



INDRI - Overview

Don Metzler

Center for Intelligent Information Retrieval
University of Massachusetts, Amherst

[Zoology 101



- *Lemurs* are primates found only in Madagascar
- 50 species (17 are endangered)
- Ring-tailed lemurs
 - *lemur catta*

[Zoology 101



- The *indri* is the largest type of lemur
- When first spotted the natives yelled "*Indri! Indri!*"
- Malagasy for "*Look! Over there!*"

[What is INDRI?]



- INDRI is a “larger” version of the Lemur Toolkit
- Influences
 - INQUERY [Callan, et. al. '92]
 - Inference network framework
 - Query language
 - Lemur [<http://www-2.cs.cmu.edu/~lemur/>]
 - Language modeling (LM) toolkit
 - Lucene [<http://jakarta.apache.org/lucene/docs/index.html>]
 - Popular off the shelf Java-based IR system
 - Based on heuristic retrieval models
- No IR system currently combines all of these features

[Design Goals



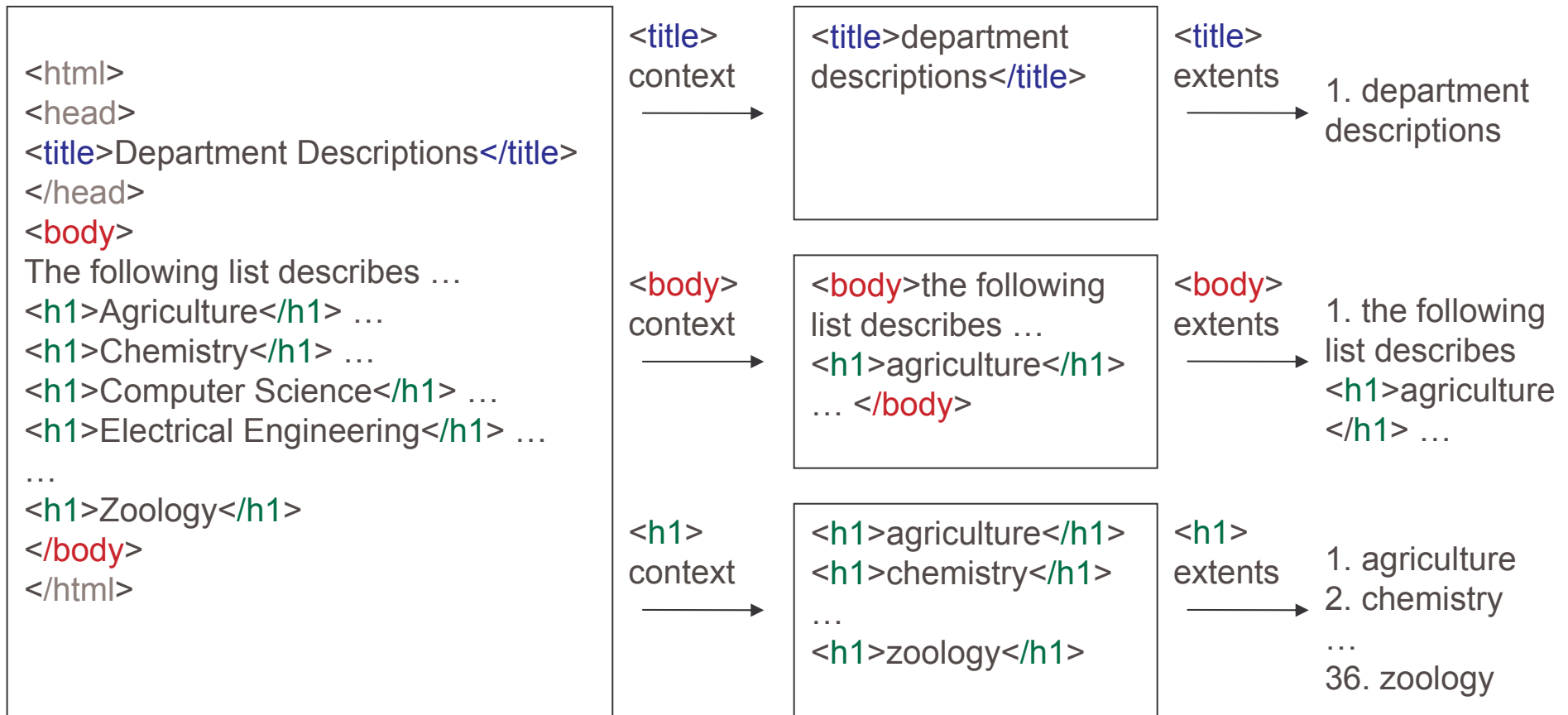
- Off the shelf (Windows, *NIX, Mac platforms)
 - Separate download, compatible with Lemur
 - Simple to set up and use
 - Fully functional API w/ language wrappers for Java, etc...
- Robust retrieval model
 - Inference net + language modeling [Metzler and Croft '04]
- Powerful query language
 - Extensions to INQUERY query language driven by requirements of QA, web search, and XML retrieval
 - Designed to be as simple to use as possible, yet robust
- Scalable
 - Highly efficient code
 - Distributed retrieval

[TREC Terabyte Track



- Initial evaluation metric for INDRI
- Task: *ad hoc* retrieval on a web corpus
- Goals:
 - Examine how a larger corpus impacts current retrieval models
 - Develop new evaluation methodologies to deal with hugely insufficient judgments
- Data:
 - .GOV2 collection (crawl of entire .gov domain)
 - 25 million documents (html, pdf, ps, Word, text)
 - Not even half of a TB of documents (426 GB)

Document Representation



⋮

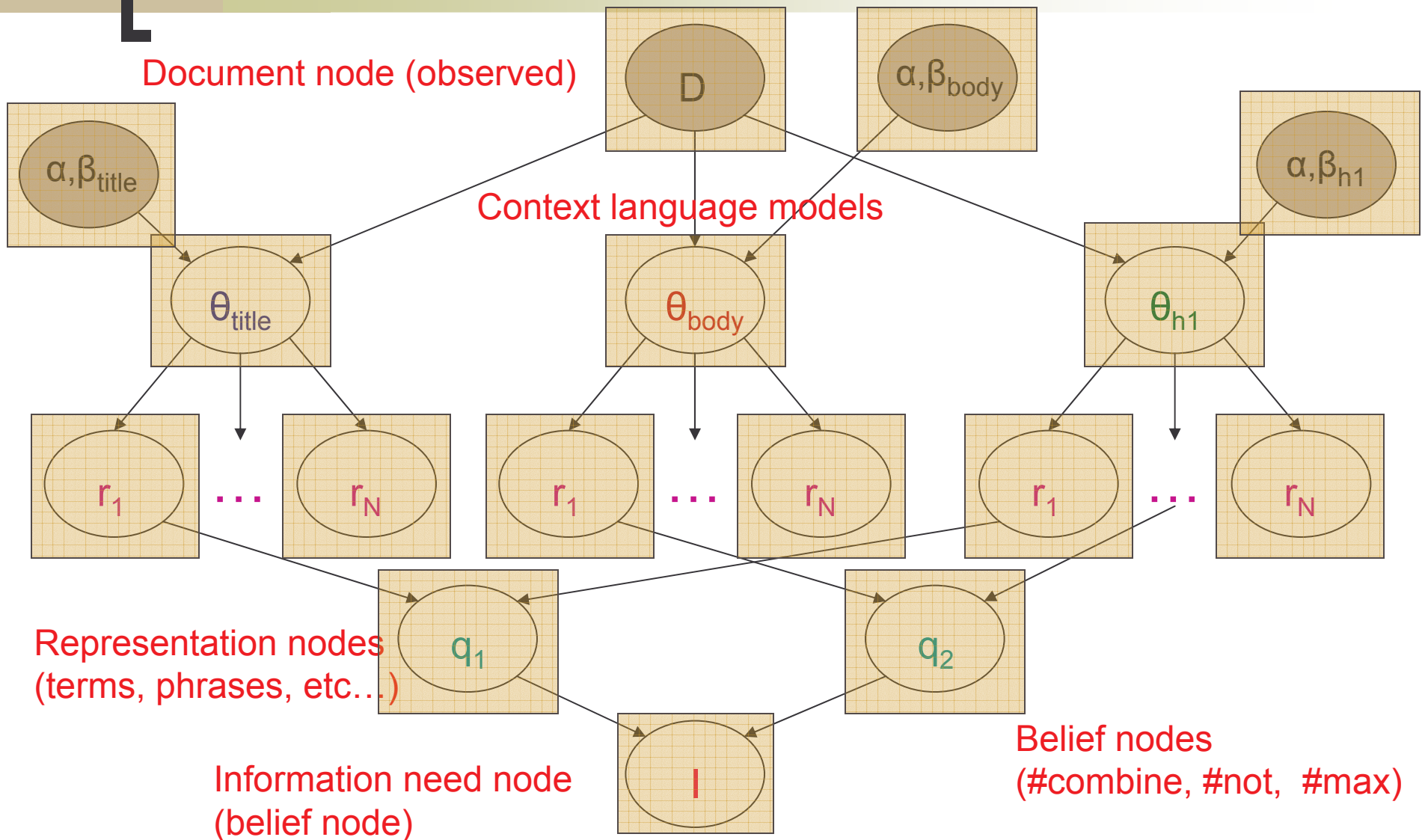
[Model



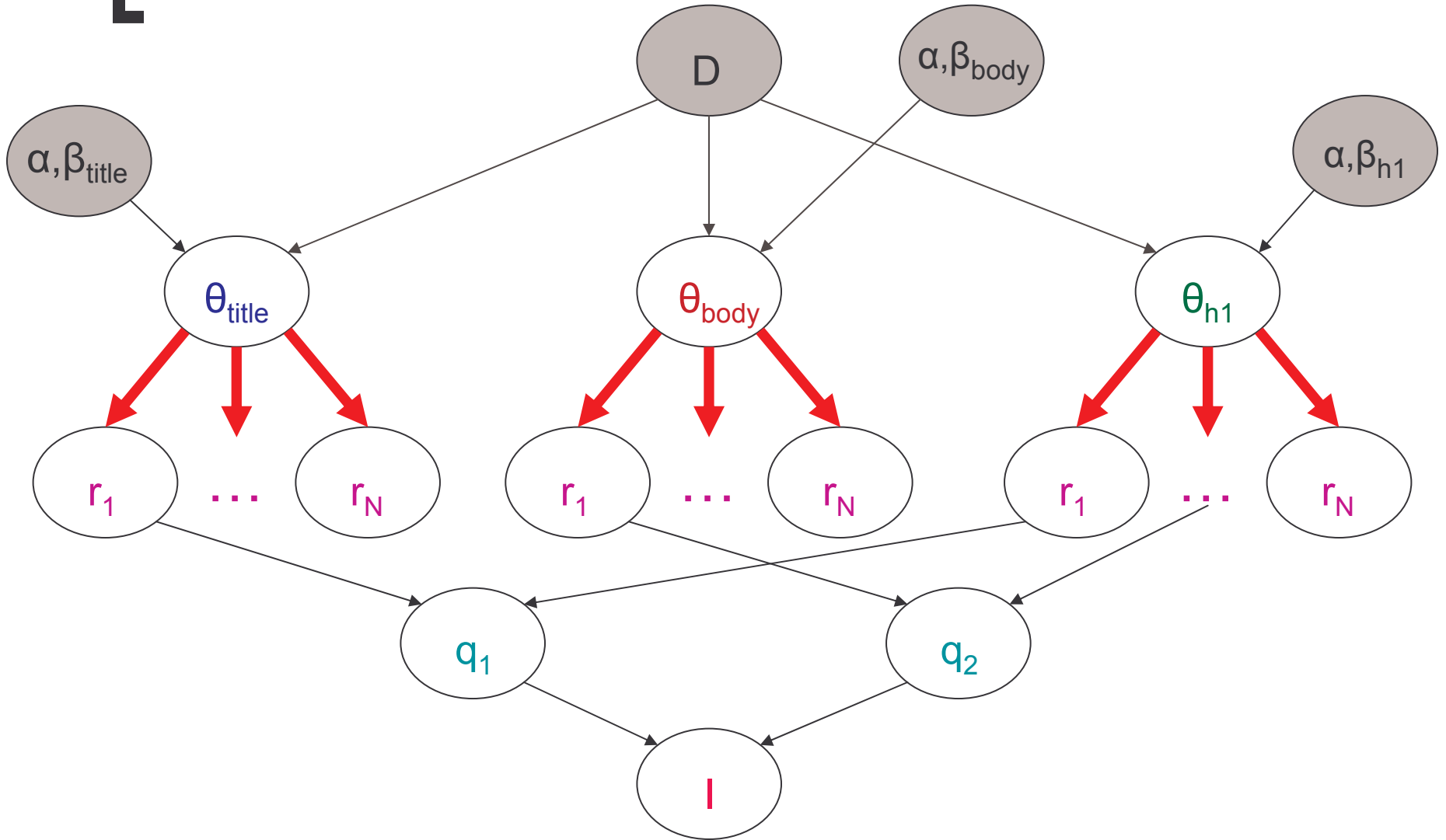
- Based on original inference network retrieval framework [Turtle and Croft '91]
- Casts retrieval as inference in simple graphical model
- Extensions made to original model
 - Incorporation of probabilities based on language modeling rather than *tf.idf*
 - Multiple language models allowed in the network (one per indexed context)

Model

Model hyperparameters (observed)



[Model

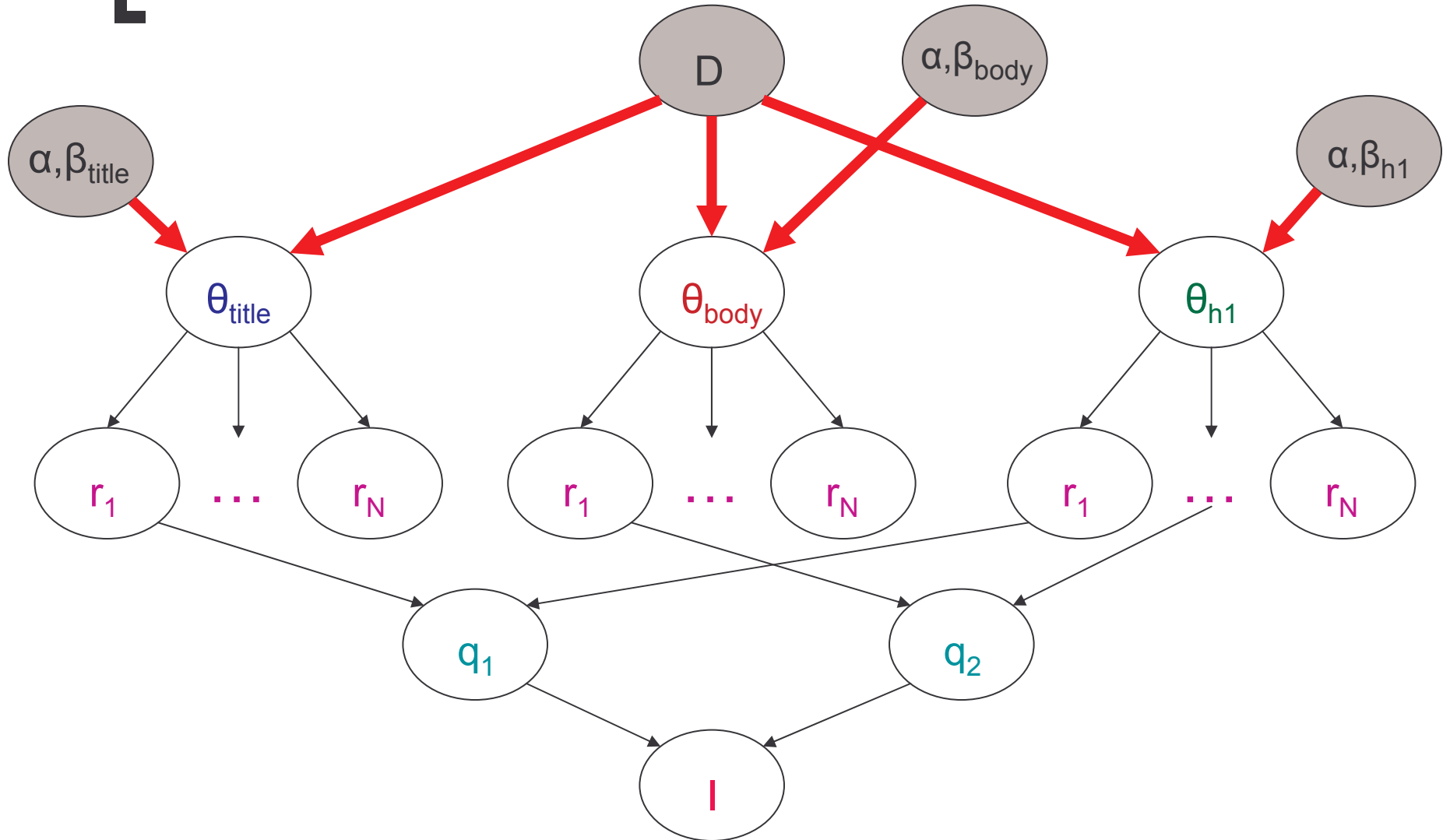


$$[P(r | \theta)$$



- Probability of observing a term, phrase, or “concept” given a context language model
 - r_i nodes are binary
- Assume $r \sim \text{Bernoulli}(\theta)$
 - “Model B” – [Metzler, Lavrenko, Croft ’04]
- Nearly any model may be used here
 - *tf.idf*-based estimates (INQUERY)
 - Mixture models

[Model



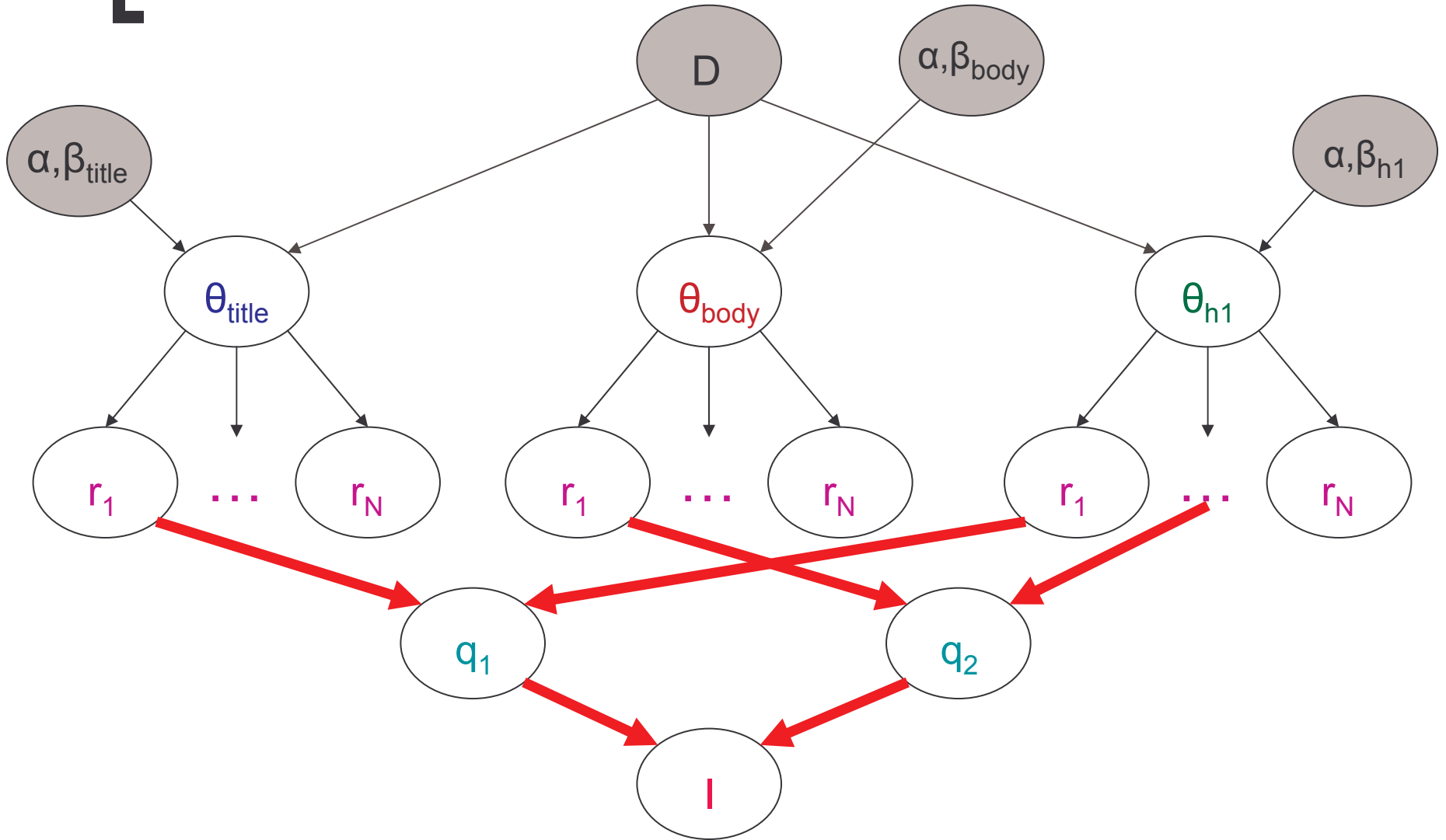
[$P(\theta | \alpha, \beta, D)$]



- Prior over context language model determined by α, β
- Assume $P(\theta | \alpha, \beta) \sim \text{Beta}(\alpha, \beta)$
 - Bernoulli's conjugate prior
 - $\alpha_w = \mu P(w | C) + 1$
 - $\beta_w = \mu P(\neg w | C) + 1$
 - μ is a free parameter

$$P(r_i | \alpha, \beta, D) = \int_{\theta} P(r_i | \theta) P(\theta | \alpha, \beta, D) = \frac{tf_{w,D} + \mu P(w | C)}{|D| + \mu}$$

[Model



[$P(q | r)$ and $P(I | r)$]

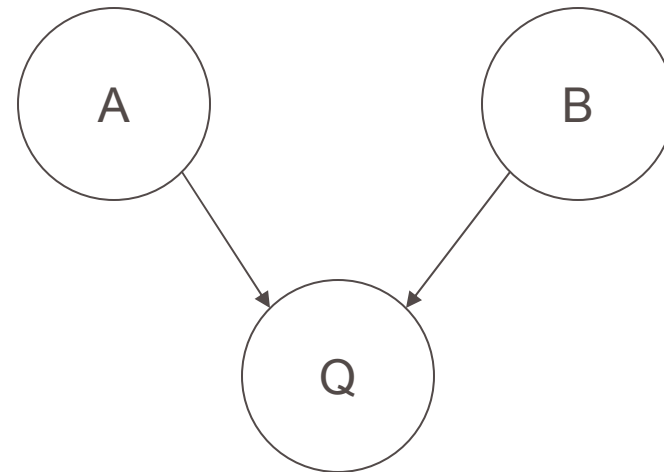


- Belief nodes are created dynamically based on query
- Belief node CPTs are derived from standard link matrices
 - Combine evidence from parents in various ways
 - Allows fast inference by making marginalization computationally tractable
- Information need node is simply a belief node that combines all network evidence into a single value
- Documents are ranked according to:
$$P(I | \alpha, \beta, D)$$

Example: #AND



$P(Q=\text{true} a,b)$	A	B
0	false	false
0	false	true
0	true	false
1	true	true



$$P_{\#and}(Q = \text{true}) = \sum_{a,b} P(Q = \text{true} | A = a, B = b) P(A = a) P(B = b)$$

$$= P(t | f, f)(1 - p_A)(1 - p_B) + P(t | f, t)(1 - p_A)p_B + P(t | t, f)p_A(1 - p_B) + P(t | t, t)p_A p_B$$

$$= 0(1 - p_A)(1 - p_B) + 0(1 - p_A)p_B + 0p_A(1 - p_B) + 1p_A p_B$$

$$= p_A p_B$$

[Query Language



- Extension of INQUERY query language
- Structured query language
 - Term weighting
 - Ordered / unordered windows
 - Synonyms
- Additional features
 - Language modeling motivated constructs
 - Added flexibility to deal with fields via contexts
 - Generalization of passage retrieval (extent retrieval)
- Robust query language that handles many current language modeling tasks

[Terms



<i>Type</i>	<i>Example</i>	<i>Matches</i>
Stemmed term	dog	All occurrences of <i>dog</i> (and its stems)
Surface term	"dogs"	Exact occurrences of <i>dogs</i> (without stemming)
Term group (synonym group)	<"dogs" canine>	All occurrences of <i>dogs</i> (without stemming) or <i>canine</i> (and its stems)
POS qualified term	<"dogs" canine>.NNS	Same as previous, except matches must also be tagged with the <i>NNS</i> POS tag

[Proximity



Type	Example	Matches
$\#odN(e_1 \dots e_m)$ or $\#N(e_1 \dots e_m)$	$\#od5(\text{dog cat})$ or $\#5(\text{dog cat})$	All occurrences of <i>dog</i> and <i>cat</i> appearing ordered within a window of 5 words
$\#uwN(e_1 \dots e_m)$	$\#uw5(\text{dog cat})$	All occurrences of <i>dog</i> and <i>cat</i> that appear in any order within a window of 5 words
$\#phrase(e_1 \dots e_m)$	$\#phrase(\#1(\text{willy wonka})$ $\#uw3(\text{chocolate factory}))$	System dependent implementation (defaults to $\#odm$)
$\#syntax:xx(e_1 \dots e_m)$	$\#syntax:np(\text{fresh powder})$	System dependent implementation

[Context Restriction



<i>Example</i>	<i>Matches</i>
dog.title	All occurrences of <i>dog</i> appearing in the <i>title</i> context
dog.title,paragraph	All occurrences of <i>dog</i> appearing in both a <i>title</i> and <i>paragraph</i> contexts (may not be possible)
<dog.title dog.paragraph>	All occurrences of <i>dog</i> appearing in either a <i>title</i> context or a <i>paragraph</i> context
#5(dog cat).head	All matching windows contained within a <i>head</i> context

[Context Evaluation



<i>Example</i>	<i>Evaluated</i>
dog.(title)	The term <i>dog</i> evaluated using the <i>title</i> context as the document
dog.(title, paragraph)	The term <i>dog</i> evaluated using the concatenation of the <i>title</i> and <i>paragraph</i> contexts as the document
dog.figure(paragraph)	The term <i>dog</i> restricted to <i>figure</i> tags within the <i>paragraph</i> context.

[Belief Operators



<i>INQUERY</i>	<i>INDRI</i>
#sum / #and	#combine
#wsum*	#weight
#or	#or
#not	#not
#max	#max

* #wsum is still available in INDRI, but should be used with discretion

[Extent Retrieval



<i>Example</i>	<i>Evaluated</i>
<code>#combine[section](dog canine)</code>	Evaluates <code>#combine(dog canine)</code> for each extent associated with the <i>section</i> context
<code>#combine[title, section](dog canine)</code>	Same as previous, except is evaluated for each extent associated with either the <i>title</i> context or the <i>section</i> context
<code>#sum(#sum[section](dog))</code>	Returns a single score that is the <i>#sum</i> of the scores returned from <code>#sum(dog)</code> evaluated for each <i>section</i> extent
<code>#max(#sum[section](dog))</code>	Same as previous, except returns the maximum score

Extent Retrieval Example



Query:
#combine[section](dirichlet smoothing)

```
<document>
<section><head>Introduction</head>
Statistical language modeling allows formal
methods to be applied to information retrieval.
...
</section>
<section><head>Multinomial Model</head>
Here we provide a quick review of multinomial
language models.
...
</section>
<section><head>Multiple-Bernoulli Model</head>
We now examine two formal methods for
statistically modeling documents and queries
based on the multiple-Bernoulli distribution.
...
</section>
...
</document>
```

→ 0.15

→ 0.50

→ 0.05

1. Treat each *section extent* as a “document”
2. Score each “document” according to #combine(...)
3. Return a ranked list of *extents*.

<i>SCORE</i>	<i>DOCID</i>	<i>BEGIN</i>	<i>END</i>
0.50	IR-352	51	205
0.35	IR-352	405	548
0.15	IR-352	0	50
...

[Example Tasks



- *Ad hoc* retrieval
 - Flat documents
 - SGML/XML documents
- Web search
 - Homepage finding
 - Known-item finding
- Question answering
- KL divergence based ranking
 - Query models
 - Relevance modeling
- Cluster-base language models

[Ad Hoc Retrieval



- Flat documents
 - Query likelihood retrieval:
 $q_1 \dots q_N \equiv \#combine(q_1 \dots q_N)$
- SGML/XML documents
 - Can either retrieve documents or extents
 - Context restrictions and context evaluations allow exploitation of document structure

[Web Search



- Homepage / known-item finding
- Use mixture model of several document representations [Ogilvie and Callan '03]

$$P(w | \Theta, \Lambda) = \lambda_{body} P(w | \theta_{body}) + \lambda_{inlink} P(w | \theta_{inlink}) + \lambda_{title} P(w | \theta_{title})$$

- Example query: Yahoo!

```
#combine( #wsum(0.2 yahoo.(body)
            0.5 yahoo.(inlink)
            0.3 yahoo.(title) ) )
```

[Question Answering



- More expressive passage- and sentence-level retrieval
- Example:
 - *Where was George Washington born?*
`#combine[sentence](#1(george washington)
born #place(?))`
 - Returns a ranked list of **sentences** containing the **phrase George Washington**, the **term born**, and a **snippet of text tagged as a PLACE named entity**

[KL / Cross Entropy Ranking]



- INDRI handles ranking via KL / cross entropy
 - Query models
 - Relevance modeling
- Relevance modeling [Lavrenko and Croft '01]
 - Do initial query run and retrieve top N documents
 - Form relevance model $P(w | \theta_Q)$
 - Formulate and rerun expanded query as:

#weight $(P(w_1 | \theta_Q) w_1 \dots P(w_{|V|} | \theta_Q) w_{|V|})$

- Ranked list equivalent to scoring by: $KL(\theta_Q || \theta_D)$

[Cluster-Based LMs]



- Cluster-based language models [Liu and Croft '04]
- Smooths each document with a document-specific cluster model

$$P(w | \Theta, \Lambda) = \lambda_D P(w | \theta_D) + (1 - \lambda_D) [\lambda_{Cluster} P(w | Cluster) + (1 - \lambda_{Cluster}) P(w | C)]$$

- INDRI allows each document to specify (via metadata) a background model to smooth against
- Many other possible uses of this feature

[Conclusions



- INDRI extends INQUERY and Lemur
 - Off the shelf
 - Scalable
- Geared towards tagged (structured) documents
- Employs robust inference net approach to retrieval
- Extended query language can tackle many current retrieval tasks