TEXT RETRIEVAL
AND
SEARCH ENGINES

The basic concepts, principles, and the major techniques in text retrieval, which is the underlying science of search engines.

Course author:

ChengXiang Zhai

University of Illinois at Urbana-Champaign
&
Coursera

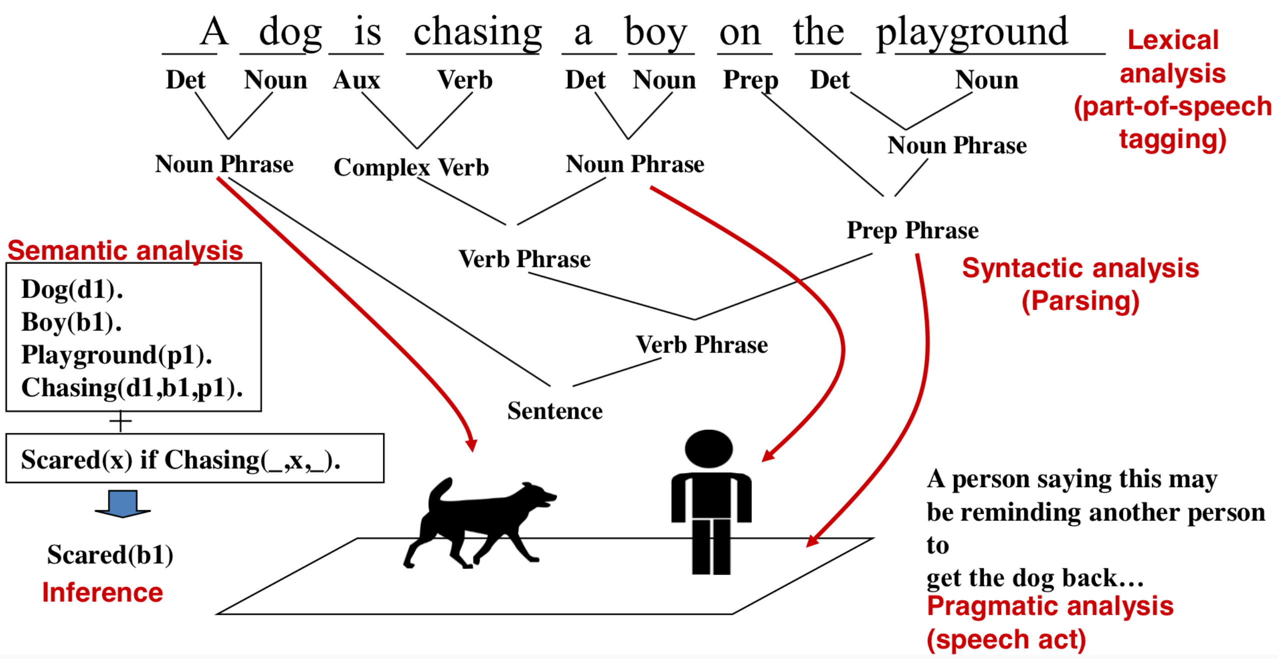
2015

Contents

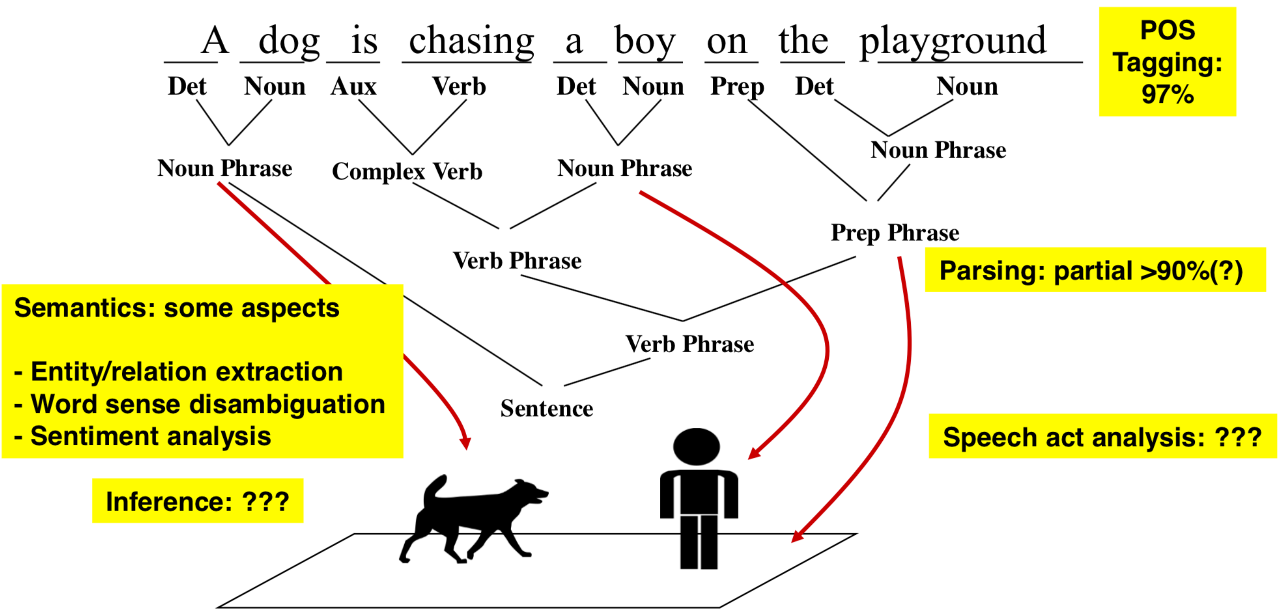
1 Natural Language Content Analysis

NLP = Natural Language Processing

1.1 An Example of NLP



1.2 The State of the Art



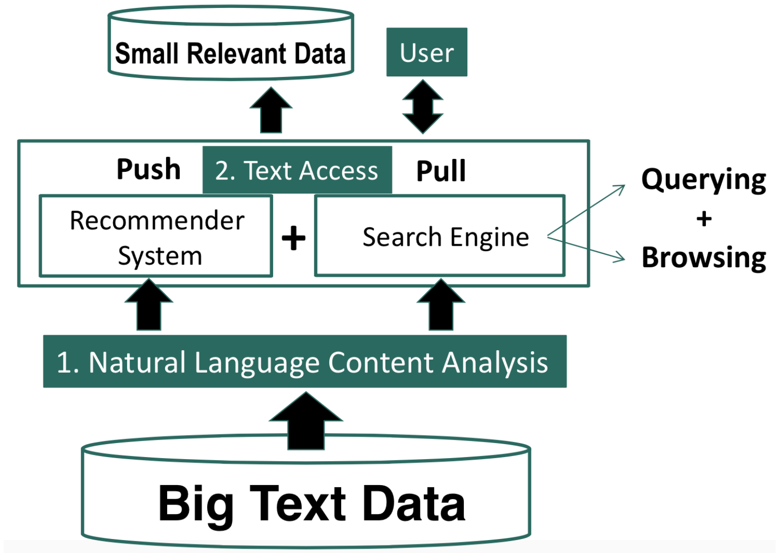
1.3 Recommended reading

- Chris Manning and Hinrich Schütze, «Foundations of Statistical Natural Language Processing», MIT Press. Cambridge, MA: May 1999.

2 Text Access

2.1 Two Modes of Text Access: Pull vs. Push

- Pull Mode (search engines) – Users take initiative
  - Ad hoc information need
- Push Mode (recommender systems)
  - Systems take initiative
  - Stable information need or system has good knowledge about a user's need



2.2 Pull Mode: Querying vs. Browsing

- Querying
  - User enters a (keyword) query
  - System returns relevant documents
  - Works well when the user knows what keywords to use
- Browsing
  - User navigates into relevant information by following a path enabled by the structures on the documents
  - Works well when the user wants to explore information, doesn't know what keywords to use, or can't conveniently enter a query

2.3 Recommended reading

- N. J. Belkin and W. B. Croft. 1992. «Information filtering and information retrieval: two sides of the same coin?» Commun. ACM 35, 12 (Dec. 1992), 29-38.

3 Text Retrieval Problem

3.1 What Is Text Retrieval?

TR = Text Retrieval[1]

- Collection of text documents exists
- User gives a query to express the information need
- Search engine system returns relevant documents to users
- Often called "information retrieval" (IR), but IR is actually much broader
- Known as «search technology» in industry

TR is an empirically defined problem:

- Can't mathematically prove one method is better than another
- Must rely on empirical evaluation involving users!

3.2 Formal Formulation of TR

- **Vocabulary**: $V = \{w_1, w_2, \dots, w_N\}$ of language
- **Query**: $q = q_1, \dots, q_m$, where $q_i \in V$
- **Document**: $d_i = d_{i1}, \dots, d_{im_i}$, where $d_{ij} \in V$
- **Collection**: $C = \{d_1, \dots, d_M\}$
- **Set of relevant documents**: $R(q) \subseteq C$
  - Generally unknown and user-dependent
  - Query is a «hint» on which doc is in $R(q)$
- **Task**: compute $R'(q)$, an approximation of $R(q)$

3.3 How to Compute $R'(q)$

- Strategy 1: Document selection
  - $R'(q) = \{d \in C \mid f(d,q) = 1\}$, where $f(d,q) \in \{0,1\}$ is an indicator function or binary classifier
  - System must decide if a doc is relevant or not (absolute relevance)
- Strategy 2 (generally preferred): Document ranking
  - $R'(q) = \{d \in C \mid f(d,q) > \theta\}$, where $f(d,q) \in \Re$ is a relevance measure function; θ is a cutoff determined by the user
  - System only needs to decide if one doc is more likely relevant than another (relative relevance)

[1] Retrieval - поиск

3.4 Theoretical Justification for Ranking

Probability Ranking Principle [Robertson 77]: Returning a ranked list of documents in descending order of probability that a document is relevant to the query is the optimal strategy under the following two assumptions:

- The utility of a document (to a user) is independent of the utility of any other document
- A user would browse the results sequentially

3.5 Recommended reading

- S.E. Robertson, «The probability ranking principle in IR». Journal of Documentation 33, 294-304, 1977
- **C. J. van Rijsbergen, «Information Retrieval», 2nd Edition**, Butterworth-Heinemann, Newton, MA, USA, 1979

4 Overview of Text Retrieval Methods

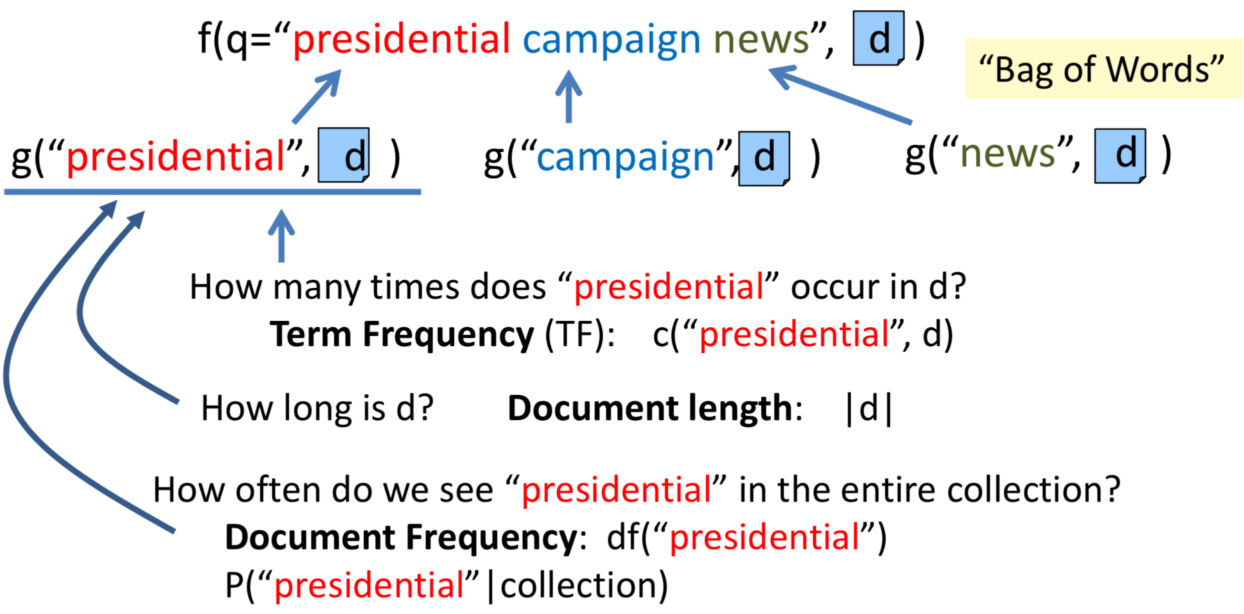
4.1 How to Design a Ranking Function

- **Query**: $q = q_1, \dots, q_m$, where $q_i \in V$
- **Document**: $d = d_1, \dots, d_n$, where $d_i \in V$
- **Ranking function**: $f(q, d) \in \Re$
- **Key challenge**: how to measure the likelihood that document d is relevant to query q
- **Retrieval model**: formalization of relevance (give a computational definition of relevance)

4.2 Retrieval Models

- **Similarity-based models**: $f(q, d) = similarity(q, d)$
  - Vector space model
- **Probabilistic models**: $f(d, q) = p(R = 1 \mid d, q)$, where $R \in 0, 1$
  - Classic probabilistic model
  - Language model
  - Divergence-from-randomness model
- **Probabilistic inference model**: $f(q, d) = p(d \to q)$
- **Axiomatic model**: $f(q, d)$ must satisfy a set of constraints

4.3 Common Ideas in State of the Art Retrieval Models



State of the art ranking functions tend to rely on:

- Bag of words representation
- Term Frequency (TF) and Document Frequency (DF) of words
- Document length

4.4 Which Model Works the Best?

When optimized, the following models tend to perform equally well [Fang et al. 11]:

- **Pivoted length normalization – BM25**
- Query likelihood
- PL2

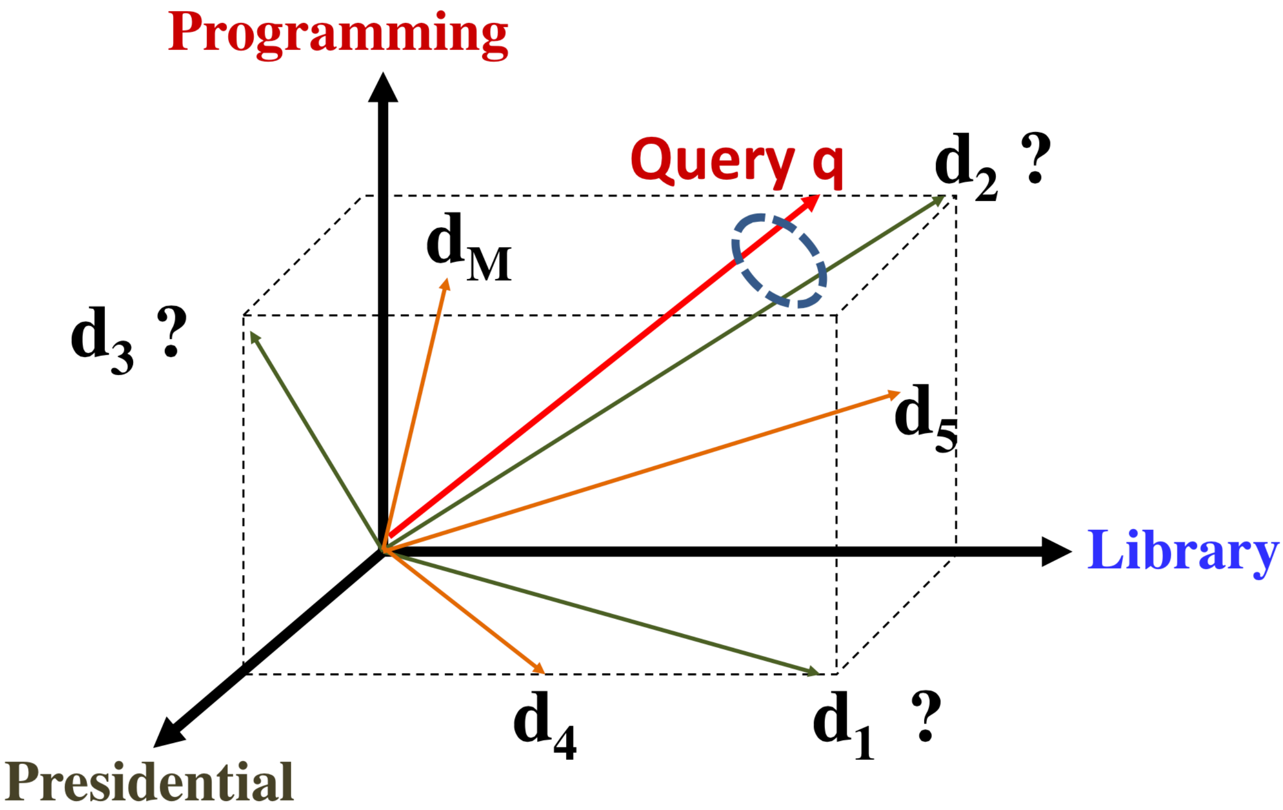
4.5 Recommended reading

- Hui Fang, Tao Tao, and Chengxiang Zhai. 2011. «Diagnostic Evaluation of Information Retrieval Models». ACM Trans. Inf. Syst. 29, 2, Article 7 (April 2011)
- ChengXiang Zhai, «Statistical Language Models for Information Retrieval», Morgan & Claypool Publishers, 2008. (Chapter 2)

5 Vector Space Retrieval Model: Basic Idea

VSM - Vector Space Model

5.1 Vector Space Model (VSM): Illustration



5.2 VSM Is a Framework

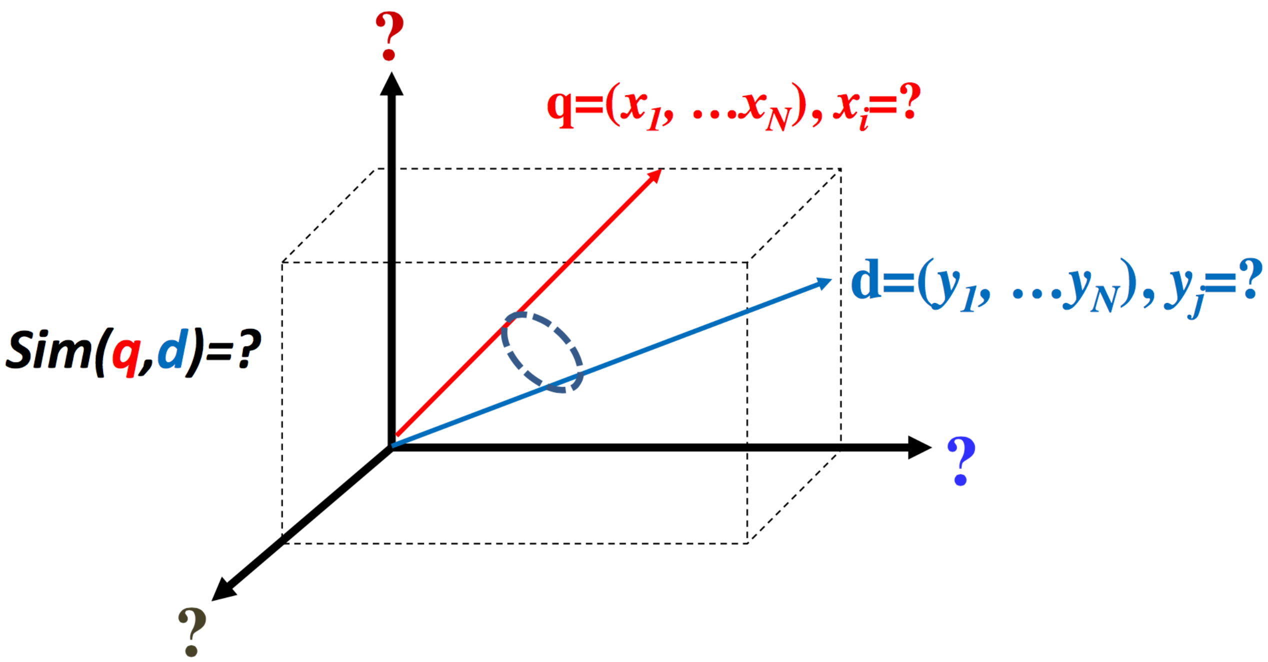
- Represent a doc/query by a term vector
  - **Term**: basic concept, e.g., word or phrase
  - Each term defines one dimension
  - N terms define an **N-dimensional space**
  - **Query vector**: $q = (x_1, \dots x_N), x_i \in \Re$ is query term weight
  - **Doc** vector: $d = (y_1, \dots y_N), y_j \in \Re$ is doc term weight
- $relevance(q, d) \propto similarity(q, d) = f(q, d)$

5.3 What VSM Doesn't Say

- How to define/select the "basic concept" – Concepts are assumed to be orthogonal
- How to place docs and query in the space (= how to assign term weights)
  - Term weight in query indicates importance of term
  - Term weight in doc indicates how well the term characterizes the doc
- How to define the similarity measure

6 Vector Space Retrieval Model: Simplest Instantiation

6.1 What VSM Doesn't Say



6.2 Simplest VSM= Bit-Vector + Dot-Product + BOW

$q=(x_1, \ldots x_N)$

$d=(y_1, \ldots y_N)$

$x_i, y_i \in \{0,1\}$

1: word W_i is **present**

0: word W_i is **absent**

$$Sim(q,d)=q.d= x_1 y_1 + \ldots + x_N y_N = \sum_{i=1}^{N} x_i y_i$$

Simplest VSM:

- Dimension = word
- Vector = 0-1 bit vector (word presence/absence)
- Similarity = dot product
- f(q,d) = number of distinct query words matched in d

7 Vector Space Retrieval Model: Improved Instantiation

Improved VSM:

- Dimension = word
- Vector = TF-IDF weight vector
- Similarity = dot product

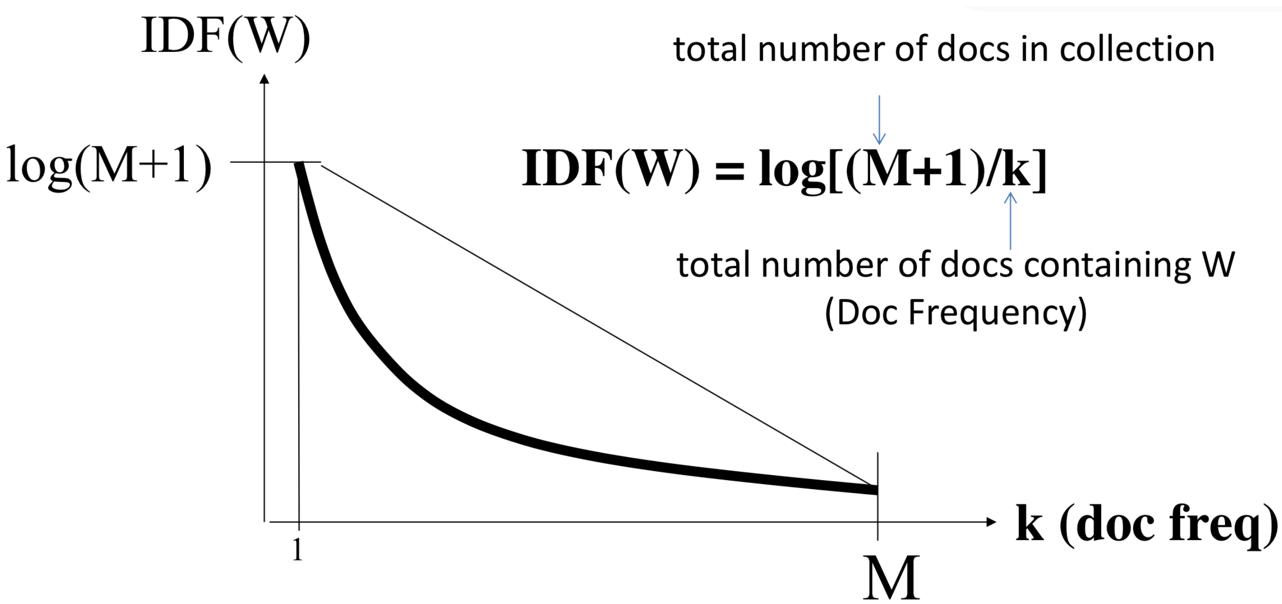
7.1 Improved VSM with Term Frequency (TF) Weighting

$q = (x_1, \ldots x_N)$ $\boxed{x_i = \text{count of word } W_i \text{ in query}}$

$d = (y_1, \ldots y_N)$ $\boxed{y_i = \text{count of word } W_i \text{ in doc}}$

$$Sim(q,d) = q.d = x_1 y_1 + \ldots + x_N y_N = \sum_{i=1}^{N} x_i y_i$$

7.2 IDF Weighting: Penalizing Popular Terms



7.3 Adding Inverse Document Frequency (IDF)