Mining the Biomedical Literature
State of the Art, Challenges and Evaluation

Hagit Shatkay
School of Computing, Queen’s University

ISMB, June, 2005
Biomedical Literature (and text) Mining Publications

BLIMP - See it all from here!

A Forum for collection, compilation and exchange of publications on biomedical text mining

http://blimp.cs.cs.queensu.ca
A Quick Introduction to Hidden Markov Models
The Many Facets of HMMs...

Natural Language Processing

DNA/Protein Analysis
Let me check my HMM...

Medical Decision Making

Robot Navigation and Planning
What are HMMS?

Models that are:

• **Stochastic** (probability-based)

• **Generative**
  Provide a putative production process for generating data.

• **Satisfying the Markov Property**
  The *present* state *summarizes* the past.
  *Future events* depend only on the *current situation* – not on the preceding ones.
Example: Weather modeling and prediction

Fair
\[ \Pr(T<5^\circ)=0.01 \]
\[ \Pr(Overcast)=0.02 \]
\[ \Pr(Precipitation)=0.1 \]
\[ \Pr(Wind>15\text{mph})=0.2 \]

Breezy & Cold
\[ \Pr(T<5^\circ)=0.8 \]
\[ \Pr(Overcast)=0.5 \]
\[ \Pr(Precipitation)=0.3 \]
\[ \Pr(Wind>15\text{mph})=0.9 \]

Rainy
\[ \Pr(T<5^\circ)=0.3 \]
\[ \Pr(Overcast)=0.8 \]
\[ \Pr(Precipitation)=0.95 \]
\[ \Pr(Wind>15\text{mph})=0.6 \]

Winter Storm
\[ \Pr(T<5^\circ)=0.999 \]
\[ \Pr(Overcast)=0.9 \]
\[ \Pr(Precipitation)=0.95 \]
\[ \Pr(Wind>15\text{mph})=0.85 \]
The Building Blocks of HMMs

An HMM is a tuple: \( \lambda = \langle S, V, A, B, \pi \rangle \)

- **States** \((S = s_1, \ldots, s_N)\) – Hidden
- **Observations** \((V = v_1, \ldots, v_M)\)

**Parameters:**

- **A:** Transition matrix \( A_{ij} = \Pr( q_{t+1} = s_j \mid q_t = s_i ) \)
- **B:** Observation matrix \( B_{ik} = \Pr( o_t = v_k \mid q_t = s_i ) \)
- **\( \pi \):** Initial distribution \( \pi_i = \Pr( q_1 = s_i ) \)

\[
\sum_{j=1}^{N} A_{ij} = 1 \\
\sum_{k=1}^{M} B_{ik} = 1 \\
\sum_{j=1}^{N} \pi_i = 1
\]
Examples Revisited

States: Possible parts of speech (e.g. noun, verb)
Observations: Uttered words
Transitions: Order of sentence fragments

Natural Language Processing

States: Positions for nucleotides deletion/matching/insertion
Observations: Nucleotides
Transitions: Nucleotides order in the DNA

States for modeling purpose.
Medical Decision Making

**States:** Patient's (varying) condition
**Observations:** Instrument Readings
**Transitions:** Changes due to treatment

Robot Navigation and Planning

**States:** Robot's position
**Observations:** Sensors Readings
**Transitions:** Changes due to movement

* A physical notion of state.
HMMs: The Three Problems

**Problem 1:** Given a model $\lambda$ and a sequence of observations $O=O_1,...,O_T$, find $O$’s probability under $\lambda$, $Pr(O|\lambda)$.

**Problem 2:** Given a model $\lambda$ and a sequence of observations $O=O_1,...,O_T$, find the best state sequence $Q=q_1,...,q_T$ explaining it.

**Problem 3:** Given a sequence of observations $O=O_1,...,O_T$, find the best model $\lambda$ that could have generated it.
HMMs in NLP

Modeling Documents [Seymore et al. 99, Freitag & McCallum 99]

**Task:** Segment a document header and extract title/author etc.  
**Typically,** document headers have (in this order):  
*Title; Author; Affiliation; Date; Abstract;*

+ some variation.

**HMM captures this Typical+variation structure:**
Modeling Parts of Speech tags and Sentence Structure

[Example, Charniak 93, Allen 95]

Again, based on *typical* word use and sentence structure.

The dog walked.

**The dog walked.**

- Det **Noun** **Verb**

Walk the dog!

**Walk the dog!**

- **Verb** **Det** **Noun**
Back to

Mining the BioMedical Literature

©Hagit Shatkay
Literature can be used to explain and predict:

Pathways:

Correlation in expression level of genes/proteins:

http://www.ana.ed.ac.uk/rnusse/wntwindow.html

Eisen et al. PNAS, 95;25, 1998

gene ↔ symptom ↔ disease relationships

TNFRSF1B
Insulin Resistance
Type 2 diabetes
Overview

➡ Text Processing and Mining
  - NLP
  - Information Extraction
  - Information Retrieval

• Applications in Bio-Medical Literature
• Detailed Examples
• Evaluation Issues
• Conclusion
Disciplines Handling Text

• Natural Language Processing (NLP)
• Information Extraction
• Information Retrieval
• All aspects of automated natural-language communication: Processing and understanding spoken, handwritten and *printed* language.

• Stages in natural language communication:
  Speaker: *Intention, Generation, Synthesis*
  Hearer: *Perception, Analysis, Disambiguation, Incorporation*

• Stages relevant to mining on-line text:
  1. *Analysis*: Part of speech tagging, Parsing, Semantic Interpretation.
  2. *Disambiguation*.

[Russell&Norvig95] Ch. 22,23, and references therein.
[Charniak93, Allen95, Manning&Schutze99]
Natural Language Issues

1. **Ambiguity** (sentence/syntax level)
   - “…flies like an eagle…” (Verb Prep Det Noun vs. Noun Verb Det Noun)
   - “…country chicken egg dispenser…”
   - “Children who won’t eat their vegetables won’t know the difference if you force them through a sieve…”
     [Second Helping Please, a Canadian cookbook from 1968]

2. **Polysemy** (word level)
   Multiple meaning for the same word. (context dependent)
   Town mouse, Country mouse, Computer mouse…

3. **Synonymy** (word level)
   Multiple words for the same meaning  [See thesaurus…]
Information Extraction

Automated identification of certain kinds of facts from text. Typically, populating a relational table from the identified facts.

Example:

<table>
<thead>
<tr>
<th>Kinase</th>
<th>Protein</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC-kinase</td>
<td>X</td>
</tr>
</tbody>
</table>

Phosphorylation

“...BES1 is phosphorylated and appears to be destabilized by the glycogen synthase kinase-3 (GSK-3) BIN2...”

<table>
<thead>
<tr>
<th>Kinase</th>
<th>Protein</th>
</tr>
</thead>
<tbody>
<tr>
<td>glycogen synthase kinase-3 (GSK-3)</td>
<td>BES1</td>
</tr>
</tbody>
</table>

[Added references: Cowie&Lehnert96, Cardie97, Grishman 97]
Information Extraction (cont.)

What it takes:

• Identify the relevant sentences
• Parse to extract the relationships
• Obtain domain-specific information
• Assume “well-behaved” fact sentences

However:

• Missing an ill-formed fact is acceptable if the database is big and redundant.
• Using co-occurrence relationships alone does not require parsing or good fact-structure.
Information Retrieval

Setting:

- A lot of documents
- Specific information needed about a subject (posed as a query)

Goal:

Retrieve* exactly those documents satisfying the information need.

*Automatically(!!!)
Means:

• Boolean query, Index based (e.g. “Gene and CDC”)
  😞 Polysemy
  (Not interested in “Center for Disease Control”)
  😞 Synonymy
  (PR55, won’t be retrieved)

• Similarity query, Vector based.
**Representing Text**

- **Lexical Analysis**: Tokenization, breaking the text into basic building blocks. (*Paragraphs, Sentences, Phrases, Words*)

* Required for both IE and IR
The HXT genes (HXT1 to HXT4) of the yeast Saccharomyces cerevisiae encode hexose transporters. We found that transcription of these genes is induced 10- to 300-fold by glucose. Analysis of glucose induction of HXT gene expression revealed three types of regulation: (i) induction by glucose independent of sugar concentration (HXT3); (ii) induction by low levels of glucose and repression at high glucose concentrations (HXT2 and HXT4); and (iii) induction only at high glucose concentrations (HXT1).

The lack of expression of all four HXT genes in the absence of glucose is due to a repression mechanism that requires Rgt1p and Ssn6p. GRR1 seems to encode a positive regulator of HXT expression, since grr1 mutants are defective in glucose induction of all four HXT genes. Mutations in RGT1 suppress the defect in HXT expression caused by grr1 mutations, leading us to propose that glucose induces HXT expression by activating Grr1p, which inhibits the function of the Rgt1p repressor.
Representing Text *

• **Lexical Analysis**: Tokenization, breaking the text into basic building blocks. (*Paragraphs, Sentences, Phrases, Words*)

• **Parsing/Processing**: (*Identify Noun-phrases, Verb-phrases, names and/or other proper nouns, stem/normalize*)

* Required for both IE and IR
The **HXT genes** (**HXT1** to **HXT4**) of the yeast *Saccharomyces cerevisiae* encode hexose transporters. We found that transcription of these genes is induced 10- to 300-fold by glucose. Analysis of glucose induction of HXT gene expression revealed three types of regulation: (i) induction by glucose independent of sugar concentration (HXT3); (ii) induction by low levels of glucose and repression at high glucose concentrations (HXT2 and HXT4); and (iii) induction only at high glucose concentrations (HXT1).

The lack of expression of all four HXT genes in the absence of glucose is due to a repression mechanism that requires Rgt1p and Ssn6p. GRR1 seems to encode a positive regulator of HXT expression, since grr1 mutants are defective in glucose induction of all four HXT genes. Mutations in RGT1 suppress the defect in HXT expression caused by grr1 mutations, leading us to propose that glucose induces HXT expression by activating Grr1p, which inhibits the function of the Rgt1p repressor.
Representing Text *

• **Lexical Analysis**: Tokenization, breaking the text into basic building blocks. (*Paragraphs, Sentences, Phrases, Words*)

• **Parsing/Processing**: (*Identify Noun-phrases, Verb-phrases, names and/or other proper nouns, stem/normalize*)

• **Storage, supporting quick access to facts/documents**: (*Implementation issues, create a database or a compact file structure, hold for each term/word all documents containing it or for all documents all the structures that occur in it, etc.*)

* Required for both IE and IR
Information Extraction - Details

What is Extracted?

- **Entities**: Specific terms in the text. (e.g. Genes, Proteins, Disease names)

- **Relationships**: Text indicating interactions between entities. (e.g. Phosphorylation, Activation of a gene by a transcription factor.)

- **Events**: Text reporting a *temporal* occurrence, such as a company acquisition or the translation of a gene.

Why is it Hard?

- **Entities** may occur under many different names (synonyms) and reference forms (anaphora - this/that/the former). Unrelated entities may share a name (polysyms).

- **Relationships and events** can be stated in various styles and indirect ways.
Information Extraction – Details (cont.)

Stages in Information Extraction

• **Tokenization.** Breaking the text into its basic building blocks.

• **Possible: Morphological and Lexical Analysis.** Tagging the sentences with parts-of-speech, and disambiguating sense of ambiguous terms.

• **Syntactic Analysis.** Identifying the explicit phrases constituting the sentence (deep vs. shallow parsing.)

• **Domain Analysis.** Using the above components to actually find the entities, facts and events in the text, relate them to each other, and produce the desired output.
Information Extraction – Details (cont.)

Rule-Based

Explicit rules for identifying entities and relations are defined.

```
NamedPerson() :-
  <Title> <Name> <Name>  | // Mr. John Smith
  <Person Name> <Person Name>  | // John Smith
  <Name> <CapL>'.' <Name>  | // John J. Smith
  <Name> <Name> <Suffix> ; // John Smith Jr.
Title :- {‘Mr.’, ‘Mrs.’, ‘Ms.’ ‘Dr.’ ‘Prof.’ ‘Col.’ …} ;
Name :- <Person Name> | <CapL> <LowL>+ ;
Person Name :- {Albert, Arthur, Alex,….};
CapL :- {A-Z}; LowL :- {a-z}; Suffix :- {'Jr.', 'Sr.' …}
```

Rules can be:

• **Hand-coded** [e.g. Appelt et al. 95, Hobbs et al. 97]

• **Automatically acquired from examples** (Supervised learning)
  [e.g. Riloff 93, Califf 98, Soderland 99]

• **Automatically acquired, no examples** (Unsupervised learning)
  [e.g. Yangarber et al. 2000]
Information Extraction – Details (cont.)

Statistical Machine-Learning (HMMs)

Probabilistic models for “typical” sequences denoting entities and relations (with some noise around them).

Example:
[Bikel et al. 97,99]

For every name-class, C, a bi-gram, Markov probability $Pr^C(W_i|W_{i-1})$.

Induces: $Pr(C|W_1…W_n) \propto Pr(W_1…W_n|C) = \prod Pr^C(W_i|W_{i-1})$

High-level, transition and emission probabilities: $Pr(C|C, W), Pr(W|C, W)$
Information Extraction – Details (cont.)

Low-level issues

Part-of-Speech (PoS) Tagging
Initial assignment of syntactic categories to a sequence of words. Tags may include: noun, verb, determiner, adjective, adverb, proper noun, etc.

PoS Tagging Paradigms:
• Rule based [Brill 92, Brill 99]
• Stochastic [Charniak93, Allan95…]
• Supervised vs. Unsupervised [Schütze 93], more…
Low-level issues

Parsing

Complete assignment of syntactic structure to a word sequence. Including: Noun phrase, Verb phrase, Prepositional phrase, etc.

• **Full parsing**: Full assignment of every word and phrase to a syntactic structure. [e.g. Grishman95, NYC PROTEUS MUC-6] Slow, and error-prone...

• **Shallow/Partial parsing**: Partial assignment of syntax structures. “… parse conservatively – only if there is strong evidence of the correctness of a reduction.” [Grishman97 ] [Appelt et al. 95, Appelt & Israel 99, Feldman et al. 2000]
Information Retrieval – Details

Boolean Queries

**DB:** Database of documents.

**Vocabulary:** $\{t_1, \ldots, t_M\}$ (Terms in DB, produced by the tokenization stage)

**Index Structure:** Holds for each term, the identifiers of all the documents containing it.
Information Retrieval – Details

The Vector Model

**DB:** Database of documents.

**Vocabulary:** \( \{ t_1, \ldots, t_M \} \) \{Terms in DB\}

**Document** \( d \in DB \): Vector, \( \langle w_1^d, \ldots, w_M^d \rangle \), of weights.

Weighting Principles

- **Document frequency:** Terms occurring in a *few* documents are *more useful* than terms occurring in *many*.
- **Local term frequency:** Terms occurring *frequently* within a document are likely to be *significant* for the document.
- **Document length:** A term occurring the same # of times in a long document and in a short one has *less significance* in the *long* one.
- **Relevance:** Terms occurring in documents judged as *relevant* to a query, are *likely to be significant* (WRT the query).

[Sparck Jones et al. 98]
Some Weighting Schemes:

\[ W_i^d = \begin{cases} 1 & \text{if } t_i \in d \\ 0 & \text{otherwise} \end{cases} \]

Binary

\[ W_i^d = f_i^d = \text{# of times } t_i \text{ occurs in } d \]

TF

\[ W_i^d = \frac{f_i^d}{f_i} \quad \text{(where } f_i = \text{# of docs containing } t_i) \]

TF X IDF

(one version...)

Example:

Medium body beans have made Colombia famous for its flavorful coffee with a slightly dry acidity.

\(<..., \text{ bean, beer, cat, coffee, colombia, ... } >\)

\(<..., 1, 0, 0, 1, 1, ... >\)
Information Retrieval - Details (cont.)

Vector-Based similarity

Document $d = <w_1^d, \ldots, w_M^d> \in DB$

Query $q = <w_1^q, \ldots, w_M^q>$ (q could itself be a document in DB…)

$$\text{Sim}(q,d) = \cosine(q,d) = \frac{q \cdot d}{|q||d|}$$

[Salton89, Witten et al99] Introductory IR.
Information Retrieval - Details (cont.)

**Probabilistic Models**

Query $q$; Document $d$, with representation $D$

- **Goal:** Find all $d$’s such that $\Pr(d \text{ is relevant to } q \mid D)$ is high

**Maximize log-odds:**

$$\log \left[ \frac{\Pr(d \text{ is relevant to } q \mid D)}{\Pr(d \text{ is Irrelevant to } q \mid D)} \right]$$

**Using likelihoods:**

$$\log \left[ \frac{\Pr(D \mid d \text{ relevant to } q)}{\Pr(D \mid d \text{ Irrelevant to } q)} \right]$$

- **View** $d= <w_1^d, \ldots, w_M^d>$ as a probability distribution over terms, (Language Model, $LM(d)$)

Query $q$ is viewed as a sample generated from a distribution.

**Sim**$(q,d) \propto \Pr(q \mid LM(d))$

[Sparck Jones et al. 98, Sahami98, Ponte&Croft 98, Hoffman 99] and references therein.

Motivation: Overcoming synonymy and polysemy.
Reducing dimensionality.

Idea: Project from “explicit term” space to a lower dimension, “abstract concept” space.

Methodology: Singular Value Decomposition (SVD) applied to the document-term matrix.
Highest singular values are used as the features for representing documents.

[Dumais et al. 88, Deerwester et al. 90, Papdimitriou et al. 98, Jiang and Littman 2000]
Information Retrieval- Details (cont.)

Text Categorization

Placing documents in their “right drawer”, making them easy-to-find for the user.

Either manually by indexers, or automatically, through classification.
Rule-Based Text Classification

A knowledge-engineering approach.

Boolean rules (DNF), based on the presence/absence of specific terms within the document, decide its membership in the class. (e.g. the CONSTRUE system [Hayes et al. 90,92]

Example:

If ( (<GENE_Name> ∧ transcript) ∨
    ((<GENE_Name> ∧ Western Blot) ∨
     ((<GENE_Name> ∧ Northern Blot))

Then GeneExpressionDoc
Else GeneExpressionDoc
Information Retrieval-Details (cont.)

Machine Learning for Text Classification *(supervised)*

- Take a *training set* of pre-classified documents
- Build a model for the classes from the training examples
- Assign each new document to the class that best fits it (e.g. closest or most-probable class.)

**Types of class assignment:**

*Hard:* Each document belongs to exactly one class

*Soft:* Each document is assigned a “degree of membership” in several classes
Information Retrieval-Details (cont.)

Machine Learning for Text Classification (supervised)

It's all to do with the training: you can do a lot if you're properly trained.

Elizabeth II. TV documentary, BBC1, 6 Feb. 1992

* Training set: pre-classified documents
  \[ T = \langle \text{document}, C(\text{class}) \rangle; \langle \text{document}, C(\text{class}) \rangle; \ldots \langle \text{document}, C(\text{class}) \rangle; \]

* Learning task: Infer \( C \) from the experience data
Examples of Supervised Learning methods

**Nearest Neighbor**
- Memorize the classes of all documents
- Choose a distance/similarity function between documents
- Assign a new document to the class of the closest document

⚠️ **Main drawback:** Does not summarize the data; classification of a new document can be excruciatingly slow, comparing it against ALL documents in the dataset.
Information Retrieval-Details (cont.)

Machine Learning Methods for Text Classification

Rocchio (Based on [Rocchio,71])

Builds a class term-weight vector, \( \vec{C} \), summarizing the document vectors in the class (and those not in it).

Assigns new document vector \( \vec{d} \), to class \( C = \arg\max_{\vec{c}} [\cos(\vec{C}, \vec{d})] \)

[Rocchio71, Buckley et al. 94, Lewis et al. 96, Joachims97, Schapire et al. 98]
Naïve Bayes Classifier

• Assumes conditional independence among features, given the class

• Builds for each class, $C$, a term distribution model, $C^M$

  $C^M : <Pr(t_1 \in d | d \in C), Pr(t_2 \in d | d \in C), \ldots, Pr(t_n \in d | d \in C)>

  (where: $d$ – document, $t_i$ – term)

• Assigns new document $d'$ to class $C'$ s.t. $C' = \text{argmax}_{C^M} [Pr(d' | C^M)]$

  Using Bayes’ rule and the “Naïve” assumption:

  \[
  Pr(C^M | d') = \frac{Pr(d' | C^M) Pr(C^M)}{Pr(d')} = kPr(d' | C^M) \prod_{t_i \in d'} Pr(t_i \in d | d \in C^M)
  \]

  [McCallum & Nigam 99]

© Hagit Shatkay
Information Retrieval-Details (cont.)

Machine Learning Methods for Text Classification (cont.)

Support Vector Machine
Partitions the training documents by a maximum-margin hyperplane. Classifies new document-vector, $\tilde{d}$, based on its position WRT the separating hyperplane.

Boosting/Mixtures
Combines multiple weak classifiers.

[Joachims 98, Yang and Liu 99, Yang 2001, Schapire et al. 98]
Evaluating Extraction and Retrieval

To say how good a system is we need:

1. Performance metrics (numerical measures)
2. Benchmarks, on which performance is measured (the gold-standard).
Evaluating Extraction and Retrieval (cont.)

Performance Metrics

_N items_ (e.g. documents, terms or sentences) in the collection

**REL**: Relevant items (documents, terms or sentences) in the collection. These SHOULD be extracted or retrieved.

**RETR**: Retrieved items (e.g. documents, terms or sentences) are actually extracted/retrieved

Some _correctly_ \((A = |\text{REL} \land \text{RETR}|)\),

Some _incorrectly_ \((B = |\text{RETR} - \text{REL}|)\)

\(|\text{RETR}| = A + B\)
Evaluating Extraction and Retrieval \(^{(\text{cont.})}\)

**Performance Metrics** \(^{(\text{cont.})}\)

\[
|\text{Collection}| = N
\]

\[
|\text{REL} \cap \text{RETR}| = A
\]

\[
|\text{REL} - \text{RETR}| = D
\]

\[
|\text{NotREL} - \text{RETR}| = C
\]

\[
|\text{RETR} - \text{REL}| = B
\]
Performance Metrics (cont.)

**Precision:** \( P = \frac{A}{A+B} \)
How many of the retrieved/extracted items are correct

**Recall:** \( R = \frac{A}{A+D} \)
How many of the items that should be retrieved are recovered

**Accuracy:** \( \frac{A+C}{N} \) (Ratio of Correctly classified items)

**Combination Scores:**

**F-score:** \( \frac{2PR}{P+R} \)
Harmonic mean, in the range [0,1]

**F_β-score:** \( \frac{(1+\beta^2)PR}{\beta^2 \cdot P + R} \)
\( \beta > 1 \) Prefer recall, \( \beta < 1 \) Prefer precision

**E-measure:** \( 1 – F_\beta \)-score
Inversely proportional to performance (Error measure).
Performance Metrics (cont.)

Precision-Recall Curves

4 relevant documents in the collection.
7 retrieved and ranked.

Recall

Precision

25% Recall
50%
75%
100%

100 90 80 70 60 50 40 30 20 10 0
0 25 50 75 100

Recall
Performance Metrics (cont.)

Accounting for Ranks
For a given rank \( n \), \( P_n \): Precision at rank \( n \) (\( P@n \))

\textbf{R-Precision}: \( P_R \) where \( R \) is the number of relevant documents

Average Scores

\textbf{Average Precision}: Average the precision over all the ranks in which a relevant document is retrieved.

\textbf{Mean Average Precision}: Mean of the Average \textit{Precision} over all the queries.

\textbf{Micro-Average}: Average over individual items across queries

\textbf{Macro-Average}: Average over queries

[Lewis 95, Yang 99, Witten \textit{et. al.} 99, web sites e.g.  
Evaluating Extraction and Retrieval (cont.)

Benchmarks and Training/Test Sets

- **MUC**: Message Understanding Conference (ARPA, past)
- **LDC** (Linguistic Data Consortium) initiatives:
  - **ACE**: Automatic Content Extraction. Detecting Entities, Relationships and Events in newswires and scanned text in Arabic, Chinese and English.
  - **TIDES**: Translingual Information Detection, Extraction & Summarization. TDT – Topic Detection and Tracking (NIST & DARPA)
- **TREC**: Text Retrieval Conferences (NIST, past & present)
- **Reuters-21578 collection** (Document classification)
- **OHSUMED collection** (Document classification/retrieval)
- **Corpora of tagged text**: Brown corpus, Penn Treebank
Overview

✓ Text Processing and Mining
► Applications in Bio-Medical Literature
  – Information Extraction
  – Information Retrieval
  – Hybrid Methods

• Detailed Examples
• Evaluation Issues
• Conclusion
Text Sources

- MEDLINE abstracts, via PubMed

- Full text documents (e.g. Elsevier, Nature, Biomed Central)

- Text annotations (e.g. Swiss-prot, GeneCards)

- Internal documents, patent information

- Ontologies (e.g. GO, UMLS, MeSH)
Biomedical Documents

Unlike newswires or web sites:

• Not written for laypeople or for general consumption
• Not written by professional writers
• Not written by fluent/native English writers/speakers…
Information Extraction

T. Leek, MSc thesis, 1997 [Leek97]

• **Gene localization on chromosomes.**
  

• **HMMs** characterize sentences describing localization relationships.

• Gene names and Chromosomes are identified through heuristics.

• Words denoting location (e.g. *mapped, located, discovered*) and methods (e.g. *southern, blot*) are pre-defined.

• Trained and tested on sentences from OMIM abstracts.

• Scored based on *correct population* of the relation slots.

> “Southern analysis shows that HKE4 is located on human chromosome 6.”
Information Extraction (cont.)

M. Craven et al [Craven&Kumlien99, Ray&Craven01, Skounakis, Craven&Ray03]

- **Protein sub-cellular localization** and **gene-disorder associations**.
  
  **Examples**: Protein: Enzyme UBC6 *localized to* Endoplasmic Reticulum. 
  Gene: PSEN1 *associated with* Alzheimer Disease

- **HMMs** (and their extensions) characterize sentences describing 
  sub-cellular localization, and disease association. 
  (Other models for sub-cellular localization in [Craven&Kumlien99])

- HMMs’ *states* represent **structural segments** (e.g. *NP_segment*)

- Training: Sentences whose gene/protein/location/disease words are 
  tagged, based on information from YPD and OMIM. 
  (Protein and localization lexicon was provided in [Craven&Kumlien99])

- Scored on correct classification of relevant/irrelevant sentences.
Information Extraction (cont.)

A. Blaschke, A. Valencia et al, 1999 [Blaschke et al 99]

• Protein-protein interactions.

  Example: Protein: Spatzle Activates Protein: Toll

• Based on co-occurrence of the form “... p1...I1...p2...” within a sentence, where $p_1, p_2$ are proteins and $I1$ is an interaction term.

• Protein names and interaction terms (e.g. activate, bind, inhibit) are provided as a “dictionary”.

• Does not use formal modeling or machine-learning.

• Applied to two systems in Drosophila:
  - The Pelle system (6 proteins, 9 interactions) and
  - The cell-cycle control system (91 proteins), without quantitative analysis of the results.
Information Extraction (cont.)

T. Jenssen, E. Hovig et al, 2001 (PubGene) [Jenssen et al 01]

• Gene-gene relations.

   Example: \[ \text{NR4A2 -- NR4A3} \]

• Based on co-occurrence of the form “... g1...g2...” within a Pubmed abstract, where \( g1, g2 \) are gene names.

• Gene names are provided as a “dictionary”, harvested from HUGO, LocusLink, and other sources.

• Does not use formal modeling or machine-learning.

• Applied to 13,712 named human genes and millions of PubMed abstracts (Most extensive!)

• No extensive quantitative results analysis.

[Pearson01] Discussion of this approach.
Information Extraction (cont.)

Extracting Gene and Protein Names [Fukuda et al 98, Krauthammer et al 00, Tanabe & Wilbur 02, Hanisch et al 03]

• **Fukuda et al**: PROPER. Rule-based (hand-coded);
  Identifies protein and domain names, *without a dictionary*.
  Tested on two sets of 30 and 50 *domain-specific abstracts*;
  93-98% recall, 91-94% precision.

• **Krauthammer et al**: Uses gene/protein name *dictionary*,
  extracted from GenBank;
  Encodes text and dictionary as *nucleotide sequences*, and
  matches text to the dictionary using *BLAST*.
  Tested on 1, *full-text, review article* (1162 gene/protein occurrences);
  78.8% recall, 71.7% precision.
Extracting Gene and Protein Names (Cont.)

- **Tanabe&Wilbur**: ABGene. Retrain Brill POS tagger, (on 7000 hand-tagged sentences from Medline abstracts), to identify a new tag Gene. Augment with rules to eliminate false positives and recover false-negatives. Use a Naïve Bayes classifier (IR), to rank documents by likelihood to contain genes/proteins. Tested on 56,469 Medline abstracts; Results for low-scoring documents, Precision <66% Recall <60%; for high-scoring gene-documents: Precision >85%, Recall >70%.

- **Hanisch et al**: Built a dictionary of ~38,000 proteins, and ~152,000 synonyms, combining multiple resources (OMIM, Swissprot, HUGO) and pruning/cleaning process. Extraction uses the dictionary and a scoring+optimization method to do better than naïve pattern matching. Tested over 470 abstracts with 141 proteins occurring in them; Best case recall 90%, precision: 95%.

[Schwartz&Hearst PSB03, BioCreative04, Mika & Rost ISMB04]
Information Extraction (cont.)

Early Link Analysis Approach: ARROWSMITH (and follow-ups)

- Fish Oil reduces and co-occurs with Blood Viscosity.
- Blood Viscosity increases and co-occurs with Platelet aggregability.
- Platelet aggregability increases and co-occurs with Vascular Reactivity.
- Vascular Reactivity increases and co-occurs with Raynaud’s Syndrome.
- Fish Oil co-occurs with Raynaud’s Syndrome.
- Fish Oil can reduce Raynaud’s Syndrome.

- Based on transitivity of relationships in co-occurrence graph.
- This idea can be used to discover new facts by co-occurrence.

Information Extraction (cont.)

Summary

• Extracts specific kinds of stated entity-entity relations.

• Identifies relations by either:
  – Syntactic/semantic model for “relational” sentence (NLP), or
  – Co-occurrence of entity/relation terms

• Requires a dictionary of entities and relations terms, or well-defined rules for identifying them

[Fukuda et al 98, Rindflesch et al 00, Friedman et al 01]

PSB proceedings (2000 and on).
The latter is hard to satisfy, as nomenclature for genes/proteins suffers from:

1. Non-uniformity, Incompleteness
2. Synonymy/Aliases (AGP1, aka, Amino Acid Permease1)
3. Polysemy: What might be a gene name in one context is a storage device in another…

Genes A and B may share a similar function but not explicitly related to each other through a publication.
Information Retrieval

Medline (via PubMed)

- ~15,000,000 biomed-related abstracts
- Boolean queries supported
- Similarity queries supported per-abstract (neighboring)
- Fast, accurate, and handles heavy traffic well
- Not meant to offer topical, task-specific, text-mining
  
  *(This is traditionally an information extraction task)*

Challenge

Create biomedical, task-specific, text retrieval systems.

useful
Information Retrieval

Finding functional relations among genes [Shatkay et al., 00,02]

Three main ideas:

1. Analysis of large-scale gene expression data can benefit from the co-analysis of the text related to the genes.

2. Terms occurring in texts which discuss the gene can be used as an alternative representation for the gene.

3. When genes are represented by their text, this representation can be clustered to find sets of related genes. This clustering is similar to, and complements, the clustering of gene sequences or gene expression data.

More – later.
Information Retrieval (cont.)

Finding Protein-Protein-Interaction Abstracts [Marcotte et al., 01]

Yeast Medline (YM)
665,807 abstracts

260 Protein-Protein Interaction (PPI) abstracts

Find Discriminating Words (DW)

\[ Pr(w \in A | A \in YM) \neq Pr(w \in A | A \in PPI) \]

Train Classifier

Abstract

\[Pr(A \in PPI | \text{the DW's in } A) > Pr(A \notin PPI | \text{the DW's in } A)\]

Naïve Bayes

Reaches about 65% precision at 65% recall on a set of 325 abstracts.
Information Retrieval (cont.)

Predicting Protein Subcellular Location

[Stapley et al. 02]

Hypothesis: Medline abstracts can characterize each organelle.

Genes/proteins of various organelles

Abstracts Mentioning Genes.

Term-vector Representation

Train Classifier

SVM

Organelle for

\(<w_{\text{New}1}, \ldots, w_{\text{New}M}>\)

Cell Membrane genes/proteins

Mitochondria genes/proteins

Nucleus genes/proteins

Information Retrieval (cont.)

**Textpresso – Domain Specific Search Engine** [Müller et al., 04]

- A retrieval engine for the C-Elegans literature
- Database: 3,800 file papers, 16,000 curated abstracts
- Extraction methods mark up entities of interest (e.g. *alleles, processes, functions*)
- “Interesting” entities are indexed and can be queried on
- Supports domain-specific queries, not generally supported by PubMed.
Hybrid Methods

Predicting Protein Subcellular Location [Stapley et al. 02]

Hypothesis: Medline abstracts + Amino Acid Composition can characterize each organelle.

Method (extension from previous slide):

• Extend the term-vectors: include for each amino acid its relative contribution to the protein(s) combination. (20 additional dimensions)

• Train and use SVM as before.

• Works much better than amino-acid combination alone, and slightly better than text alone.
Hybrid Methods (cont.)

Protein Homology by Sequence and Text [Chang et al. 01]

Find sequence homology using Psi-Blast, multiple alignment.

Create "documents" from all text associated with each protein (annotations and abstracts)

Discard alignments with low document similarities. Create profile.
Hybrid Methods (cont.)

Gene Clustering by Expression and Text [Glenisson et al. 03,04]

Find pair-wise dissimilarity between expression profiles

Create text-vectors based on all the abstracts curated for the genes (in SGD)

Find pair-wise dissimilarity in text

Combine into a single gene-dissimilarity measure

Hierarchically cluster the genes
Hybrid Methods (cont.)

Protein-Protein Interaction, PreBind/Textomy [Donaldson et al. 03]
Combining information retrieval with information extraction

400 No-Interaction abstracts (NI)

700 Protein-Protein Interaction (PPI) abstracts

Find PPI abstracts:

Find PPI Sentences:
Sentence $S \in A$

Train Classifier

SVM

$A \in NI$

$A \in PPI$

$S \in NI$ Sentences

$S \in PPI$ Sentences

Extract Protein Names

Manual Curation!
Overview

- Text Processing and Mining
- Applications in Bio-Medical Literature
- Detailed Examples
  - Functional relations among genes through IR
  - The KDD 2002 experience, IE
- Evaluation Issues
- Conclusion
Sub-Overview

- Functional relations among genes through IR
  - The Information Retrieval Model
  - Themes and How to find them
  - From Documents to Genes and Back
  - Experiments and Results
Finding functional relations among genes [Shatkay et al., 00,02]

Functionally Related Genes

Similar bodies of relevant literature

The original motivation: [ISMB 2000 “Genes, Themes and Microarrays”]

Finding functional relationships among genes, to explain and predict the results of large-scale expression experiments
The IR Model Used [Shatkay&Wilbur00]

- **Collection** → **Set** of documents from one **broad domain** (e.g. yeast genes, AIDS, food…)
- **Information need** → **Documents with a unifying theme**
- **Query** → **Example document**
- **Means** → **Probabilistic Similarity Search**

**Produce**

- Theme **documents**
- Summarizing **terms**

Documents are generated by a **hidden, stochastic process**
Modeling Themes

**Theme T:**
A collection of documents discussing a common (specific) subject.
Characterized by a family of $M$ Bernoulli Distributions, $Pr(t_i \in d|d \in T)$. 

**Term Distribution for the theme Coffee**

<table>
<thead>
<tr>
<th>Term</th>
<th>Pr(Term)</th>
</tr>
</thead>
<tbody>
<tr>
<td>acidity</td>
<td>0.5</td>
</tr>
<tr>
<td>apple</td>
<td>0.1</td>
</tr>
<tr>
<td>cake</td>
<td>0.3</td>
</tr>
<tr>
<td>coffee</td>
<td>0.9</td>
</tr>
<tr>
<td>ginger</td>
<td>0.1</td>
</tr>
<tr>
<td>grind</td>
<td>0.6</td>
</tr>
<tr>
<td>roast</td>
<td>0.8</td>
</tr>
<tr>
<td>squash</td>
<td>0.1</td>
</tr>
</tbody>
</table>
Modeling Themes (cont.)

Model Components for theme $T$:

• $\Pr(\text{doc. } d \text{ is in theme } T) : P_d \overset{\text{def}}{=} \Pr(d \in T)$.

• $\Pr(\text{Term } t_i \text{ to occur in a theme doc. } d) : P_i \overset{\text{def}}{=} \Pr(t_i \in d | d \in T)$.

• $\Pr(\text{Term } t_i \text{ to occur in an off-theme doc. } d) : Q_i \overset{\text{def}}{=} \Pr(t_i \in d | d \notin T)$.

• $\Pr(\text{Term } t_i \text{ to occur in any doc. } d) : DB_i \overset{\text{def}}{=} \Pr(t_i \in d)$.

  $DB_i \approx \frac{\# \text{ of docs containing } t_i}{|DB|}$

• $\Pr(\text{Term } t_i \text{ to be generated according to } DB_i) : \lambda_i$

Model $R = \{ P_d, \{ P_i \}, \{ Q_i \}, \{ DB_i \}, \{ \lambda_i \} \}$
Stochastic Generation of document $d$: 

For each term, $t_i$ 

$P_i$ 

$DB_i$ 

$Q_i$ 

Include $t_i$ in $d$? 

Theme / Off-Theme? 

From DB or P/Q?
GenTheme: Finding Themes

Task: Starting from a single document $d$, find a theme model $R$, that maximizes the likelihood, $Pr(DB|R)$.

Input: Kernel $d$ (PubMed ID), and document collection $DB$

Output: 
- Top (10) documents, with highest theme probability, $Pr(d \in T| d, DB, R)$
- Top (10) key terms, with highest ratio $Pr(t_i \in d|d \in T)/Pr(t_i \in d|d \notin T)$.

Method: Expectation Maximization (EM)
Finding Themes

**GOAL:** \( R_{\text{best}} = (P_i, Q_i, \lambda_i) = \underset{R}{\arg\max} [Pr(DB|R)] \)

If we knew which are the **theme docs**, and which terms are generated by **theme-specific** distributions…

Strong evidence links consumption of **French roast coffee** with Hepatocellular Carcinoma

Most roasters would use a Costa Rican coffee bean and **roast** it to the French darker degree

Medium body beans have made Colombia famous for its flavorful coffee with a slightly dry **acidity**.

\[
P_{\text{bean}} \leftrightarrow \frac{2}{3} \quad P_{\text{coffee}} \leftrightarrow \frac{3}{3} \quad P_{\text{roast}} \leftrightarrow \frac{2}{3}
\]

But we do not know… Use **Expected** values instead!!!
GenTheme in context: (From Genes to Documents)

For each $g \in G$

Documents discussing $g$

Terms summarizing $g$'s function

PubMed queries

Mapping

Kernels

G

DB

Broad Domain

© Hagit Shatkay
Finding Related Genes
(From Documents back to Genes)

*Similar* document sets represent *related* genes.

Thus, Find *similar* sets of documents
Similarity between *sets* of documents

• Space $S$ of *relevant* documents:

$$S \approx \bigcup_i S_i = \{ID_1, \ldots, ID_{Mr}\} \quad |S| = Mr$$

• Represent each set $S_i$, (kernel $k_i$) as $Mr$-dimensional vector:

$$<v_i^1, \ldots, v_i^{Mr}>$$

$$v_i^j = \begin{cases} 
1 & \text{if } ID_j \in S_i \\
0 & \text{otherwise}
\end{cases}$$

• *Similarity* between two vectors: The cosine of the angle between them.

---

Set, $S_i \rightarrow$ similar sets $\{S_i^1, \ldots, S_i^p\}$

Kernel, $k_i \rightarrow$ similar kernels $\{k_i^1, \ldots, k_i^p\}$

Gene, $g_i \rightarrow$ related genes $\{g_i^1, \ldots, g_i^p\}$

---

© Hagit Shatkay

92
From Genes to Documents and Back
Summary

Per Kernel:
Find related docs. + descriptive terms

Information Retrieval Engine

Find Similarity among Bodies of Literature

Bodies of Literature (+ Term Sets.)

Sets of Inter-related Genes
Experiments and Results

**Domain:** Yeast genes

★ Accessible information sources (*SGD*, *YPD*)

★ Grouping and functional analysis presented by *Spellman et al* [1998]

★ Hence, the quality of our results is easily verifiable
Experimental Setting

- **Kernel documents:** 344 PubMed abstracts. (Curated references from SGD as representatives for 408 cell-cycle regulated genes).

- **Database:** 33,700 Pubmed abstracts, generated through iteratively neighboring the 344 abstracts.

- **GenTheme algorithm:** Produce a theme for each kernel

- **Cosine-based method:** Find groups of related genes based on related kernels

- Compare the results with the functionality of genes according to Spellman et al. [Spellman et al 98].

<table>
<thead>
<tr>
<th>Biological Function</th>
<th>G1</th>
<th>S</th>
<th>G2</th>
<th>M</th>
<th>M/G1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Replication Initiation</td>
<td>CDC45</td>
<td>ORC1</td>
<td>CDC47 CDC54</td>
<td>MCM2 MCM6</td>
<td>CDC6 CDC46 MCM3</td>
</tr>
<tr>
<td>Fatty Acids/Lipids/Sterols/Membranes</td>
<td>EPT1 LPP1 PSD1 PSD1</td>
<td>AUR1 ERG3 LCB3</td>
<td>ERG2 ERG5 PMA1 PMA2 PMP1</td>
<td>ELO1 FAA1 FAA3 FAA4 FAS1</td>
<td></td>
</tr>
<tr>
<td>Nutrition</td>
<td>BAT2 PHO2</td>
<td>AGP1 BAT1 GAP1</td>
<td>DIP5 FET3 FTR1 MEP3 PFK1 PHO3 PHO5 PHO11 PHO12 PHO84 RGT2 SUC2 SUT1 VAO1 VCX1 ZRT1</td>
<td>AUA1 GLK1 HXT1 HXT2 HXT4 HXT7</td>
<td></td>
</tr>
</tbody>
</table>

©Hagit Shatkay
1. Qualitative Evaluation

Kernel PMID: 8702485, Gene: ELO1, Function: Fatty Acids/Lipids/Sterols/Membranes

<table>
<thead>
<tr>
<th>Keywords</th>
<th>Genes</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fatty acid,</td>
<td>OLE1</td>
<td><em>Fatty Acid, Sterol Metabolism</em></td>
</tr>
<tr>
<td>Fatty,</td>
<td>FAA4</td>
<td>FA/Lipids/Sterols/Membranes</td>
</tr>
<tr>
<td>Lipids,</td>
<td>FAA3</td>
<td>FA/Lipids/Sterols/Membranes</td>
</tr>
<tr>
<td>Acid,</td>
<td>SUR2</td>
<td>FA/Lipids/Sterols/Membranes</td>
</tr>
<tr>
<td>Carbon,</td>
<td>ERG2</td>
<td>FA/Lipids/Sterols/Membranes</td>
</tr>
<tr>
<td>Grown,</td>
<td>FAA1</td>
<td>FA/Lipids/Sterols/Membranes</td>
</tr>
<tr>
<td>Medium,</td>
<td>PSD1</td>
<td>FA/Lipids/Sterols/Membranes</td>
</tr>
<tr>
<td>Synthase,</td>
<td>CYB5</td>
<td><em>Fatty Acid, Sterol Metabolism</em></td>
</tr>
<tr>
<td>Strains,</td>
<td>PGM1</td>
<td>Carbohydrate Metabolism*</td>
</tr>
<tr>
<td>Deficient</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Another Example:

Starting point: **2233722** (Curated in SGD for HXT2):

Top ranking docs: **10336421, 7862149, 8594329, 2660462**, …

Top ranking terms: *glucose transport, glucose, transporter*, etc.

Related Genes:

- Spellman Verified
- YPD Verified
- Spurious Relation
### K. PMID: 6323245, Gene MCM2, Function: Replication Initiation

<table>
<thead>
<tr>
<th>Keywords</th>
<th>Genes</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARS, Autonom. replicating, Replicating sequence, Autonomously, Minichromosomes, Replicating, Centrometric, leu2, Plasmids, ura3</td>
<td><strong>CDC10</strong></td>
<td>Site selection, Morphogenesis</td>
</tr>
<tr>
<td></td>
<td><strong>PHO3</strong></td>
<td>Nutrition</td>
</tr>
<tr>
<td></td>
<td><strong>EST1</strong></td>
<td>DNA Synthesis</td>
</tr>
<tr>
<td></td>
<td><strong>MIF2</strong></td>
<td>Chromatin</td>
</tr>
<tr>
<td></td>
<td><strong>PHO12</strong></td>
<td>Nutrition</td>
</tr>
<tr>
<td></td>
<td><strong>POL3</strong></td>
<td>DNA Synthesis</td>
</tr>
<tr>
<td></td>
<td><strong>DHS1</strong></td>
<td>DNA Repair</td>
</tr>
<tr>
<td></td>
<td><strong>SNQ2</strong></td>
<td>DNA Repair</td>
</tr>
<tr>
<td></td>
<td><strong>SMC3</strong></td>
<td>Chromatin</td>
</tr>
<tr>
<td></td>
<td><strong>EXG2</strong></td>
<td>Cell Wall Synthesis</td>
</tr>
</tbody>
</table>
2. Quantitative Evaluation of Summaries

- **Kernels:** 105 abstracts discussing the biological function of distinct yeast genes.

- **Expert Thesaurus Construction:**
  - List 5 top ranking terms for each of the 105 kernels. 330 Terms, alphabetically sorted.
  - 4 Independent judges assign terms to one of 22 functional categories (+ “uninformative”).

![Diagram with terms: chromatin, fatty, telomere, acid]
2. Quantitative Evaluation (cont.)

Expert Thesaurus - Example:

<table>
<thead>
<tr>
<th>Function</th>
<th>Associated Terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chromatin</td>
<td>Chromatids, Chromatin, Chromosome, sister Chromatids, telomere, telomeric</td>
</tr>
<tr>
<td>Secretion</td>
<td>Acid phosphatase, coatomer, endoplasmic, endoplasmic reticulum, er, golgi apparatus, golgi complex, golgi transport, golgi</td>
</tr>
</tbody>
</table>

Count how many of the top 5 summary terms are assigned to a thesaurus entry matching the function discussed in the query document; 
**average over 105 queries.**

<table>
<thead>
<tr>
<th>Correct</th>
<th>Wrong</th>
<th>Don’t Care</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.27 terms</td>
<td>1.12 terms</td>
<td>0.61 terms</td>
</tr>
</tbody>
</table>

\[ P << 0.005 \]
General

• Similarity Search + EM: Independent of explicit query terms (synonymy, polysemy).
• Provides a key-term list, justifying the relevance of retrieved documents.

Application Specific

• Independent of explicit gene/protein names, makes no assumption about standard nomenclature.
• Independent of sentence structures (e.g. “A interacts with B”).
• Can foreshadow putative, undiscovered relationships
• Results depend on the initial kernel document
Acknowledgements

Collaborators
Stephen Edwards (Rosetta)
Mark Boguski (Allen Institute of Brain Science)
John Wilbur (NCBI)

Yeast Experts (NCBI):
Jan Fassler
Ken Katz
Steven Sullivan
Tyra Wolfsberg

EM Discussions:
Luis Ortiz (Brown, UPenn)
Overview

✓ Text Processing and Mining
✓ Applications in Bio-Medical Literature
✓ Detailed Examples
  ✓ Functional relations among genes through IR
    ➔ The KDD 2002 experience, IE
• Evaluation Issues
• Conclusion
KDD Cup Task 1
Information Extraction from Biomedical Articles

System Description

Summer 2002

© Hagit Shatkay
The Task:
A system for automatic analysis of scientific papers regarding the Drosophila fruit fly.

**Input:**
- A set of full-text papers
- A list of gene names and synonyms (per paper)

**Output:**
- Papers containing experimental evidence for *wild-type gene expression*. (Order documents by relevance to gene expression; *Curate vs. Don’t Curate* decision)
- Genes mentioned in the paper that had their expression *actually* discussed in it.
- Classification of the genes based on expression product: *Protein or Transcript*. 
Why is it hard?

- Often papers discuss expression of mutated genes during a functional study, rather than wild-type genes.

- Many of the genes within an experimental paper are auxiliary or “just mentioned” - with no reported expression results.

- The Transcript vs. Protein distinction is tricky (they usually have same/similar name … )

- Human curators rely on images. The system is only given the text version without images.
The problem as an Information Extraction Task

The scientific papers given are lengthy and complex … But

• They have a very regular structure
• Indications for the results of interest can be found in: 
  *Abstract, Figure Captions, Titles, Footnotes*
• The results are obtained through well-known techniques with relatively limited characteristic terminology (e.g. *northern* and *western* blot.)
• Knowledge-Based Information Extraction - finding frequent patterns relevant to the domain - can be applied.
Evidence in Figure Captions

Positive Example:
"Fig. 2. Expression of *damm* mRNA during Drosophila development."

*Evidence for wild type gene expression.*

Negative Example:
"Fig. 2. Ectopic expression of *dNSF1* in the nervous system rescues the phenotypes of *dNSF1* mutations."

*Evidence for mutated, unnatural gene expression – ignore!*
Evidence in Footnotes

Footnotes indicate deposit of a transcript sequence GeneBank database:

“Data deposition: The sequences reported in this paper have been deposited in the GeneBank database (accession no. AF348329).”

Evidence of a newly derived gene transcript.
Positive and Negative Evidence Examples

Positive Evidence

“Northern blot analysis of @norpA@ transcripts in adult @Drosophila@ tissues”

Negative Evidence

“Figure 2. Ectopic expression of @dNSF1@ in the nervous system rescues the phenotypes of @dNSF1@ mutations”
Curation Decision:

Based on total scores of extracted evidence.

• Extract evidence from Title, Abstract, Figure Caption and GenBank footnotes.

• Keep a Score entry for the whole document and for each product (transcript/protein) of a candidate gene.

• At the end of the document, use the scores to decide regarding the curation of the document and the products of the candidate genes.

• If a gene scores above a certain threshold, mark it as having an experimental result, and the whole document as curatable.
Implementation: DIAL Rulebook

• The System is implemented in **DIAL** (Declarative Information Analysis Language), a general IE language developed at ClearForest.

• DIAL is based on **matching patterns** within the text and **imposing constraints** on the patterns.

• Patterns combine **syntactic** and **semantic** elements.

• Relies on provided/built-in data:
  Gene names, symbols and synonyms, analysis techniques, positive keywords, negative keywords
  **Infrastructure libraries** of simple tokens/phrases (numbers, capital sequences) and NLP patterns.
Example of a DIAL Rule

Rule for detecting “induced expression”. (Expression is discussed but not as an indication for a transcript/protein of a gene of interest) “DAC does not antagonize hth expression”

//Lexicon for nouns indicating “expression”
wordclass wcExpressionNoun =
  expression transcription localization detection;

//Lexicon for verbs indicating induction/interaction between genes
wordclass wcInducedVerbs =
  reduce inhibit activate induce repress antagonize

//Extract Noun Phrase (NG-NounGroup) incorporating a gene
GeneExpressionNG() :-
  ExtractedGene(Gene, Product) //”hth”
  NounGroup(Article,Head,Stem) //”hth expression”
  verify(InWC(Head,@wcExpressionNoun)); //verify that the Head is relevant

//Rule for the induced expression itself
Induced_ExpressionNG() :-
  ExtractedGene(Gene, Product,Mutant) //”DAC” (the gene)
  VerbGroup(Stem,Tense,Aspect,Voice,Polarity) //”does not antagonize”
  GeneExpressionNG //”hth expression”
  verify(InWC(Stem,@wcInducedVerbs)) //verify that the Stem is a relevant verb
Evaluation and Results (By the KDD Cup)

- Training Set: 862 Full text articles
- Test Set: 213 Full text articles
- Metric: The F-measure

Limitations:
- Small test set.

Success rate: (Best among 32 entries)

- Document Curation: 78% F-Measure
  Median was 58% for all systems submitted
- Gene Products: 67% F-Measure
  Median was 35% for all systems submitted.
Results Summary

Information Extraction is very effective for the task since:

• Most papers share the same narrow discourse domain and use the same vocabulary. (Text categorization / IR is difficult in this case. F-Measure: 62-64% in KDD.)

• Many curatable papers contain results that are indicative of wild-type expression only for some of the genes occurring in them. Phrases must be examined and extracted for a per-gene decision.

• Protein vs. Transcript distinction requires fine-grain pattern analysis at the phrase/word level.

• Bonus: Phrase extraction provides explicit evidence to human curators.
The ClearForest-Celera Team

**ClearForest**
Ronen Feldman
Yizhar Regev
Michal Finkelstein
Maya Gorodetsky

**Celera**
Hagit Shatkay
Xin Zheng
Samuel Levy
Rosanne Charlab
Charles Lawrence
Ross Lippert
Qing Zhang
Overview

✓ Text Processing and Mining
✓ Applications in Bio-Medical Literature
✓ Detailed Examples

➡ Evaluation Issues
   ✓ The KDD 2002 experience, IE (done)
     – TREC Genomics Track, IR
     – BioCreAtIvE, IE

• Conclusion
Evaluation

Question: **How Good is a Biomedical Text-Mining System?**

Answer: **Good for *What***?

**Evaluation Procedure:**

- Set task
- Create Gold Standard of *correct answers for the task*
- Decide on evaluation metric (precision/recall/combinations)
- Compare systems performance on *task* WRT *Gold Standard*
- Which system does best?
  How well is *the best* currently doing?
- How different (*statistical significance*) is the best from the others?

**Aftermath**

- This exercise is Per Task and Data Sets…
- How do the results generalize?
- How do we estimate the latter?
TREC Genomics

TREC: Text REtrieval Conference
Focus: Evaluation of document retrieval

• Tasks (2003):
  – Primary: Given a gene, find all PubMed abstracts discussing its function
  – Secondary: Given 139 full-text curated GeneRIF documents, recover
    GeneRIF text snippet summarizing the function. (GeneRIF: LocusLink’s Reference Into Function annotation).

• Evaluation metrics:
  Mean Average Precision for retrieval; Dice string-overlap for extraction

• Gold Standard: Current GeneRIFs (References & text-snippets)
  Not ideal, contains many false negatives, but easy to obtain…
TREC Genomics (cont.)

Primary task 2003, outcome

- 29 teams, submitted a total 49 official runs, of 50 queries each.
- Best results: MAP: 0.4165
  - Mean # of relevant documents @ 10: 3.16
  - Mean # of relevant documents @ 20: 4.84
- Differences between highest scoring teams not statistically significant.
- Top teams: NLM, NRC (Canada), UC Berkeley, U of Waterloo.
  - Used MeSH and rules to identify species
  - Used MeSH terms and substance fields – not just the text
  - Identified Gene synonyms
Secondary task 2003, outcome

• 14 teams, submitted 24 official runs each.

• Analysis by Aronson and Mork from NLM:
  95% of GeneRIF snippets contain title/abstract words.
  42% of those are direct cut-and-paste from title/abstract.
  25% contained significant word runs from the title/abstract.

• Top teams (Erasmus and Berkeley):
  – Used classifiers to identify likely GeneRIF sentences
  – Performed about the same as those teams who simply provided the title of
    the GeneRIF reference.
TREC Genomics 2004

• Tasks:
  – **Ad Hoc Retrieval**: Given 50 information-need statements collected from bio-medical professionals, find all documents satisfying that need.
  – **Categorization**:
    a. **Triage**: Given a gene, find all documents relevant to it that are experimental and contain evidence for a GO top category, function/process/locus.
    b. **Annotation**: Given a gene and a document discussing it, assign the GO category for the gene based on the paper. Secondary task – assign the GO evidence code as well.

• **Gold Standard**:
  – **Ad Hoc**: Relevance judgments performed by 2 judges on top ranks of all runs (MAP measure)
  – **Categorization**: MGI curated papers for the 3 main GO categories (Utility score)
TREC Genomics (cont.)

Ad-Hoc Task 2004, outcome

• 27 teams, submitted a total 47 official runs, of 50 queries each.
• Best results: MAP: 0.4075
  
  Mean # of relevant documents @ 10: 6.04
  Mean # of relevant documents @ 100: 41.96

• Top teams (PATOLIS, Japan; Waterloo, Canada):
  – Used MeSH terms and LocusLink alias symbols to expand queries
  – Used synonym expansion
  – Used pseudo relevance feedback (expands queries by terms from the top docs)
TREC Genomics (cont.)

Categorization 2004, outcome

• 20 teams, submitted 59 official Triage runs, 36 Annotation runs. (Only 3 runs were submitted for the secondary annotation task)

• Best results: Utility: 0.65  
  Precision: 0.16  
  Recall: 0.89  
  F-score: 0.27

• Top team (Rutgers/DIMACS):
  – First phase classification: Check for MeSH term “Mice”
  – Second phase classification: Bayesian logistic regression classifier
  – The first phase is almost all it takes. Checking for “Mice” in MeSH terms: Utility: 0.64, precision:0.15, recall:0.89, F-score:0.26
    Still better than all other systems!
TREC Genomics (cont.)

TREC Genomics 2005

Happening now.

• Tasks:
  – Ad Hoc Retrieval: Given 50 queries based on templates (e.g. “Find documents describing the role of Gene X in Disease Y”), instantiated by bio-medical professionals, find all documents satisfying each query.
  – Categorization: Given a gene, find all documents relevant to it that are experimental and contain evidence for a GO top category, function/process/locus, or to tumor biology, mutant-phenotype, embryologic gene expression.

• Gold Standard:
  – Ad Hoc: Relevance judgments performed on the submitted runs
  – Categorization: MGI curated papers for each category
BioCreAtlE

Critical Assessment of IE systems in Biology

Focus: Evaluation of Information Extraction

Task 1: Entity extraction

a. Identify all gene/protein names in given sentences
b. Given genes and an abstract, identify all references (synonyms) to each given gene in the text
   Gene references must match the species discussed in the paper

Text sources:
– Sentences mentioning genes from Abstracts (7500 training, 5000 test)
– Abstracts for Drosophila, Mouse and Human papers (5000 training, 250 test per organism)
– Lists of gene names

Gold standard: Text (sentences, abstracts) annotated with gene names
**Task 2: Functional Curation**

a. Given the GO code for proteins in a paper, find the evidence for the GO annotation and highlight it.

b. Given a paper and the proteins in it, deduce the GO codes for the proteins. 

Output tuples: <Protein, Paper, GO code, Evidence text>

**Text sources:**
- Full text of journal articles as used by SwissProt annotators (Training: 636 documents)
- GO, which is freely available, and any other public information

**Gold standard:** Relevance judgment by a team of SwissProt annotators
Outcome will be published.

Entity extraction task 1A: 12 participants with 40 official entries.

<table>
<thead>
<tr>
<th>Measure</th>
<th>Highest</th>
<th>Median</th>
<th>Lowest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>0.86</td>
<td>0.8</td>
<td>0.17</td>
</tr>
<tr>
<td>Recall</td>
<td>0.85</td>
<td>0.74</td>
<td>0.42</td>
</tr>
<tr>
<td>F-score</td>
<td>0.83</td>
<td>0.78</td>
<td>0.25</td>
</tr>
</tbody>
</table>

Entity extraction/normalization task 1B: 8 participants with 15 official entries.

<table>
<thead>
<tr>
<th>Measure</th>
<th>High(Y)</th>
<th>Median(Y)</th>
<th>High(F)</th>
<th>Median(F)</th>
<th>High(M)</th>
<th>Median(M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>0.969</td>
<td>0.94</td>
<td>0.831</td>
<td>0.659</td>
<td>0.828</td>
<td>0.765</td>
</tr>
<tr>
<td>Recall</td>
<td>0.962</td>
<td>0.848</td>
<td>0.841</td>
<td>0.732</td>
<td>0.898</td>
<td>0.730</td>
</tr>
<tr>
<td>F-score</td>
<td>0.921</td>
<td>0.858</td>
<td>0.815</td>
<td>0.661</td>
<td>0.791</td>
<td>0.738</td>
</tr>
</tbody>
</table>
Summary: BioMed Lit-Mining in a Nutshell

Define a Text Collection

Tokenize

Store in a Readily Accessible Format

Retrieval

Extraction

Classification

Feature Selection

Suggested tools on Alex Morgan’s page:
http://www.tufts.edu/~amorga02/bcresources.html
Conclusion

Information Extraction:
• Extracts *well-defined facts* from the literature
• Requires domain *vocabulary* or *rules* to identify these facts
• Finds *explicitly stated* facts
• Looks for facts stated within a *sentence, a paragraph or a single document* (Fine granularity)

Information Retrieval:
• Looks for relevant documents
• Does not give a tidy *fact statement* (Coarse granularity)
• Can find relations among *documents* or *document collections*
• Can create a *coherent context* for performing Information Extraction
• Can foreshadow putative, yet-undiscovered relationships
• Less sensitive to vocabulary and terminology
Conclusion (cont.)

Challenges:

• Reduce dependency on vocabularies and nomenclature
• Automate fact-finding about gene-disease interaction
• Reconstruct metabolic, signaling or regulatory pathways
• Augment analysis of large-scale experiments with data from the literature. (e.g. [Chang et al 01, Glenisson et al 03,04])

• Establish evaluation standards for testing the utility of literature mining tools.

No single method can address all the needs.

A combined approach is likely to get us closer to our goal.