Chemical−Text Hybrid Search Engines

Yingyao Zhou,*,† Bin Zhou,† Shumei Jiang,† and Frederick J. King†,‡

Genomics Institute of the Novartis Research Foundation, 10675 John Jay Hopkins Drive, San Diego, California 92121, and, Developmental and Molecular Pathways, Novartis Institutes for BioMedical Research, 250 Massachusetts Avenue, Cambridge, Massachusetts 02139

Received September 30, 2009

As the amount of chemical literature increases, it is critical that researchers be enabled to accurately locate documents related to a particular aspect of a given compound. Existing solutions, based on text and chemical search engines alone, suffer from the inclusion of "false negative" and "false positive" results, and cannot accommodate diverse repertoire of formats currently available for chemical documents. To address these concerns, we developed an approach called Entity-Canonical Keyword Indexing (ECKI), which converts a chemical entity embedded in a data source into its canonical keyword representation prior to being indexed by text search engines. We implemented ECKI using Microsoft Office SharePoint Server Search, and the resultant hybrid search engine not only supported complex mixed chemical and keyword queries but also was applied to both intranet and Internet environments. We envision that the adoption of ECKI will empower researchers to pose more complex search questions that were not readily attainable previously and to obtain answers at much improved speed and accuracy.

INTRODUCTION

Search has undoubtedly become the buzzword of the digital age. The life sciences have seen an explosion of data resulting from such innovations as combinatorial chemistry, automated DNA sequencing, and high throughput small molecule screening technologies. These advancements have created challenges related to finding useful information in a comprehensive and time efficient manner. For example, the SureChem database (http://surechem.org) contains over 5.1 million patents and 4.7 million MEDLINE articles. There have been over 400 000 patents and 200 000 MEDLINE articles added annually in the past few years, which provides an estimate as to how these types of databases will increase in size in the future.

To navigate the vast array of chemical files, a search engine capable of accurately locating all documents that describe a particular aspect of a compound of interest is of critical importance to researchers. Currently available search engines can be segregated into two groups: text search engines (e.g., Google search (http://google.com), Microsoft search (http://bing.com), etc.) and structure-based chemical search engines (e.g., SciFinder search (http://scifinder.cas.org)). Search strategies based on these technologies in isolation are suboptimal because of the inclusion of "false negative" and "false positive" hits. For example, according to Google Insights for Search (http://www.google.com/insights/search), 57% of the text searches for the chemical compound Gleevec have been performed using the term "Gleevec", whereas 32% use the Gleevec synonym "Imatinib". Furthermore, "Gleevec" retrieves only ∼11% of the articles in PubMed relevant to compounds that are structurally identical to Gleevec, yet are referred to using Imatinib or even a different compound identifier. This false-negative problem is inherent to text search engines because they rely on chemical names but have no understanding of chemical synonyms. In contrast, chemical search engines that rely on structure drawings overcome the issue associated with synonyms. However, structure-based searches are typically unaware of the text context within which a chemical entity is described. A search using the structure of Gleevec will identify all such documents regardless of whether the intended search is to identify information related to Gleevec's ability to target the Bcr-Abl oncoprotein, treatment of patients with Chronic Myelogenous Leukemia (CML), etc. If a researcher is interested in using a structure-based search of "Gleevec" to identify documents describing the biomedical application of Gleevec to CML, ∼60% of the articles will not be relevant, since this subset does not have information relevant to CML (based on the results of "Gleevec" and "Gleevec CML" Pubmed searches). Therefore, this subset, can be considered as a false positive hit for structure-based search approaches. In addition to the above-described challenges in search accuracy, mining in-house chemical documents is further complicated by the diverse repertoire of file storage protocols and data formats within a corporate intranet, for example, currently no search tool can discovery all in-house documents containing Gleevec-analogs.

Our analyses of the architectures of existing text and chemical search tools indicated new search platforms could be built by adding chemical intelligence on top of existing text search engines. The key to the implementation lies in an approach called Entity-Canonical Key Indexing (ECKI). ECKI maps a chemical entity onto its unique Canonical Key (CK), which is an indexable text identifier. As illustrated below, ECKI enables text search engines to reduce the impact related to their false-negative deficiencies, while preserving

† Genomics Institute of the Novartis Research Foundation.
‡ Novartis Institutes for BioMedical Research.

10.1021/ci900380s
Published on Web 01/04/2010

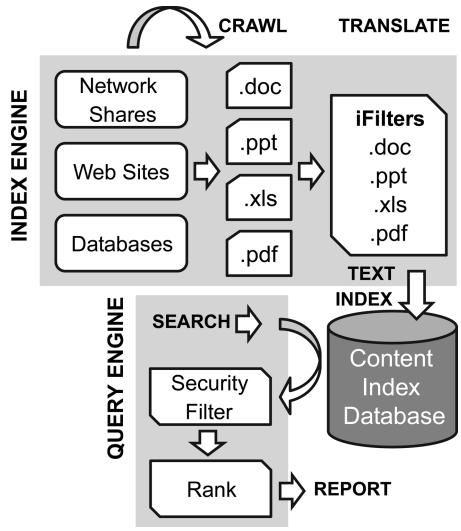

Figure 1. Architecture of the MOSS text search engine.

all the desirable features that have been optimized for text searches over the decade.

The ECKI-based hybrid approach can score highly in nearly all critical features required for chemical document searching, which is demonstrated below using three examples: two Internet and one intranet applications. Sophisticated queries such as finding all documents describing the application of Gleevec-analogs (nominally >90% structural similarity) to CML disease can be supported. By enabling queries containing both chemical and biological conditions, ECKI now empowers researchers to pose questions of greater relevance to the functions of chemicals and to obtain answers of better precision.

RESULTS

Text Search Engines. Companies such as Google and Microsoft are leaders in the text search market, which has provided text search engines that are optimized for crawling, indexing and ranking data contents both within a corporate intranet and in the Internet.[1]Using the Microsoft Office Sharepoint Server (MOSS) search engine as an example,[2] the architecture of a text search engine is outlined in Figure 1. The search engine relies on a Crawler to continually identify newly available data files. The Crawler is able to incorporate a wide range of data transfer protocols for fetching documents from network folders, Web sites, database tables, etc. Note that the data files can be in various formats, such as binary MS Office 2003 formats (.doc, .ppt, .xls, etc.). A component called iFilter is responsible for converting the file content into blocks of text and forwarding it to an Indexer to construct a content index database. iFilters are supplied by various parties, usually by the vendor, who defines the corresponding format. Indices are organized in such a way that proximity search, that is, words appearing near each other in a document, is possible. When a user enters a keyword query, all security-filtered hits are scored and ranked in descending level of predicted interest. This facilitates the rapid identification of desired information. All these features make text engines a desirable robust search solution for most applications.

Text search engines also are used widely to search chemical-containing documents, as seen in Google Scholar (http://scholar.google.com), Google Patent (http://www.google.com/patents), and Windows Academic Search (discontinued). Nevertheless, their power is very much limited because of their lack of "chemical intelligence". For example, these engines cannot recognize that "Gleevec" and "Imatinib" are synonyms for the same chemical compound. Consequently, 57% of the Google searches using the term "Gleevec" failed to identify ~89% of the documents that contained reference to compound structures identical to Gleevec! This problem is exacerbated by the many different representations that a compound can take. For example, Gleevec is also known by its corporate identifier "STI571" (widely used before the drug name was assigned) and CAS numbers (152459-95-5 or 220127-57-1). For compounds without a popular code name, IUPAC names, SMILES strings, InChi string, etc., can all be used as an identifier, which makes the problem even more challenging.

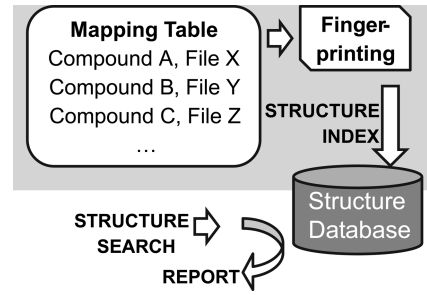

Figure 2. Architecture of a typical chemical search engine.

Chemical Search Engines. More advanced queries, such as finding documents for chemicals that are structurally similar or superstructures for a given chemical of interest, are certainly beyond the scope of what typical queries for text search engines were designed to undertake. A potential strategy to address this challenge is "chemical indexing". Systematic efforts in indexing chemical literature started as early as a century ago.[3] However, despite an earlier start in history the tools available for the text mining of chemical documents lags significantly compared to the success of text mining the biomedical literature (such as PubMed).[4] Reliable and accurate recognition of chemical entities in the literature presents the major obstacle.[5] Current chemical search engines utilize compound structure information, but the literature often refers to chemical entities by means of a variety of chemical names (including all forms of compound synonyms), which have been difficult to translate. In recent years, strides have been made within the cheminformatics community regarding name-to-structure translation; currently there are multiple tools available for this application (see tools reviewed in ref 5). The current state-of-art application of name-to-structure technology is exemplified by the SureChem patent database, a computer-assembled database consisting of over 6.6 million structures and 5.1 million patents.

Cheminformatics indexing technologies exist for the speedy identification of structures of relevance among tens of millions of potential candidates (e.g., the SureChem database). In fact, one potentially can search a database of several million structures per second using the ChemAxon chemical cartridge (http://chemaxon.com). The architecture of a typical chemical structure-based search engine is outlined in Figure 2. Starting with a structure-file location mapping table, the engine takes each structure and creates a binary

Table 1. Feature Comparison Among Text, Chemical, and Later-Described Hybrid Search Engines[a]

Features	Text Search Engines	Chemical Search Engines	Hybrid Search Engines	Important to Intranet	Important to Internet
Crawling and indexing	3	1	3	3	3
Support web pages	3	2	3	2	3
Support files in file system, database, SharePoint, etc.	3	1	3	3	1
Support non-text file formats	3	1	3	3	3
Support file meta data	3	1	3	3	2
Page ranking	3	1	3	3	3
Document security	3	1	3	3	1
Understand chemical names	1	3	3	2	3
Understand proprietary IDs	1	2	3	3	2
Understand chemical drawings	1	2	1	1	3
Search chemicals	1	3	3	3	3
Search chemicals and text	1	1	3	3	3

[a] For the three search engine columns, 1 means non-existent or limited, 2 means somewhat limited, and 3 means optimized. For the two intranet and Internet requirement columns, 1 means not-required or optional, 2 means desirable-to-have, and 3 means critical.

fingerprint. Both the structure and its fingerprint are stored within a structure database. Users then draw a query chemical structure, which is first filtered within the fingerprints, and then graph-matched to any surviving structure candidates. Fingerprints are binary strings that serve two roles: as an accelerator that filters out the majority of obvious nonmatches in full and substructure searches and as a measure for the degree of similarity within similarity searches.

The greatest advantage of chemical search engines is their ability to enable users to retrieve documents that contain a query structure either as an exact match, a substructure, or as a "similar" structure (as defined by the user). However, chemical search engines typically ignore the nonchemical text content. Search systems used by SciFinder, SureChem, and RSC Prospect (http://www.rsc.org/Publishing/Journals/ProjectProspect), etc., allow keyword searches, but these keyword and structure searches are implemented independently.

We have not found a chemical search engine that provides a query interface that unites both structure and keyword conditions, for example, a search of documents that contains the Gleevec structure along with the disease name "CML", both occurring within a text neighborhood. In principle, one could approximate this query by the Boolean-AND union of two hit lists: a Gleevec structure search hit list and another "CML" keyword search hit list. However, this approach would lead to false positive occurrences, for example, where Gleevec appears in the beginning of the document and "CML" appears in a different context at the end. In addition, hits are not ranked on the basis of a combined relevance score on both Gleevec and "CML". Besides lacking the ability to support context-specific searches, chemical search engines also lack components such as the Crawler, iFilter, and Ranker described above. This limits their usage to specialized database applications where crawling and ranking are not required. They are typically inappropriate for crawling various data sources in intranet and Internet settings.

Comparison of Search Engines. The challenge of constructing an effective search engine for chemical-containing documents can be represented by reviewing a set of features of importance to intranet and Internet applications, as well as the extent of support offered by existing text and chemical search engines alone (summarized in Table 1). The scores in Table 1 were assigned by the authors and, admittedly, are subjective. For example, we consider web page indexing less critical for an intranet application (score 2) because the majority of documents created within a company setting are probably not presented in a HyperText Markup Language (HTML) format. Metadata support for files is considered as "nice-to-have" in the context of Internet applications (score 2). Internet applications generally do not have to crawl file shares, SharePoint sites, or databases, and document security is unlikely to be an obstacle (score 1).

Text search engines have rather optimized strategies in crawling and indexing document sources, understand a battery of file formats, take advantage of file metadata, combine ownership information with people profiles in order to rank hits, and apply stringent security filters for data protection (Table 1). Almost all of these features are critical for corporate intranet applications.

Chemical search engines typically are built on top of name-to-structure recognition modules, which take a text document, parse and identify chemical entities, and create a database table that stores the linkage between chemical entities and data paths. They provide a robust structure search interface for users to retrieve documents by similarity, substructure, and full structure searches. With modest additional programming effort they can be modified to recognize proprietary corporate chemical IDs. Presently, there are efforts underway in recognizing chemical drawings directly. Although the quality of optical-compound recognition is still primitive,[5] we nevertheless score it as a 2 to distinguish it from nonexistent recognition (score 1) endemic to text search engines. Chemical search engines also may support keyword search, but typically this functionality is independent from the structure search and often lacks advanced components such as a stemmer, thesaurus, or noise word filter (score 1).

On one hand, chemical search engines typically score poorly for features where text search engines are optimized. We score chemical search engines a 2 in supporting web pages, because crawling and indexing web pages is what text search engines are superior for after years of keen market competition. In contrast, for features related to chemical intelligence, support within text search engines is essentially nonexistent. The above comparison suggests text search engines and chemical search engines are rather complementary in terms of features critical for intranet and Internet applications. This insinuates that the merging of both search engines into one hybrid system would provide a superior approach. The implementation of such an approach is described below.

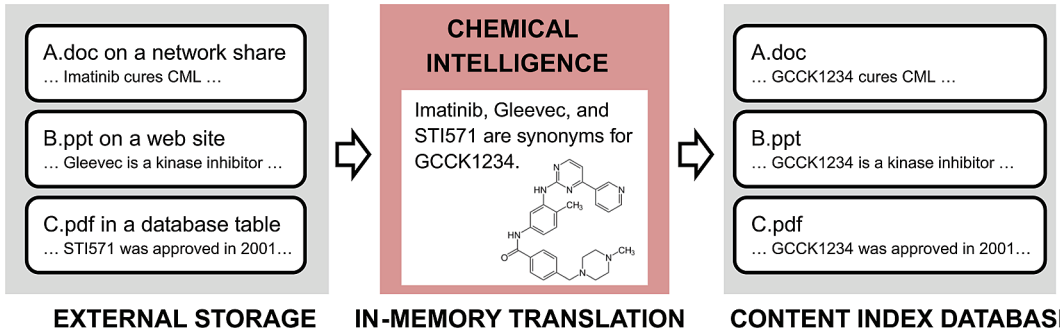

Figure 3. ECKI eliminates the need for text search engines to understand complex chemical synonyms because chemical entities are replaced by their canonical keywords before indexing.

Entity-Canonical Keyword Indexing (ECKI). From our preliminary attempts, it quickly became clear that adding all the text intelligence to a chemical search engine required reinventing many crawling, indexing, and ranking solutions that previously had been elegantly addressed by text search engines such as Microsoft Office SharePoint Server (MOSS) search.[2] Instead, we focused on looking for ways to incorporate chemical intelligence into existing text search engine by adapting chemical structure into a native language to engines such as MOSS Search. We propose an approach called Entity-Canonicalization Keyword Indexing (ECKI). An "entity" refers to a chemical structure, for example, "Gleevec" and "Imatinib" are names of the same entity. A canonical key (CK) is an indexable keyword that uniquely identifies an entity, that is, a form of primary key for structures. The term "indexing" refers to the fact that the entity to CK conversion takes place as part of the overall indexing processes run by text search engines, which has significant implications as detailed below.

The main concepts of ECKI are illustrated by the examples that are described below and summarized in Figure 3. When a text search engine crawls for "Gleevec", the crawler identifies documents where the different chemical names "Gleevec", "Imatinib", and "STI571" are included in the text. The Crawler initiates a chemically intelligent filter to convert the data source (either a file, a database object, or an embedded object within a document) to text. The filter may optionally delegate the format conversion task to other preregistered filters; it then performs chemical entity recognition (by names, proprietary IDs, embedded chemical objects, or even drawings). The three synonyms are recognized to be aliases of the same entity and their unique canonical keyword (CK) representation, GCCK1234, is generated and used to replace the original chemical names before the text stream is forwarded to an indexing component. For illustration purposes, our CK is made of a unique integer plus the prefix GCCK, standing for GNF Compound Canonical Key (where GNF is our institute name).

From the text search engine's point of view, it is now as if only GCCK1234 were used in the text associated with the three different documents. A keyword search using GCCK1234, therefore, will identify all three documents, since that is how the Gleevec synonyms are presented inside the content index database where the actual search is taken place. The entity to CK translation process occurs "on the fly" in computer memory. This allows the three documents to remain physically unmodified in the external storage, which preserves all the security settings that will be used before returning the search hits. With ECKI technology, users are free to use an arbitrary chemical synonym for an entity in their writings, as long as it can be recognized by name-to-structure translation code in search engines.

We implemented the ECKI idea using the MOSS search platform (Figure 4). The core component is our custom-written iFilter program. The custom iFilter is preregistered to be associated with multiple file extensions; it delegates format conversion tasks to other third-party Office and PDF iFilters. Once the iFilter obtains the text, it scans for chemical entities using regular expressions for recognizing corporate IDs, using synonym dictionaries obtained from ChemAxon, SureChem, Word Drug Index by Thomson Reuters (http://www.thomsonreuters.com), MeSH (http://www.nlm.nih.gov/mesh/), and using the ChemAxon name-to-structure toolkit that translates IUPAC names, etc. If a chemical entity is recognized, its CK is identified in a backend compound database; in our case, an Oracle database bundled with the JChem Cartridge. If the entity has been encountered earlier, a full structure search will retrieve its previously assigned CK. Otherwise, it is inserted into the structure database and a new CK is generated. Note that canonical SMILES are not valid CKs, since they contain nonindexable characters. The content index database contains all text information with entities already replaced by their CKs.

Query Containing Mixed Chemical and Keyword Conditions. Assigning and utilizing a CK for compounds addresses the issue of identifying false-negatives, since all synonyms of a particular entity are now replaced by a unique keyword. Therefore, a user is no longer negatively impacted by the choice of "Gleevec", "Glivec", "Imatinib", "STI-571", or other forms in search term selection.

The power of the ECKI-based hybrid search engine is more apparent when it comes to complex queries that mix both chemical and keyword conditions. Consider the simplest mixture search, such as "Gleevec AND CML". The hybrid search engine simply uses "GCCK1234 AND CML", regardless of which synonym of Gleevec was used. As explained above, chemical search engines can accurately fetch all Gleevec-containing files, and then join with the results from a distinct "CML" keyword search. However, there are complicating limitations to implementing Boolean operations by manipulating the two hit lists obtained from separate structure and keyword searches. For instance, (1) search engines only return a finite number of top hits, so one may miss valid final matches because there are too many "CML-only" intermediate matches; (2) ranking is solely based on CML-containing documents, which is not as

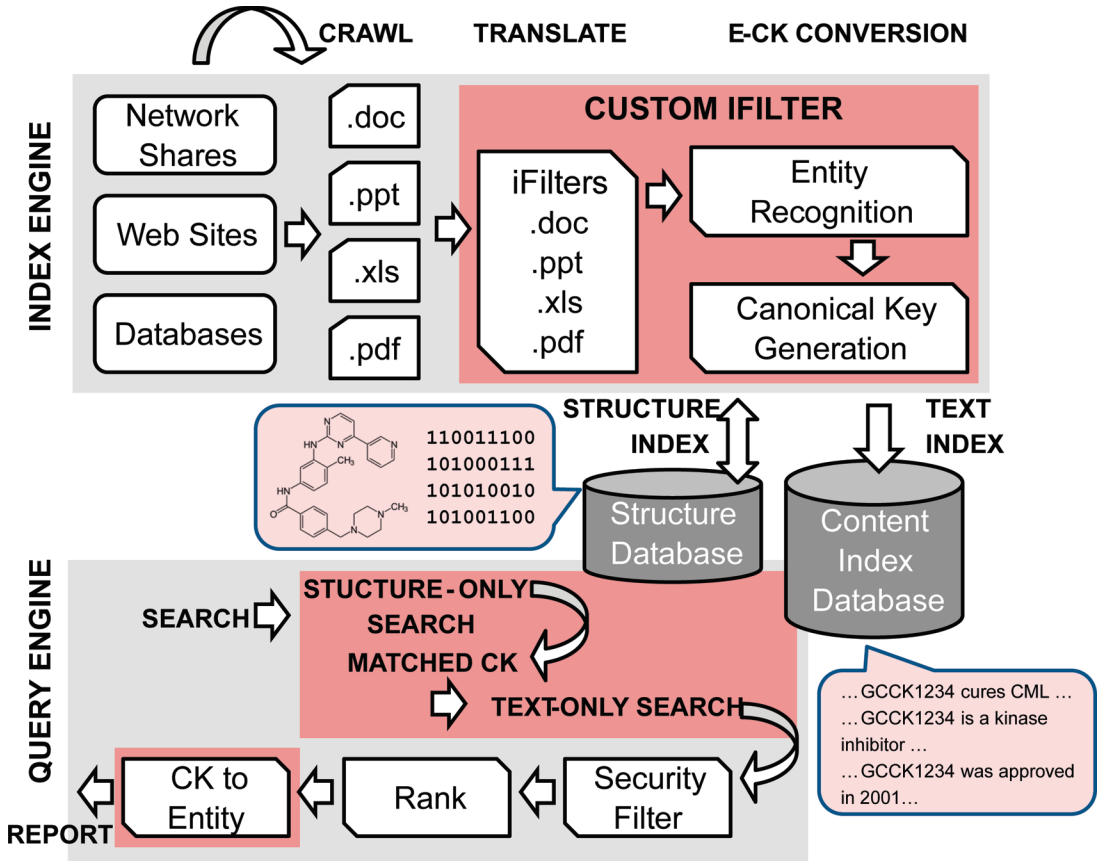

Figure 4. Architecture of the GNF MOSS-based ECKI search engine, where salmon-colored boxes highlight the custom components added to the MOSS search engine.

desirable as the ranking calculated on files containing both "Gleevec" and "CML". In fact, it is known that when there are multiple keywords, the ranking based on all keywords is superior to the combined ranking based on individual keywords alone.[2] (3) When multiple similar matches are found, text search engines often return one representative document, so a document containing Gleevec may be assigned lower significance than another "CML"-only documents. It should be noted that although chemical search engines in principle can support simultaneous chemical and keyword queries at the document level as outlined above, it is challenging to find a service on the Internet where it is actually implemented.

A "Gleevec AND CML" text search does not guarantee a match for a document that describes a relationship between the drug and the disease. False positive occurrences of "Gleevec AND CML" can be minimized by using the proximity search natively supported by MOSS, assuming that this is the user's goal. One would instead search using "GCCK1234 NEAR CML" and obtain higher-quality results in the hybrid search engine. In contrast, chemical search engines alone are unable to support any contextual search.

Queries containing similarity or substructure search conditions, such as "Gleevec-like", cannot be supported by text-search engines. For hybrid search engines such as ECKI, this query can first be carried out in the backend structure database with Tanimoto similarity score >0.9, just as in a chemical search engine. A list of matched CKs are returned and concatenated into a new query. This takes on the appearance of a query such as "GCCK1234 OR GCCK5678 OR ...", which can then be executed on the underlying text search engine (Figure 4). It is then straightforward to extrapolate this strategy toward the application of more complex queries, such as translating "Gleevec-like NEAR CML" into "(GCCK1234 NEAR CML) OR (GCCK5678 NEAR CML) OR ...".

Search and Report Interface. It is convenient if users can construct their queries using their favorite compound synonyms or, alternatively, employing chemical drawing tools, while the CK remains an identifier internal to the search engine itself. For reference purposes, we here outline the web interface implemented at GNF that enables this process (Figure 5).

In our convention, the user may use brackets to highlight the chemical search conditions. While "Gleevec" means the word itself and will not be interpreted as a chemical entity; "[Gleevec]" triggers a name-to-structure conversion and the actual query sent to the backend text search engine will become "GCCK1234" instead. Similarly "[Gleevec >0.9] NEAR CML" triggers a structure similarity search first and the query string is mapped into "(GCCK1234 NEAR CML) OR (GCCK5678 NEAR CML)" in the backend. The backend Microsoft SQL server can take a query string of 256 kilobytes or longer. Chemical conditions can also be specified via the drawing component: a query like "[SQ1] NEAR CML" indicates that the structure query condition "[SQ1]" is defined by the drawing component shown in the left panel of Figure 5. This drawing is capable of describing full, substructure, or similarity search conditions. The interface can in principle be extended to support multiple chemical conditions, such as "([SQ1] OR [SQ2]) NEAR CML". Additional examples of queries are described below.

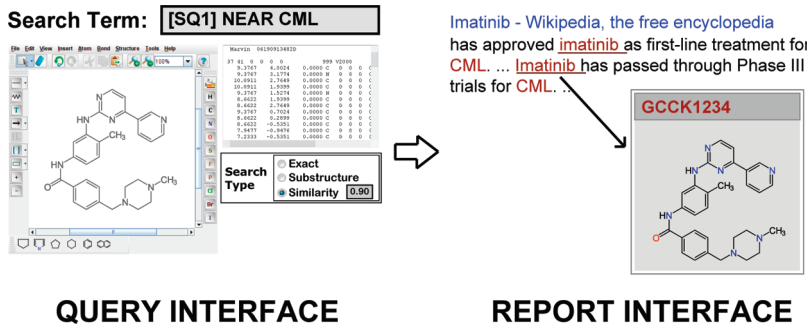

Figure 5. GNF web interface for ECKI search and report.

Table 2. Example Complex Queries Supported by Our Hybrid Engine Using Wikipedia Data

question	query	matched wiki entry	text-search engines alone
all compounds related to GIST(s)	GCCK* AND GIST*	imatinib, sunitinib	return all GIST-containing pages, could be false positives
use STI571 to advocate rational drug design	[STI571] NEAR rational NEAR design	imatinib	none; false negatives because "STI-571" was used in the context
everolimus-analogs	[everolimus >0.95]	everolimus, sirolimus	cannot support

MOSS Search Center provides an out-of-the-box search result page consisting of multiple rendering web parts. Following some additional customized coding, we converted the CKs in the search report into dynamic hyperlinks, where mouse-overs were able to display structure, key data, and additional links for the chemical entities. Adopting additional customization strategies as suggested in the Internet, our MOSS Search Center also supports wild card queries. For example "GCCK*" would be used to search all compounds within our database.

Applications of ECKI-Based Search Engines. With ECKI adapting any chemical search into a keyword search query that text search engines are optimized to handle, a researcher can conceive many search scenarios previously not possible. We outline some of the most interesting applications of the ECKI search engines below.

Intranet Application: Beginning of an Automatic Corporate Compounds Wikipedia. Wikipedia (http://wikipedia.org) is a phenomenal knowledgebase built through community efforts. At present, there are efforts in replicating its success in the chemical field applications, such as the ~7000 drug wiki pages in Wikipedia.org. However, attempts to build compound Wiki sites on larger sets of compounds, such as ChemSpider (http://www.chemspider.com), have not accumulated large amounts of compound curations in parallel as determined by the authors by performing random searches. One may explain this observation by the plethora of available compounds (tens-of-millions) or the majority of nontrivial chemical information being proprietary. Even when legal constraints are no longer a barrier in annotating in-house compounds, we found lack of input from researchers the determining bottleneck for similar compound curation efforts.[6] We therefore resorted to a more "automated" solution.

Our approach was to crawl various data sources to collect compound annotations from corporate researchers. For example, researchers probably have commented on a given compounds structure in their electronic lab notebooks, collected measurements in spreadsheets, given summaries and presentations, and even written publications before (in MS Office files, PDF files). By enabling the ECKI iFilter to recognize internal GNF compound identifiers, our hybrid MOSS engine crawled company file shares and generated structure-file links. Since the information in presentations and publications typically was carefully reviewed by the authors, the file links were generally of higher quality compared to some spontaneous comments collected through a Web site such as a "Discussion board". This exercise resulted in the identification of ~20 000 Microsoft Word, PowerPoint, and Adobe PDF files that were associated with ~90 000 in-house compounds. It was interesting that the most frequently cited compounds were, indeed, our most advanced preclinical and clinical candidates. It is worth reiterating that the ECKI approach enables the use of existing Microsoft iFilters to reliably parse all binary Office formats, as well as controlling file access based on the native security settings of individual documents discovered. Both are critical aspects for the proper application of any Intranet search solution.

Internet Application: Sophisticated Queries on Drug Wikipedia and MEDLINE. ECKI can be applied to support Internet searches as well. For illustration purpose, we first downloaded the ~7000 drug Wikipedia pages and then indexed locally by the hybrid MOSS search engine. We list here a few interesting search examples that outline the capability of such a system (Table 2).

SureChem has also applied the name-to-structure technology for the identification of over 200 000 chemicals among 4.7 million MEDLINE entries. We then took the subset of 163 925 chemical articles published after 2008 and created individual web pages that each contained paper title, abstract, and other summary data. The search results summarized in Table 3 illustrate how the improved search technology can lead to interesting matches of direct biomedical relevance.

DISCUSSION

Comparing ECKI to Other Related Technologies. ECKI is fundamentally different from technologies such as InChiKey (http://www.iupac.org/inchi/) or semantic markup (e.g., the RSC Prospect project). InChiKey uses a nearly unique identifier for chemical entities, with

Table 3. Example Complex Queries Supported by the Hybrid Engine Using MEDLINE Data

question	query	matched entry (PubMed ID, compound)
non-Gleevec CML compounds	"Chronic myelogenous leukemia" AND "GCCK*" AND NOT "GCCK1234"	18537755, nelarabine, forodesine; 18705753, vincristine, quinacrine; 18644865, doxorubicin
use Gleevec to show rational drug design can work	[Gleevec] NEAR rational NEAR design	18616236, imatinib
all compounds related to GIST(s)	"GCCK*" AND "gastrointestinal stromal tumor" AND "GIST"	18708414, imatinib; 18294292, imatinib; 17729245, imatinib, sunitinib

approximately one replicate in every 75 billion structures. Therefore, it has many desirable properties for use as a standard code for all chemicals to facilitate search of chemical information.[7] However, InChiKey alone is not a viable search solution for several reasons. First, InChiKey only enables a search engine to identify the exact query structure: it cannot be used to support substructure or similarity searches (The same argument applies to technologies that rely on structure keywords describing molecule fragments[8]). Second, it is technically difficult and unrealistic to retrospectively reinsert InChiKeys into all previously written documents, especially for documents in an Intranet environment. Third, InChiKey relies on a set of structure standardization rules. Considering the fact that the definition of aromatic bonds and tautomerism can be ambiguous, the standardization rules can evolve and result in changes in InChiKey. Lastly, chemical structures may not even be available at the time of documentation. For projects extensively using compounds such as natural products, using an in-house compound identifier may be the only option. Bearing these limitations in mind and incorporating some modifications, InChiKey could potentially serve as a specialized form of canonical key for ECKI. The dynamic indexing nature of ECKI generates CKs on the fly, and therefore does not require a CK to be used by authors in the first place. Any change in the CK generation algorithm simply requires a recrawl and reindex procedure. Therefore, the CK used by ECKI does not have to be an InChiKey, but can be any indexable primary key that developers feel most appropriate to use. Many of the limitations of the InChiKey could be eliminated when it is used as an ECKI canonical key.

The RSC Prospect project attempts to markup chemical entities by converting an original text document into an enhanced HTML format, while tagging the entities using SMILES, InChiKeys, etc. However, there are limitations to this approach that are difficult to overcome. First, although semantic markup is supported in some word processing software, the majority of existing corporate documents do not contain such markups, and it would be resource consuming for authors to tag all chemical entities during future writings. Second, markups are annotations, which are not part of the original text body. Most search engines do not query markups, or at least do not support mixing markup tags with keywords in one query. Last but not least, the RSC Prospect method creates a new document for each file crawled, while ECKI does not create a second set of documents. Therefore, the original security settings that are essential for an Intranet environment are preserved.

Future Directions. The implementation of ECKI depends upon the application programming interface (API) that text search engines provide, explaining why the selection of the MOSS search engine for this study. Chemical entities are sometimes represented as object linking and embedding (OLE) binary objects embedded in an Office document, but the interface to OLE chemical entities is not clear to us and, therefore, not implemented in this study. Being able to recognize OLE objects is very desirable, because many documents contain structure objects from ChemDraw, ISIS Draw, etc. Nevertheless, this limitation is somewhat offset in intranet applications, because corporate compound identifiers are often present.

For Internet applications, such as Google Scholar, Google Patent, and SureChem patent database, one big challenge is to improve the accuracy of current optical-compound recognition (OCR) tools. We were unable to evaluate this approach, because the current PDF iFilter skips embedded pictures. In light of some recent advances in OCR,[9,10] it is an interesting future topic to test the maturity of this technology in search applications.

Although we focused on chemical searches in this study, ECKI is essentially a generic approach for indexing entities in disparate domains. Biologists face a similar challenge, where a macromolecule can be presented by multiple identifiers. For example, the Abl gene has multiple identifiers, NM_080104, CG4032, NP_524843, let alone numerous additional associated research tools from various microarray probe sets, siRNA sequences, etc. Searching for the Abl gene in text search engines by a finite set of synonyms always presents the risk of missing critical hits. ECKI has the capability to be used to construct a text search engine with biological intelligence, for example, to find all the papers supporting the connections between "Gleevec" and "Abl". The query would be formulated as "GCCK1234 NEAR GGCK4321", where you have probably guessed that we use GGCK4321 for the GNF gene canonical keyword of "Abl". The concept of a structure database then naturally maps into a sequence database. ECKI can be easily applied to entities in other domains as well.

CONCLUSIONS

ECKI is a viable approach to add domain-specific intelligence to existing text search engines. Here, we demonstrate how this idea can be implemented on top of the Microsoft MOSS search engine and result in a ready-to-use solution to effectively crawl all chemical-containing documents in a corporate intranet (availability will be updated on companion Web site http://carrier.gnf.org/publications/ECKI). We envision that the adoption of ECKI will eventually empower sophisticated chemically aware searches over the Internet to query literature, patents and other valuable data sources. With such chemical intelligence available, the researchers themselves, rather than the search engines, will become the limiting factor of what type of chemical-related queries one can pose.

ACKNOWLEDGMENT

The authors would like to thank Warren Hall and Christopher Hodge in helping set up and customize the

MOSS search environment used for this study. We would also like to thank William Smith for valuable suggestions to improve the manuscript.

REFERENCES AND NOTES

(1) Wikipedia. Web search engines. http://en.wikipedia.org/wiki/Search_engines (accessed Nov 10, 2009).
(2) Tisseghem P.; Fastrup L. *Inside the Index and Search Engines: Microsoft Office SharePoint Server 2007*, 1st ed.; Microsoft Press: Redmond, WA, 2007.
(3) Amato, I. A century of CAS. *Chem. Eng. News* **2007**, *85*, 38–39.
(4) Zhou, Y.; Zhou, B.; Chen, K.; Yan, S. F.; King, F. J.; Jiang, S.; Winzeler, E. A. Large-scale annotation of small-molecule libraries using public databases. *J. Chem. Inf. Model.* **2007**, *47*, 1386–1394.
(5) Banville, D. L. Mining chemical structural information from the drug literature. *Drug Discovery Today* **2006**, *11*, 35–42.
(6) Walkdrop, M. M. Science 2.0—Is open access science the future? *Scientific American Magazine.* **2008**, *298*, 68–73.
(7) Wikipedia. International Chemical Identifier. http://en.wikipedia.org/wiki/International_Chemical_Identifier (accessed Nov 10, 2009).
(8) James C. A.; Gubernator, K. Molecular keyword indexing for chemical structure. U.S. Patent Appl. 0016612 A1, 2007.
(9) Filippov, I. V.; Nicklaus, M. C. Optical structure recognition software to recover chemical information: OSRA, an open source solution. *J. Chem. Inf. Model.* **2009**, *49*, 740–743.
(10) Valko, A. T.; Johnson, A. P. CLiDE Pro: the latest generation of CLiDE, a tool for optical chemical structure recognition. *J. Chem. Inf. Model.* **2009**, *49*, 780–787.

CI900380S