SEMEX: Mining for Personal Information Integration

Xin Dong, Alon Halevy, Ema Nemes, Stephan B. Sigurdsson, and Pedro Domingos

University of Washington, Seattle
{lunadong, alon, enemes, stebbi, pedrod}@cs.washington.edu

Abstract. Personal information management is one of the key applications of the semantic web. Whereas today's devices store data according to applications, ideal personal information management system should treat all data as a set of meaningful objects and associations between the objects. To ensure extensibility, a personal information management system should automatically incorporate associations generated in multiple ways: mining specific personal data sources, or integrating with external data. As a first step in this direction, we describe the SEMEX system that provides a logical and integrated view of one's personal information.

1 Introduction

The advent of modern networking technology has enabled numerous opportunities for sharing data among multiple parties. Today, data sharing and integration is crucial in large enterprises, government agencies, collaborative scientific projects, and in our personal information management where individuals need to share data from various sources. The pervasive applications of data sharing and integration have led to a very fruitful line of research and recently to a significant industry as well. The vision of the Semantic Web is even more ambitious: web-scale data and knowledge integration.

Despite the immense progress, building an information integration application is still a major undertaking that requires significant resources, upfront effort, and technical expertise. Today, information integration projects proceed by identifying needs in an organization and the appropriate set of data sources that support these needs, typically focusing on frequently recurring queries throughout the organization. As a result, current information integration systems have two major drawbacks. First, evolving the system is hard as the requirements in the organization change. Second, many smaller-scale and more transient information integration tasks that we face on a daily basis are not supported. In particular, integration that involves personal data sources on one's desktop or in one's laboratory is not supported. On the Semantic Web front, it has been observed on several occasions that the growth of the Semantic Web is rather slow, and that personal information management has the potential of fueling faster growth [**?**].

The vision of *on-the-fly information integration* is to fundamentally change the cost-benefit equation associated with integrating information sources. The goal is to aid non-technical users to easily integrate diverse information sources. To achieve this goal, we posit that information integration systems should incorporate two principles:

- The information integration environment should be closely aligned with and be an extension of users' *personal* information space, *i.e.*, the information they store on the desktop (*e.g.*, files, emails, contact lists, spreadsheets, personal databases). In that way, users can extend their personal information views with public data resources.
- Information integration should happen as a *side effect* of people doing their daily jobs, by continuous accumulation of the solutions they produce for their needs of the moment, and by leveraging experiences from previous integration tasks. In short, information integration should be *woven into the fabric* of the organization.

We are building the SEMEX System (short for SEMantic EXplorer), that embodies the vision of on-the-fly integration. With SEMEX, users can access a set of information sources, spanning from personal to public, and from unstructured to structured. Users interact with SEMEX through a domain ontology that offers a set of meaningful domain objects and relationships between these objects. Information sources are related to the ontology through a set of mappings, thereby enabling queries that span multiple sources. Users can personalize their domain models, share domain models with other users, and import fragments of public domain models in order to increase the coverage of their information space. When users are faced with an information integration task, SEMEX aids them by trying to leverage from previous tasks performed by the user or by others with similar goals. Hence, the effort expended by one user later benefits others.

There are three main thrusts to the SEMEX System. This paper focuses on the first of these.

Personal information management (PIM) and integration: Today, the personal information on our desktop is organized by applications (*e.g.*, email, calendar, files, spreadsheets). Finding a specific piece of information involves either searching a file directory or employing a particular application. Integration of multiple pieces of information can only be done manually. Nevertheless, even as early as 1945, Vannevar Bush pointed out in his vision of the *Personal Memex* [Bus45] that our mind works by connecting disparate data items with *associations*, which are not naturally supported by directory and application structures. Hence, an ideal personal information management system should provide a *logical view* of our data so that it can support search through associations between multiple items. A key for its success is that personal information should be populated automatically. This requirement poses an important challenge to the data mining and information extraction communities. The bulk of this paper describes a system that automatically creates such a view, and describes the main technical challenges in doing so.

Personal information as a platform for information integration: Once we have a logical view of our personal information, we can relate external sources to it, thereby facilitate personal tasks that require integration of multiple external sources. Using an architecture such as peer-data management [TIM+03,TH04], we can share data among multiple users. The challenges involved in building this component of SEMEX are to develop tools that make it easy to incorporate external sources (by non-technical users), to personalize the domain model of one's data, and to share these personalized views of data.

Leveraging previous integration tasks: Information integration tasks are often repetitive or closely related to each other. Hence, the final component of SEMEX is to leverage previous integration tasks to facilitate future ones. In this way, users can benefit from integrations performed by colleagues interacting with the same data sources. Our past work on schema matching using machine learning [DDH01] has shown that previous experience can be used to boost the performance of semi-automatic schema matching. Following the same line, mining previous information integration tasks poses several exciting challenges to the data mining community.

In the remainder of the paper we discuss how SEMEX creates a database of instances and associations from one's personal information, thereby offering a logical view of this data. This database complements current storage of personal information, and will form the basis for a variety of services relating to personal information and to information integration. The main technical challenge we address in this component of SEMEX is to reconcile multiple references to the same real-world data item. In contrast to previous work on object-matching (a.k.a. *record linkage, reference reconciliation*), here the references we need to consider (1) do not conform to a single schema, (2) may have multiple values for a single attribute, and (3) typically have very few attributes, thereby exacerbating the challenges involved.

The paper is organized as follows. Section 2 describes the architecture of SEMEX. Section 3 describes the reference reconciliation algorithm and discusses the experimental results on a significant personal data set. Section 4 discusses related work and concludes.

2 Personal Information Management

The first goal of SEMEX is to create a database that consists of objects and relationships between objects obtained from one's personal information (see Figure 1). Objects come from a variety of sources, such as email, contacts, calendar, Latex and Bibtex, Word documents, Powerpoint presentations, pages in the user's web cache, other files in a person's personal or shared file directory, and data in more structured sources, such as spreadsheets and databases. Associations are binary relationships between objects, such as AuthorOf, Sender, Cites, *etc.* Given this logical model of one's personal information, users can seamlessly browse or query their data.

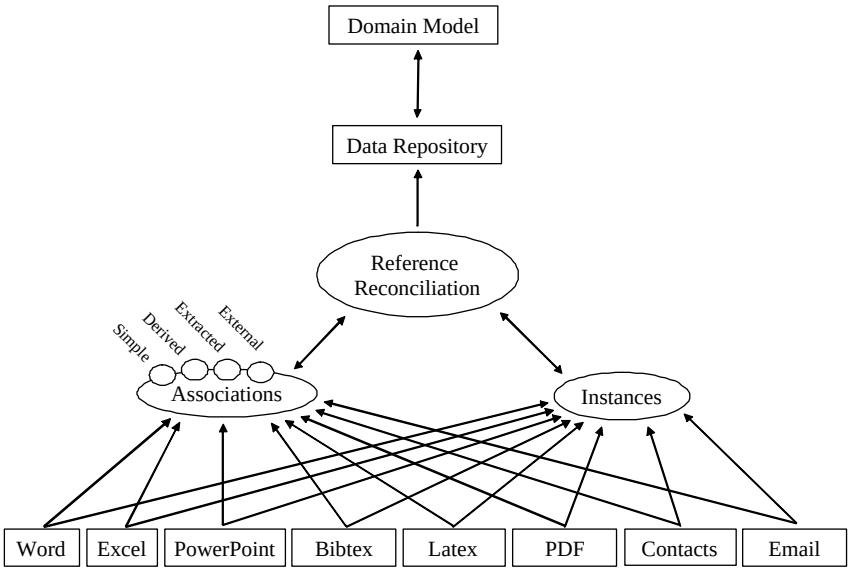

Fig. 1. The architecture of SEMEX. SEMEX begins by extracting data from multiple sources. Such extractions create instances of classes in the domain model. SEMEX employs multiple modules for extracting associations, as well as allowing associations to be given by external sources or to be defined as views over other sets of associations. To combine all these associations seamlessly, SEMEX automatically reconciles multiple references to the same real-world object. The user browses and queries all this information through the domain model.

SEMEX stores the objects in a domain ontology, which includes a set of *classes* such as Person, Publication and Event, and *relationships* (which we refer to as *associations*). At the moment the SEMEX uses a simple data model of classes and associations, but there is a clear need for supporting subclasses and subproperties (*e.g.*, AuthorOf is a subclass of MentionedIn). We also note that our domain model is not a proposal for a standard schema for personal information; it will evolve from several base models by modification and personalization, and we will have to support mappings between the various schemas. The instances and associations that SEMEX extracts are stored in a separate database. While we have not implemented any sophisticated update mechanisms yet, we envision a module that periodically updates the database and makes the process transparent to the user.

Associations and instances: The key architectural premise in SEMEX is that it should support a variety of mechanisms for obtaining class and association instances. SEMEX currently supports the following:

1. *Simple:* In many cases, objects and associations are already stored conveniently in the data sources and they only need to be extracted into the

domain model. For example, a contact list already contains several important attributes of persons, and email messages contain several key fields indicating their senders and receivers.

2. *Extracted:* A rich set of objects and associations can be extracted by analyzing specific file formats. For example, authors can be extracted from Latex files and Powerpoint presentations, and citations can be computed from the combination of Latex and Bibtex files.
3. *External:* External sources can explicitly define many associations. For example, if CiteSeer were to publish a web interface, one could extract citation associations directly from there. Alternatively, a professor may wish to create a class MyGradStudents and populate the class with data in a department database.
4. *Defined:* In the same way as views define interesting relations in a database, we can define objects and associations from simpler ones. As simple examples, we can define the association coAuthor, or the concept emailFromFamily.

In a sense, the domain ontology of SEMEX can be viewed as a *mediated schema* over the set of personal information sources. Instances of the classes and the associations in the domain ontology are obtained from multiple sources. The distinguishing aspect of our context from other information integration settings is that we expect the ontology to be significantly evolved by the user through adding new classes and arbitrary associations.

To make such a system useful, we must ensure that all the data mesh together seamlessly. Specifically, if the same object in the real world (*e.g.*, a person) is referred to in two ways, the system must be able to determine that the two references are to the same object. Otherwise, we will not be able to query effectively on associations, let alone follow chains of associations. In personal data, reference reconciliation is extremely challenging. For example, in the personal data of one author of this paper, there were over 100 distinct ways in which the author was referred. The next section describes the reference reconciliation algorithm of SEMEX.

Browsing and querying interface: SEMEX offers an interface that combines intuitive browsing and a range of querying options. Figure 2 shows a sample screenshot from browsing SEMEX database. Initially, a user can simply type keywords into a search box and SEMEX will return all the objects that are somehow associated with the keyword. For example, typing `Bernstein` in the search box will produce a set of objects that mention Bernstein. Note that the answers to such a query can be a heterogeneous set of objects; SEMEX already classifies these objects into their classes (Person, Publication, *etc.*). When the `Bernstein` person object is selected, the user can see *all* the information related to the person, and the relationship is explicitly specified. (*e.g.*, AuthorOf, CitedIn). The user can then browse any of Bernstein's emails, papers (and then to the objects corresponding to other authors), *etc.* An alternative way to begin browsing is to choose a particular property in the domain model (e.g., AuthorOf) and enter a specific value, thereby specifying an association query.

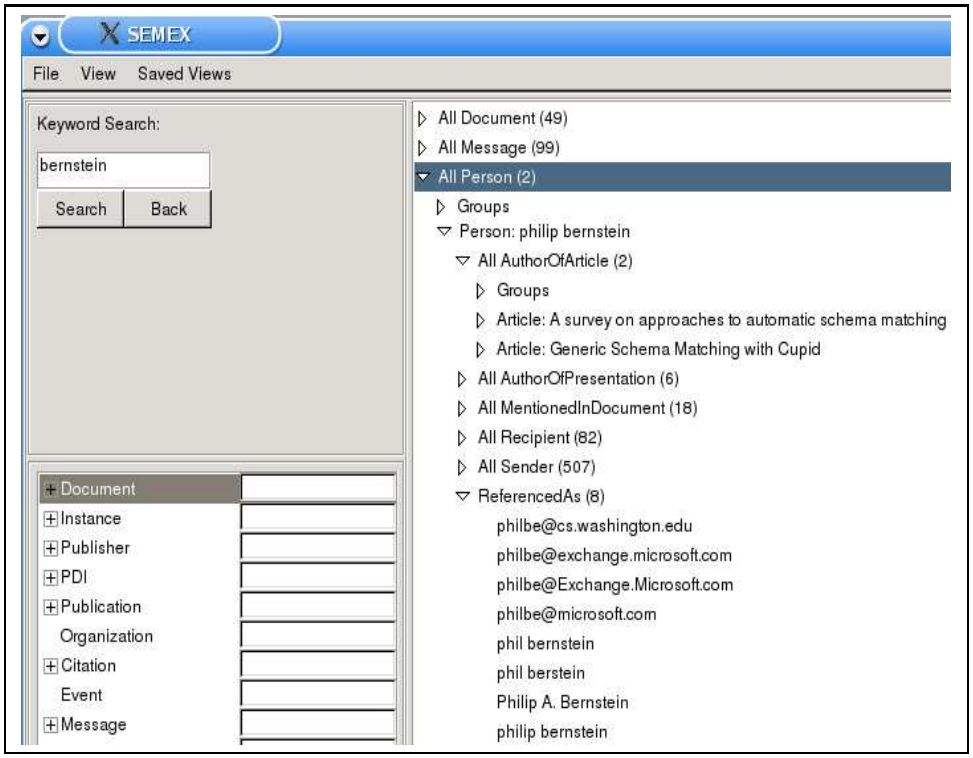

Fig. 2. A sample screenshot from browsing the SEMEX database. Note that the ReferencedAs attribute lists the different ways in which Phil Bernstein is referenced in this personal data set.

3 Reference Reconciliation in Semex

In this section we describe how SEMEX reconciles multiple references to the same real-world object. Our discussion focuses on the hardest reconciliation problem, namely references to persons. We leave the generalization of our algorithm to other objects and domains for further study.

The following example shows three references of persons derived from contact, email and Bibtex data.

$$\begin{aligned} \text{name, phone} &: \texttt{Mike Carey}, (123)456-7890 \\ \text{email} &: \texttt{carey@almaden.ibm.edu} \\ \text{name} &: \texttt{M. Carey} \end{aligned}$$

Earlier approaches (see [BMC+03] for a recent survey) to reference reconciliation focus on reconciling tuple references from a single database table; these tuples share attributes and each attribute allows a single value. These approaches

do not directly apply to SEMEX for four reasons. First, the data sources in SEMEX are heterogeneous, containing different sets of attributes; as the above example shows, the attributes of the first and the second references even do not overlap. Second, each attribute of a person object may contain multiple values: it is common for a person to have multiple email accounts and phone numbers. Furthermore, some of the statistical techniques that have been considered are difficult to apply because of the relatively small size of the personal data sets. Finally, training data is also not readily available, which limits the application of supervised learning. On the other hand, the size of the data sets allows for more computationally intensive matching algorithms.

3.1 Reference reconciliation algorithm

Traditionally, the reference reconciliation problem was solved by independently matching each pair of references, and taking a transitive closure over matching pairs. In the case of people, each single reference is rather weak (*i.e.*, contains relatively little information). To tackle this problem, our algorithm repeats the comparing-and-clustering process several times, each time considering a result cluster obtained from the preceding pass as a single reference, and recomputing the distances between new references based on a different distance measure. The stronger reference may potentially be matched with other instances with which its constitutes could not be matched before.

Specifically, the algorithm begins by assigning each reference to a class of cardinality one and then successively refines the relation in four passes.

Step 1: Reconciling based on shared keys. The first step merges references that share exact values on keys. For person instances, `name` and `email` can each serve as a key.

Step 2: Reconciling based on string similarity. The second iteration combines string matching features with domain-specific heuristics. We employ edit distance [BMC+03] to measure string similarity. In some cases we exploit the specific data types and apply domain heuristics. For example, we compare email addresses by exploiting knowledge of the different components of the address and recognizing certain mail software idiosyncrasies. In the case of phone numbers, we allow for missing area codes or additional extension numbers.

Step 3: Applying global knowledge. Now that we have grouped multiple references into clusters, we can extract global information to perform additional merging. We give two important examples of such global knowledge. In the first case, the knowledge is extracted *within* the cluster, and in the second case we use *external* information. We note that the algorithm is conservative when applying global knowledge, as we consider avoiding false positives more important to guarantee quality browsing of personal information.

- *Time-series comparison:* The time-series analyzer selects pairs that were judged similar in the previous passes, but not combined. It then collects for

each reference a set of time stamps associated with its email messages. If the time series have little or no overlap, the references are merged. This heuristic works well for detecting people who move from one institution to another. In our experiments, this method was very effective.
- *Search-engine analysis:* Our search-engine analyzer feeds the texts of two references into the Google search engine (via their web-service interface) and compares the top hits. Two references to the same person object tend to obtain similar top hits in Google search. In our experiments, this technique also helped resolve a significant number of references.

The result of the reconciliation algorithm is a high-quality reference list of people mentioned in one's personal data. We then leverage this list to obtain additional associations within the data set. For example, we search for occurrences of the names in the reference list in spreadsheets and the top portions of Word and PDF files to create associations to these types of files. We do not discuss the details of this step due to space limitations.

	Count	%	Size [kb]	%
Messages	18037	—	—	—
Contacts	240	—	—	—
Files	7085	100%	886836	100%
Latex	582	8%	7332	1%
Bibtex	25	0.9%	2236	0.3%
PDF	97	1.3%	24768	2.8%
PostScript	668	9.4%	215584	24%
Plain text	51	0.7%	940	0.1%
Rich text	31	0.4%	104	0.0%
HTML/XML	666	9.4%	7060	0.8%
Word	400	5.6%	12092	1.3%
PowerPoint	777	11%	151045	17%
Excel	55	0.7%	1396	0.2%
Multimedia	539	7.6%	123521	14%
Archives	475	6.7%	15754	1.8%
Other	1809	32%	194112	22%

Table 1. The characteristics of our experimental data set.

3.2 Experiments

We describe the results of experiments applied to a personal data set of one author of this paper[1]. The data set spans six years of activities and consists of

[1] To further complicate matters, this author changed his name from Levy to Halevy a few years ago.

	Before Reconciliation	%
Instances	23318	100%
Person	5014	22%
Message	17322	74%
Document	805	3%
Publication	177	1%
Associations	38318	100%
senderOf	17316	45%
recipientOf	20530	54%
authorOf	472	1%

Table 2. The number of instances extracted from the raw data for classes in the domain model. For example, after scanning all the sources, we have 5014 person references, and these need to be reconciled.

the usual variety of personal data (though probably more Latex files than typical computer users). Table 1 details the characteristics of the raw data, and Table 2 shows the number of instances extracted from the raw data for several of the classes in the domain model.

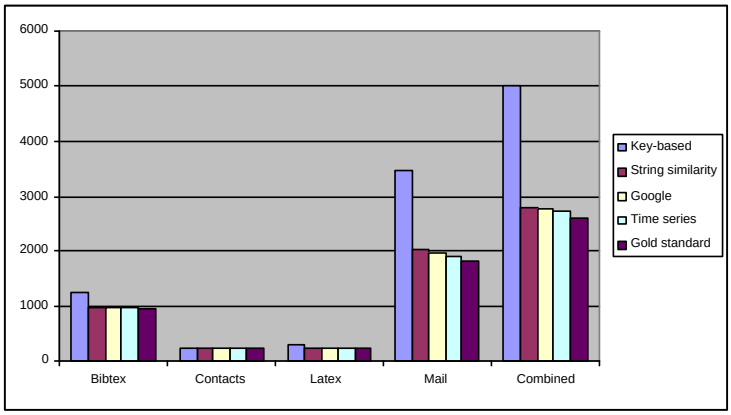

Fig. 3. This figure shows the progress of the reference reconciliation algorithm w.r.t. its different steps. The right-most set of bars concerns the entire data set, while the other sets consider individual components of the data set.

We limit the following discussion to person instances. Figure 3 shows the progress of the matching algorithm for each component of the data set in isolation (*i.e.*, for Bibtex, contacts, email, latex), and then the results for all these components combined. The rightmost column (labeled *gold standard*) in each

group indicates the *actual* number of distinct objects in the domain. The other columns report the numbers of clusters after each reconciliation step.

We observe from the experiment that the first two steps of the algorithm remove 91% of the extra references (*i.e.*, differences between the references extracted directly from the raw data set and the distinct ones in the gold standard). The time-series and Google analyzers successively remove an additional 1.7% of the beginning total of extra references each, but more importantly, these correspond to 18% and 29% of the references that still need to be reconciled. We also observed that changing the order of the time-series and Google analyzers does not change the results substantially.

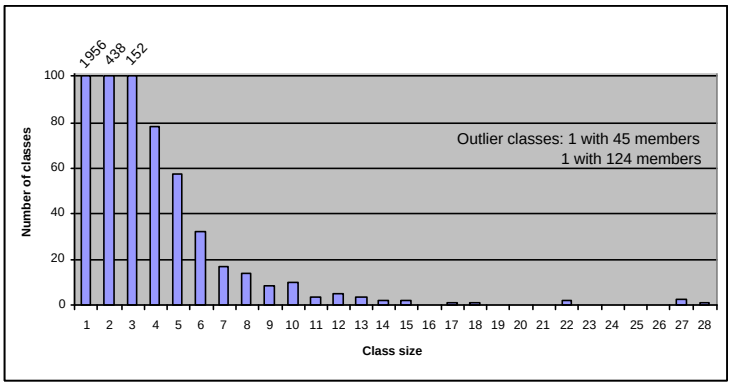

Fig. 4. The number of different references per person after the reconciliation algorithm is applied.

Another perspective on the quality of the reference reconciliation is shown in Figure 4. Each bar shows the number of persons for whom there are n references, where n labels the bar (therefore, when users browse the data they could expand the single collapsed reference to see the n original references to that person).

In conclusion, while the current reconciliation algorithm already provides a reasonable start, we believe that techniques for reference reconciliation by growing clusters of references merits additional study.

4 Related Work and Conclusions

A number of PIM projects studied the method to organize and search information effectively. They all discard the traditional hierarchical directory model. Haystack [QHK03] and MyLifeBits [GBL+02] resort to annotations in building a graph model of information; Haystack puts more emphasis on personalization. Placeless Documents [DEL+00] annotates documents with property/value pair, and group documents into overlapping collections according to the property

value. Stuff I've Seen (SIS) [DCC+03] indexes all types of information and provides a unique full-text search interface. Finally, LifeStreams [FG96] organizes documents based on a chronological order. All of the above projects manage information at the document level. Our approach distinguishes from them by taking objects as the search and organization unit and facilitating the search with associations between objects. The system uses an ontology to guide information management, allowing manipulation and personalization of the ontology.

This paper serves to bring personal information management closer to the mainstream of data management research, and as a platform for the next generation of information integration systems. Specifically, we have argued that the keys to research on personal information management are to seamlessly integrate users' personal information views with organizational data sources and to integrate information on-the-fly. We described the current implementation of SEMEX that performs personal information management and integration. We described a novel reference reconciliation algorithm for personal information, and showed that it performs well on a sizable data set.

Personal information management is a rich area for further research. In the immediate future, our goal is to improve the reference reconciliation algorithm. We believe that rich probabilistic models hold great promise in this context because there is a clear need to combine evidences from multiple sources during the reconciliation. Further down the road, we plan to use the SEMEX database to discover useful patterns in one's data set, such as clusters of people who are related in ways that are not explicit in one's data. Finally, we will use SEMEX to coordinate multiple PIM devices and provide a flexible tool for merging multiple data sets of a user.

References

[BMC+03] Mikhail Bilenko, Raymond Mooney, William Cohen, Pradeep Ravikumar, and Stephen Fienberg. Adaptive name matching in information integration. *IEEE Intelligent Systems Special Issue on Information Integration on the Web*, September 2003.

[Bus45] Vannevar Bush. As we may think. *The Atlantic Monthly*, July 1945.

[DCC+03] Susan Dumais, Edward Cutrell, JJ Cadiz, Gavin Jancke, Raman Sarin, and Daniel C. Robbins. Stuff i've seen: A system for personal information retrieval and re-use. In *SIGIR*, 2003.

[DDH01] Anhai Doan, Pedro Domingos, and Alon Halevy. Reconciling schemas of disparate data sources: a machine learning approach. In *Proc. of SIGMOD*, 2001.

[DEL+00] Paul Dourish, W. Keith Edwards, Anthony LaMarca, John Lamping, Karin Petersen, Michael Salisbury, Douglas B. Terry, and James Thornton. Extending document management systems with user-specific active properties. *ACM TOIS*, 18(2), 2000.

[FG96] Eric Freeman and David Gelernter. Lifestreams: a storage model for personal data. *SIGMOD Bulletin*, 1996.

[GBL+02] Jim Gemmell, Gordon Bell, Roger Lueder, Steven Drucker, and Curtis Wong. Mylifebits: Fulfilling the memex vision. In *ACM Multimedia*, 2002.

[QHK03] Dennis Quan, David Huynh, and David R. Karger. Haystack: A platform for authoring end user semantic web applications. In *ISWC*, 2003.

[TH04] Igor Tatarinov and Alon Halevy. Efficient query reformulation in peer data management systems. In *Proc. of SIGMOD*, 2004.

[TIM+03] Igor Tatarinov, Zachary G. Ives, Jayant Madhavan, Alon Y. Halevy, Dan Suciu, Nilesh N. Dalvi, Xin Dong, Yana Kadiyska, Gerome Miklau, and Peter Mork. The piazza peer data management project. *SIGMOD Record*, 32(3):47–52, 2003.