GTE SuperPages: Using IR Techniques for Searching Complex Objects

Steven Whitehead
swhitehead@gte.com

Himanshu Sinha
hsinha@gte.com

Michael Murphy
mmurphy@gte.com

GTE Laboratories Incorporated
40 Sylvan Road
Waltham, MA 02254
USA

Abstract

The yellow pages service of GTE SuperPages enables Web users to flexibly search through listings of 11 million businesses in over 17000 categories. To achieve the flexibility desired it uses an Information Retrieval (IR) engine to search through complex listing objects. The objects themselves are stored in an object database. The use of the IR engine enables us to create an index that spans all the components of a complex object.

1 Introduction

1.1 The Scope of SuperPages Service

SuperPages is an interactive directory service operated on the World Wide Web by GTE Directories Corporation. The SuperPages site provides five different types of directory services: a yellow pages directory, a consumer product guide, a web-site directory, a classified advertising service, and a series of local city guides. Each of these services is a full-fledged directory in its own right and each receives a significant amount of traffic. However, for the purposes of this paper, we shall restrict our attention to the yellow pages portion of SuperPages and use the term SuperPages to refer to the yellow pages directory alone.

The yellow pages directory of SuperPages is national in scope and provides listings for over 11 million U.S. businesses in more than 17,000 categories. Since its launch in January of 1996, traffic to the site has increased steadily and SuperPages is currently one of the top yellow page directories in the world, serving tens of thousands of users daily.

The service enables users to locate businesses based on name, location, business category, phone number, and/or product and service keywords. The information it provides about each business is generically called the business' "listing". A listing includes the business' name, address, telephone and fax numbers, Internet contact information, product and service information, advertisements, etc. Search results provide listing information and also provide access to maps and driving directions.

1.2 The Schema of SuperPages Data

The data model for SuperPages is fairly complex. Each listing can have the following information associated with it:

- a business name, plus multiple aliases
- multiple physical addresses
- multiple telephone numbers
- multiple URLs
- multiple email addresses
- multiple business categories
- multiple product and service keywords
- mapping information
- advertising content (html and multimedia objects)

**Proceedings of the 23rd VLDB Conference
Athens, Greece, 1997**

- other miscellaneous control information

SuperPages also employs an elaborate network of synonyms and aliases for cities, states, and business categories. The end result is that each logical listing is defined by a fairly complex network of objects.

1.3 The Search Requirements of SuperPages

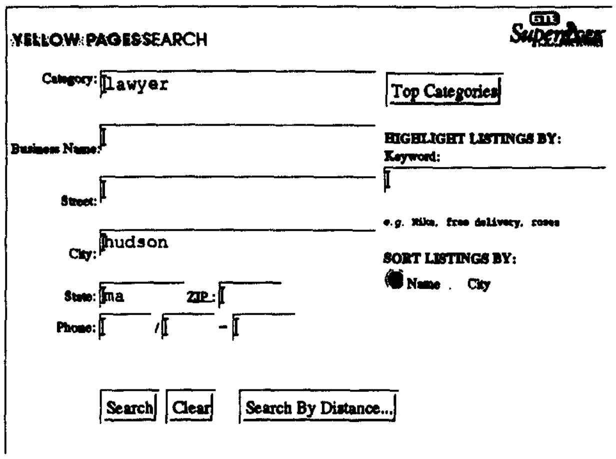
The goal of any electronic yellow pages service is to provide end-users with fast, easy access to relevant business listings. To achieve this goal, the directory must provide a query mechanism that is not only flexible and efficient, but also semantically robust with respect to the user's intended search. In SuperPages flexibility is achieved by allowing users to specify searches based on any combination of business name, location, category, phone number, and/or product and service keyword. To achieve semantic robustness, the SuperPages search engine supports the following features:

- token matching, as opposed to string matching
- boolean logical operators
- pattern and wildcard matching
- word stemming
- relevance ranking
- field/region-specific searches
- context-specific synonyms (for cities, states, categories, and business names)
- mixed-type searching (e.g., text, numeric, range, date-time)

Figure 1 shows the input form used for searching SuperPages. Figure 2 shows a sample output listing.

1.4 Use of Information-Retrieval Methods to Search Complex Data

Our need for fast, text-oriented searches combined with the size and complexity of our data set makes the native query processors in existing commercial databases look unattractive. Instead, SuperPages employs a technique based on information-retrieval technology. The essence of the approach is to map logical units of the database (e.g., listing objects and their associated sub-object networks) into an indexed collection of structured text documents. These documents are then searched using an IR search engine instead of the database's native query processor. The approach enables us to index and search across complex articulated object structures. The result is a query processor that is both robust and efficient. The remainder of the paper describes our approach in detail. We present an analysis of the strengths and weaknesses of the approach.



YELLOW PAGESSEARCH

Category: lawyer

Business Name:

Street:

City: hudson

State: ma ZIP:

Phone: / -

HIGHLIGHT LISTINGS BY:
Keyword:

e.g. Nike, free delivery, roses

SORT LISTINGS BY:
● Name . City

Figure 1: Input form for searching SuperPages

2 The SuperPages Architecture

2.1 Architectural Overview

The basic structure of the SuperPages architecture is shown in Figure 3. The architecture has two major components: an Interface Manager and an Application Database. The Interface Manager is responsible for generating and managing views into the SuperPages database — primarily web access. The Application Database is the repository for all listing data. Through an application specific API, the Application Database provides the set of search, retrieval, and update services used by the Interface Manager.

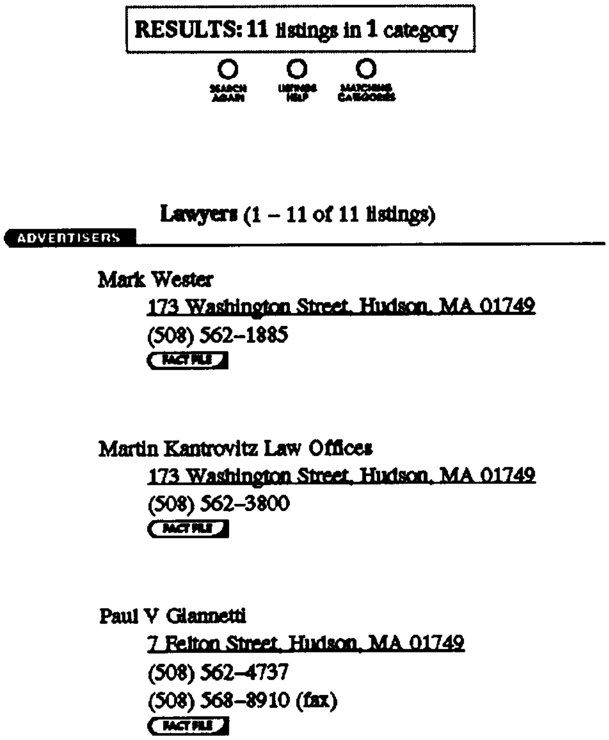
The Application Database has three main components: an IR search engine, an ODBMS, and system integration code. Both the IR search engine and the ODBMS are commercial off-the-shelf products (The IR engine is Verity's VDK engine, and the ODBMS is Object Design's ObjectStore database). The integration code is custom software that implements a gateway between the IR engine and the ODBMS and presents a coherent application API to the Interface Manager.

2.2 The Object Database

The ODBMS is used only as a persistent object repository. It contain all listings and their related sub-object networks. An object database was selected for the data repository because

- it allowed for the storage of native programming language objects (C++), obviating the need for mapping code
- it provided efficient reference (pointer) based access to listing objects.

The query processing capabilities of the ODBMS are not used.



RESULTS: 11 listings in 1 category

Lawyers (1 – 11 of 11 listings)

Mark Wester
173 Washington Street, Hudson, MA 01749
(508) 562–1885

Martin Kantrovitz Law Offices
173 Washington Street, Hudson, MA 01749
(508) 562–3800

Paul V Giannetti
7 Felton Street, Hudson, MA 01749
(508) 562–4737
(508) 568–8910 (fax)

Figure 2: A sample output listing from SuperPages

2.3 The IR-Engine

Virtually all searching in SuperPages is accomplished through the IR-engine. The IR-engine provides the core set of text-retrieval capabilities required by SuperPages. This includes: word matching, pattern matching and wildcards, stemming, relevance ranking, and mixed mode searching (text, numeric, range, date). The IR engine is used to identify the listing objects in the database which satisfy a user's query.

2.4 Indexing complex object structures using an IR-engine and proxy documents

In SuperPages, users principally search for business listings. In our data model (and in our database) each listing is represented by an articulated object structure. A simplified version of our data model is shown in Figure 4. To search these object structures, SuperPages uses a technique based on an IR-search engine and proxy documents. The essential idea is to map an articulated object structure into a hierarchically structured text document. This document is then used as a proxy for the object-structure it represents. Once a proxy document has been generated it is indexed and logically inserted into a document collection that can subsequently be searched using standard IR-techniques. [1] In SuperPages, SGML is used to provide the necessary structure to the proxy text documents. A simplified example of a proxy document is shown in Figure 5. Figure 6 shows how some of the SGML tags used in Figure 5 correspond to the objects of Figure 4. Most of the arrows in Figure 4 represent aggregation. Some (for example, the one pointing to "categories") represent reference. The indexing technique can be used for both types of association.

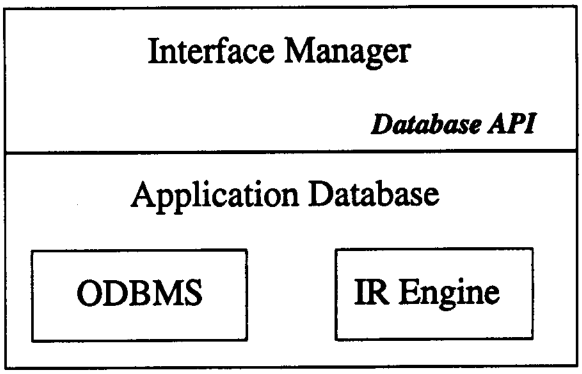


Figure 3: The architecture of the SuperPages server

2.5 Search and Retrieval of Complex Object Structures

Search and retrieval in SuperPages is accomplished through a coordinated effort involving the IR-engine and the ODBMS. Searches are accomplished as follows:

1. Input is collected from the user and transformed into a query that is submitted to the IR-engine.

2. The IR-engine returns a sorted list of matching "proxy documents".

3. Database object identifiers are extracted from the matching proxy documents.

4. The object identifiers are used to retrieve matching objects from the database.

3 Discussion

3.1 Alternative Designs

Several architectures were considered before selecting the current one. Designs based on RDBMS systems were considered unattractive because

[1] An essential requirement of this approach is that the IR-engine support searching over structured documents. This is a standard feature in most commercial IR-engines.

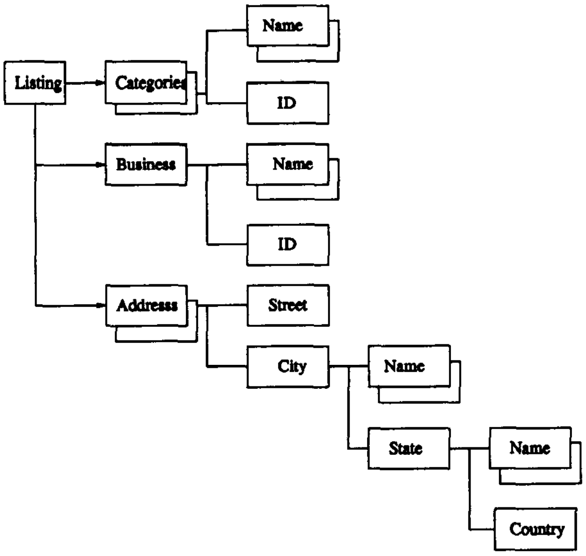


Figure 4: A listing in SuperPages is defined by a fairly complex network of articulated objects.

1. at the time, there was very limited integrated support for text-retrieval,
2. highly normalized database designs were too slow — requiring at least nine joins over large relations to retrieve listings, and
3. denormalized schema designs, though more efficient, were too difficult to maintain.

Designs based solely on the ODBMSs were dismissed because their native query processors lack the speed and flexibility needed by SuperPages. They lack sufficiently powerful IR-search capabilities. Furthermore, the ODBMS chosen could not index across complex articulated objects.

Finally, a design based solely on the use of an IR-engine was considered. In particular, an IR engine could have been used without an object database. However, this would have required us to store listing data in the file system. Coupling the IR-engine with an ODBMS permitted transactional updates to the database and simplified implementation by providing direct support for persistent C++ objects.

```
<L>
<ID>891473027</ID>
<B><BN>Lanning, Fay, Getchius & Associates</BN>
   <BN>LFG & Associates</BN>
   <BI>98987674</BI>
</B>
<H><SIC>17520447704</SIC>
   <HN>Attorneys</HN>
   <HN>Lawyers</HN>
</H>
<PA><ST>4012 Mendon Road</ST>
   <C><CN>Marlborough</CN><CN>Marlboro</CN>
   <S><SN>RI</SN><SN>Rhode Island</SN>
      <CO><CON>US</CON>
   </S>
   </C>
   <Z><Z5>02216</Z5><Z4>0023</Z4></Z>
</PA>
<VT><CC>1</CC><AC>508</AC>
   <X>555</X><LN>1212</LN></VT>
<FT><CC>1</CC><AC>508</AC>
   <X>555</X><LN>1213</LN></FT>
<EM>info@lfg.com</EM>
<URL>http://www.lfg.com</URL>
<KVP><KW>Services</KW>
   <V>Corporate Business</V><V>Criminal Justice</V>
   <V>Estate Planning</V><V>Family Law</V>
   <V>Insurance</V><V>Real Estate</V>
</KVP>
</L>
```

Figure 5: A simplified example illustrating the SGML structure of a listing's proxy document.

3.2 Strengths of our approach

The most salient benefit derived from our use of IR technology is fast and flexible searches over a large and complex listing space. The IR-engine provides support for word matching, pattern matching and wildcards, stemming, and mixed mode searching (text, numeric, range, date). Users can search over any combination of nine fields. The IR engine also provides for free-format "relevance ranked" searches. Relevance ranked searches enable users to find and order listings based on the number of matching keywords, rather than exact Boolean expressions. Another benefit of our approach is support for localized synonyms and aliases. For example, in the context of the category field, SuperPages treats the word "lawyer" as a synonym for the word "attorney". Localized synonyms are achieved by "rolling out" synonym terms into the appropriate fields of the proxy document generated for each listing. So, for example, the SGML describing the category information associated with an attorney's listing might look like this:

```
<H>
<SIC>01134848<\SIC>
<HN>Attorney<\HN>
<HN>Lawyer<\HN>
<\H>
```

The same technique is applied to define synonyms for cities and states.

3.3 Weaknesses of our approach

3.3.1 Loose integration

Integration of the IR engine and ODBMS is poor. This results in several drawbacks. First, there is no sim-

Object	SGML Tag
listing	<L>
business	<B>
business name	<BN>
physical address	<PA>
city	<C>
state	<S>
zip	<Z>
email	<EM>
voice telephone	<VT>
category	<H>

Figure 6: Correspondence between objects and SGML tags

ple way to update the IR-indices and database objects atomically. As a result, it is possible to have an update to an index succeed but to have the corresponding update to the database fail. The lack of an agreement protocol, e.g., two-phase commit, between the IR engine and the ODBMS forces us to checkpoint index-database pairs before updating them — so they can be rolled back to a consistent state if an update fails. Since creating a checkpoint is slow, we are restricted to batch updates only. Second, weak integration results in a large amount of data duplication between the index and the object database. This hurts performance because the same data has to be brought into memory twice.

3.3.2 Non-Text Searches

A peculiarity of the IR-engine we used in our implementation is that it maintains two distinct representations of the source documents: a searchable word index and a set of internal fields derived from the source document. The only operations available on the word index are token matches, phrase (sequence-of-token) matches, and substring matches ("wildcards"). A richer set of data representations (e.g., fixed and floating point numbers, in addition to strings) is possible with the internal field representation, along with a richer set of search operations (e.g., numeric equality and inequality operators). Unfortunately, searches over internal fields are slower than searches over the word indices.

4 Conclusion

In this paper we described an approach to indexing and searching complex articulated object structures based on IR-technology and the use of structured proxy documents. This approach enabled us to provide high performance access to a large database of complex objects. The main limitation of our implementation is the loose coupling between the database and the IR engine. Most RDBMS vendors have, by now, incorporated IR capabilities into their product offerings. However, these solutions only allow indices to be created over individual columns within a table. Our approach, on the other hand, can be used to create indices that span multiple columns and even multiple tables. In spite of its limitations, our solution, implemented using off the shelf products, has been quite effective. It has also scaled well to an object database of over 5 GB containing millions of objects. We look forward to the incorporation of this approach in commercial DBMS offerings.