. Consequently, early approaches focused on extracting numerical scores from images and movies that are related to human perception such as the coarseness of an image, its color distribution, and the parameters of mathematical models describing textures, so called low-level features [SWS+00]. Although these features do not correspond directly to query-relevant concepts, they provide a solid foundation for further processing, in particular for defining meaningful similarity measures. In a sense, these feature spaces correspond to the coordinate spaces created by LSI and the method we presented in our second case study.

More resent research in multimedia databases, focuses on integrating high-level semantic features into the retrieval process [LZLM07]. Here, the idea is to learn relevant query concepts from the collection of extracted low-level features, thus bridging the semantic gap. This general approach follows the same spirit as conceptual views, although the latter also integrate meaningful structured data, which rarely exists in multimedia databases.

5.2 Recommender Systems

Recommender systems [AT05] aim at learning the user's preferences based on his or her previous interaction with the system. Typically, this is done by recording which items have been bought or at least investigated in the past. The main goal of recommender systems is to present a ranked list to the user containing those items which are likely to match the user's taste. Again, there is connection to conceptual views. While conceptual views try to extract concepts that are meaningful for all users, the only concepts relevant to recommender systems are of the general type *well-liked by user X*. Although the task of learning those concepts is much more focused than the extraction problem underlying conceptual views, it also limits the possibilities of recommender systems. For conceptual views, we intentionally chose not to perfectly fit any user's needs and taste but provide a semantically meaningful conceptual description of all entities. In particular, this enables the system to respond to spontaneous changes in the user's mood and taste (e.g., the lover of documentary movies that sometimes just wants to view a comedy). However, as we have seen in our second case study, research in recommender systems offers a wide variety of methods that can be adapted to construct conceptual views.

6 Cognitive Psychology's View on Concepts

Before concluding this paper, we would like to offer cognitive psychology's perspective on concepts. Until now, our motivation and handling of vague concepts has been backed mainly by intuition and common knowledge. While this approach is perfectly valid and in line with previous work on handling concept-related queries in database and information retrieval systems, we next show that many ideas presented are backed by research performed on cognitive psychology over the last forty years.

From a psychological perspective, concepts are mental representations of classes, and their primary function is to enable *cognitive economy* [Ros78]. By dividing the world into categories, we decrease the amount of information to be perceived, learned, remembered, communicated, and reasoned about. Concepts are considered to be formed through the discovery of correlations between features/attributes (that is, clear properties of the entities under consideration). For example, the concept *bird* if formed when noticing a correlation between the features *has wings* and *has feathers*. Features largely correspond to precise attributes that are typically modeled explicitly in databases.

When investigating how people think about concepts and categories, psychologists came to differentiate two kinds of categories: *precise concepts* (also called classical or crisp concepts) and *vague concepts* (also called fuzzy or probabilistic concepts). Precise concepts can be defined through logically combining defining features, e.g., the concept *prime number* can be defined this way. Vague concepts cannot be so easily defined, a popular example is *game*. Their borders tend to be fuzzy. Some concepts may even appear both in a precise as well as in a vague shape. Biologists may suggest that we use the word *fruit* to describe any part of a plant that has seeds, pulp, and skin. Nevertheless, our natural, vague

concept of fruit usually does not easily extend to tomatoes, pumpkins, and cucumbers. The notion of vagueness as used in this context is clearly to be distinguished from the problem of missing knowledge. Vague concepts are inherently fuzzy, that is, even with perfect knowledge of the world they do not allow crisp classifications of all entities. Vague concepts are primarily based in human intuition and often cannot be made explicit in terms of logical rules.

Another central property of human concepts is typicality, that is, within a category, items differ reliably regarding their "goodness-of-example." While both penguins and robins clearly are considered birds, the former are judged as being untypical instances of this concept. This phenomenon can even be observed in precise concepts. For example, people generally consider 13 to be a better example of a prime number than 89 [Mur04].

Cognitive psychology also offers formal models for capturing the essential properties of real-world concepts. These models can roughly be classified into two groups: models based on features, and models based on semantic spaces. In feature-based models, each entity is represented by a list of features; the process of categorization often is modeled by complex interactions between these features, e.g., by means of neural network models. Models based on semantic spaces represent each entity as a point in some high-dimensional space, where individual coordinate axes do not necessarily have an interpretable meaning; categorization is modeled by means of similarity measures in this space, e.g., distance of an entity to the concept's prototype. Both types of models have been found to be highly accurate; some researchers even propose to create hybrid models to get the best of both worlds [RJ10].

Now, the connection to the models used in our case studies becomes apparent. The model used in the first study (mobile phones) models the target concept using a feature-based approach; classification is performed based on relative feature frequency. In our second study (movies), the underlying model uses a semantic space for categorization, which has been extracted from rating data. Therefore, our work provides a solid foundation for further research and may readily be extended to incorporate the important notion of typicality.

7 Conclusion and Outlook

In this work, we identified and discussed one of the major problems of entity-centric search, namely, the lack of support for querying natural concepts in a structured fashion. We demonstrated that most of the relevant information is already present in current database systems and only needs to be made visible to applications.

To this end, we proposed the notion of conceptual views, which provide a systematic and unified interface between the broad range of data types available in current database systems and existing query processing algorithms, thus bridging the semantic gap between the vague concepts characterizing the users' information need and the database schema.

In two extensive case studies from different domains we have been able to demonstrate that our vision of an automated extraction process for vague concepts can indeed be put into practice. We also showed that our work is backed by theories from cognitive psychology.

However, this paper also revealed that we have just started our journey towards an effective, efficient, and reliable construction and maintenance of conceptual views. In future work, we will develop a detailed theory of conceptual views that is closely interwoven with models and theories from cognitive psychology. Since we are primarily dealing with natural, vague concepts, there needs to be a stronger focus on research results from the humanities. Moreover, we are going to both continue the work on our existing concept extractors as well as creating new ones for application scenarios that are not covered well enough yet.

References

[AT05] Gediminas Adomavicius and Alexander Tuzhilin. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[BHJ+10] Dominik Benz, Andreas Hotho, Robert Jäschke, Beate Krause, and Gerd Stumme. Query Logs as Folksonomies. *Datenbank-Spektrum*, 10(1):15–24, 2010.

[Che76] Peter Pin-Shan Chen. The Entity-Relationship model—Toward a unified view of data. *ACM Transactions on Database Systems*, 1(1):9–36, 1976.

[Cho03] Jan Chomicki. Preference Formulas in Relational Queries. *ACM Transactions on Database Systems*, 28(4):427–466, 2003.

[CRS+07] Michael J. Cafarella, Christopher Ré, Dan Suciu, Oren Etzioni, and Michele Banko. Structured Querying of Web Text: A Technical Challenge. In *Proceedings of CIDR 2007*, pages 225–234, 2007.

[DDF+90] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[GUP06] José Galindo, Angélica Urrutia, and Mario Piattini. *Fuzzy databases: Modeling, Design, and Implementation*. Idea Group, 2006.

[HLES06] Jonathon S. Hare, Paul H. Lewis, Peter G. B. Enser, and Christine J. Sandom. Mind the Gap: Another Look at the Problem of the Semantic Gap in Image Retrieval. In *Multimedia Content Analysis, Management and Retrieval 2006*, volume 6073 of *Proceedings of SPIE*. SPIE, 2006.

[HPF+09] Ming Hua, Jian Pei, Ada W. C. Fu, Xuemin Lin, and Ho Fung Leung. Top-K Typicality Queries and Efficient Query Answering Methods on Large Databases. *The VLDB Journal*, 18(3):809–835, 2009.

[IBS08] Ihab F. Ilyas, George Beskales, and Mohamed A. Soliman. A Survey of Top-K Query Processing Techniques in Relational Database Systems. *ACM Computing Surveys*, 40(4), 2008.

[KBV09] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer*, 42(8):30–37, 2009.

[Kie02] Werner Kießling. Foundations of Preferences in Database Systems. In *Proceedings of VLDB 2002*, pages 311–322. Morgan Kaufmann, 2002.

[KT09] Ravi Kumar and Andrew Tomkins. A Characterization of Online Search Behavior. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 32(2):3–11, 2009.

[LHC05] Bing Liu, Minqing Hu, and Junsheng Chen. Opinion Observer: Analyzing and Comparing Opinions on the Web. In *Proceedings of WWW 2005*, pages 342–351. ACM, 2005.

[LZLM07] Ying Liu, Dengsheng Zhang, Guojun Lua, and Wei-Ying Ma. A Survey of Content-Based Image Retrieval with High-Level Semantics. *Pattern Recognition*, 40(1):262–282, 2007.

[Moe06] Marie-Francine Moens. *Information Extraction: Algorithms and Prospects in a Retrieval Context*. The Information Retrieval Series. Springer, 2006.

[Mot88] Amihai Motro. VAGUE: A User Interface to Relational Databases that Permit Vauge Queries. *ACM Transactions on Office Information Systems*, 6(3):187–214, 1988.

[MS06] Imran R. Mansuri and Sunita Sarawagi. Integrating Unstructured Data into Relational Databases. In *Proceedings of ICDE 2006*. IEEE Computer Society, 2006.

[Mur04] Gregory L. Murphy. *The Big Book of Concepts*. MIT Press, 2004.

[RIK07] Tapani Raiko, Alexander Ilin, and Juha Karhunen. Principal Component Analysis for Large Scale Problems with Lots of Missing Values. In *Proceedings of ECML 2007*, volume 4701 of *LNAI*, pages 691–698. Springer, 2007.

[RJ10] Brian Riordan and Michael N. Jones. Redundancy in Perceptual and Linguistic Experience: Comparing Feature-Based and Distributional Models of Semantic Representation. *Topics in Cognitive Science*, to appear, 2010.

[Ros78] Eleanor Rosch. Principles of Categorization. In Eleanor Rosch and Barbara L. Lloyd, editors, *Cognition and Categorization*, pages 27–48. Lawrence Erlbaum, 1978.

[SB10] Joachim Selke and Wolf-Tilo Balke. Extracting Features from Ratings: The Role of Factor Models. In *Proceedings of M-PREF 2010*, pages 61–66, 2010.

[SS04] Alex J. Smola and Bernhard Schölkopf. A Tutorial on Support Vector Regression. *Statistics and Computing*, 14(3):199–222, 2004.

[Sur04] James Surowiecki. *The Wisdom of Crowds*. Doubleday, 2004.

[SWS+00] Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-Based Image Retrieval at the End of the Early Years. *IEEE Transactions on Pattern Analysis ans Machine Intelligence*, 22(12):1349–1380, 2000.

[Wei07] Gerhard Weikum. DB & IR: Both Sides Now. In *Proceedings of SIGMOD 2007*, pages 25–29, 2007.

[WKQ+08] Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou, Michael Steinbach, David J. Hand, and Dan Steinberg. Top 10 Algorithms in Data Mining. *Knowledge and Information Systems*, 14(1):1–37, 2008.

[WKRS09] Gerhard Weikum, Gjergji Kasneci, Maya Ramanath, and Fabian Suchanek. Database and Information-Retrieval Methods for Knowledge Discovery. *Communications of the ACM*, 52(4):56–64, 2009.

[ZG09] Xiaojin Zhu and Andrew B. Goldberg. *Introduction to Semi-Supervised Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool, 2009.

[ZGBN07] Xuan Zhou, Julien Gaugaz, Wolf-Tilo Balke, and Wolfgang Nejdl. Query Relaxation using Malleable Schemas. In *Proceedings of SIGMOD 2007*, pages 545–556, 2007.