Extracting Entity Profiles from Semistructured Information Spaces

Robert A. Nado Scott B. Huffman

Price Waterhouse Technology Centre
68 Willow Road
Menlo Park, CA 94025-3669
{nado, huffman}@tc.pw.com

Abstract

A semistructured information space consists of multiple collections of textual documents containing fielded or tagged sections. The space can be highly heterogeneous, because each collection has its own schema, and there are no enforced keys or formats for data items across collections. Thus, structured methods like SQL cannot be easily employed, and users often must make do with only full-text search. In this paper, we describe an approach that provides structured querying for particular types of *entities,* such as companies and people. Entity-based retrieval is enabled by *normalizing* entity references in a heuristic, type-dependent manner. The approach can be used to retrieve documents and can also be used to construct entity profiles – summaries of commonly sought information about an entity based on the documents' content. The approach requires only a modest amount of meta-information about the source collections, much of which is derived automatically.

1 Introduction

Decentralized information sharing architectures like the World Wide Web and Lotus Notes make it easy for individuals to add information, but as the space grows, retrieval becomes more and more difficult. *Semistructured* information sharing systems, including Lotus Notes™ and a variety of meta-tagging schemes being developed for the World Wide Web (e.g. Apple's Meta Content Framework [Guh97]), address part of this problem by providing the ability to structure local parts of the information space. In a semistructured information space, documents are sectioned into weakly-typed fields according to user specifications, and documents with the same field structure can be grouped into collections. Within a collection, field values can be used as indexes for easier retrieval.

Unfortunately, semistructuring document collections does not solve the problem of retrieving information across a large information space. Even if individual collections are well designed for retrieval, users can be overloaded with the sheer number of collections. Retrieval across the entire space is difficult because it is highly heterogeneous. Each collection has its own local schema, and there are no enforced keys or formats for data items within or across collections.

Our work addresses the problem of finding and integrating useful information across collections in large semistructured information spaces. Our goal is to provide querying that is more powerful and precise than full-text search, but without requiring the collections to be strongly typed, data normalized, and fully mapped to a global schema, as methods like multidatabase SQL require. In this paper, we focus on the retrieval of integrated summaries of useful information (entity profiles), drawn from multiple, heterogeneous document collections.

Our approach is to provide high quality retrieval of information related to important *entities* in the information space. In our organization (a large professional services firm), important types of entities include people, companies, and consulting skills. A review of our largest collections revealed that nearly always, references to important entities are fielded rather than buried in free-running text. Because the same entity can be referred to in many different ways across a heterogeneous information space, our entity retrieval system *normalizes* references to entities in a heuristic, type-dependent manner. For instance, the person names "Mr. Bob Smith", "Smith, Robert", and "R. J. Smith" are normalized such that a query for any

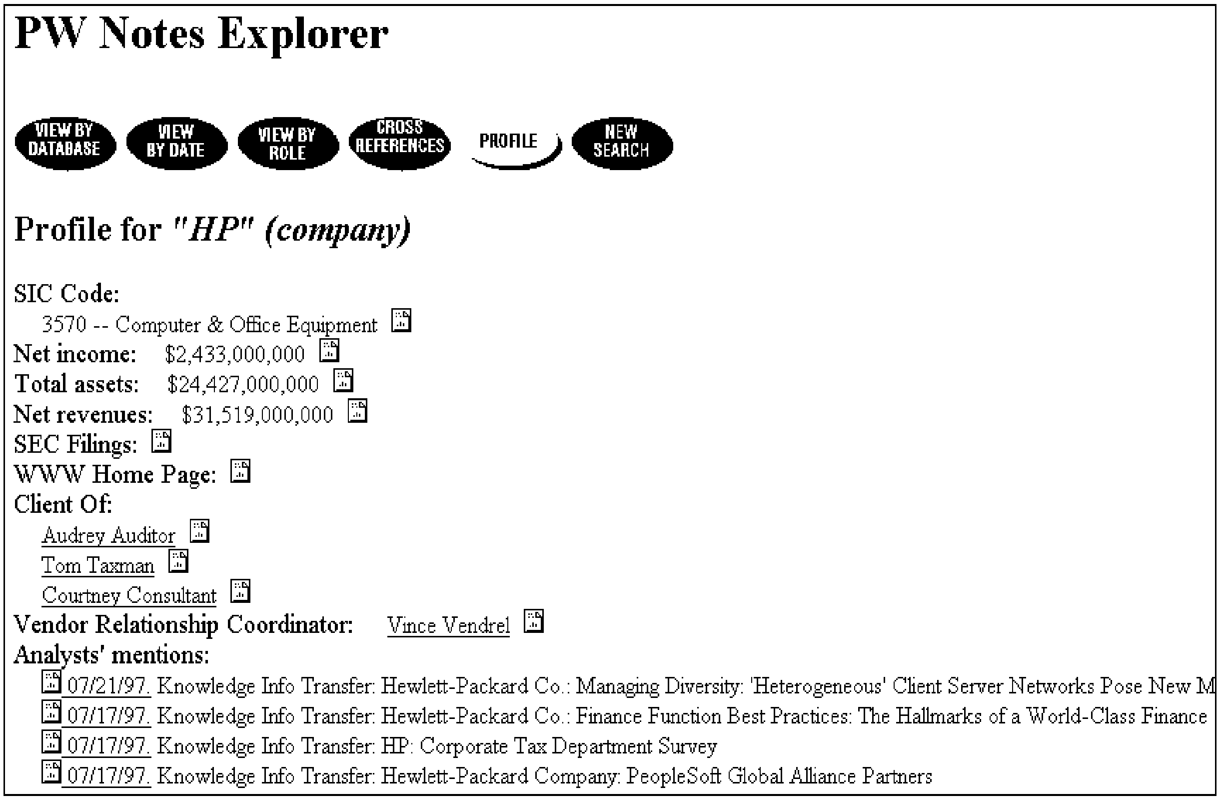

Figure 1: Results of NX Profile Search

one (or a number of other possible forms) will retrieve documents containing any of them.

We have implemented an entity-based retrieval system called *NX* (for Notes Explorer) that operates over a large semistructured information space. The space currently includes over one hundred corporate Lotus Notes collections and a small set of web collections, together containing about 300,000 documents. NX provides full-text search, entity-based document retrieval for people, companies, and skills, and profile extraction for people and companies. It is delivered over an intranet using HTML.

A key hypothesis behind this work is that *a relatively small amount of meta-information* – much less than that required to normalize and map collections to a comprehensive global schema – *can give a large gain* in query power and precision over knowledge-free methods like full-text search. NX is one illustration of this hypothesis. It requires only a modest amount of meta-information about each collection – an indication of fields containing entities in various semantic categories and pairs of fields that stand in specific semantic relations – and uses it to produce a dramatic improvement in retrieval quality for entity-related queries. Much of the required meta-information can actually be inferred automatically based on field names and data within the collections, using a simple heuristic classifier.

In what follows, we first motivate the task of generating entity profiles with a real-world example. Next, we describe the main components of our retrieval system. We conclude by discussing related and future work.

2 Entity-based retrieval

In a corporate setting, information in different documents is frequently linked through references to entities with business importance, such as people and companies. Often, users search for information about *particular* entities (e.g., "What is Bob Smith's phone number?" or "Who's the manager for the XYZ Co. account?") as opposed to ungrounded, aggregate queries across sets of entities (e.g. "Show me all managers with more than five clients over $5 million in sales"). We designed NX to support this type of search.

Consider a typical example from our organization. A staff member is writing a proposal to XYZ Company for some consulting work. She needs answers to questions like:

(a) How large is XYZ Company? E.g., what are their assets, revenues, etc.?

(b) Does our organization have a prior relationship with XYZ? Have we done other consulting work for them in the past?

(c) If so, who did that work, and how can they be contacted?

Each question refers to entities of various types – XYZ Company, staff members, phone numbers, etc. – and these entities may be referred to differently in different documents. Some questions involve information that may be found in many collections of the same type – e.g., information about prior work for XYZ (b) might be found in numerous collections containing client engagements. Others involve linking information about XYZ with information about another entity -- e.g., question (c) requires finding staff names in documents that list XYZ engagements, and then finding contact information for those staff names.

Figure 1 displays the results of a profile search in NX given "HP" as a company name search string.[1] Normalization allows NX to retrieve information from documents that mention "Hewlett Packard", "Hewlett-Packard, Inc.", etc., as well as "HP". The headings (e.g., "SIC Code:" and "Client Of:") list specific values that have the specified relationship to the company of interest. These values are drawn from multiple documents in different collections; the square document icons are hyperlinks to the source documents. In the case of values representing people and companies, the value (e.g., "Audrey Auditor") is also displayed with a hyperlink that initiates a profile search on that value. This allows, for example, contact information to be found for people who have "HP" as a client. Other headings (e.g., "SEC Filings:" and "Analysts' mentions:") are followed only by document links, as it is the document as a whole that is of interest -- not specific information extracted from it.

3 Extracting Entity Profiles in NX

This section describes the major components of NX that are used to support its profile search capability:

- Semi-automatic field classification.
- Entity normalization.
- Definition of a partial global schema
- Extraction of profile information from entity indexes
- Detection and resolution of profile ambiguity

3.1 Semi-automatic field classification

To build an index of entity references of different types, we must identify where those types occur within collections. NX's field classifier uses field names and sample values from a collection to classify fields as containing entity types (people's names, company names, phone numbers, dollar amounts, etc.) and identifiable semantic *roles* that they play within the collection, e.g., partner on an engagement, client company, or vendor company. The current version recognizes person names, company names, telephone numbers, geographic locations, office names, and dollar amounts. As classification is not 100% accurate or complete, a Web browser interface is provided to alter the entity and role types for each collection's fields.

3.2 Entity Normalization

In a standard relational database, tuples from different tables that contain information about the same entity each contain a *key* for that entity allowing the tables to be joined. In a semistructured document space, however, there are rarely unique keys shared by collections. Rather, entities are referred to within text strings in a variety of formats, with a variety of synonyms and abbreviations.

Therefore, to allow search over entities, entity references must be normalized and matched (as in [HS95]). For maximum retrieval speed, NX normalizes entity references at indexing time. The normalization is heuristic, using formatting knowledge and synonym tables specific to each entity type. NX's entity index stores both the original form and a normalized form of each entity reference. At retrieval time, a normalized form of the user's search string is created and used to retrieve matches from the normalized entity index. In some cases, values are only partially normalized, and the original forms of retrieved matches and the search string are compared to verify the match.

In addition, pre-processing is required to find the portions of the input string containing entity references. Often, a field will contain multiple entity values in a single string, with spurious information interspersed. For example, a typical person name field value might be "Bob J. Smith Jr. – managing partner; Sue Jones, 415-555-1212, Palo Alto." NX's normalization routines extract "Bob J. Smith Jr." and "Sue Jones" out of this field value.

NX's field classification and normalization routines are described in more detail in [HB97].

[1] Actual people names have been replaced in the HTML generated by Notes Explorer to preserve privacy.

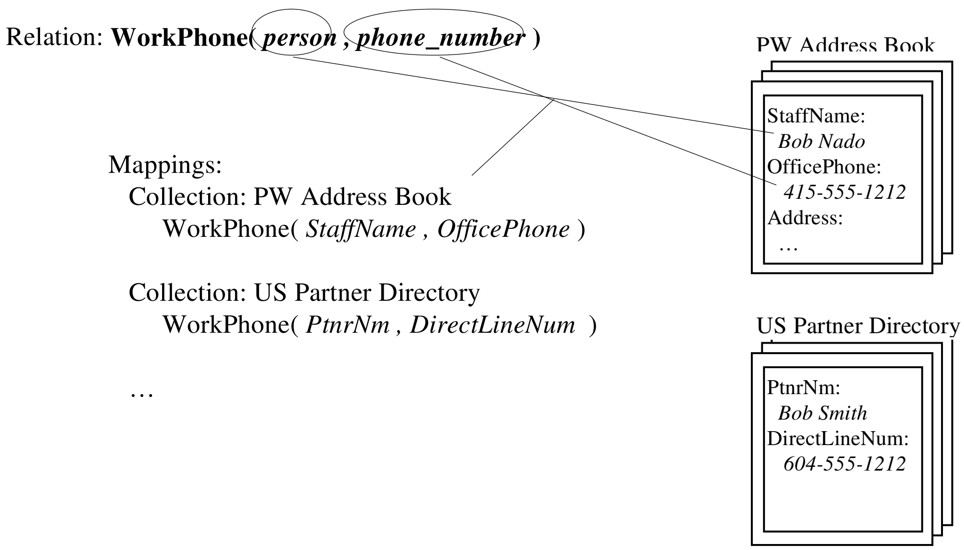

Figure 2: Mapping a Predicate to Collection Fields

3.3 Definition of a Partial Global Schema

The profile search capability of NX is based on a global vocabulary for describing the types of information that may be found about an entity in the different information sources that are available. No attempt is made to define a complete global schema characterizing all of the relations that might be extracted from individual collections. Rather, the global schema used by NX is partial – containing only enough meta-information to support the desired entity profiles. Currently, the global vocabulary includes sorted, binary predicates of two types. An *entity predicate* represents a relationship between two entities. For example, "Work Phone" is an entity predicate representing the relationship between a person and a phone number where that person may be reached at work. Sorts (entity types) are assigned to the domain and range arguments of the "Work Phone" predicate -- "Person" and "Phone Number" -- restricting the applicability of the predicate. The other type of predicate, called a *document predicate*, represents a relationship between an entity and a document that is "about" that entity. For example, "Resume" is a document predicate relating a "Person" and a resume document.

In addition to declaring domain and range sorts, each predicate must be mapped to the relevant fields in collections that locally instantiate the predicate. For example, in the *PW Address Book* collection, the "Work Phone" predicate is mapped to a domain field called "StaffName" and a range field called "OfficePhone". Other collections may also have information relevant to the "Work Phone" predicate but use different fields to record the person name and the phone number (see Figure 2). An entity predicate may be mapped to multiple pairs of domain and range fields in a single collection. Document predicates have a simpler mapping, requiring only a domain field in each relevant collection.

Currently, the mapping of predicates to fields in collections is performed manually using a Web browser interface. The interface narrows the set of candidate collections and fields for each predicate by exploiting the entity types assigned to fields by NX's field classifier. A collection can be ignored when mapping a predicate if does not contain fields with entity types matching both the domain and range sorts of the predicate. Given an eligible collection, candidates for the domain and range fields are narrowed to those whose entity types match the domain and range sorts of the predicate. The interface allows the predicate mapping process to be performed in a small amount of time, typically less than a half hour per collection.

3.4 Extraction of profile information from entity indexes

A profile for a particular category of entity is defined by listing the global predicates that should make up the profile in the order in which they should be displayed in the results page of a profile search. Information can be associated with individual predicates through a Web browser interface to control the formatting, number, and sorting of profile results displayed for the predicates.

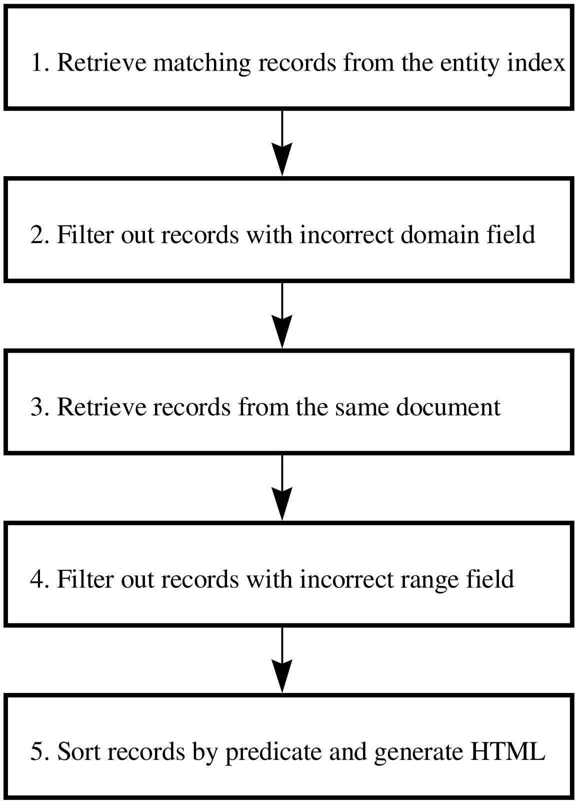

Figure 3: Profile Information Extraction

Retrieving an entity profile involves five steps as depicted in Figure 3. First, NX retrieve records from the entity index whose normalized field values match the normalized forms of the search string and that have the correct entity type. Next, NX filters out records retrieved in Step 1 whose field is not a domain field for any predicate in the profile. For each record *A* resulting from step 2, NX retrieves records from the entity index that originate in the same document. In step 4, NX filters out records retrieved for *A* in Step 3 whose field is not a range field corresponding to *A's* field as a domain field for one of the profile predicates. Finally, NX sorts the remaining records by profile predicate and generates an HTML page displaying the results for each predicate.

Because they have been normalized, the results found for a particular profile predicate can be properly grouped independently of how they were referred to in the source documents. In essence, this uses the normalized entity index as a simple data warehouse, enabling an aggregation over entities in document sets.

3.5 Profile ambiguity

Given a particular search string, NX may find references in documents to more than one distinct entity, each of whose names match the search string. For example, when "Bob Smith" is supplied as the search string, matches may be found that give information about both Robert A. Smith and Robert S. Smith. This problem of ambiguous profile searches can only be addressed heuristically, as entities do not have unique keys across collections.

In some cases, however, reference lists of entities can be used to aid in disambiguation. For person names within our firm, for instance, there are “address books” mapping each person’s name to a unique email address. Generally, a PW staff member will have exactly one entry in one of the address books that exist for each PW firm around the world. If more than one match is found for the search string in the collection of address books, the user is asked to select one of the address book entries in order to refine the search . This is illustrated in Figure 4 for a profile search on “Bob Smith”.

The selected address book entry often gives a more specific search string with which to continue the search. In addition, the address book entry may give other information about the chosen person (such as work office) that may be used to filter out documents that contain conflicting information.

4 Discussion and Future Work

As described in [HB97], we have evaluated NX's entity-based retrieval through a comparison to standard full-text search, finding that it produces much more precise result sets than full-text search for important classes of queries. To date, we have not explicitly evaluated the entity profiling capability. It may be difficult to use traditional IR evaluation metrics like precision and recall over such a large and diverse information space. Rather, we plan to evaluate profiles' usefulness to end users, through user feedback and surveys.

The goal of our work is to provide better information retrieval across a large semistructured space than full-text search, while avoiding excessive meta-information overhead. Our approach is based on observing that in an information space used by a particular organization, important entity types link information together and can be used as a central retrieval cue. This data-driven approach can be contrasted with schema-driven approaches used by multidatabase systems (e.g., [ACHK93]), and similar systems attempting to integrate structured world-wide web sources [LRO96, FDFP95]. In schema-driven approaches, each local schema is mapped to a central global schema, and mapping rules are used to translate between data formats used by different

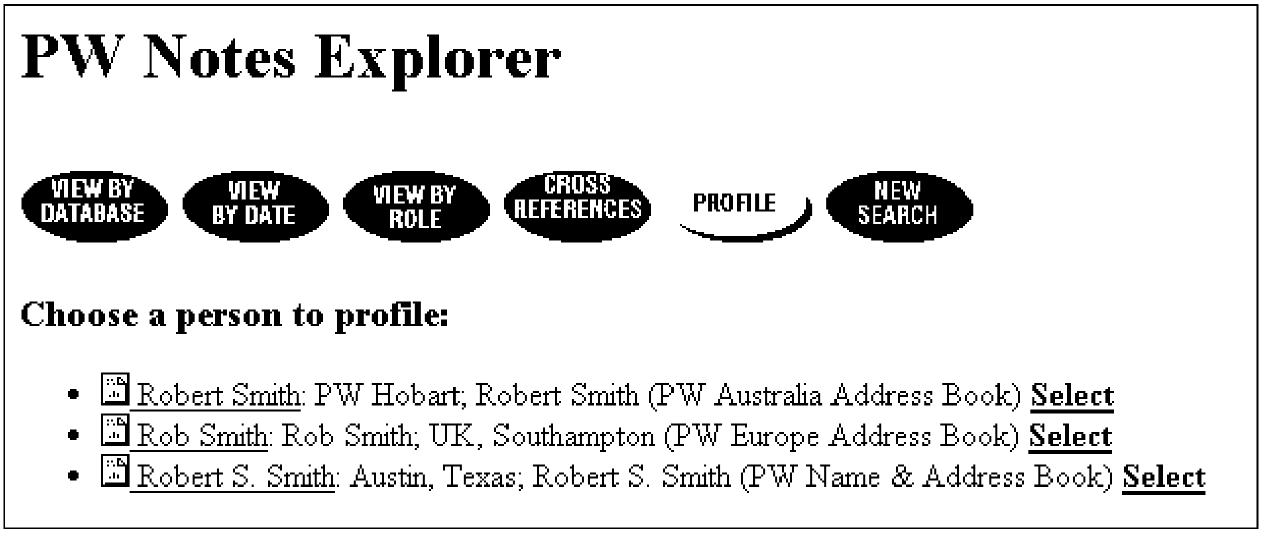

Figure 4: Ambiguous Profile Search

sources (e.g. [CHS91]). These approaches are appropriate for relatively small numbers of tables where the data within each table is well-specified; however, semistructured information spaces can include hundreds of sources, and data even within single sources can have multiple formats. A schema integration phase would be burdensome in such a large space [GMS94]. Instead, NX relies on heuristics to categorize fields into a small number of entity and role types, and normalizes entity values for retrieval. The resulting retrieval system makes it practical to encompass a greater number and variety of data sources than multidatabase systems, although the query language is less general because queries must refer to a specific entity.

Topics to be addressed by future work include:

- extending profiles to other entity types such as service lines and skills,
- customizing profiles to meet the requirements of particular classes of users,
- using information about recency and reliability to resolve conflicts in information retrieved as part of a profile, e.g., multiple office phone numbers retrieved for a person
- performing inference in the determination of profile results that combines information from several documents, e.g., determining a person's office telephone number from his assigned office and that office's main switchboard number
- developing automated techniques for mapping global schema predicates to pairs of collection fields by exploiting abstract classifications of collections, e.g., "directory" collections are more likely to contain a person's phone number; client engagement archives are more likely to contain the names of a person's clients.
- extending the information available as part of a profile by developing additional extraction and summarization methods, e.g., producing a summary of a person's key skills from resume documents

6 Conclusion

Semistructured systems are an intermediate point between unstructured collections of textual documents (e.g., untagged Web pages) and fully structured tuples of typed data (e.g., relational databases). Based on observing how information is typically retrieved and used within our organization, we have developed an entity-based retrieval system over a large semistructured information space. The system incorporates semi-automatic classification of fields, normalization of field values, and structured retrieval of commonly required information in the form of entity profiles. For typical queries containing entities, the system provides much more focused and normalized retrieval than full-text search.

References

[ACHK93] Y. Arens, C.Y. Chee, C.N. Hsu, and C.A. Knoblock. Retrieving and integrating data from multiple information sources. *Intl Journal on Intelligent and Cooperative Information Systems,* 2(2):127-158, 1993.

[CHS91] C. Collet, M.N. Huhns, and W. Shen. Resource integration using a large knowledge base in Carnot. *IEEE Computer*, pp. 55-62, December 1991.

[FDFP95] A. Farquhar, A. Dappert, R. Fikes, and W. Pratt. Integrating information sources using context logic. In *Working Notes of the AAAI Spring Symposium on Information Gathering from Heterogeneous Distributed Environments.* AAAI, 1995.

[GMS94] C.H. Goh, S.E. Madnick, and M.D. Siegel. Context Interchange: Overcoming the challenges of large-scale interoperable database systems. In *Proceedings of the 3rd International Conference on Information and Knowledge Management.* 1994.

[Guh97] R.V. Guha. Meta Content Framework: A Whitepaper (Draft). Apple Computer. Available at URL http://mcf.research.apple.com/wp.html, 1997.

[HB97] S. Huffman and C. Baudin. Toward Structured Retrieval in Semi-structured Information Spaces. In *Proceedings of the 1997 International Joint Conference on Artificial Intelligence,* 1997.

[HS95] S. Huffman and D. Steier. Heuristic joins to integrate structured heterogeneous data. In *Working notes of the AAAI Spring Symposium on Information Gathering in Heterogeneous Distributed Environments*. AAAI, 1994.

[LRO96] A. Levy, A. Rajaraman, and J. Ordille. Query-answering algorithms for information agents. In *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI-96)*, pp. 40-47, 1996.