Chapter IR:V

V. Retrieval Models

Combining Evidence

- Effective retrieval requires the combination of many pieces of evidence about a document's potential relevance
  - Until now: focus on simple word-based evidence
  - Many other types of evidence
    - Words: structure, proximity of words, relationships among words
    - Metadata: PageRank, publication date, document type
    - Scores from different models
- Variant 1: Adapt BM25 or Query Likelihood with additional factors
  - Difficult to maintain, understand and tune
  - But there is a well-understood variant BM25F
- Variant 2: Inference network model to combine evidence
  - Probabilistic model
  - Uses Bayesian network formalism
  - Mechanism to define and evaluate operators in a query language
    - Operators to specify evidence
    - Operators to combine evidence

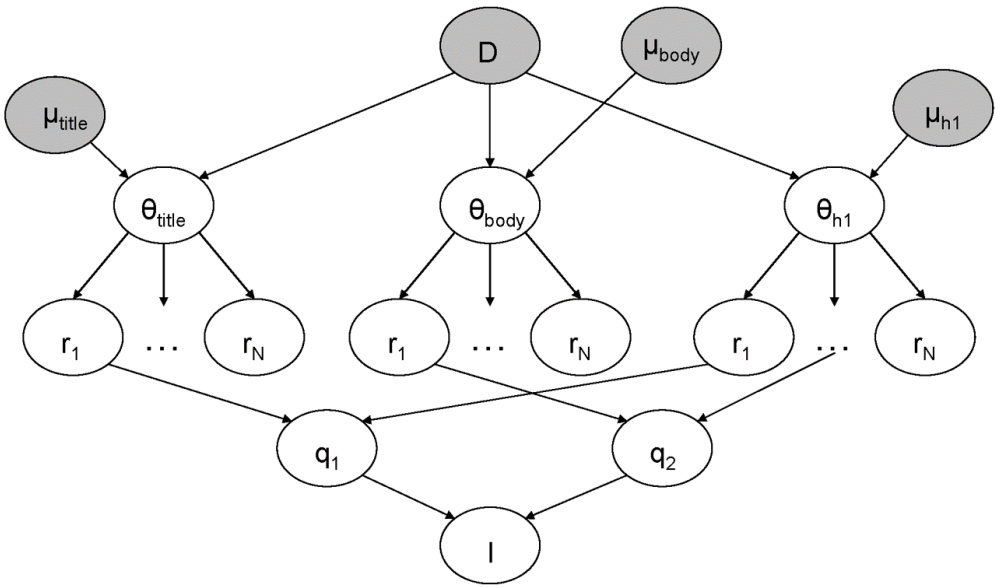
Combining Evidence

Bayesian Networks

- Probabilistic model
- Specifies set of events and dependencies between them
- Modeled as DAG – directed acyclic graph
  - Nodes: events
    - Here: observing a particular document or piece of evidence or some combination of evidences
    - All binary
  - Arcs: probabilistic dependencies between events

Combining Evidence

Inference Network

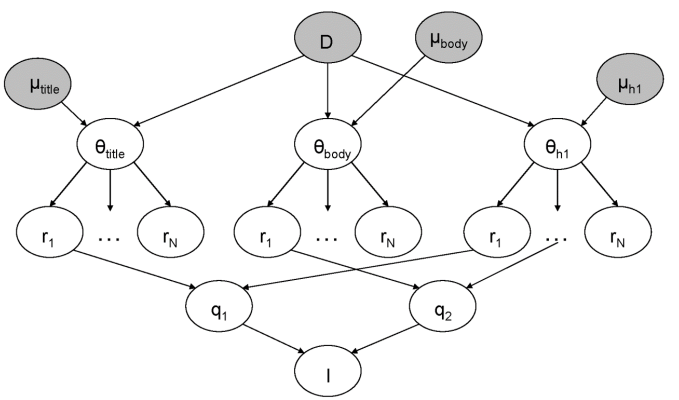


- One document node (D) per document in the collection
- Example combines evidence about web page title, body, and <h1> headings
- θ-nodes are language models with μ-parameters
- r_i are representation nodes (features, evidence); probabilities depend on θ
- Query nodes q_i combine more complex evidence
- Information need node I combines query evidence

Combining Evidence

Inference Network

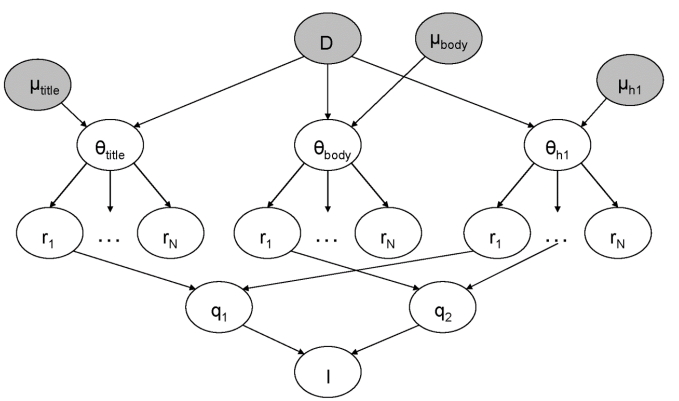
- Document node (D) corresponds to the event that a document is observed
- Representation nodes (r_i) are document features (evidence)
  - Probabilities associated with those features are based on language models θ estimated using the parameters μ
  - One language model for each significant document structure
  - r_i nodes can represent proximity features, or other types of evidence, e.g., date



Combining Evidence

Inference Network

- Query nodes (q_i) are used to combine evidence from representation nodes and other query nodes
  - Represent the occurrence of more complex evidence and document features
  - A number of combination operators are available
    - AND, OR, ...
- Information need node (I) is a special query node that combines all of the evidence from the other query nodes
  - In all, network computes $P(I|D, \mu)$
  - = probability that an information need is met given the document and the parameters μ
  - Used to rank documents



Combining Evidence

Inference Network

- Connections in an inference network are defined by the query and the representation nodes
- Probabilities for representation nodes estimated using a relevance model
  - Reflect the probability that a feature is characteristic for a document
    - Not probability of occurrence
  - Node for `lincoln` represents the binary event that a document is about that topic
  - Relevance model used to calculate the probability that that event is TRUE
- Document is represented by binary vector

Combining Evidence

Inference Network

- To calculate probabilities:

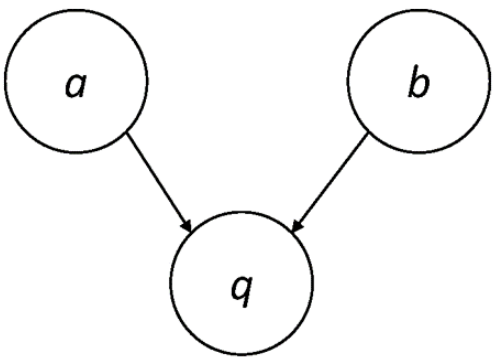
$$P(r_i|D,\mu) = \frac{f_{r_i,D} + \mu P(r_i|C)}{|d| + \mu}$$

  - Same as before – Dirichlet smoothing
  - $f_{r_i,D}$ is number of times feature r_i occurs in D
  - $P(r_i|C)$ is collection probability for feature r_i
  - μ is Dirichlet smoothing parameter
    - Specific to the document structure of interest
- Example: $f_{i,D}$ is number of times `lincoln` appears in title
  - Collection probability calculated based on all collection titles
  - μ is title-specific

Combining Evidence

Example: AND Combination

- Query nodes are basis for operators of query language
  - Restricted to combinations that can be efficiently calculated
  - Calculate probability of each outcome (true or false) given all possible states of parent nodes
- Example for Boolean AND:



a and b are parent nodes for q

$P(q = TRUE \| a, b)$	a	b
0	FALSE	FALSE
0	FALSE	TRUE
0	TRUE	FALSE
1	TRUE	TRUE

Combining Evidence

Example: AND Combination

- Combination must consider all possible states of parents
- Some combinations can be computed efficiently
- Let p_{xy} denote probability that q is TRUE given state x and y of parents
  - p_a is probability that a is TRUE
- Calculate *belief value* (probability) from an AND combination:

$$
\begin{aligned}
bel_{\text{AND}}(q) = \quad & p_{00} P(a = \text{FALSE}) P(b = \text{FALSE}) \\
& + p_{01} P(a = \text{FALSE}) P(b = \text{TRUE}) \\
& + p_{10} P(a = \text{TRUE}) P(b = \text{FALSE}) \\
& + p_{11} P(a = \text{TRUE}) P(b = \text{TRUE}) \\
= \quad & 0 \cdot (1 - p_a)(1 - p_b) + 0 \cdot (1 - p_a) p_b + 0 \cdot p_a (1 - p_b) + 1 \cdot p_a p_b \\
= \quad & p_a p_b
\end{aligned}
$$

Combining Evidence

Inference Network Operators

- Other operators can also be calculated efficiently
- Let q have n parents
  - each with probability p_i of being true
  - and some weight wt_i to indicate relative importance

$$bel_{\text{NOT}}(q) = 1 - p_1$$

$$bel_{\text{OR}}(q) = 1 - \prod_i^n (1 - p_i)$$

$$bel_{\text{AND}}(q) = \prod_i^n p_i$$

$$bel_{\text{WAND}}(q) = \prod_i^n p_i^{wt_i}$$

$$bel_{\text{MAX}}(q) = \max p_1, p2, \ldots, p_n$$

$$bel_{\text{SUM}}(q) = \frac{\sum_i^n p_i}{n}$$

$$bel_{\text{WSUM}}(q) = \frac{\sum_i^n wt_i p_i}{\sum_i^n wt_i}$$

Combining Evidence

Query Language Example

- Given description of underlying model and combination operators, an internal query language can be used to produce rankings based on complex combinations of evidence
- Example: Galago/Indri
- Query: `pet therapy` compiled to Galago query

```
#weight(
0.1 #weight( 0.6 #prior(pagerank) 0.4 #prior(inlinks))
1.0 #weight(
    0.9 #combine(
        #weight(1.0 pet.(anchor) 1.0 pet.(title)
         3.0 pet.(body) 1.0 pet.(heading))
        #weight(1.0 therapy.(anchor) 1.0 therapy.(title)
         3.0 therapy.(body) 1.0 therapy.(heading)))
    0.1 #weight(
        1.0 #od1(pet therapy).(anchor)
        1.0 #od1(pet therapy).(title)
        3.0 #od1(pet therapy).(body)
        1.0 #od1(pet therapy).(heading))
    0.1 #weight(
        1.0 #uw8(pet therapy).(anchor)
        1.0 #uw8(pet therapy).(title)
        3.0 #uw8(pet therapy).(body)
        1.0 #uw8(pet therapy).(heading))))
```