

# Representing Queries as Structures

Michael Bendersky

*Joint work with*

*W. Bruce Croft, David A. Smith*  
*University of Massachusetts Amherst*

*Donald Metzler*  
*University of Southern California*

*“The meaning of a representation can be nothing but a representation.”*

**Charles Sanders Peirce (1839 -1914)**

# **STRUCTURAL REPRESENTATION OF QUERIES**

# The Challenge of Query Representation

- User inputs a string of characters
- Query structure is never explicitly observed and is difficult to infer
  - Short and ambiguous search queries

*new york times square* 

- Idiosyncratic grammar

*do grover cleveland have kids* 

- No capitalization and punctuation

*talking to heaven movie* 

# Existing Query Representations

- Term-Based (a.k.a. *bag-of-words*)
  - BM25 (*Robertson et al., 2000*)
  - Query Likelihood (*Ponte & Croft, 1998*)
  - DFR (*Amati, 2003*)
  - ...
- Concept-Based
  - Markov Random Fields for IR (*Metzler & Croft, 2005*)
  - BM25 with term proximities (*Song et al., 2008*)
  - DFR-SD, DFR-FD (*Peng et al., 2007*)
  - ...

# Structural Query Representation

- A query  $Q$  has a *hierarchical representation*
  - A query is a set of structures  $\Sigma = \{\sigma_1, \dots, \sigma_n\}$
  - Each structure is a set of concepts  $\sigma = \{\kappa_1, \kappa_2, \dots\}$
- Hierarchical representation allows to
  - Model arbitrary term dependencies as concepts
  - Group concepts by structures
  - Assign weights to concepts/structures

Structures

Terms

*members rock group nirvana* 

Bigrams

[members] [rock] [group] [nirvana]

Chunks

[members rock] [rock group] [group nirvana]

Key Concepts

[members] [rock group] [nirvana]

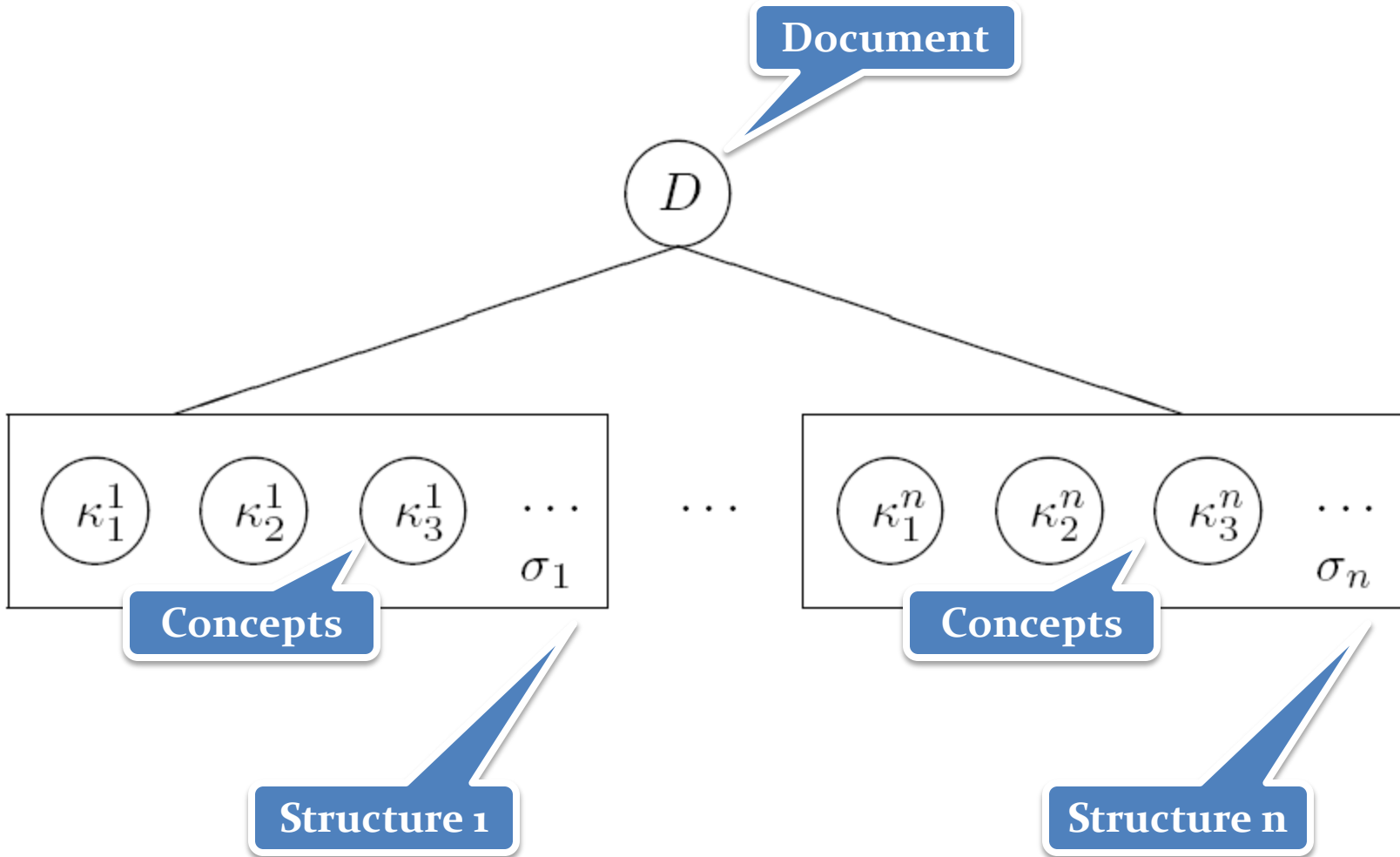
Dependence

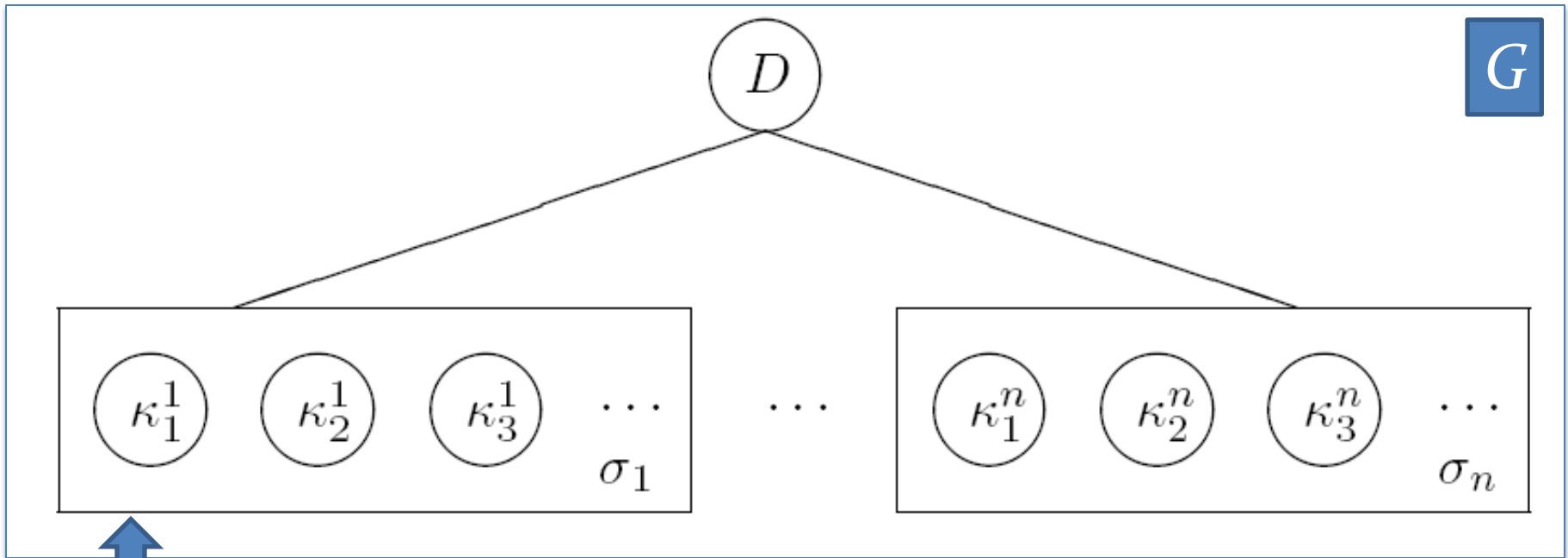
[members] [nirvana]

[members nirvana] [rock group]

Concepts

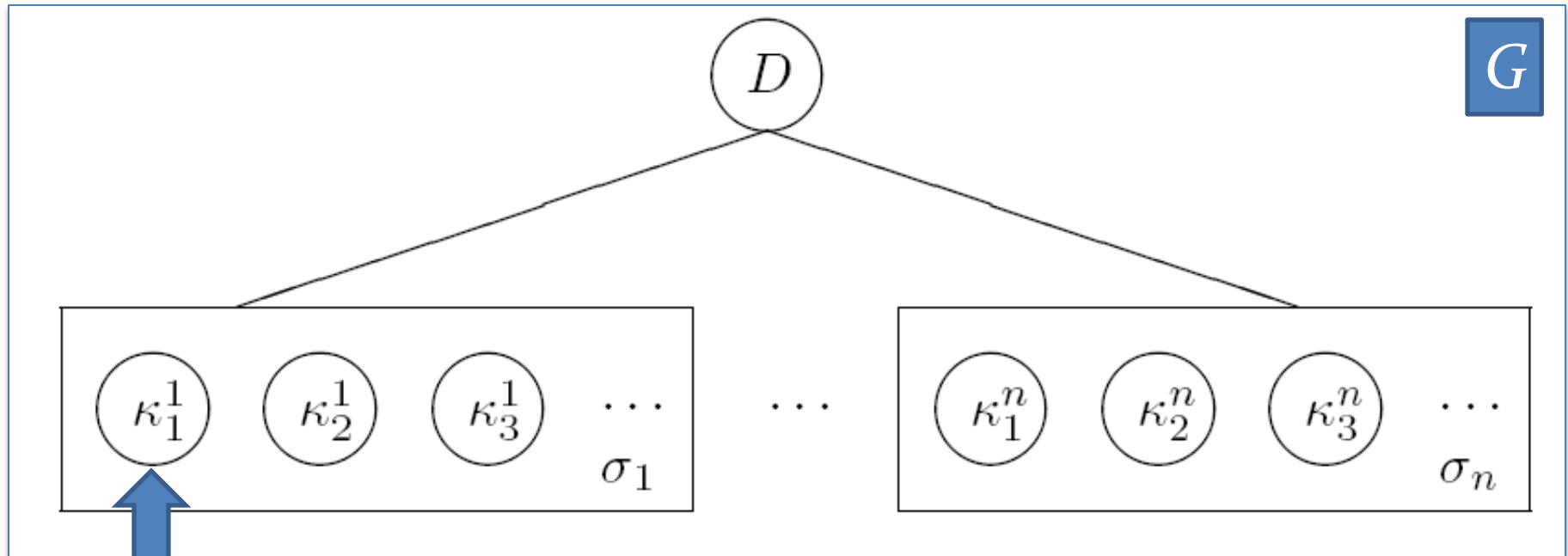
# Encoding query structure ( $G$ )



 $\lambda_\sigma$ 

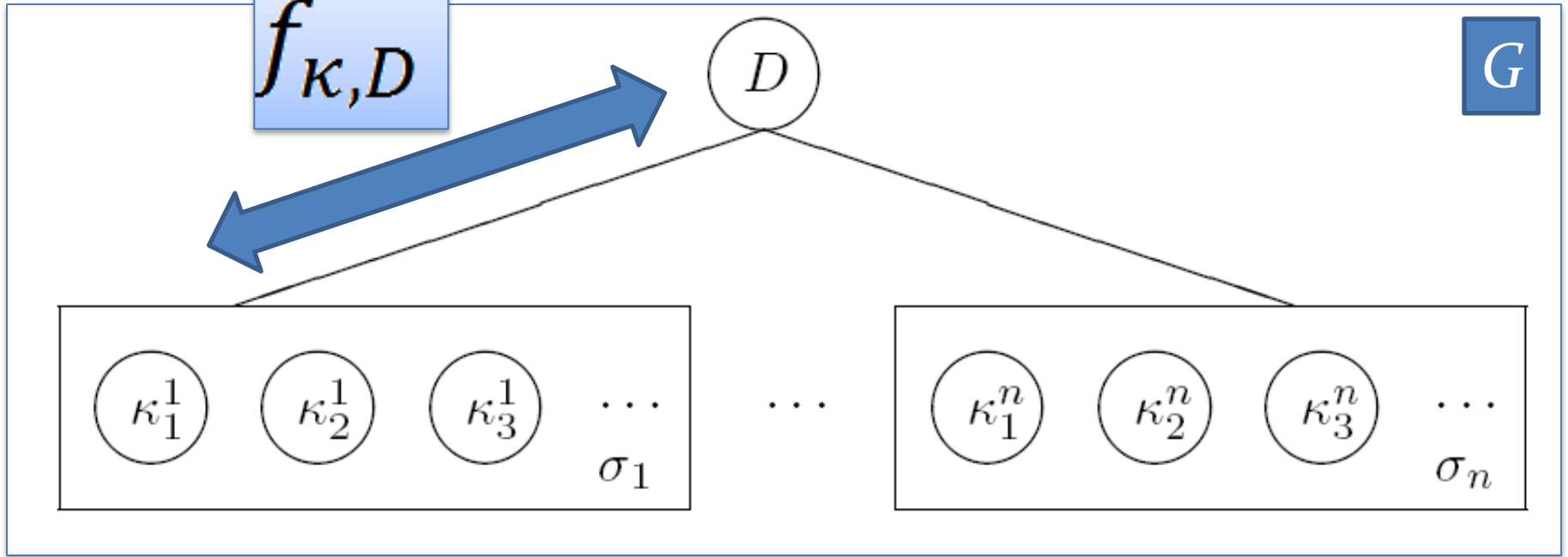
Representation fitness of the structure  $\sigma$

- Structures with higher value can be viewed as more accurate representations of the query  $Q$

 $\lambda_\kappa$ 

## Importance of the concept $\kappa$

- How imperative is the concept for conveying the query intent to the search engine?

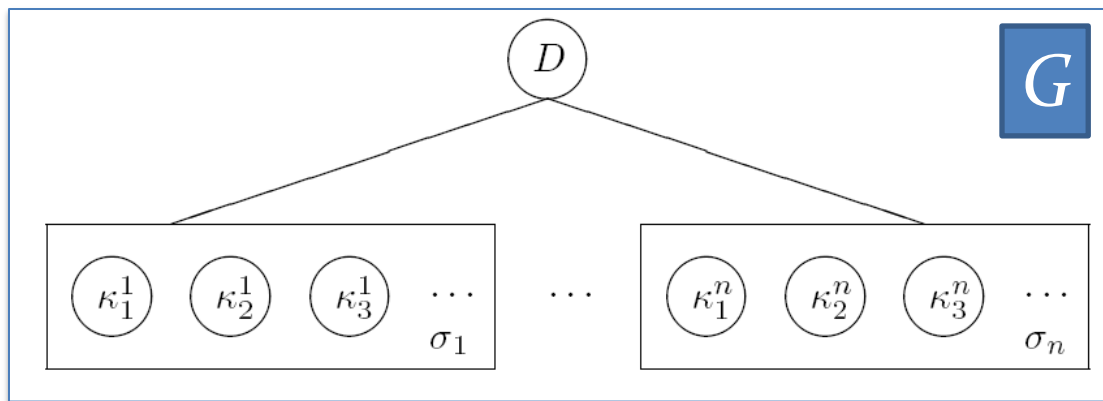


## Concept scoring function

- Scoring the occurrence of the concept in document  $D$

# Document Score Derivation

$$sc(Q, D) = \sum_{\sigma \in \Sigma} \lambda_{\sigma} \sum_{\kappa \in \sigma} \lambda_{\kappa} f_{\kappa, D}$$

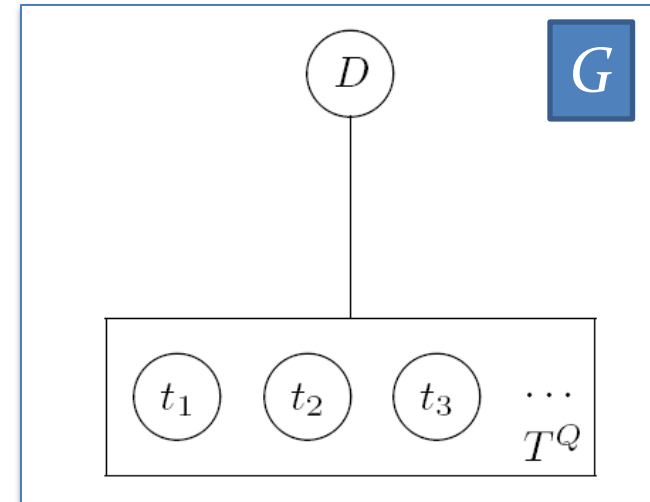


# RETRIEVAL WITH STRUCTURAL QUERY REPRESENTATIONS

# Bag-of-Words Models

members rock group nirvana 

[members] [rock] [group] [nirvana]



Assuming  
uniform  
concept  
weights

$$sc(Q, D) = \sum_{t \in T^Q} f(t, D)$$

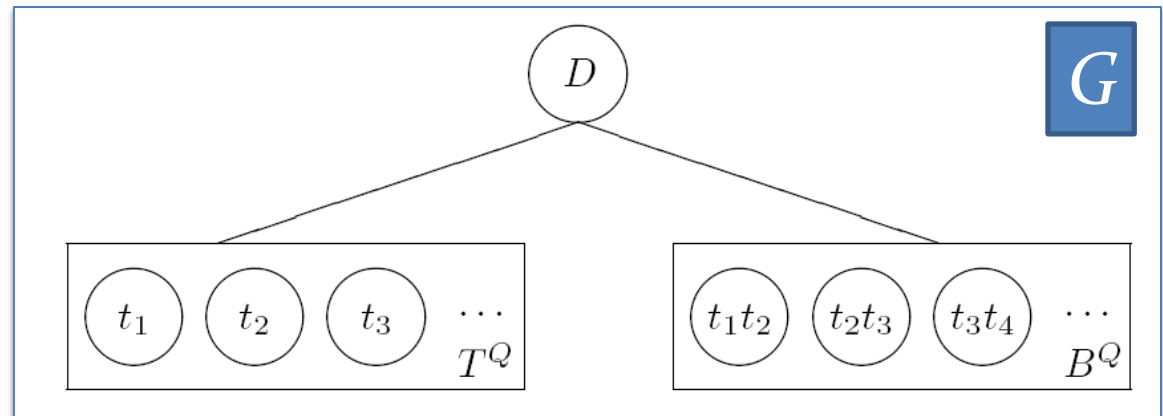
# Sequential Dependence Model (Metzler & Croft, 2005)

members rock group nirvana 

[members] [rock] [group] [nirvana]

[members rock] [rock group] [group nirvana]

$$sc(Q, D) = \lambda_{T^Q} \sum_{t \in T^Q} f(t, D) + \lambda_{B^Q} \sum_{b \in B^Q} f(b, D)$$



# Weighted Sequential Dependence

(Bendersky, Metzler & Croft, 2010)

- Similar to the sequential dependence model, but with non-uniform concept importance weights

$$sc(Q, D)$$

$$= \lambda_{T^Q} \sum_{t \in T^Q} \lambda_t f(t, D)$$

$$+ \lambda_{B^Q} \sum_{b \in B^Q} \lambda_b f(b, D)$$

Concept weights

# Key Concepts

(Bendersky & Croft, 2008)

$$sc(Q, D) = \lambda_{T^Q} \sum_{t \in T^Q} f(t, D) + \lambda_{N^Q} \sum_{n \in N^Q} \lambda_n f(n, D)$$

Query terms

Weighted noun phrases

*members rock group nirvana* 

[members] [rock] [group] [nirvana]

**0.3** [members] **0.7** [nirvana]

# Query Segmentation

(Bendersky, Croft & Smith, 2009)

$$sc(Q, D) = \lambda_{TQ} \sum_{t \in T^Q} f(t, D) + \sum_{s \in S^Q} \lambda_s \sum_{c \in C^s} f(c, D)$$

Query terms

Segments

Chunks

*members rock group nirvana* 

[members] [rock] [group] [nirvana]

**0.8**

[members] [rock group] [nirvana]

**0.2**

[members] [rock group nirvana]

# Directions for Future Work

- Integrating multiple latent query structures
  - *Segments*
  - *Noun phrases*
  - *Dependencies*
  - ...
- Expanding queries with structures
- Structures of sequences of queries