TREC Chemical IR Track 2009: A Distributed Dimensional Indexing Model for Chemical Patent Search

Jay Urbain
Elec. Eng. & Comp. Sci. Department
Milwaukee School of Engr.
Milwaukee, WI
urbain@msoe.edu

Ophir Frieder
Department of Computer Science
Georgetown University
Washington, DC
ophir@cs.georgetown.edu

Abstract

For the TREC-2009 Chemical IR Track, we explore development of a distributed information retrieval system based on a dimensional data model. The indexing model supports named entity identification and aggregation of term statistics at multiple levels of patent structure including individual words, sentences, claims, descriptions, abstracts, and titles.

The system was deployed across 15 Amazon Web Services (AWS) Elastic Cloud Compute (EC2) instances and 15 Elastic Block Storage (EBS) database shards to support efficient indexing and query processing of the relatively large index generated from indexing each individual word (sans stop words) in the 100G+ collection of chemical patent documents.

The query processing algorithm for *technology survey search* and *prior art search* uses information extraction techniques and locally aggregated term statistics to help disambiguate candidate entities and terms in context. Query processing for *prior art search* automatically generates a structured query based on the relative distinctiveness of individual terms and candidate entity phrases from the query patent's claims, abstract, and title sections. For both the *technology survey* and *prior art search*, we evaluated several probabilistic retrieval functions for integrating statistics of retrieved named entities with term statistics at multiple levels of document structure to identify relevant patents.

1. Introduction

The TREC Chemistry Track for 2009 was organized to evaluate the statistical significance on the ranking of information retrieval (IR) systems and the scalability of IR systems when dealing with chemical patents [1]. A test collection was assembled from approximately 1.2M patent files (approximately 100G) of full-text chemical patents and research papers to evaluate two ad-hoc retrieval tasks common to patent investigation: *technology survey* and *prior art search*.

The goals of *technology survey search* and *prior art search* are fundamentally different. *Technology survey* search is similar to ad-hoc retrieval targeting patent documents using a natural language query to satisfy an information need. Systems are required to return a set of documents that is relevant to this information need. The goal of the technology survey evaluation is to identify how current IR methods adapt to text containing chemical names and formulas. Systems for the *technology survey* task are evaluated using a pooling, sampling, and expert evaluation methodology.

The goal of *prior art search* is to evaluate the validity of a patent claim. In this task, systems attempt to identify prior art that may invalidate a patent claim. The query set for this evaluation consists of 1000 patent files with prior art references removed. Systems are required to return a set of documents relevant to the prior art of claims stated in the patent. Of special interest in this task was to consider three types of topics: full text patents, description only, and claims only. Systems for this task are evaluated automatically using the known references for each patent.

Chemical and patent information retrieval are challenging tasks. Chemical IR requires a chemical named entity identification strategy for dealing with large multiword terms, synonyms, acronyms, and morphological variants used for identifying the same chemical concept. For example, *Dipeptidyl peptidase-IV* inhibitor can also be referred to as *Dipeptidyl peptidase-4, DPP4, DPP-4*, or *dipeptidylaminopeptidase*. *Guggulsterone* can be identified as *Pregna-4,17-diene-3,16-dione*, *Guggulsterone-E, Guggulsterone-Z, trans-guggulsterone*, or as the *guggulu* steroid extract. Many chemical terms also have hierarchical hyponym-hypernym relationships. For example, *silver halides, AgX*, could include *silver bromide (AgBr), silver iodide (AgI)*, or *silver fluorides*.

Chemical patent retrieval requires not only the identification of named entities, but the relationships between entities in the context of how they are applied. The structure of patent documents is important for identifying and validating specific claims as they serve as the legal basis of the patent, however descriptions in other portions of the documents may provide important context and alternative nomenclature [2].

To meet the needs of chemical IR, patent IR, and the scalability needs of large patent collections, we have developed a distributed search engine based on a dimensional data model. The model supports chemical named entity identification and aggregation of term statistics at multiple levels of patent structure including individual words, sentences, claims, descriptions, abstracts, and titles [3].

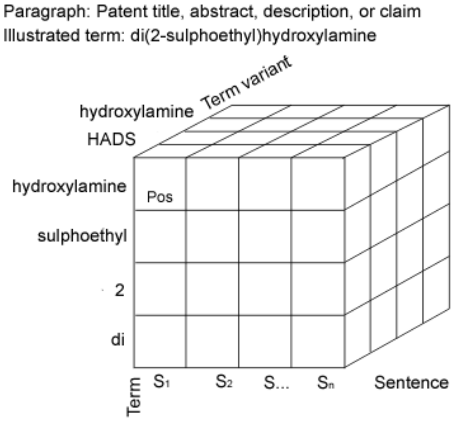
The query processing algorithm uses information extraction and locally aggregated term statistics to identify and disambiguate candidate named entities in context. One of our future goals is to use candidate entities to identify additional synonymous chemical terms using public chemical databases. Processing for prior art search automatically generates a structured query based on the relative distinctiveness of individual terms and candidate chemical entities from the query patent's claims, abstract, and title sections. Finally, for both the technology survey and prior art search, we evaluated several probabilistic retrieval functions for integrating statistics of retrieved named entities with term statistics at multiple levels of document structure to identify relevant patents. Our primary objective in this research was to develop a scalable and flexible system for further research.

We first describe our distributed indexing model, followed by the system description, indexing process, query processing, retrieval functions, and our results.

2. Distributed Dimensional Data Model

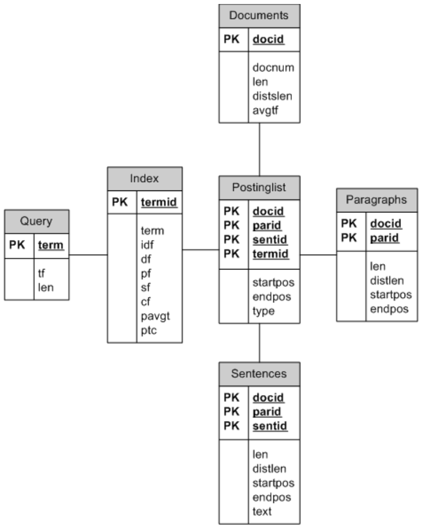
Paragraphs, sentences, and terms, representing complete topics, thoughts, and units of meaning respectively, provide a logical breakdown of document lexical structure into finer levels of meaning and context [4]. We capture these hierarchical relationships within a search index based on a dimensional data model. As shown in Figure 1, this dimensional model can be logically represented as an n-dimensional cube. Where each patent document is represented as a series of paragraphs (title, abstract, descriptions, and claims). Each paragraph is represented as a series of sentences, and each sentence is represented as a sequence of individual terms. Such a model facilitates search for multi-word terms and efficient aggregation of term statistics within multiple levels of patent structure.

Figure 1. Search index based on dimensional model.


Paragraph: Patent title, abstract, description, or claim
Illustrated term: di(2-sulphoethyl)hydroxylamine
Term variant
hydroxylamine
HADS
hydroxylamine
Pos
sulphoethyl
2
di
Term
S_1
S_2
S...
S_n
Sentence

As shown in **Figure 2**, we represent the dimensional index as a star schema [5, 6] with a *dimension* table for each level of document structure (document, paragraph, sentence, term) and one *central fact* table or *postinglist*. The “grain”, i.e., the smallest non-divisible element of the database, is the individual word. Sentences aggregate words in sequence by position, paragraphs aggregate sentences, and documents aggregate paragraphs.

Figure 2. Search index based on dimensional model.



The index can be extended to include additional dimensions, and allows for efficient formulation of SQL search queries. By indexing each individual word, queries can be developed for searching single- and multi-word terms, and term statistics can be aggregated over different levels of document structure.

3. System Description

Indexing, retrieval, and analysis applications were developed in Java. The *MySQL* 5.0 database with the *MyISAM* storage engine was used for index storage and retrieval. 15 Amazon Elastic Cloud Compute (EC2) *m1.small* instances based on the *Ubuntu* Hardy base (ami-ef48af86) machine image were allocated for processing each of 15 database shards [7]. Each shard was roughly equivalent to the Chemical IR track collection distribution. For example, one shard for EP 000000, one shard for EP 000001, one shard for US 020060, etc. Each database shard included a dimensional data model for its portion of the collection, and a dimensional index of *PubChem* [8] terminology for synonym identification. Elastic Block Storage (EBS) volumes of 350G were allocated for each compute instance to accommodate the size of the index and the need to insure persistence of the database if a compute instance was restarted. An *m1.small* EC2 compute unit consisted of 1.7 GB memory, 1 32-bit virtual core, and 160 GB of storage. Experiments with larger dual-core compute instances improved indexing performance 2-fold per instance, but did not significantly improve query performance. It takes approximately 2 days to construct the entire index. Each *m1.small* instance cost $0.10 per hour. Additional charges are encountered for data loading and storage. Each compute instance performed roughly equivalent to a standard Pentium 4 laptop with 2G of memory. The total cost of the experiment was approximately $1,000 though this included a fair amount of trial and error to get things running.

4. Indexing Process

The indexing process includes the following:

1. *Lexical Partitioning*: Documents are parsed into title, abstract, descriptions, and claims. Each is subsequently parsed into paragraphs, and these paragraphs are parsed into sentences.
2. *Tokenization*: Sentence terms are tokenized, stop words removed, and lexical variants are normalized. Porter stemming [9] is used on each token with the following exceptions: all upper case, mixed case, alpha-numeric terms. Small “s” is also stripped from all upper-case terms.
3. *Indexing*: Each individual word is stored in the index with positional information and its paragraph type (title, abstract, description, or claim).

4. Query Processing

Structured query generation for both *technical survey* and *prior art* search is illustrated with the following abbreviated example: *“We are a new pharmaceutical company that is interested in entering the area of Dipetidyl peptidase-IV inhibitors…”*

1. Sentences are extracted.
2. Part-of-speed tagging is performed.
3. Candidate entities are identified by locating non-recursive noun phrases: *[pharmaceutical_NN company_NN], [Dipetidyl_NN peptidase-IV inhibitors_NNS]*.
4. Candidate entities are verified in the index, and their normalized IDF (Inverse Document Frequency normalized to between 1 and 0) is verified against a minimum threshold of 0.15. Note: Table 1 illustrates resolved entities for our example.
5. Stop and function words are removed.

Table 1. Resolved Entity

Resolved entity	Synonyms
[Dipetidyl peptidase-IV inhibitor]	[Dipeptidyl peptidase-4 inhibitor] [DPP-4] [dipeptidylaminopeptidase]

Paragraph Queries

For *technology survey*, the topic is used as the basis of the paragraph query. For *prior art search*, paragraph queries are generated from the query patent title, abstract, and for each of (up to) the first 20 claims. The top 7 terms by NIDF (normalized inverse document frequency) above a minimum threshold of 0.10 (0.0 to 1.0) is used to generate a query for each paragraph.

The top 500 paragraphs are retrieved using the probabilistic BM25 retrieval function [10] shown in equation (1). BM25 is implemented using standard SQL.

BM25: (1)

$$\sum_{wq} \ln\left(\frac{N - df + 0.5}{df + 0.5}\right)\left(\frac{(k_1 + 1) * tf_d}{k1 * (1 - b) + b * (\frac{docLen}{avgDocLen}) + tfd}\right)\left(\frac{(k_3 + 1) * tfq}{k_3 + tf_q}\right)$$

Note: We used k1=1.4, k2=0, k3=7, and b=0.75

Entity Queries

From the cached list of candidate entities from across title, abstract, and claim paragraphs, entities are selected as follows:

1. Entities must occur in at least 2 paragraphs and have a NIDF > 0.15. We have found that this drastically reduces the number of spurious entity phrases.
2. Entity phrases are ranked by NIDF and their log frequency of occurrence.
3. The top 20 remaining entity phrases are searched within the context of title, abstract, description, and claims paragraphs of target patent documents. We do not place a limit on retrieved results.

The following abbreviated query illustrates entity search for

“*Dipetidyl peptidase-IV inhibitor*”. All queries are distributed across all database shards and results are aggregated:

```
select i1.term, p1.docid, p1.parid, p1.sentid, p1.seq, p1.section, d.docnum
    from invertedindex_qc i1, postinglist_qc p1, paragraphresults_qc d
    where i1.term=' dipetid’
    '  and i1.termid=p1.termid   and p1.docid=d.docid   and
    d.parid=p1.parid
and exists (
    select *   from invertedindex_qc i2, postinglist_qc p2
    where i2.term=' peptidas'  and i2.termid=p2.termid   and
    p1.docid=p2.docid   and p1.parid=p2.parid   and p1.sentid=p2.sentid
    and p1.section=p2.section   and abs(p2.seq-p1.seq)<=5
and exists (…
```

Document Query

The top 2000 target patent documents are retrieved using a BM25 formulated query of the top 20 individual terms (by NIDF) selected from the top 20 candidate entities. The idea is to select a relatively even distribution of terms across all patent claims.

5. Retrieval Functions

Table 2 shows the similarity coefficients (SC) computed from the paragraph, entity, and document query results for each retrieved patent document. Each individual score is normalized.

Table 2. Retrieval Function Similarity Coefficients

Similarity Coefficient	**Definition**
DocScNorm	Document normalized BM25
ParScNorm	Paragraph (Title, Abstract, Claims, Description) normalized BM25
ConceptDocCountNorm	Distinct count of candidate entities per document
ConceptDocIdfSumNorm	Normalized IDF summation of all distinct entities per document
ConceptMaxParCountNorm	Max distinct count of candidate entities per paragraph
ConceptMaxParIdfSumNorm	Max normalized IDF summation of all distinct entities per paragraph

A linear weighted sum (2) is used to generate various retrieval functions by weighting and combining similarity coefficients (SC) for each target document.

$$SC_{composite} = w_1SC_1 + w_2SC_2 + \ldots + w_nSC_n \qquad (2)$$

Table 3 shows probabilistic model weighting for each search coefficient.

Table 3. Similarity Coefficients

	Parameters	**Similarity Coefficient**
x111111	1 1 1 1 1 1	DocScNorm ParScNorm ConceptDocCountNorm ConceptDocIdfSumNorm ConceptMaxParCountNorm ConceptMaxParIdfSumNorm
x100000	1 0 0 0 0 0	DocScNorm ParScNorm ConceptDocCountNorm ConceptDocIdfSumNorm ConceptMaxParCountNorm ConceptMaxParIdfSumNorm
x010000	0 1 0 0 0 0	DocScNorm ParScNorm ConceptDocCountNorm ConceptDocIdfSumNorm ConceptMaxParCountNorm ConceptMaxParIdfSumNorm
x110000	1 1 0 0 0 0	DocScNorm ParScNorm ConceptDocCountNorm ConceptDocIdfSumNorm ConceptMaxParCountNorm ConceptMaxParIdfSumNorm
x111000	1 1 1 0 0 0	DocScNorm ParScNorm ConceptDocCountNorm ConceptDocIdfSumNorm ConceptMaxParCountNorm ConceptMaxParIdfSumNorm
x110100	1 1 1 0 0 0	DocScNorm ParScNorm ConceptDocCountNorm ConceptDocIdfSumNorm ConceptMaxParCountNorm ConceptMaxParIdfSumNorm
x110010	1 1 1 0 0 0	DocScNorm ParScNorm ConceptDocCountNorm ConceptDocIdfSumNorm ConceptMaxParCountNorm ConceptMaxParIdfSumNorm
x110001	1 1 1 0 0 0	DocScNorm ParScNorm ConceptDocCountNorm ConceptDocIdfSumNorm ConceptMaxParCountNorm ConceptMaxParIdfSumNorm

Table 4. Results

Retrieval Function	**111111**	**100000**	**010000**	**110000**	**111000**	**110100**	**110010**	**110001**
num_ret	98300	98300	98300	98300	98300	98300	98300	98300
num_rel	2860	2860	2860	2860	2860	2860	2860	2860
num_rel_ret	1508	1291	1298	1492	1509	1507	1503	1491
map	0.0118	0.0063	0.0074	0.0080	0.0086	0.0094	0.0077	0.0076
gm_map	0.0047	0.0016	0.0030	0.0041	0.0043	0.0041	0.0040	0.0040
Rprec	0.0094	0.0029	0.0058	0.0051	0.0057	0.0062	0.0042	0.0047
bpref	***0.5458***	**0.4705**	**0.4552**	**0.5369**	**0.5433**	**0.5423**	**0.5406**	**0.5365**
recip_rank	0.0444	0.0228	0.0240	0.0284	0.0275	0.0281	0.0228	0.0177
iprec_at_recall_0.00	0.0546	0.0320	0.0354	0.0379	0.0377	0.0371	0.0312	0.0271
iprec_at_recall_0.10	0.0391	0.0137	0.0192	0.0173	0.0225	0.0276	0.0187	0.0172
iprec_at_recall_0.20	0.0267	0.0127	0.0165	0.0160	0.0209	0.0260	0.0159	0.0160
iprec_at_recall_0.30	0.0159	0.0122	0.0152	0.0148	0.0151	0.0151	0.0147	0.0147
iprec_at_recall_0.40	0.0147	0.0119	0.0117	0.0135	0.0140	0.0139	0.0135	0.0135
iprec_at_recall_0.50	0.0121	0.0109	0.0090	0.0115	0.0117	0.0117	0.0116	0.0115
iprec_at_recall_0.60	0.0107	0.0098	0.0073	0.0102	0.0105	0.0105	0.0104	0.0102
iprec_at_recall_0.70	0.0074	0.0072	0.0041	0.0067	0.0072	0.0072	0.0075	0.0067
iprec_at_recall_0.80	0.0039	0.0043	0.0023	0.0035	0.0040	0.0039	0.0035	0.0035
iprec_at_recall_0.90	0.0022	0.0030	0.0015	0.0022	0.0022	0.0022	0.0022	0.0022
iprec_at_recall_1.00	0.0005	0.0017	0.0001	0.0003	0.0003	0.0003	0.0003	0.0003
P_5	0.0101	0.0061	0.0081	0.0061	0.0061	0.0061	0.0061	0.0020
P_10	0.0091	0.0030	0.0091	0.0051	0.0051	0.0030	0.0061	0.0040
P_15	0.0088	0.0034	0.0074	0.0054	0.0040	0.0040	0.0054	0.0047
P_20	0.0081	0.0030	0.0056	0.0051	0.0040	0.0051	0.0056	0.0045
P_30	0.0071	0.0030	0.0074	0.0044	0.0047	0.0054	0.0064	0.0040
P_100	0.0064	0.0038	0.0069	0.0047	0.0054	0.0056	0.0052	0.0047
P_200	0.0061	0.0037	0.0072	0.0064	0.0057	0.0057	0.0056	0.0064
P_500	0.0069	0.0045	0.0078	0.0065	0.0064	0.0064	0.0062	0.0065
P_1000	0.0152	0.0130	0.0131	0.0151	0.0152	0.0152	0.0152	0.0151

6. Results

Results for the first 100 query patents are shown in Table 4. The document retrieval similarity coefficient (SC) resulted in a *bpref* measurement of **0.4705**, competitive with the top TREC results. The paragraph SC was **0.4552**. Integrating document with paragraph SC's improved the result to **0.5369**. Integrating individual entity SC's further improved the results to **0.5484**. These results clearly demonstrate the efficacy of integrating.

From these results we have noted a number of issues and opportunities for further development of these models:

- Limiting individual paragraph retrieval and subsequent named entity retrieval to the top 500 paragraphs is too small for a collection of greater than 1M documents. Cursory analysis of relevant document not included in our final results were often identified by our document retrieval SC, but were not included in the relatively short list of paragraphs.
- Identifying candidate entities is a work in progress, though selecting document retrieval terms from individual entity terms appears to be an effective strategy.
- In this work sought to demonstrate the efficacy of multievidentiary contextual models. In future work we plan to develop more refined probabilistic models for integrating SC's.
- Ablation studies to identify the most effective sections of source and target documents for technology survey and prior art search.
- Identifying the optimal size and number of shards for indexing and query model performance.
- More effective probabilistic models.

7. Conclusion

We explored development of a distributed multidimensional indexing model to enable efficient search and aggregation of entities and terms at multiple levels of document context and distributed across a cloud computing cluster.

Several probabilistic retrieval models for integrating term statistics with entity search using multiple levels of document context to improve the performance of chemical patent invalidity search. Relevance measurements were integrated within a probabilistic retrieval model for re-ranking of results. Results from our integrated approach outperformed baseline results and exceeded the top results reported at the TREC forum, demonstrating the efficacy of our approach.

References

1. TREC-CHEM 2009 Track Guidelines, http://wiki.ir-facility.org.
2. Fujii, Atsushi, Iwayama, M., Kando, N. "Introduction to the special issue on patent processing," Information Processing and Management 43 (2007) 149-1153.
3. Urbain, J., Frieder, O., & Goharian, N. (2008b). Probabilistic Passage Models for Semantic Search of Genomics Literature, *Journal of the American Society of Information Science and Technology*.
4. Urbain, J., Frieder, O., & Goharian, N. (2008). A Dimensional Retrieval Model for Integrating Semantics and Statistical Evidence in Context for Genomics Literature Search. *Computers in Biology and Medicine*.
5. J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M Venckatrao, F Pells, "Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals. Data Mining and Knowledge Discovery," Volume 1, Issue 1, 1997.
6. R. Kimball, "The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses," Ralph, John Wiley, 1996.
7. Amazon Web Services, http://aws.amazon.com/documentation/
8. PubChem, National Center for Biotechnology Information (NCBI), http://pubchem.ncbi.nlm.nih.gov.
9. M.F. Porter, "An algorithm for suffix stripping," Program, 14:130–137, 1980.
10. S. Robertson, S. Walker, "Okapi/Keenbow at TREC-8," NIST Special Publication 500-246, 2000.