ARTICLES

Harvesting Chemical Information from the Internet Using a Distributed Approach: ChemXtreme

M. Karthikeyan,* S. Krishnan, and Anil Kumar Pandey

Digital Information Resource Center Information Division, National Chemical Laboratory, Pune 411008, India

Andreas Bender†

Unilever Centre for Molecular Science Informatics, Department of Chemistry, University of Cambridge, Lensfield Road, Cambridge CB2 1EW, United Kingdom

Received August 16, 2005

The Internet is a comprehensive resource of chemical information which is at the same time largely unstructured. It provides a wealth of scientific information such as experimental data and requires a suitable automated data mining and analysis tool for its meaningful exploration. The Java based software presented here, ChemXtreme, is developed for harvesting chemical information from the Internet employing the Google API in combination with a distributed client/server text analysis architecture based on JavaRMI. It represents the first and until now the only toolkit for automated structured data retrieval from the Internet which is itself open source. ChemXtreme employs the "search the search engine" strategy, where the URLs returned from the search engine are analyzed further via textual pattern analysis. This process resembles the manual analysis of the hit list, where relevant data are captured and, by means of human intervention, are mined into a format suitable for further analysis. ChemXtreme on the other hand transforms chemical information automatically into a structured format suitable for storage in databases and further analysis and also provides links to the original information source. The query data retrieved from the search engine by the server is encoded, encrypted, and compressed and then sent to all the participating active clients in the network for parsing. Relevant information identified by the clients on the retrieved Web sites is sent back to the server, verified, and added to the database for data mining and further analysis. The distributed further analysis of URLs in a client/server architecture scales very favorably, thus producing only minimal overhead.

1. INTRODUCTION

Quick and reliable access to appropriate sources of data is crucial in all areas of science, as for example in chemistry and life sciences. Due to patenting issues in particular in the medicinal and pharmaceutical industries information needs to be located in the shortest time possible. If at hand, dedicated databases such as the MDL Drug Data Repository[1] (MDDR) are usually the first choice for information retrieval since the information therein can be trusted the most and it is readily indexed. On the other hand, it should be kept in mind that the introduction of errors is possible at any stage of the database curation process, be the errors experimental, occurring during transcription of data from journals or patents or at any other stage. Still, carefully curated databases should allow some degree of consistency checking in order to maximize the trustworthiness of data contained in them.

Another option for locating relevant data is to employ information retrieval tools based on traditional keyword searches, such as Google[2] or PubMed,[3] which are able to retrieve the most relevant documents containing query *keywords*. Keyword-based search tools show severe limitations both with respect to the query that can be employed as well as the output format of retrieved results. As an example, the keywords "melting point" and "water" will retrieve a very large number of documents in which those words are present (which is 642 000 as of June 2005, employing the Google search engine). Still, to investigate what the melting point of water is manual analysis of the search results is required. Keyword-based search engines thus only retrieve a (possible) *location* of the desired data (here the URL), *not the data itself* (here the melting point of water). Various enhancements to the basic keyword search paradigm were proposed which include for example query expansion, where keywords are augmented by related keywords to narrow or broaden the search by employing the Boolean operators AND, OR, and NEAR. Another option is to cluster results for easier browsing, as performed by some search engines such as Accumo.[4] Also the application of structured queries has been reported[5] which promises to return more meaningful

* Corresponding author phone: +91 (20) 2590 2483; fax: + 91 (20) 258 93 355; e-mail: m.karthikeyan@ncl.res.in.

† Current address: Discovery Technologies, Lead Discovery Informatics (LDI), Novartis Institutes for BioMedical Research Inc., 250 Massachusetts Ave., Cambridge, MA 02139.

10.1021/ci050329+ CCC: $33.50
Published on Web 02/10/2006

search results. In structural queries, both the relative location of search terms to each other (such as "same sentence") and their structural relationship (such as "interacts with") can be defined by the user. Still, the basic proposition remains which is the input of keywords and the retrieval of documents in which the keyword is found, with subsequent *manual* analysis of the results retrieved.

Automated information extraction, on the other hand, is based on an understanding of the structure of natural language and its automated processing in order to extract relevant information. Algorithms of this category are rapidly gaining popularity with the advent of information in electronic formats. An automated experimental data checker[6] (OSCAR) was developed which is able to extract experimental data (such as NMR spectra and physicochemical data) from organic chemistry manuscripts. OSCAR alerts the user in case inconsistent or incomplete data are found. Consistently over 80% recall as well as over 80% precision was achieved over a wide range of NMR spectra data, elemental analysis, and MS data as well as melting points.[7] ChemDig[8] represented an indexing engine of various kinds of molecular information across a broad range of formats. Given that about 40 structural formats are encountered on the Web today defining parsers for these data represented a formidable challenge. Mining protein−protein interactions from Medline abstracts[9,10] was able to retrieve a significant portion of the cell cycle network of *Drosophila*. From the same source also biological terms associated with different gene expression levels could be identified,[11] leading to a quality of annotations which is comparable to the quality achieved by manual, labor-intensive annotation. Directly relevant to drug discovery is the EDGAR approach[12] which is able to extract drug−protein interactions from biomedical literature. More recently also commercial information mining approaches such as the ontology-based interactive information extraction (OBIIE) system have been reported,[13] which was as a case study employed to compile enzyme cofactors from EMBASE and MEDLINE. A recent review deals with information extraction in the realm of molecular biology,[14] and it includes a historical overview of the field. A current introduction[15] outlines general concepts in text mining, while a more comprehensive overview[16] summarizes the most relevant approaches as well as a number of applications.

The most obvious data source for knowledge extraction today is, due to its sheer size, the World Wide Web (WWW). Based on the capabilities it provides, the information on cancer provided by various governmental organizations was mined by Hopfield networks and Self-Organizing Maps (Kohonen networks), providing improved navigation of information.[17] A distributed Web crawler design for data mining has been proposed[18] which at peak times involved up to 100 nodes; acknowledging the superior crawling power that Google provides in the work presented here no indexing of Web sites was performed from our side. Most relevant to the work presented here Pichl et al.[19] present the "Networked Mining of Atomic and Molecular Data from Electronic Journal Databases on the Internet"; while the context is scientific as well, attention was only paid to journal databases, and no distributed analysis of documents was performed.

With respect to more chemically oriented information, the amount provided by the Internet is growing rapidly, and the advancement of not only open access initiatives such as the Budapest Open Access Initative[20] but also specifically chemical initiatives such as ChemBank,[21] Chemical Entities of Biological Interest[22] (ChEBI), and the World Wide Molecular Matrix (WWMM)[23] indicate that this trend will continue. In addition to mainly biological data, movements such as the Southampton Crystals Reports[24] provide structural data in an open access manner to the scientific community, also strongly emphasizing the importance of metadata interfaces such as the Open Archives Initiative.[25] There are already various academic institutions and organizations indexing such information, for example from Ph.D. theses, such as the National Chemical Laboratory, Pune.[26]

Still, extracting the reusable information remains difficult. This is mainly due to the nature in which chemical information is usually presented, as chemical structures, but furthermore different nomenclatures and data formats exist which are difficult to analyze, for example chemical structures in graphical formats such as GIF.

Currently the search engine Google[2] is an obvious choice for searching any information located on the Internet since it currently indexes about 8 billion Web pages (8 058 044 651 is the official number of pages indexed, as of July 1, 2005), which is more than any other search engine. In the current work the ranking mechanism is implicitly accepted since only the top URLs for the queries are retrieved only. Depending on the information required, different ranking mechanisms might turn out to be more suitable in the future. However care has to be taken in finding a suitable query input format. Chemical structures are no suitable query structure for Google since a keyword-based input format needs to be employed. Trivial names of chemicals are not ideal either since for most compounds multiple trivial names are known. This fact is in some cases also true for 'systematic' chemical nomenclature. For the reasons presented, in this work the limited CAS identifier[27] is employed to identify substances. The CAS identifier is a ubiquitous form, which is used to identify chemical substances uniquely, distinguishing for example between stereoisomers and different counterionic forms. Shortcomings of the CAS format include that polymers differing only in chain length or molecular weight may not be differentiated. Also in some cases CAS numbers were assigned to mixtures of compounds without specification of the compound ratio, leading to nonunique identifiers.

In the future it is likely that molecular representations such as the Google-searchable IChI/INChI-format[28,29] will gain importance, which represents molecular structures in the form of a unique string amenable to text-based database querying. Databases such as PubChem[30] as well as publishers such as the Royal Society of Chemistry, among others, are current users of this molecular format. Still, currently the INChI representation is not (yet) widely enough used for this purpose due its comparative recentness. If in the future more Web pages are going to contain both INChI identifiers and associated data a subsequent study, based on INChI-queries, would be a suitable follow-up study of the work presented here.

After retrieval of relevant Web sites via Google, pattern analysis needs to be performed on the retrieved Web pages. In the work presented here, this analysis step is performed using a client/server architecture employing the JavaRMI[31] environment. Details of this open-source distributed comput-

ing environment were presented in the first paper of the series.[32] Reasons for employing JavaRMI as well as computational details are also presented in the Material and Methods section. Earlier work employing the JavaRMI in the computational chemistry context[33] focused on the authentication of Internet-based computing resources. In addition, three applications were presented: COS, which is a database application based on JavaRMI, MoldaNet, which invokes Java3D to create a molecular visualization tool, and JSpec, which delivers analytical spectral data.

After retrieval of relevant Web pages via the Google API, text parsing is performed, after which information relevant to the user query is displayed to the user and/or stored in a local database.

The work presented here is based on the single-machine ChemReader approach which also employs the Google API for information retrieval, with the additional distributed analysis of retrieved Web pages employing a JavaRMI client/server architecture.[32] Due to the distributed approach the time requirements for analyzing retrieved Web pages shrinks enormously (practically inversely linear with the number of clients employed for text parsing). It should be noted that this gain in processing power can be obtained without any additional cost, by just using empty cycles of employees' desktop PCs during the night or times of little computational load.

All harvested data are stored in a local database for free access over the Internet for the benefit of the academic community. For security reasons and in order to avoid bandwidth congestion the well-known LZW or ZIP compression method and a custom-built encoding method are implemented in the system. To facilitate performance analysis, timestamps are employed at every step where data are sent or received. Further details are given in the following Material and Methods section.

In the following sections, we will first outline computational details such as the information flow in section 2. Sample results are presented and discussed in section 3, for both biological and physical data mined from the Internet, while conclusions and further work are summarized in section 4.

2. MATERIAL AND METHODS

The ChemXtreme environment is implemented in a JavaRMI[31] based distributed architecture which is powered by the Google API for locating information sources in the first place. We will now explain the two broad components of the ChemXtreme system, which are, first, its Google API-based retrieval of relevant URLs and, second, its text parsing component. A flowchart of the system is shown in Figures 1 and 2.

The Google Web Application Programming Interface (Google API), which requires an access key for its utilization, enables the user to query 8 billion documents from its database on a limited basis and to retrieve hits automatically. Limitations are given with respect to the number of queries per day, which is currently restricted to be 1000, and the number of retrieved documents from each hit list, currently restricted to 10. The API employs the standard Web Services Description Language[34] (WSDL) and the Simple Object Access Protocol[35] (SOAP), allowing for a variety of programming languages on the client side. In the work presented here Java is employed for querying the Google API. The reason to employ the Google API instead of querying the Web interface is that it represents the only way officially approved by Google to query its databases on an automated basis since it produces less computational overhead. While this poses restrictions on the data retrieved, we expect those restrictions to be gradually alleviated in the future. The way Google handles access to its API is also an additional reason to employ a parallel architecture, as described in this work. Since only a certain number of queries per time unit is allowed *per client*, the total number of queries sent can be increased by employing a parallel architecture for this step.

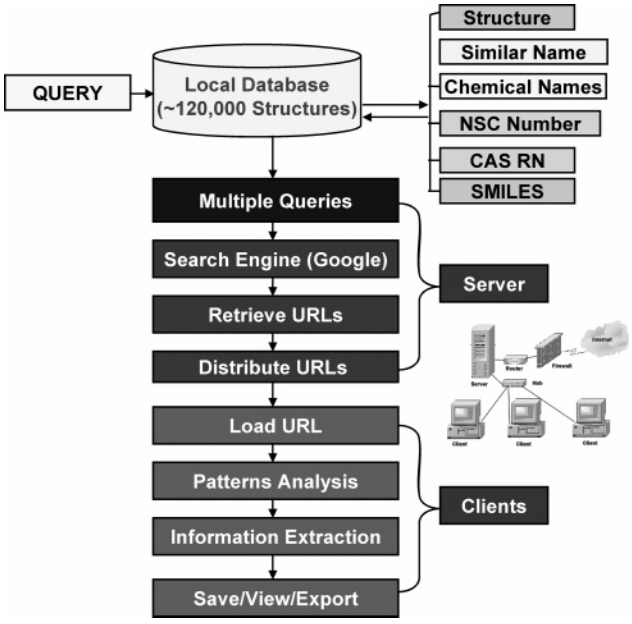


Figure 1. Flowchart of the ChemXtreme approach. Queries are preprocessed and, if present, the chemical structure additional information corresponding to the text query is retrieved from a local database. Next, the Google API is used to retrieve additional information about the compounds of interest from the Web. The textual content of the retrieved Web sites is analyzed further in a distributed environment. Finally, results are displayed to the user for further analysis.

After retrieving URLs relevant to the query via the Google API, every Web page contents retrieved from the URLs is analyzed by Java regular patterns for associated molecular properties. Since there are no universal abbreviations for molecular properties, flexible queries have to be used to extract as much information as possible, avoiding at the same time the extraction of false positive information. For example, the molecular property "melting point" may be abbreviated m.pt, m pt, melt point, melting point, melting pt, and so on. Some properties such as boiling points may also be given in different units such as C (centigrade) or F (Fahrenheit). To retrieve most of the unstructured (free text) information from Web pages, we designed regular expression patterns for the most common physicochemical and biological properties often cited in chemical documents. Those patterns are fully user-definable and may be easily adapted to additional properties. Currently used patterns are given in Table 1.

URLs retrieved from Google are distributed to a number of clients to perform text analysis in a distributed fashion. A detailed description of the communication between client-side and server-side necessary for this purpose is given in

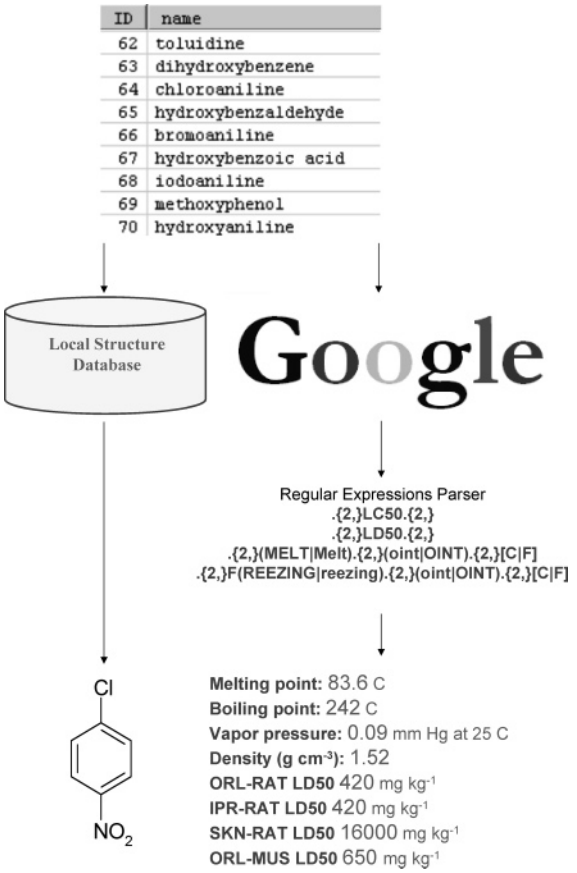


Figure 2. Flowchart representing the 2-fold analysis of the query entered by the user: Structural information about the compound of interest is retrieved from a local database, while additional information about the compound is retrieved from the Internet via a search engine, here Google. Both results are finally shown to the user.

Figures 3 and 4. Figure 3 shows the ChemXtreme communication process between server, Google API, and an (in principle unlimited) number of clients on a coarse-grained level. Figure 4 shows the invoked processes on the server side and the client side, respectively, in a more detailed fashion. A detailed publication on the distributed computing environment using the JavaRMI was presented earlier.[32]

The general communication between the user, server, and clients proceeds as follows (Figure 3). The user enters a query into a Web form which is then sent to the server (1). The server forwards the query to the Google API (2) and retrieves the most relevant URLs from it (3), as ranked by Google-internal algorithms. Parallel to submission to Google, the internal database is queried (2) to provided synonyms, CAS ID, and additional information such as corresponding INChI identifiers. The URLs are encrypted and distributed over the network to the clients (4), which decrypt the URLs and retrieve the URL contents via their individual connections to the Internet (5). Client-sided text parsing is performed (6), upon which the results are sent back to the server (7). Data are validated and stored in a storage medium (8), and finally results are displayed to the user (9). As an alternative route, and if each individual client is supplied with a valid Google API key, it is also possible to distribute the queries themselves to the clients. They will then independently query the Google API, retrieve a list of URLs, and perform the text parsing. In any case, results will finally be sent back to the server and stored in the local database as well as displayed to the user.

Distinguishing on a finer scale between steps required on the server and on the client side the following steps are necessary to perform distributed analysis of Google API-retrieved results for a query given by the user (see Figure 4). First, on the server side the number of available clients on the LAN is determined which also need to run the required client software. On the server side, queries are retrieved from the user, and via the Google API the 10 highest ranked Web pages are retrieved for each query and inserted into the database. From the total set of URLs retrieved from the Google API a set of input URLs for each client is constructed. This input list of URLs is encrypted and LZW or ZIP-compressed for safe and reliable information distribution over the network. Threads are then generated on the server for each client, and the encrypted and compressed data generated in the previous step (basically, a list of URLs and some additional data) is distributed to the clients. On the client side the received data is decompressed, decrypted, and processed. Processing in this context means that first the actual contents are retrieved from each URL of its assigned list. Alternatively, if each client is supplied with a valid Google API key, the list of queries are distributed to the clients which independently query the Google API and retrieve the contents of the relevant URLs. Next, pattern matching is performed on the client side for each of the retrieved documents. Pattern matching results are encrypted on the client side, compressed, and then sent back to the server within a certain time limit. The server finally decompresses and decrypts the results received from the clients, checks the results for validity, and stores them in a permanent database.

During the whole process timestamps and information package identifiers are employed in each communication task in order to detect failure of clients or failure of the network. If results are not retrieved from the clients within a set time limit, tasks are reassigned to different clients until complete analysis of the URL data is achieved.

In this work JavaRMI is employed to distribute the searching workload over multiple machines. While Java for example supports sockets for this purpose, which are in general a flexible and sufficient method, sockets require the client and the server to employ applications-level protocols for message exchange. The design of protocols of this level was found to be rather cumbersome. An alternative to sockets are Remote Procedure Calls (RPCs) which provide the programmer with the option to call remote procedures in the same way local procedures are called, when in fact the arguments of the call are transmitted to the (remote) target of the call. The problem with RPC calls is that they do not work well in distributed object settings, where communication between *program*-level objects is needed, which reside in different address spaces. To provide appropriate semantics of object invocation distributed object systems require remote method invocation, RMI, which is exactly what is provided via JavaRMI. With JavaRMI a local stub object manages the invocation of the actual remote object. The JavaRMI system takes advantage of the Java virtual machine and thus also employs the Java object model in most cases. In addition JavaRMI provides very flexible reference styles to remote objects such as persistent and nonpersistent references,

Table 1. List of Regular Expressions Employed for Extracting Physicochemical Properties from HTML Pages

regular expression	range and sample harvested data	details of regular expression
[0−9]{2,10}-[0−9]{2}-[0−9]{1}	50-00-0, 123456-78-9	CAS registry number
.{2,}LC50.{2,}	LC50	biological activity
.{2,}LD50.{2,}	LD50	toxicology data
.{2,}(MELT|Melt).{2,}(oint|OINT).{2,}[C|F]	melting point	melting point (C/F)
.{2,}F(REEZING|reezing).{2,}(oint|OINT).{2,}[C|F]	freezing point	freezing point (C/F)
.{2,}B(OIL|oil).{2,}(oint|OINT).{2,}[C|F]	boiling point	boiling point (C/F)
.{2,}V(APO|apo).{2,}(ressure|RESSURE).{2,}	vapor pressure	vapor pressure+EOL
.{2,}D(ensity|ENSITY).{2,}	density	density+EOL
(Physica|PHYSICA).{2,8}S(tate|TATE).{2,50}	physical state	physical state+EOL
A(ppearance|PPEARANCE).{2,50}	appearance	appearance+EOL
O(dor|DOR).{2,50}	odor	odor+EOL
E(vaporation|VAPORATION).{2,50}	evaporation	evaporation+EOL
V(iscosity|ISCOSITY).{2,50}	viscosity	viscosity+EOL
D(ecomp|ECOMP).{2,10}T(emp|EMP).{2,50}[C|F]	decomposition temperature (C/F)	decomposition temperature (C/F)
S(olubility|OLUBILITY).{2,50}	solubility	solubility+EOL
Speci.{2,10}(g|G)ravity.{2,50}	specific gravity	specific gravity+EOL
S(ynony|YNONY).{2,50}	synonym	synonym+EOL

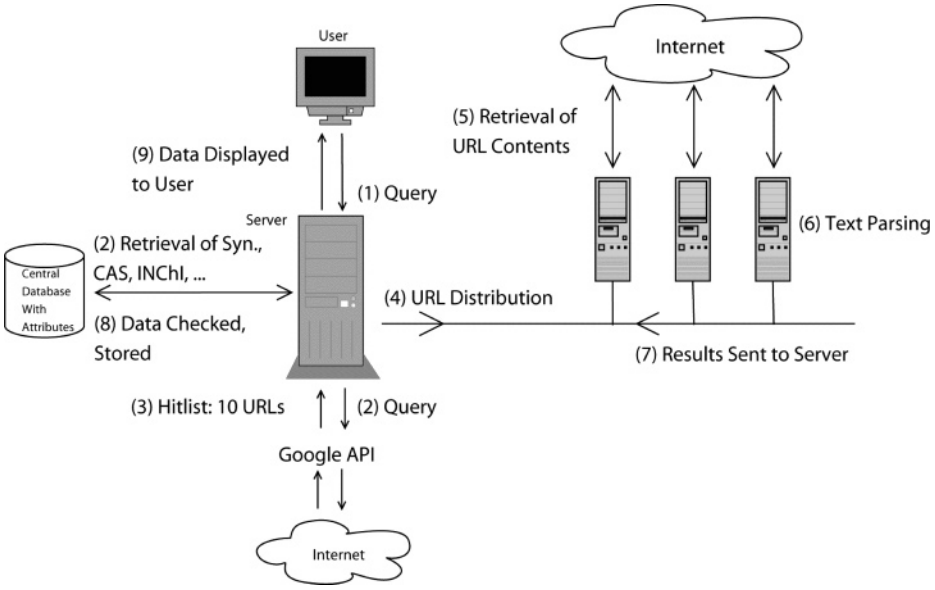


Figure 3. ChemXtreme communication process between server, Google API, and an (in principle unlimited) number of clients on a coarse-grained level. Each communication step is checked for possible errors, such as software, network, or hardware failures.

among others. Overall, JavaRMI was chosen to provide a very flexible and efficient way of providing distributed searching tasks in a client/server architecture.

ChemXtreme is additionally able to translate a number of about 400 000 CAS identifiers or chemical names into chemical structures by querying a local database (see Figures 1 and 2). Currently this database contains about 5 000 000 unique molecules collected from various Web resources such as the National Cancer Institute Developmental Therapeutics Program, of which about 400 000 structures contain a CAS RN. Information about CAS numbers and associated molecular names and structures was harvested from the Internet. For this particular project the CAS Registry number extraction from public resources for example from MSDS datasheets to build an open access molecular database was communicated to Chemical Abstract Service, as building a database containing more than 10 000 CAS registry numbers is not usually permitted by the recent policies of CAS. This route was necessary for legal reasons and ensures adherence to the CAS requirements as well as integrity of the data.

When using ChemXtreme CAS IDs are thus, in addition to being submitted to the Google API, also used to query the local database for molecular structures (see Figure 1), and results for both the Google search and querying the local database can be displayed to the user simultaneously. Querying can be performed by chemical name or CAS number. A sample result is shown in Figure 2, where on the left-hand side chemical structures are retrieved from the database, while on the right-hand side information retrieved via Google is displayed. At the current state of ChemXtreme chemical structures are displayed from the database-retrieved SMILES using MarvinViewer[36] developed by ChemAxon.[37] While MarvinViewer is not open source software itself the integration with other viewers (such as JChemPaint[38]) would not be difficult to realize.

3. RESULTS AND DISCUSSION

ChemXtreme is able to perform text parsing in a distributed fashion. To establish the efficiency of this approach, first a speed comparison between the single-machine ChemReader and the distributed version presented here was performed. All data given are for P-4 machines with 2.8 GHz and 512 MB of RAM, running the Windows XP operating system. The distributed version ChemXtreme was tested on

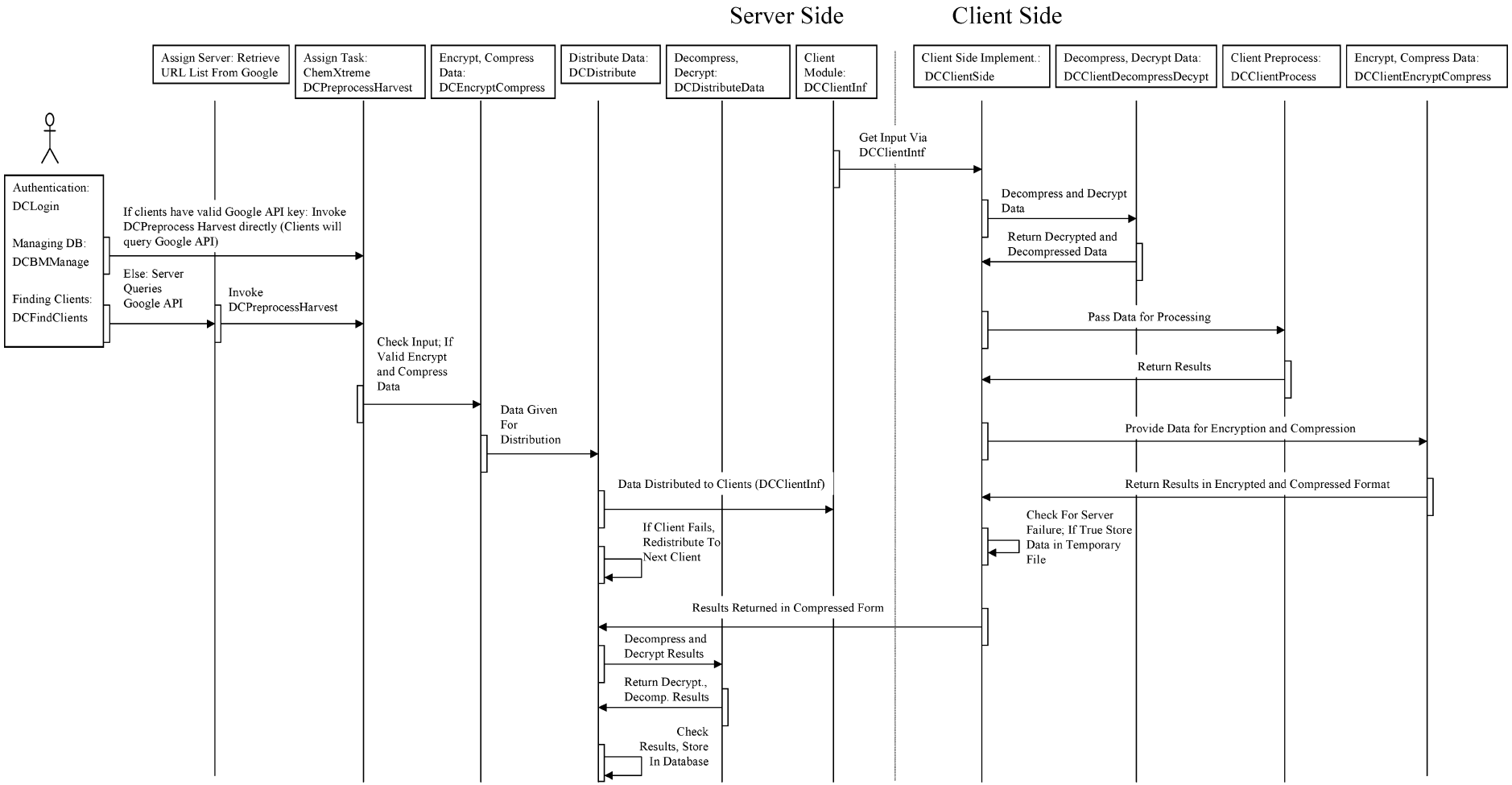
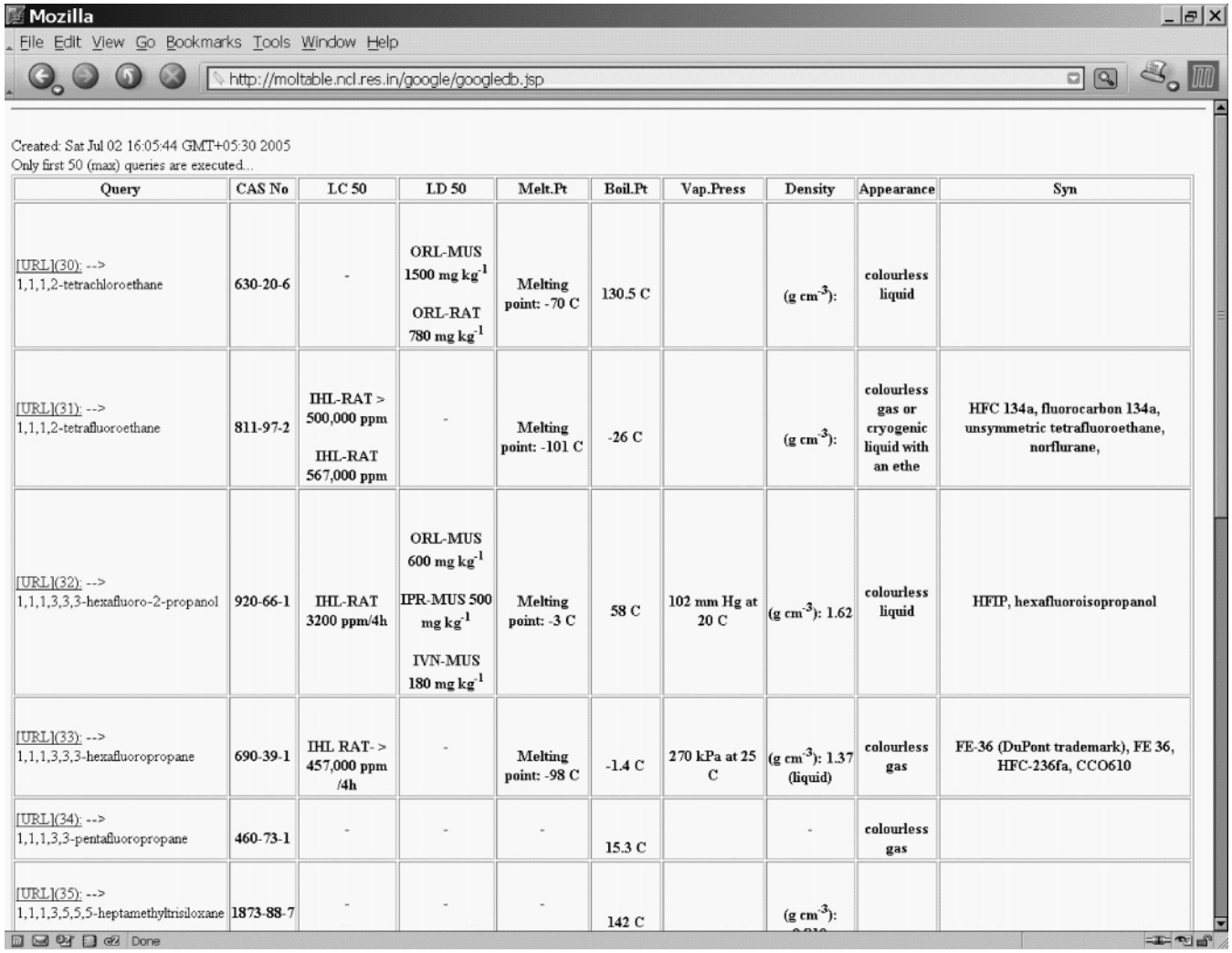


Figure 4. Computational steps employed for distributed searching on the client and server side.



Query	CAS No	LC 50	LD 50	Melt.Pt	Boil.Pt	Vap.Press	Density	Appearance	Syn
[URL](30): --> 1,1,1,2-tetrachloroethane	630-20-6	-	ORL-MUS 1500 mg kg^{-1} ORL-RAT 780 mg kg^{-1}	Melting point: -70 C	130.5 C		(g cm^{-3}):	colourless liquid	
[URL](31): --> 1,1,1,2-tetrafluoroethane	811-97-2	IHL-RAT > 500,000 ppm IHL-RAT 567,000 ppm	-	Melting point: -101 C	-26 C		(g cm^{-3}):	colourless gas or cryogenic liquid with an ethe	HFC 134a, fluorocarbon 134a, unsymmetric tetrafluoroethane, norflurane,
[URL](32): --> 1,1,1,3,3,3-hexafluoro-2-propanol	920-66-1	IHL-RAT 3200 ppm/4h	ORL-MUS 600 mg kg^{-1} IPR-MUS 500 mg kg^{-1} IVN-MUS 180 mg kg^{-1}	Melting point: -3 C	58 C	102 mm Hg at 20 C	(g cm^{-3}): 1.62	colourless liquid	HFIP, hexafluoroisopropanol
[URL](33): --> 1,1,1,3,3,3-hexafluoropropane	690-39-1	IHL RAT- > 457,000 ppm /4h	-	Melting point: -98 C	-1.4 C	270 kPa at 25 C	(g cm^{-3}): 1.37 (liquid)	colourless gas	FE-36 (DuPont trademark), FE 36, HFC-236fa, CCO610
[URL](34): --> 1,1,1,3,3-pentafluoropropane	460-73-1	-	-	-	15.3 C		-	colourless gas	
[URL](35): --> 1,1,1,3,5,5,5-heptamethyltrisiloxane	1873-88-7	-	-	-	142 C		(g cm^{-3}):		

Figure 5. Results page displayed to the user after retrieval of relevant Web pages and automated analysis of the results. The original Web site is linked from the results page for validation by the user in the first column, followed by CAS-ID and, if found, information about physicochemical and biological properties of the compound. For the compounds shown here both biological as well as physicochemical data could be retrieved. If multiple Web sites are found for compounds, all individual data are displayed to the user.

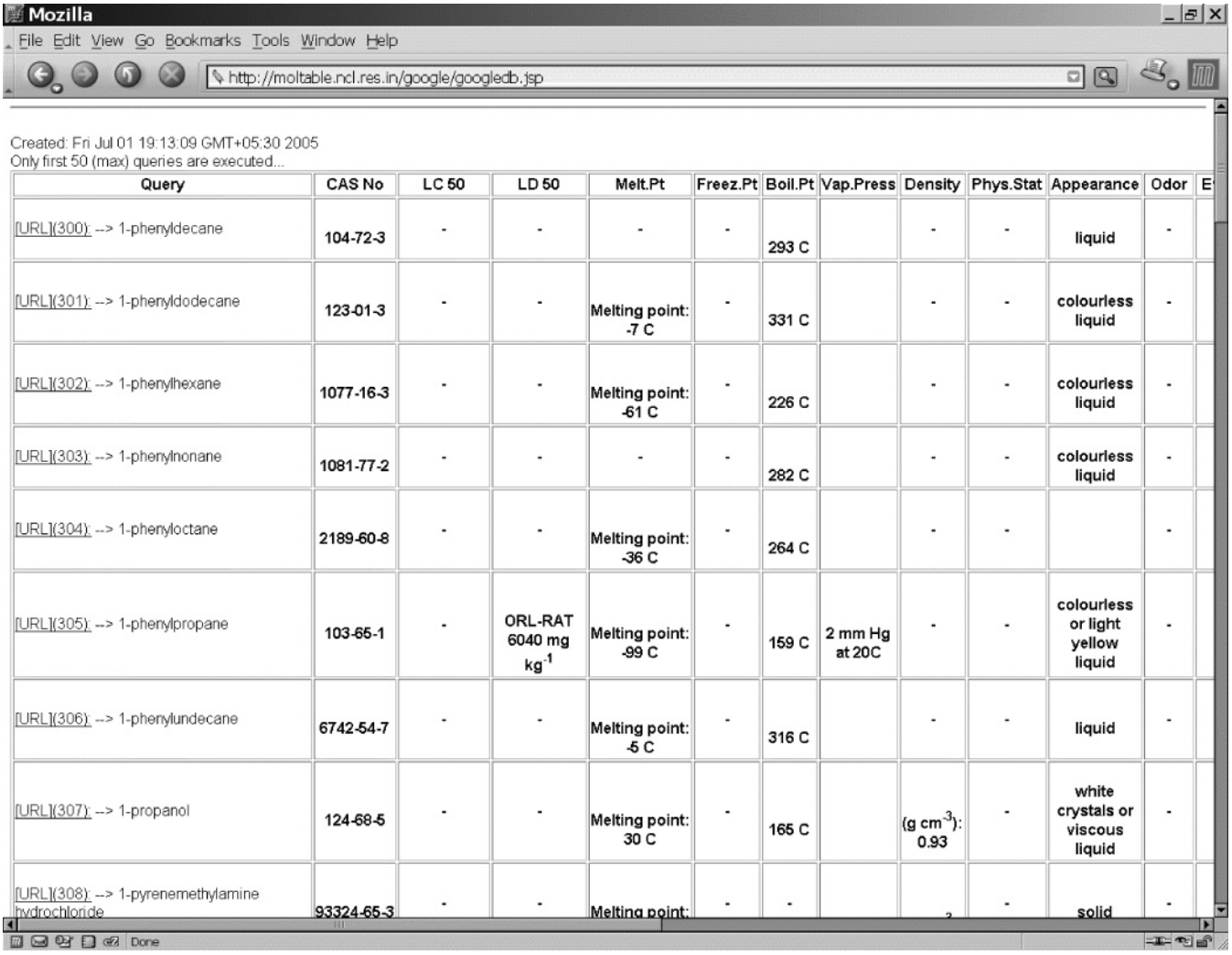
30 machines connected to 2 Mbps bandwidth for Internet in a high-speed Ethernet network.

The time required to process a single query (retrieve the Web page from the URL, perform pattern matching, and, in case of the distributed version, transmit results back to the server) was found to be about 6 s in both cases. For the case study presented here, where 100 queries were sent off to Google and 10 URLs were retrieved for each of the queries, a time requirement of between 12 and 15 min was found for the distributed (ChemXtreme) approach, while a time of about 105 min (1 h 45 min) was determined for the standalone (ChemReader) approach. The time requirement for text parsing scales is inversely linear with the number of clients employed, producing minimal computational overhead. This clearly shows the effectiveness of employing distributed text analysis, although it should be admitted that at larger numbers of results retrieved the integration of extracted data on the server might pose a computational bottleneck.

As an example of how to retrieve molecular physicochemical properties using the Google API the sample halogenated alkanes 1,1,1,2-tetrachloroethane (CAS ID 630-20-6), 1,1,1,2-tetrafluoroethane (CAS ID 811-97-2), 1,1,1,3,3,3-hexafluoro-2-propanol (920-66-1), 1,1,1,3,3,3-hexafluoropropane (CAS ID 690-39-1), and 1,1,1,3,3-pentafluoropropane (CAS ID 460-73-1) were employed which retrieve the information shown in Figure 5. It comprises many important biological and physicochemical properties such as LC 50 (lethal concentration 50%) and LD 50 (lethal dose 50%) data, melting, and boiling points as well as vapor pressure, density, and appearance at ambient temperature and pressure. In addition synonymous names for the compound are presented, given they exist and are present. If multiple information sources are identified, all extracted data are displayed to the user who is then in the position to judge the quality of the retrieved data.

The results retrieved for a series of substitutes alkanes are given in Figure 6. As opposed to the previous example, biological data such as LC50/LD50 values are rarely found; physicochemical data such as melting points and boiling points, on the other hand, are retrieved for the majority of query structures.

To demonstrate the effectiveness of extracting chemical information from Web resources on a larger scale about 6000 chemical names of commonly available chemicals were compiled. This list was distributed to the participating clients



Created: Fri Jul 01 19:13:09 GMT+05:30 2005
Only first 50 (max) queries are executed...

Query	CAS No	LC 50	LD 50	Melt.Pt	Freez.Pt	Boil.Pt	Vap.Press	Density	Phys.Stat	Appearance	Odor	E
[URL](300): --> 1-phenyldecane	104-72-3	-	-	-	-	293 C		-	-	liquid	-	
[URL](301): --> 1-phenyldodecane	123-01-3	-	-	Melting point: -7 C	-	331 C		-	-	colourless liquid	-	
[URL](302): --> 1-phenylhexane	1077-16-3	-	-	Melting point: -61 C	-	226 C		-	-	colourless liquid	-	
[URL](303): --> 1-phenylnonane	1081-77-2	-	-	-	-	282 C		-	-	colourless liquid	-	
[URL](304): --> 1-phenyloctane	2189-60-8	-	-	Melting point: -36 C	-	264 C		-	-		-	
[URL](305): --> 1-phenylpropane	103-65-1	-	ORL-RAT 6040 mg kg^{-1}	Melting point: -99 C	-	159 C	2 mm Hg at 20C	-	-	colourless or light yellow liquid	-	
[URL](306): --> 1-phenylundecane	6742-54-7	-	-	Melting point: -5 C	-	316 C		-	-	liquid	-	
[URL](307): --> 1-propanol	124-68-5	-	-	Melting point: 30 C	-	165 C		(g cm^{-3}): 0.93	-	white crystals or viscous liquid	-	
[URL](308): --> 1-pyrenemethylamine hydrochloride	93324-65-3	-	-	Melting point:	-	-			-	solid	-	

Figure 6. Results page displayed to the user after retrieval of relevant Web pages and automated analysis of the results for a series of substitutes alkanes. As opposed to the previous example, biological data such as LC50/LD50 values are rarely found; physicochemical data such as melting points and boiling points on the other hand are retrieved for the majority of query structures.

in order to obtain relevant URLs through Google API systematically. As before, documents corresponding to the retrieved URLs from the Google API were downloaded to the client side for pattern analysis, comprising physicochemical data and the CAS Registry number as well as biological data on the molecules. The extracted data can then optionally be written as (1) an HTML page containing a table with the harvested data, (2) as a plain text file with comma separated values, or (3) transferred directly to other databases via JDBC connectivity. In the case presented here an HTML page was finally obtained from the database

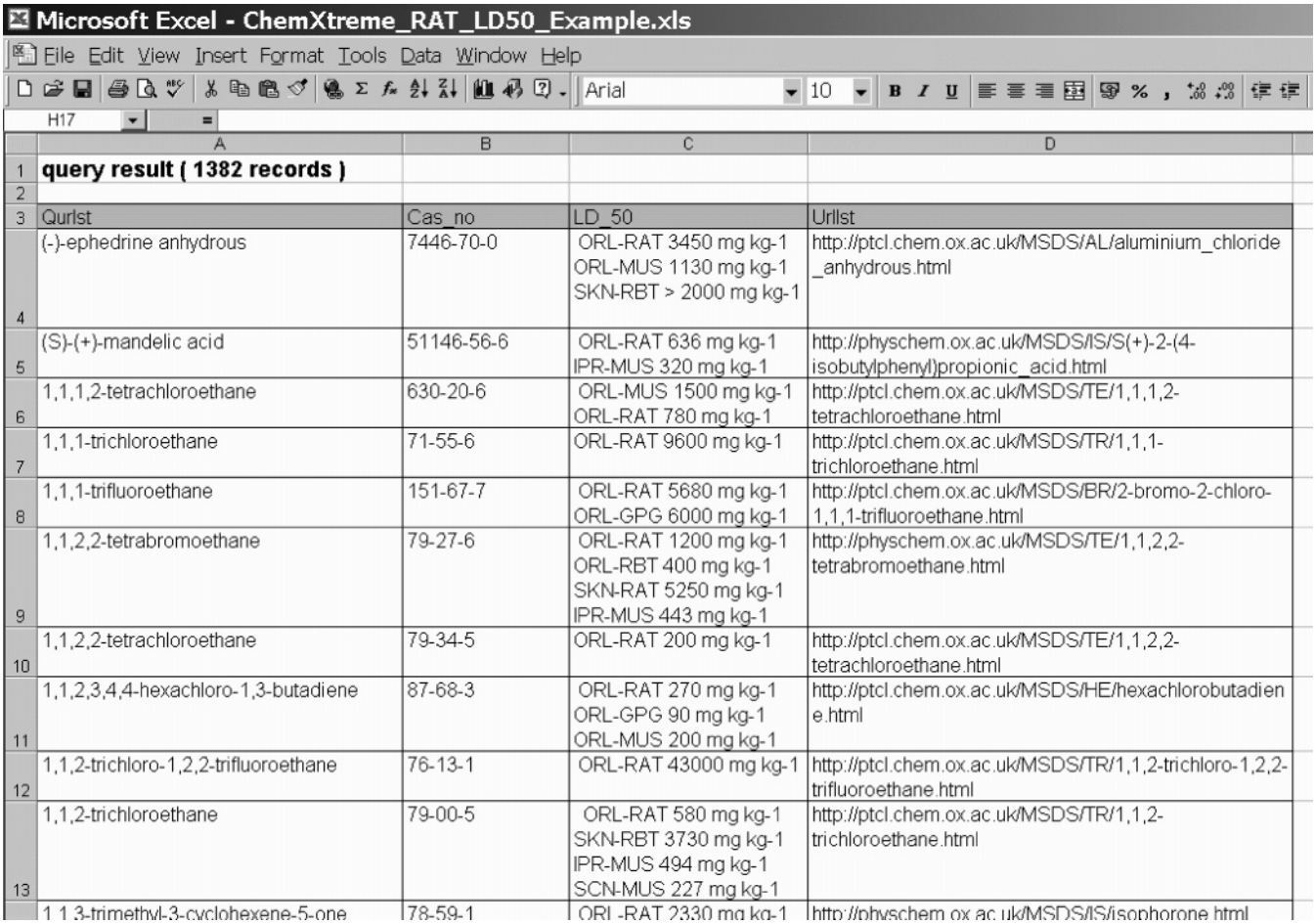
A multitude of biological information was harvested in this way, and giving a more specific example ChemXtreme could be efficiently employed to build a database of rat oral LD50 values for 1382 chemical names in this case, part of which is shown in Figure 7. This complete sample data set is available as Supporting Information along with the URLs where the information was retrieved from and the CAS-RN. The CAS-RN can then in a real-world study be further linked to other chemical structure databases to facilitate QSAR related studies in combination with molecular descriptor calculation and appropriate statistical tools. A comprehensive study on data mining of biological and physical data from the Internet for SAR and related studies is still on the way, also including a sample data set of molecular properties compiled for several thousand compounds.

The query form of ChemXtreme also allows for specification of additional queries, apart from the compound name information is searched for. For example, terms such as msds (for MSDS datasheets) or toxicology (for LD50 and related data) were found to retrieve more meaningful results in most cases. Since only a certain number of Google API searches are allowed per day, a local queuing system has been set up to allow fair use of the interface to all users. Alternatively users have the opportunity to apply for a Google API license which is provided free of cost by Google. If a valid license key is provided to the ChemXtreme server, then retrieval of results is greatly accelerated.

4. CONCLUSIONS

The work presented here is a proof-of-concept study for automatically mining chemical information from the Internet. It also represents the first exploitation of the Google API in combination with JavaRMI-based distributed information analysis for scientific purposes.

The Google API represents a versatile and easy-to-use interface to the largest number of indexed Web pages



Microsoft Excel - ChemXtreme_RAT_LD50_Example.xls

File Edit View Insert Format Tools Data Window Help

query result (1382 records)

Qurlst	Cas_no	LD_50	Urllst
(-)-ephedrine anhydrous	7446-70-0	ORL-RAT 3450 mg kg-1 ORL-MUS 1130 mg kg-1 SKN-RBT > 2000 mg kg-1	http://ptcl.chem.ox.ac.uk/MSDS/AL/aluminium_chloride_anhydrous.html
(S)-(+)-mandelic acid	51146-56-6	ORL-RAT 636 mg kg-1 IPR-MUS 320 mg kg-1	http://physchem.ox.ac.uk/MSDS/IS/S(+)-2-(4-isobutylphenyl)propionic_acid.html
1,1,1,2-tetrachloroethane	630-20-6	ORL-MUS 1500 mg kg-1 ORL-RAT 780 mg kg-1	http://ptcl.chem.ox.ac.uk/MSDS/TE/1,1,1,2-tetrachloroethane.html
1,1,1-trichloroethane	71-55-6	ORL-RAT 9600 mg kg-1	http://ptcl.chem.ox.ac.uk/MSDS/TR/1,1,1-trichloroethane.html
1,1,1-trifluoroethane	151-67-7	ORL-RAT 5680 mg kg-1 ORL-GPG 6000 mg kg-1	http://ptcl.chem.ox.ac.uk/MSDS/BR/2-bromo-2-chloro-1,1,1-trifluoroethane.html
1,1,2,2-tetrabromoethane	79-27-6	ORL-RAT 1200 mg kg-1 ORL-RBT 400 mg kg-1 SKN-RAT 5250 mg kg-1 IPR-MUS 443 mg kg-1	http://physchem.ox.ac.uk/MSDS/TE/1,1,2,2-tetrabromoethane.html
1,1,2,2-tetrachloroethane	79-34-5	ORL-RAT 200 mg kg-1	http://ptcl.chem.ox.ac.uk/MSDS/TE/1,1,2,2-tetrachloroethane.html
1,1,2,3,4,4-hexachloro-1,3-butadiene	87-68-3	ORL-RAT 270 mg kg-1 ORL-GPG 90 mg kg-1 ORL-MUS 200 mg kg-1	http://ptcl.chem.ox.ac.uk/MSDS/HE/hexachlorobutadiene.html
1,1,2-trichloro-1,2,2-trifluoroethane	76-13-1	ORL-RAT 43000 mg kg-1	http://ptcl.chem.ox.ac.uk/MSDS/TR/1,1,2-trichloro-1,2,2-trifluoroethane.html
1,1,2-trichloroethane	79-00-5	ORL-RAT 580 mg kg-1 SKN-RBT 3730 mg kg-1 IPR-MUS 494 mg kg-1 SCN-MUS 227 mg kg-1	http://ptcl.chem.ox.ac.uk/MSDS/TR/1,1,2-trichloroethane.html
1,1,3-trimethyl-3-cyclohexene-5-one	78-59-1	ORL-RAT 2330 mg kg-1	http://physchem.ox.ac.uk/MSDS/IS/isophorone.html

Figure 7. Results retrieved employing a more specialized query, retrieving rat LD50 values. For a set of 1382 chemical structures appropriate values could be retrieved which are also given as Supporting Information.

(currently about 8 billion). Until now retrieved documents had to be browsed manually for relevant information. We analyze those documents automatically by employing Java regular expressions. This step greatly speeds up text analysis and also allows the use of clearly (explicitly) defined queries. Performance of text parsing using Java regular expressions in a distributed client/server architecture was found to produce only minimal computational overhead over single-machine performances, resulting in a parsing time that scales inversely linear with the number of clients employed. All information from retrieved Web pages that is relevant to the query is compiled and presented to the user in a structured format. In addition it may be submitted to a local database. Thus, we present the first chemical informatics approach to retrieve *answers* to chemical questions by utilizing knowledge available on the Internet, instead of retrieving only a list of (possible) *locations* of it. The biological activity data retrieved for a list of chemical substances can be directly used for QSAR or other kinds of analyses, as illustrated by a list of several hundred rat LD50 values compiled this way.

The approach is easily extendable (by redefining search patterns) to any other kind of information residing on the Internet, such as biological systems, where depending on the context search terms occur in conclusions can be drawn as to not only the family of the organism but also any other information related to it which the user is interested in. In addition, information retrieved can be analyzed with respect as to how it is related to each other. While not the purpose of the current work, this is envisaged for the future.

A Web demo version of ChemXtreme is publicly accessible at http://moltable.ncl.res.in/google/googledb.html. Analysis of HTML documents containing CAS identifiers and retrieval of according structures from a database can be performed at http://moltable.ncl.res.in/google/urlcas.html. The JavaRMI open source code of distributed computing environment integrated with ChemXtreme module to harvest chemical information is available from one of the authors (M.K.).

ACKNOWLEDGMENT

The authors thank the Director of the NCL-Pune, Department of Science and Technology and DSIR, New Delhi for the financial support for this project. M.K. thanks S. K. Sidhu, S. G. Nandkumar, B.C. Sachin, I. Deepali, and S. Rashmi for technical support. A.B. thanks the Gates Cambridge Trust and Unilever for financial as well as Robert C. Glen (Unilever Centre) for academic support. The anonymous referees are thanked for helpful comments on the manuscript.

Supporting Information Available: A set of 1382 chemical structures with appropriate values and a database of rat oral LD50 values for 1382 chemical names along with the URLs (where the information was retrieved from) and the CAS-RN. This material is available free of charge via the Internet at http://pubs.acs.org.

REFERENCES AND NOTES

(1) MDL Drug Data Report; MDL ISIS/HOST software, MDL Information Systems, Inc.

(2) Google, I. Mountain View, CA. http://www.google.com. Google, Inc., Mountain View, CA. http://www.google.com.
(3) http://www.pubmed.org.
(4) Accumo Search Engine. http://www.accumo.com/.
(5) Milward, D.; Thomas, J. *From Information Retrieval to Information Extraction*; Hong Kong University of Science and Technology, 2000; pp 85−97.
(6) Adams, S. E.; Goodman, J. M.; Kidd, R. J.; McNaught, A. D.; Murray-Rust, P. et al. Experimental data checker: better information for organic chemists. *Org. Biomol. Chem.* **2004**, *2*, 3067−3070.
(7) Townsend, J. A.; Adams, S. E.; Waudby, C. A.; de Souza, V. K.; Goodman, J. M. et al. Chemical documents: machine understanding and automated information extraction. *Org. Biomol. Chem.* **2004**, *2*, 3294−3300.
(8) Gkoutos, G. V.; Leach, C.; Rzepa, H. S. ChemDig: new approaches to chemically significant indexing and searching of distributed web collections. *New J. Chem.* **2002**, *26*, 656−666.
(9) Blaschke, C.; Andrade, M. A.; Ouzounis, C.; Valencia, A. Automatic extraction of biological information from scientific text: protein−protein interactions. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* **1999**, 60−67.
(10) Thomas, J.; Milward, D.; Ouzounis, C.; Pulman, S.; Carroll, M. Automatic extraction of protein interactions from scientific abstracts. *Pac. Symp. Biocomput.* **2000**, 541−552.
(11) Blaschke, C.; Oliveros, J. C.; Valencia, A. Mining functional information associated with expression arrays. *Funct. Integr. Genomics* **2001**, *1*, 256−268.
(12) Rindflesch, T. C.; Tanabe, L.; Weinstein, J. N.; Hunter, L. EDGAR: Extraction of drugs, genes, and relations from the biomedical literature. *Proc. Pacific Symp. Biocomput.* **2000**, 514−525.
(13) Milward, D.; Bjareland, M.; Hayes, W.; Maxwell, M.; Oberg, L.; et al. Ontology-based interactive information extraction from scientific abstracts. *Compar. Funct. Genom.* **2005**, *6*, 67−71.
(14) Blaschke, C.; Hirschman, L.; Valencia, A. Information extraction in molecular biology. *Brief. Bioinform.* **2002**, *3*, 154−165.
(15) Hale, R. Text mining: getting more value from literature resources. *Drug Discovery Today* **2005**, *10*, 377−379.
(16) Krallinger, M.; Erhardt, R. A.; Valencia, A. Text-mining approaches in molecular biology and biomedicine. *Drug Discovery Today* **2005**, *10*, 439−445.
(17) Houston, A. L.; Chen, H. C.; Hubbard, S. M.; Schatz, B. R.; Ng, T. D.; et al. Medical data mining on the Internet: Research on a cancer information system. *Artif. Intell. Rev.* **1999**, *13*, 437−466.
(18) Thelwall, M. A web crawler design for data mining. *J. Inf. Sci.* **2001**, *27*, 319−325.
(19) Pichl, L.; Suzuki, M.; Joe, K.; Sasaki, A. Networked mining of atomic and molecular data from electronic journal databases on the Internet. *Databases in Networked Information Systems, Proceedings*; 2005; pp 159−170.
(20) http://www.soros.org/openaccess/.
(21) http://chembank.broad.harvard.edu/.
(22) Chemical Entities of Biological Interest (ChEBI). http://www.ebi.ac.uk/chebi/.
(23) World Wide Molecular Matrix (WWMM). http://wwmm.ch.cam.ac.uk/gridsphere/gridsphere.
(24) http://ecrystals.chem.soton.ac.uk/. S. C. R. Southampton Crystal Reports. http://ecrystals.chem.soton.ac.uk/.
(25) http://www.openarchives.org/. O. A. I. Open Archives Initiative. http://www.openarchives.org/.
(26) http://dspace.ncl.res.in/dspace/index.jsp.
(27) Chemical Abstracts Service. http://www.cas.org.
(28) Stein, S. E.; Heller, S. A. *Abs. Pap. Am. Chem. Soc.* **2001**, *222*, CINF-005.
(29) Murray-Rust, P.; Rzepa, H. S. Chemical markup, XML, and the World Wide Web. 4. CML schema. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 757−772.
(30) PubChem - Part of the NIH Molecular Libraries Roadmap Initiative. http://pubchem.ncbi.nlm.nih.gov/.
(31) http://www.javacoffeebreak.com/articles/javarmi/javarmi.html.
(32) Karthikeyan, M.; Sangade, V.; Pandey, A. K.; Bender, A. Design of a Java RMI Based Architecture for Distributed Chemical Informatics Applications. **2006**, submitted for publication.
(33) Tonge, A. P.; Rzepa, H. S.; Yoshida, H. Authentication of Internet-based distributed computing resources in chemistry. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 483−490.
(34) Web Services Description Language. The W3C Consortium, http://www.w3.org/TR/wsdl.
(35) Simple Object Access Protocol. The W3C Consortium, http://www.w3.org/TR/soap/.
(36) Csizmadia, F. JChem: Java applets and modules supporting chemical database handling from web browsers. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 323−324.
(37) ChemAxon, Inc. http://www.chemaxon.com.
(38) Krause, S.; Willighagen, E.; Steinbeck, C. JChemPaint−Using the Collaborative Forces of the Internet to Develop a Free Editor for 2D Chemical Structures. *Molecules* **2000**, *5*, 93−98.

CI050329+