

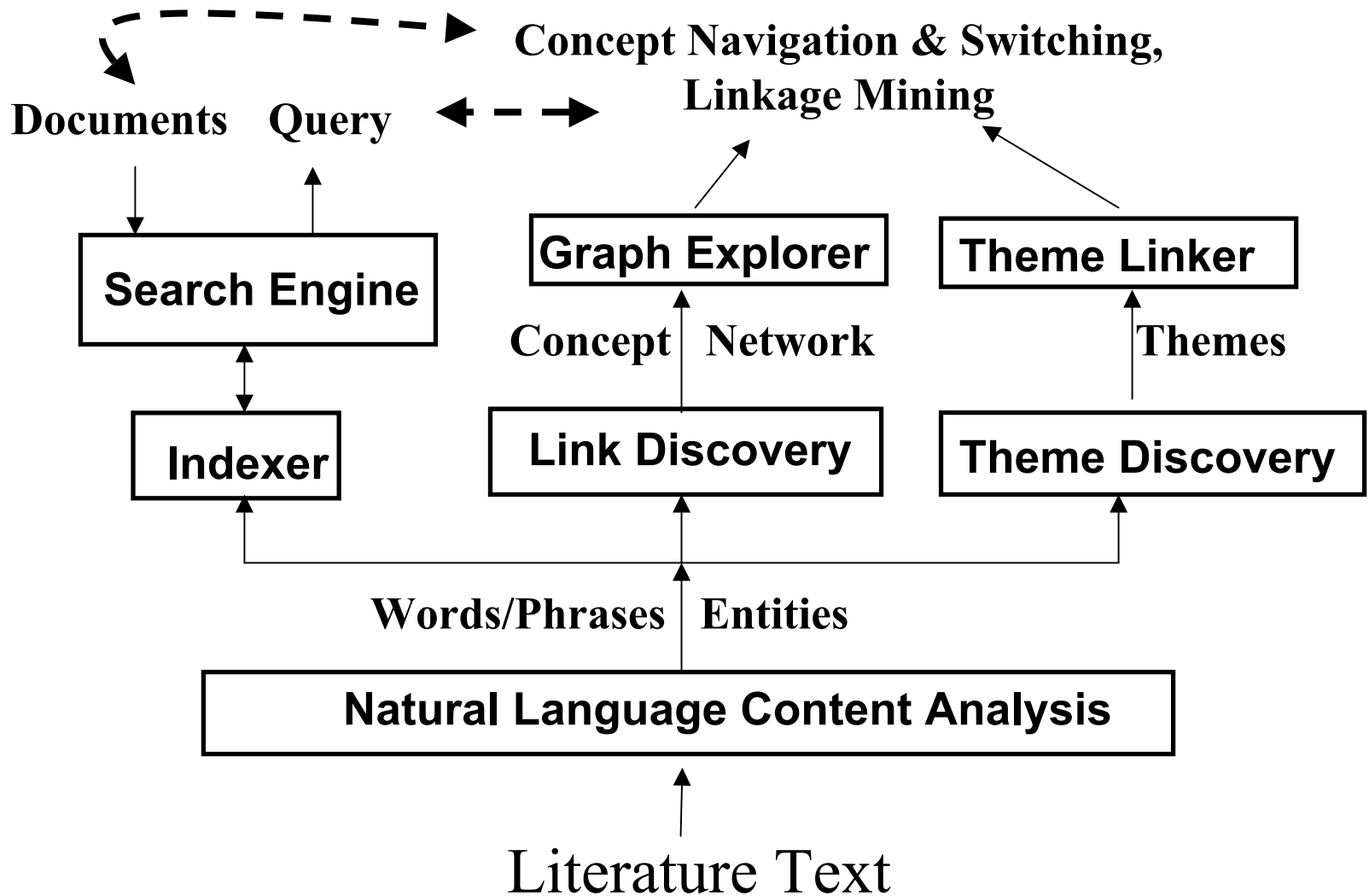
# **BeeSpace Analysis Environment**

**ChengXiang Zhai**

**BeeSpace Workshop, June 6, 2005**

**Department of Computer Science  
University of Illinois at Urbana-Champaign**

# Overview of BeeSpace Technology



# Natural Language Content Analysis

- Part of speech recognition
- Phrase analysis
- Entity recognition
- Mostly using/adapting existing tools

<Sent><NP>We</NP> have <VP>cloned</VP> and <VP>sequenced</VP>  
<NP>a cDNA encoding <Gene>Apis mellifera ultraspiracle</Gene><NP>  
(<Gene>AMUSP</Gene>) and <VP>examined</VP> <NP>its responses to  
JH</NP>.</Sent>...

# Search Engine

- **Given a text query and a collection of documents**
- **Find documents that are relevant to the query**
- **Standard methods are available**
  - **Estimate a query language model (i.e., word distr.)**
  - **Estimate a document language model**
  - **Compute the distance between two language models**
- **Use the Lemur toolkit**

# (Concept) Link Discovery

- Exploit co-occurrence information to discover strongly associated concept pairs
- Many techniques available
- We use the mutual information (MI) measure

**Random Var.**  $C_i = \begin{cases} 1 & \text{Concept } i \text{ observed} \\ 0 & \text{not observed} \end{cases}$        $C_j = \begin{cases} 1 & \text{Concept } j \text{ observed} \\ 0 & \text{not observed} \end{cases}$

**MI Measure:**  $I(C_i; C_j) = \sum_{x \in \{0,1\}, y \in \{0,1\}} p(C_i = x, C_j = y) \log \frac{p(C_i = x, C_j = y)}{p(C_i = x)p(C_j = y)}$

Chances of seeing them together

Chances of seeing each

# Theme Discovery

- Assume  $k$  themes, each being represented by a word distribution
- Use a  $k$ -component mixture model to fit the text data
- The estimated  $k$  component word distributions are taken as  $k$  themes

Likelihood: 
$$\log p(C | \Lambda) = \sum_{D \in C} \sum_{i=1}^{|D|} \log[\lambda p(D_i | \theta_B) + (1 - \lambda) \sum_{j=1}^k \pi_j p(D_i | \theta_j)]$$

Maximum likelihood estimator: 
$$\Lambda^* = \arg \max_{\Lambda} p(C | \Lambda)$$

Bayesian estimator: 
$$\Lambda^* = \arg \max_{\Lambda} p(\Lambda | C) = \arg \max_{\Lambda} p(C | \Lambda) p(\Lambda)$$

# Theme Linker/Retrieval

- **Theme link discovery**
  - Given two themes, measure their similarity
  - Add a link if the similarity is high enough
- **Theme retrieval**
  - Given any query, construct a theme
  - Compute the similarity of the query theme with other themes
  - Retrieve top-k most similar themes
- **Theme similarity: divergence-based measures**

# Weighted Entity-Relation Graph Explorer

- **Given a weighted entity-relation graph (e.g., a concept network with weights for association)**
- **Support interactive exploration of the graph**
  - Find best neighbors
  - Find best paths
  - Operators can be combined to perform complex exploration
- **Applications:**
  - Ad hoc exploration of concepts/themes (navigation)
  - Linkage discovery
  - Concept switching