横浜国立大学　大学院　環境情報学府
情報メディア環境学専攻(前期)

言語情報処理原論(12)

森　辰則

mori@forest.eis.ynu.ac.jp

今後の予定

□6/27(水)　講義 第12回
□7/ 4(水)　講義 第13回
□7/11(水)　講義 第14回　最終日
□7/18(水)　大学院入試のため休講
□7/25(水)　予備日(授業は行ないません)

□8/ 3(金)　期末レポート提出

期末レポート(1)

- あなたの研究に最も関連する自然言語処理・文書処理に関連するトピックを一つ選択し，それに関連する学術論文を2編以上読み，以下の両者を含む報告書を作成しなさい.
  - 論文中の技術について，目的，方法，長所，課題を論じなさい。ただし、各項目は自分の言葉で要約し述べること。元論文の単なる抜粋は認めない。
  - あなたの研究を簡単に説明した後，前項で述べた技術が如何に活用できるかについて論じなさい。また、その活用における問題点も考察しなさい.

期末レポート(2)

□提出方法

- ○締切: 2012年8月3日(金) 17:00
- ○提出先: 環境情報研究院等学務係のポスト(今後設置予定)
- ○書式: A4 縦置，横書，4～10ページ程度(表紙を除く). 表紙には学籍番号，氏名，ならびに，参照した論文の出典を明記のこと
- ○注意: 必ず自分で内容を記述すること．原論文の丸写しや，他のレポートと類似する記述が判明した場合には，採点を行なわない．

概要

- 新聞記事やWeb文書等テキスト情報は，重要な情報編纂の対象の一つである.
- 複数のテキストを編纂し，利用者が望む情報を創出するために，情報検索，情報抽出，自動要約等，テキストを対象とした情報アクセス技術が研究されている.

目次

- 今回
  - 情報編纂とテキスト処理
  - 想定される処理
  - 情報検索(文書検索)
- 次回以降
  - Webに特化した情報検索
  - 情報抽出
  - 自動(文書)要約
  - 文書分類

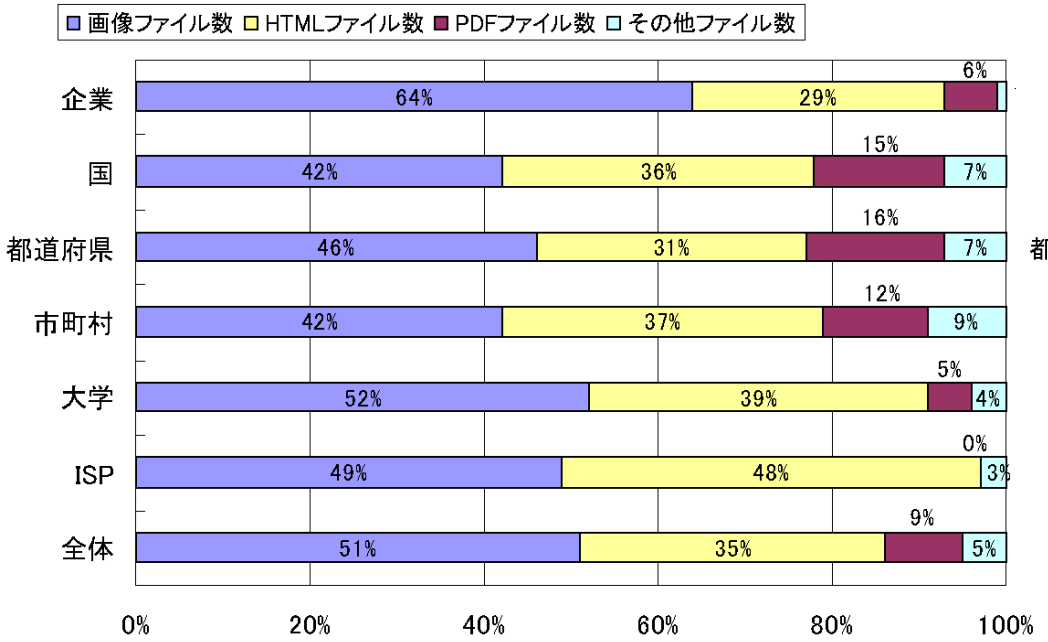
※いずれも入門的な事項を中心に紹介

情報編纂とテキスト処理

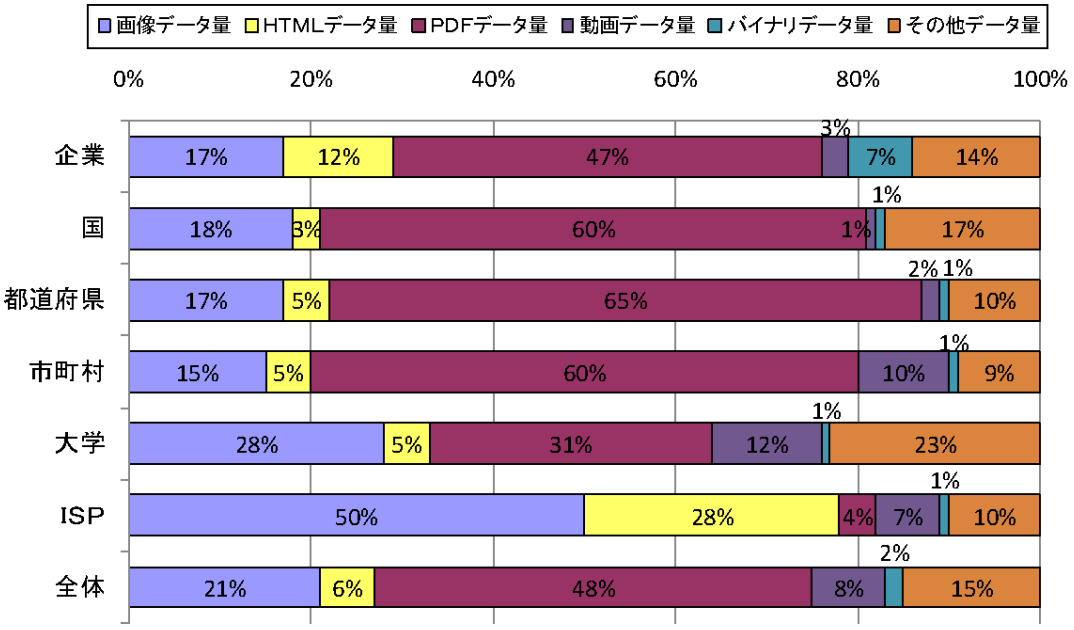
- 情報爆発時代の要の技術の一つ
  - 多量のテキストから、
  - 利用者が注目するであろう情報を抽出し、
  - コンパクトに纏め上げて、提示

WWWコンテンツ統計調査報告書 [総務省 07]

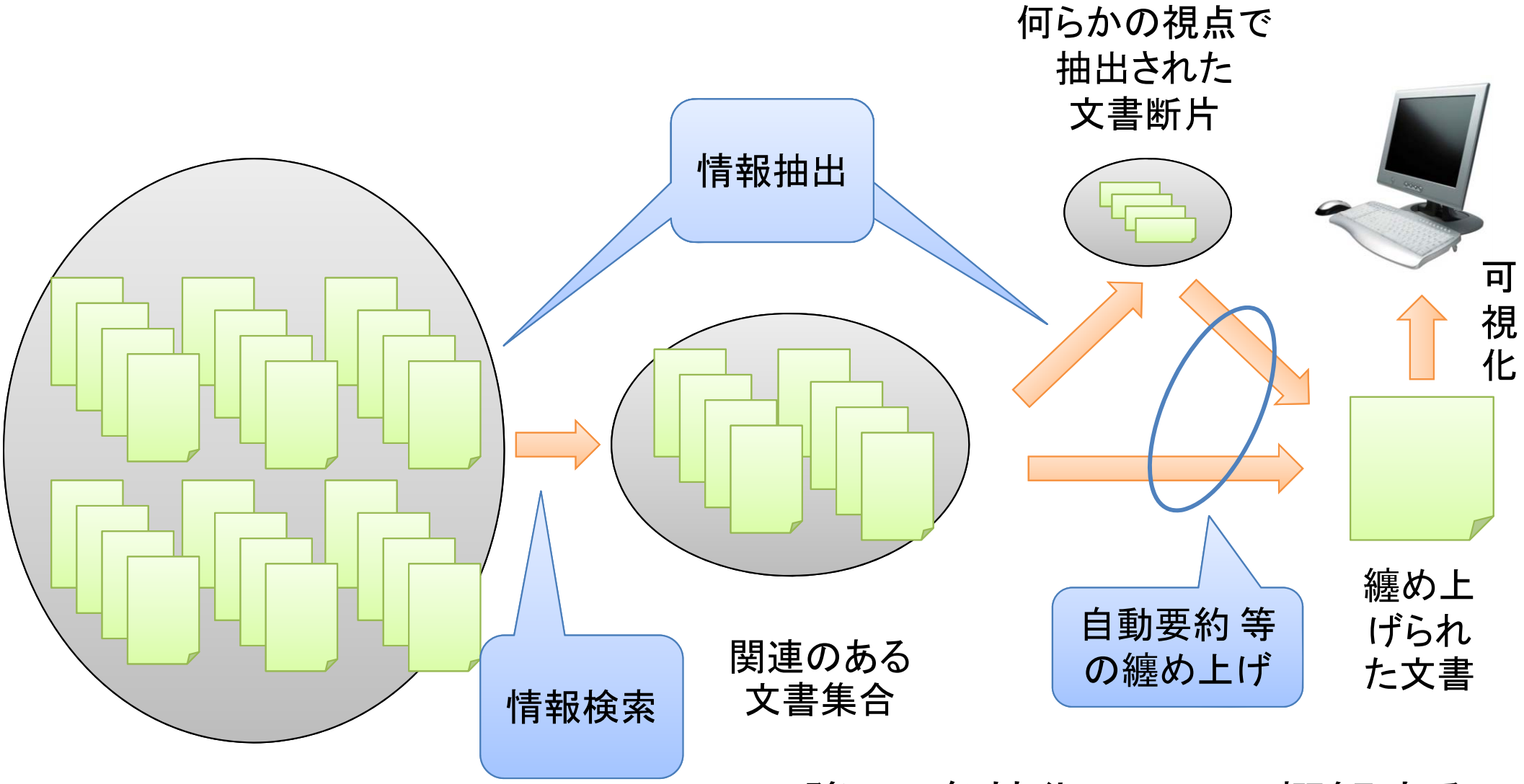
図表 3-5　発信されるコンテンツ情報のファイル数におけるファイル種別の構成比



図表 3-6　発信されるコンテンツ情報のデータ量におけるファイル種別の構成比



想定される処理



以降で、各技術について概観する

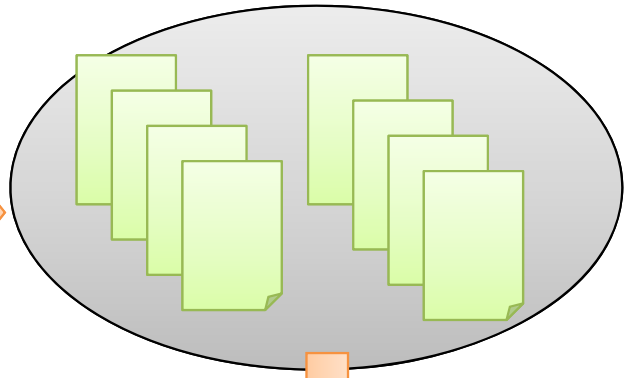
文書検索(Information Retrieval)

- 処理のスキーム
  - 「情報要求」⇒「文書群」
- 情報要求(Information need)
  - 利用者がどのような要求をだせるか
  - 以下の各点により決まる
    - 取り出す文書をどのように定義するか
    - 索引に含める情報の選別
    - クエリ言語の仕様
- 文書集合
  - 手元にある文書
  - 他組織が提供する文書 (e.g. WWW)
- 教科書
  - C. Manning et al. Introduction to Information Retrieval. Cambridge University Press (2008)
    - Web上で本文、関連スライド等がよめる
    - http://nlp.stanford.edu/IR-book/information-retrieval-book.html

文書検索の仕組み

ネットワーク等

クローラ
(Crawler)



検索対象文書集合

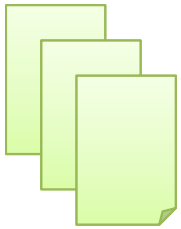
インデクサ
(Indexer)

索引情報
(Index)

クエリ
(Query)

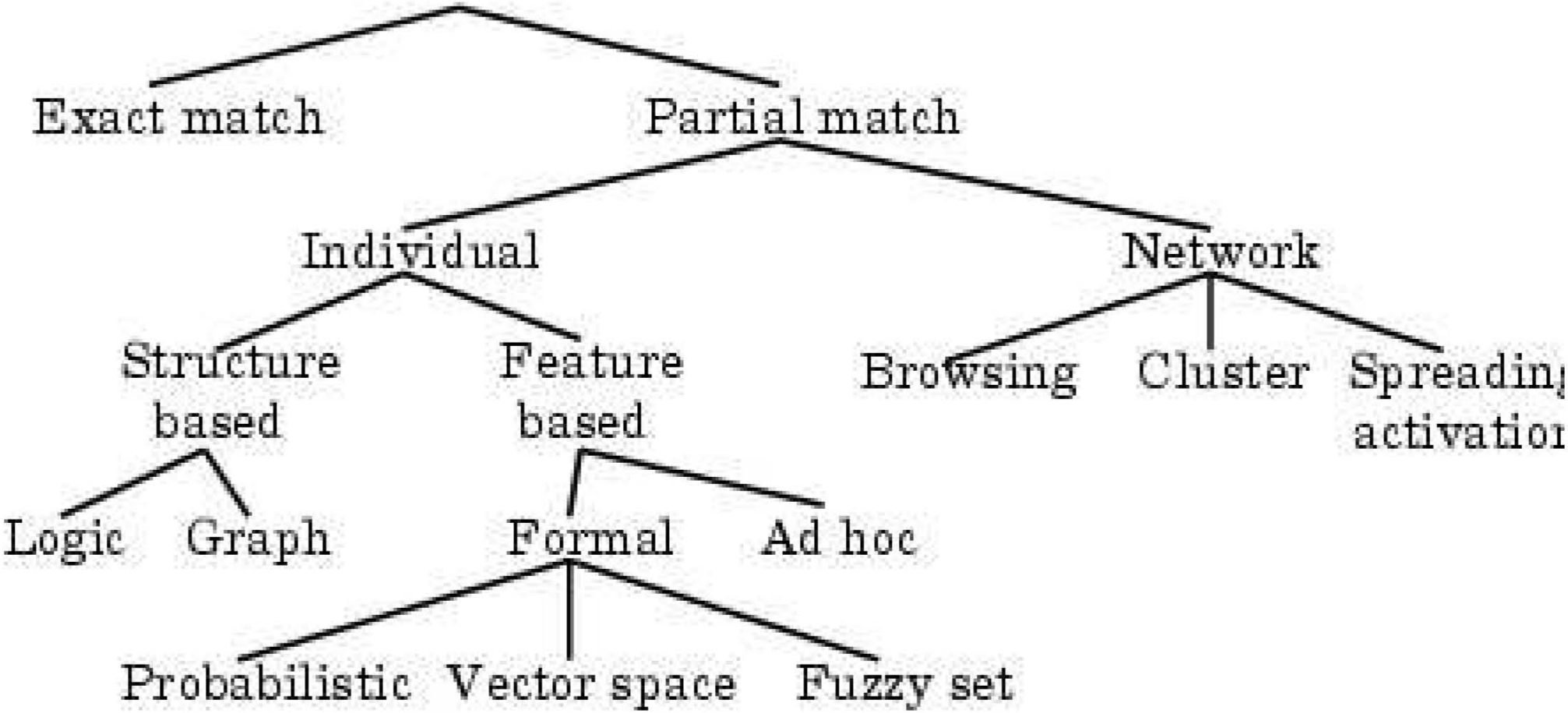
クエリパーザ
(Query parser)

検索エンジン
(Search engine)



検索された文書集合

情報検索の技術の分類


Exact match
Partial match
Individual
Network
Structure based
Feature based
Browsing
Cluster
Spreading activation
Logic
Graph
Formal
Ad hoc
Probabilistic
Vector space
Fuzzy set

代表的な検索技術

- ブーリアンモデル
  - 伝統的なキーワード検索(Exact match)
- ベクトルモデル
  - TFIDF法とベクトル空間法による検索
- 確率モデル
  - 確率計算に基づく検索

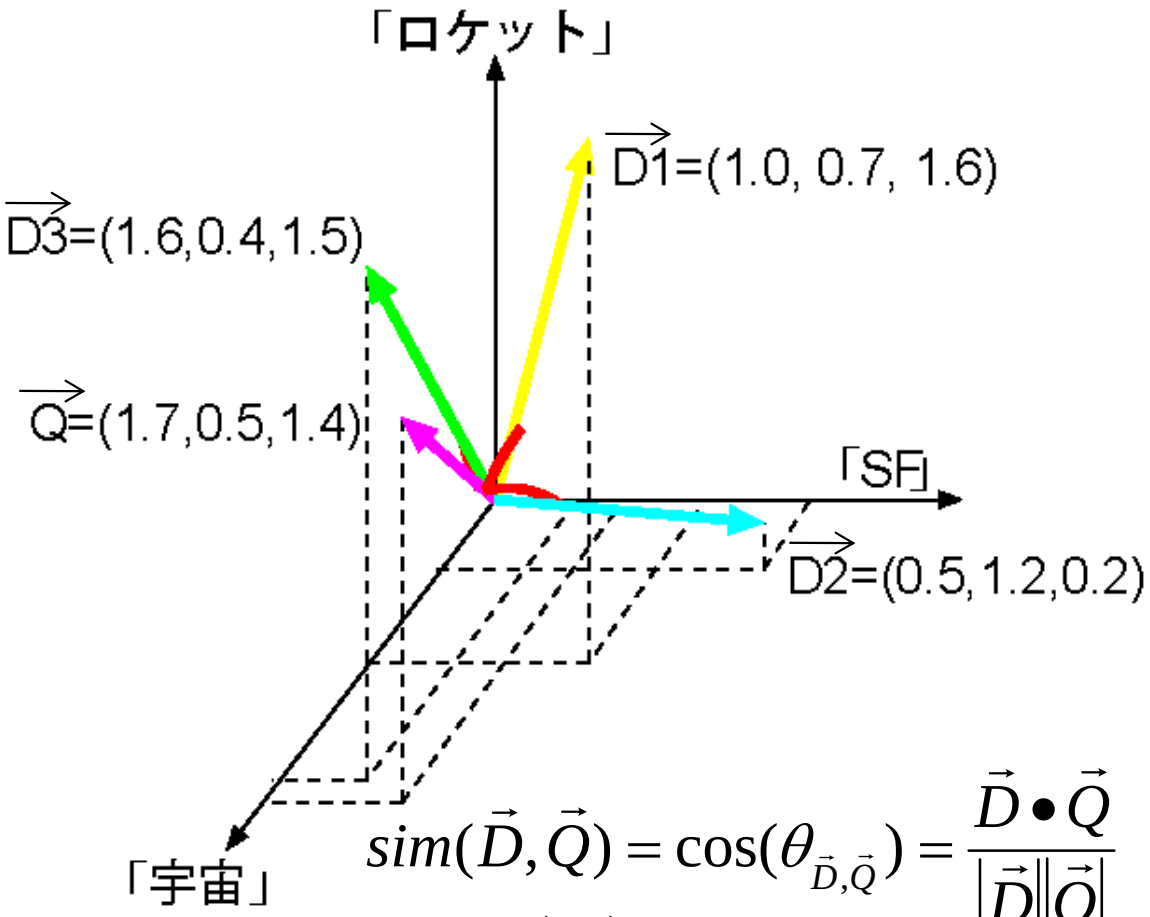
情報検索システム(1)

- いくつかのサブプログラム・ライブラリの複合体
  - (クローラ(Crawler))
  - インデクサ (Indexer)
    - 索引を作る。転置ファイル形式のものが多い。
  - クエリパーザ (Query parser)
    - 利用者が入力する検索要求を解析し、所定の検索式に変換する
  - 検索エンジン (Search engine)
    - 索引と検索式に基づき文書を検索する

情報検索システム(2)

- ベクトル空間法
  - 文書とクエリを同一空間上のベクトルとして表現
    - 特徴量をある次元(の値)とするベクトル
  - ベクトル間の類似度により、クエリと文書の間の関連度を定義し、順位付け
  - 典型的な場合
    - 各単語tを各次元に対応付け、文書Dにおける語tの重み(重要度)$w(t,D)$を各次元の値とするベクトル
    - 類似度は、ベクトルのなす角のcosine値
      - 二つのベクトルが同じ方向を向いている場合は1に近くなり、
      - 異なる方向を向いているときには0に近くなる
      - 類似度にベクトルの長さの効果を反映させない

文書Diの文書ベクトルは
(w(宇宙,Di), w(SF, Di), w(ロケット, Di))



$$sim(\vec{D},\vec{Q}) = \cos(\theta_{\vec{D},\vec{Q}}) = \frac{\vec{D}\bullet\vec{Q}}{\left|\vec{D}\right|\left|\vec{Q}\right|}$$

$$sim(\overrightarrow{D3},\vec{Q}) = 0.99$$

$$sim(\overrightarrow{D1},\vec{Q}) = 0.94$$

$$sim(\overrightarrow{D2},\vec{Q}) = 0.58$$

情報検索システム(3)

- tf-idf法
  - 語の重みを計算する一手法
  - 考慮される値
    - $tf(t,D)$: 語tの文書D内での頻度 (Term frequency)
    - $idf(t)$: $df(t)$の逆数 (Inverse document frequency)
      - $df(t)$: 文書集合における、語tの文書頻度(tが現れる文書数。Document frequency)
  - 文書Dにおける語tの重要度$w(t,D)$
    - ベクトル空間法の各次元(=語)の値。Nは全文書数。

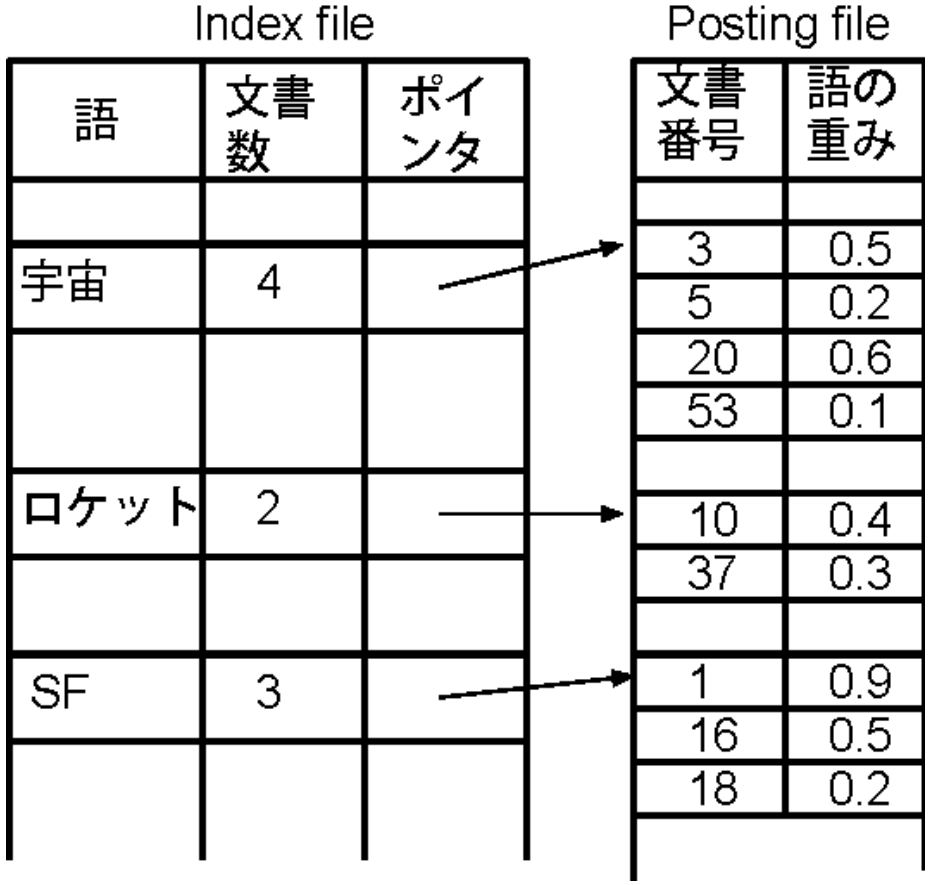
方式1 $$w(t,D) = tf(t,D) * idf(t) = tf(t,D) * \log(\frac{N}{df(t)})$$

方式2 $$w(t,D) = (1+\log(tf(t,D))) * idf(t)$$

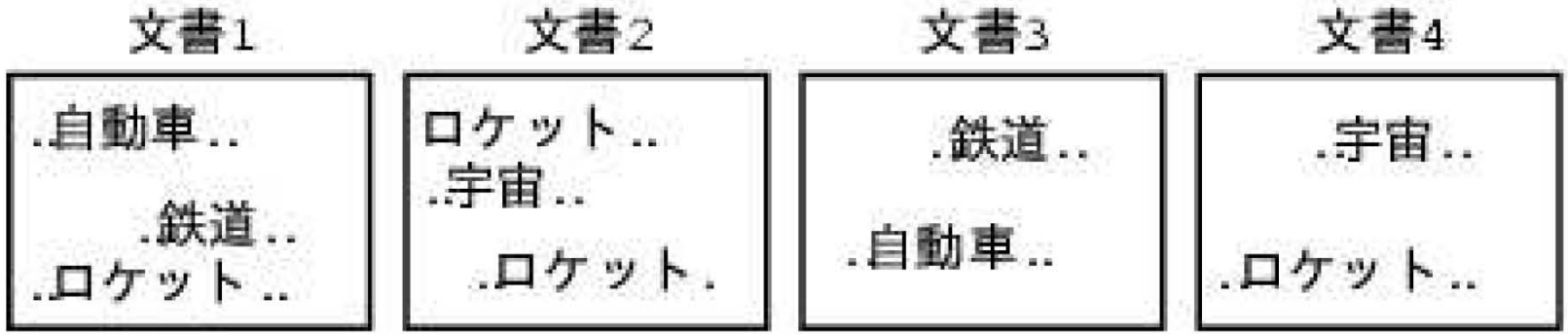
$$= (1+\log(tf(t,D))) * \log(\frac{N}{df(t)}) \qquad etc.$$

情報検索システム(4)

- 転置ファイル(Inverted file)による索引
  - Index file + Posting file



演習問題



ベクトルの各次元　(宇宙,ロケット,自動車,鉄道)

	頻度のベクトル	TFIDF値のベクトル
文書1	(0,1,1,1)	(?,?,?,?)
文書2	(1,2,0,0)	(?,?,?,?)
文書3	(0,0,1,1)	(?,?,?,?)
文書4	(1,1,0,0)	(?,?,?,?)

方法1で

□tfidf法により文書ベクトルを求めよ．類似度の高い文書はどれ?

- ○ただし log2(4/3) = 0.4 とする．
- ○また，簡単のために各ベクトルの長さをすべて1.0と仮定する．
- ○(この仮定により，ある不都合が起きる．なぜか考えよ．)

確率モデル (1)

□仮定
- ○「ある文書dの検索質問qに対する関連度」
- ○= 「ある文書 dj が検索質問qに関連している確率」

□求めるべき確率は,

$$P(R = 1 | D = d_j)$$

- ○確率変数R: 値は1もしくは0で，それぞれ，「qに関連している」「していない」
- ○確率変数D: 値は文書

確率モデル (2)

□文書の関連度の定式化

○確率P(R=1|D=dj)の順位を保存する関数LOR() (対数オッズ比, Log-odds ratio)とそれから導出される関連度関数sim()

▸ただし，○は "natural zero" (ゼロベクトル等)

$$
\begin{aligned}
LOR(d_j, q) &= \log \frac{P(R=1|D=d_j)}{P(R=0|D=d_j)} \\
&= \log \frac{P(D=d_j|R=1)P(R=1)}{P(D=d_j|R=0)P(R=0)} \\
&= \log \frac{P(D=d_j|R=1)}{P(D=d_j|R=0)} + \log \frac{P(R=1)}{P(R=0)} \\
sim(d_j, q) &= LOR(d_j, q) - LOR(O, q) \\
&= \log \frac{P(D=d_j|R=1)P(D=O|R=0)}{P(D=d_j|R=0)P(D=O|R=1)}
\end{aligned}
$$

確率モデル (3)

□文書の関連度から単語の重みへ

○文書djにおいて構成素(例えば単語i)間の独立性を仮定

$$
\begin{aligned}
sim(d_j, q) &= \log \frac{\prod_i P(T_i = t_i|R = 1) \prod_i P(T_i = 0|R = 0)}{\prod_i P(T_i = t_i|R = 0) \prod_i P(T_i = 0|R = 1)} \\
&= \sum_i \log \frac{P(T_i = t_i|R = 1)P(T_i = 0|R = 0)}{P(T_i = t_i|R = 0)P(T_i = 0|R = 1)} \\
&= \sum_i W(T_i = t_i)
\end{aligned}
$$

$$
W(T_i = t_i) = \log \frac{P(T_i = t_i|R = 1)P(T_i = 0|R = 0)}{P(T_i = t_i|R = 0)P(T_i = 0|R = 1)}
$$

○確率変数Ti : 第i番目の単語(単語i)に対する(ある文書における)統計量(例えば単語頻度など)

○W(Ti=ti) : 単語iについて，その統計量がtiであったときの単語の重要度

□確率P(Ti=ti|R=1)などをどのように推定するかが本質的な問題

確率モデル (4)

出現頻度を考慮しない場合

□次式となる

$$W(T_i = t_i) \quad = \quad \log \frac{p(1-q)}{q(1-p)}$$

○p: P(単語iが存在|R=1)

○q: P(単語iが存在|R=0)

□適当な確率推定により

$$W(T_i = t_i) \quad \sim \quad w^{(1)} = \log \frac{(r+0.5)/(R-r+0.5)}{(n-r+0.5)/(N-n-R+r+0.5)}$$

○N:全文書数， n:単語iが含まれる文書数，R:質問qに対して関連していることが既知である文書数，r:単語iが含まれ，かつ，質問qに関連していることが既知である文書数

○Robertson/Sparck Jones weight と呼ばれる

確率モデル (5)

出現頻度tf(i)を考慮する場合(1)

- 準備: ポワソン分布 (Poisson distribution)
  - 生起頻度Nが低い事象について，単位時間内に平均でラムダ回発生する事象がちょうどk回(k=0,1,2,...)発生する確率

$$p(N = k, \lambda) = \frac{e^{-\lambda}\lambda^k}{k!}$$

- 2つのポワソン分布(2-Poisson)に基づくモデル
  - 各語はある「エリート(elite)文書集合」に関連づけられている
    - エリート文書集合においては，その単語の文書内頻度tf(i)はポワソン分布に従う
    - 残りの「非エリートの」文書集合についても，ポワソン分布に従う．

$$W_i = \log \frac{(p'\lambda^{tf_i}e^{-\lambda} + (1-p')\mu^{tf_i}e^{-\mu})(q'e^{-\lambda} + (1-q')e^{-\mu})}{(q'\lambda^{tf_i}e^{-\lambda} + (1-q')\mu^{tf_i}e^{-\mu})(p'e^{-\lambda} + (1-p')e^{-\mu})}$$

  - p': P(T_iに対するエリート文書集合|R=1)
  - q': P(T_iに対するエリート文書集合|R=0)
- 問題点:パラメタが4つもある．「エリート性」は隠れ変数

確率モデル (6)

出現頻度tf(i)を考慮する場合(2)

- 大幅な近似: 先の式の外形を模擬する．つまり，以下の特徴を持つ式で近似.
  - a) tf(i)が0の時，値は0
  - b) tf(i)について単調増加するが，c) ある最大値に漸近する
  - d) Robertson/Sparck Jones weightに近い
- BM25 (BM = Best Match)
  - Okapiシステムで採用されている

$$W_i = w^{(1)} \cdot \frac{(k_1+1)tf_i}{K+tf_i} \cdot \frac{(k_3+1)qtf_i}{k_3+qtf_i}$$

$$K = k_1((1-b)+b\frac{dl}{avdl})$$

  - qtf(i): q内の単語iの頻度，dl:文書長，avdl:平均文書長
  - k1,b,k3: パラメタ．Okapiシステムでは，k1=1.2, b=0.75,k3=7もしくは1000

情報検索システムの例

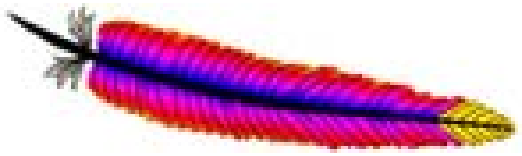
- 無償で利用できる代表的な情報検索システム (手元にインデクスを置くもの)
  - Lucene
  - Indri
- 無償で利用できるWWW用検索エンジンAPI　(Application Program Interface)
  - Yahoo! デベロッパーネットワーク
    - http://developer.yahoo.co.jp/
    - REST形式のAPI
      - REST(Representational State Transfer)
      - HTTPによるやり取り。アプリケーション側からみると、HTTPリクエストで要求し、XML形式の結果をもらう。
  - Google Custom Search API
    - https://developers.google.com/custom-search/v1/overview
    - REST形式のAPI
  - 検索エンジン基盤TSUBAKIのAPI
    - http://tsubaki.ixnlp.nii.ac.jp/
    - 自然言語文を検索質問として受け付けることができる。
    - 「標準フォーマット」とよばれる、形態素解析、係り受け解析済みのテキストを受け取れる。
    - REST形式のAPI

Lucene

- るしーん

開発者Douglas Cuttingの奥さんのミドルネームで、
奥さんの母方のおばあちゃんのファーストネーム
http://wiki.apache.org/lucene-java/LuceneFAQ

- http://lucene.apache.org/
- Apacheによる検索エンジン開発プロジェクトの成果
- ベクトル空間法に基づく検索手法
  - Space Optimizations for Total Ranking [Cutting 97]
    - http://lucene.sf.net/papers/riao97.ps



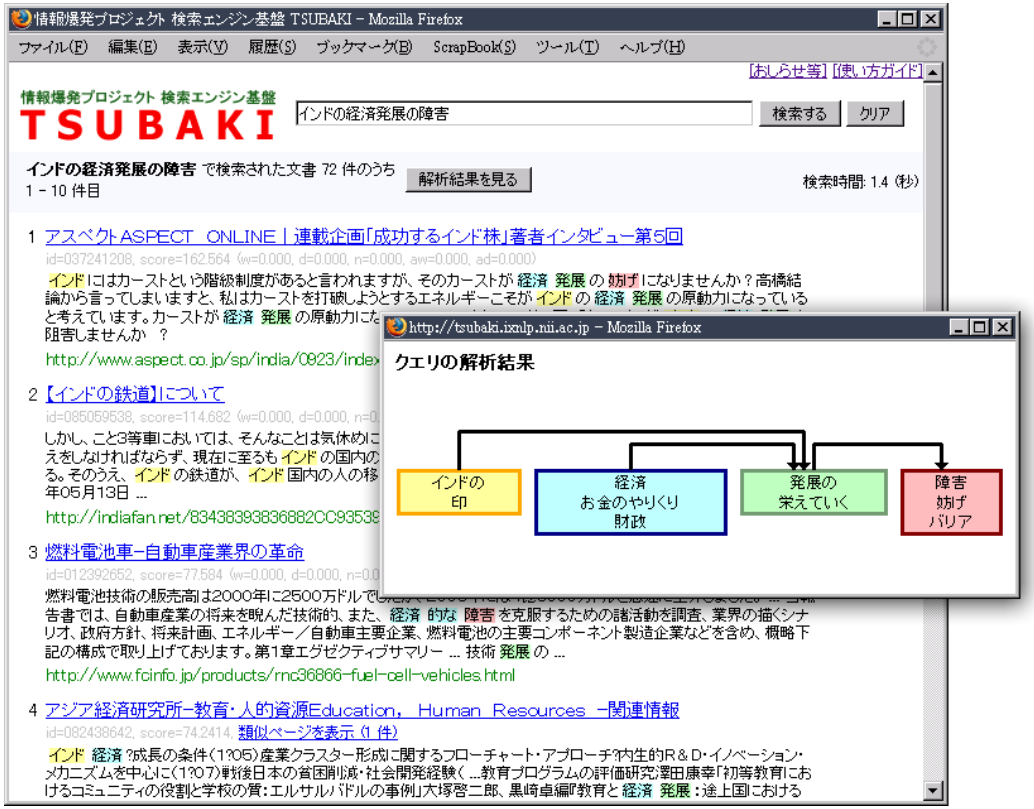
Indri

- いんどり
- http://www.lemurproject.org/indri/
- マサチューセッツ大とCMUが開発
- Lemur (れむーる)プロジェクトの検索エンジン
- ベイジアン推論ネットワークと言語モデルに基づく検索手法

Indri lemur
インドリキツネザル

検索エンジン基盤TSUBAKI

- つばき
- http://tsubaki.ixnlp.nii.ac.jp/
- 京都大学が科研「情報爆発」プロジェクトの一環として開発
- 自然言語文を検索質問として受け付けることができる。
- 通常の索引情報に加えて以下の言語的な情報が利用される。
  - 係り受け関係
  - 同義関係



http://tsubaki.ixnlp.nii.ac.jp/tutorial.html より引用

Application Program Interfaceの例
検索エンジン基盤TSUBAKIの場合

- TSUBAKI API: 典型的なREST形式のAPI
- リクエストURLでHTTPを用いて検索要求を行う。
  - http://tsubaki.ixnlp.nii.ac.jp/api.cgi?query=*検索クエリ*&*パラメタ*=*値*&*パラメタ*=*値*...
  - *検索クエリ*: utf-8で記述された検索クエリをURLエンコードしたもの
  - *パラメタ*と*値*: 次ページの表
- 結果はHTTPレスポンスとして、XML形式で得られる。
  - 例は、次々ページ

TSUBAKI APIのパラメタの例

パラメタ	値	説明
start	*integer*	取得したい検索結果の先頭位置.
results	*integer*	取得したい検索結果の数. デフォルトは10.
logical_operator	AND/OR	検索時の論理条件. デフォルトはAND.
only_hitcount	0/1	ヒット件数だけを得たい場合は1, 検索結果を得たい場合は0. デフォルトは0.
force_dpnd	0/1	クエリ中の係り受け関係を全て含む文書を得たい場合は1, そうでない場合は0. デフォルトは0.
snippets	0/1	スニペッツが必要な場合は1, スニペッツが不要な場合は0. デフォルトは0.
near	*integer*	クエリ中の単語と単語が n 語以内に出現するという条件のもと検索を実行する(近接検索). クエリ中の単語の出現順序は考慮される.
format	html/xml	オリジナルのウェブ文書, または標準フォーマット形式のウェブ文書のどちらを取得するかを指定. idを指定した際は必須.

TSUBAKI APIの出力例

```
<ResultSet time="2007-02-21 13:43:58" query="京都の観光名所" totalResultsAvailable="19586" totalResults
  <Result Id="24411919" Score="68.67424">
    <Title>関西探索</Title>
    <Url>http://www.kansaitansaku.com/</Url>
    <Snippet>
      このホームページは、京都を中心とした観光名所におとずれ、その感想を述べていくホームページです
    </Snippet>
    <Cache>
      <Url>
        http://tsubaki.ixnlp.nii.ac.jp/index.cgi?URL=INDEX_NTCIR2/24/h2441/24411919.html&KEYS=%B5%FE%
      </Url>
      <Size>619</Size>
    </Cache>
  </Result>
  <Result Id="06832429" Score="64.16455">
    <Title>京都観光タクシー　京都の名園1</Title>

...中略...

  </Result>
</ResultSet>
```

TSUBAKI APIでキャッシュを利用する

- リクエストURL
  - http://tsubaki.ixnlp.nii.ac.jp/api.cgi?format=*フォーマット名*&id=*文書ID*
- パラメタ
  - format
    - html : クロールされた元のHTML文書の形式
    - xml : 標準フォーマット(形態素解析、係り受け解析済み)
  - id
    - TSUBAKI APIが返す検索結果のうちの、Result要素のID属性の値
- 結果はHTTPレスポンスとして、HTML形式もしくはXML形式で得られる。

情報検索と情報編纂

- キーワード等で検索するだけで文書を絞り込むことができる場合
  - 検索システムが提供する索引作成手法で十分

- 特別な情報で文書を絞りこみたい場合
  - 情報編纂のために検索を使用する場合
  - 何らかの情報抽出結果で検索したい
    - 例: 日時、場所、特定の統計量、etc.
  - フィールドの利用
    - 文書=複数のフィールドの複合体
  - 文書の内容を保存するフィールド以外に、抽出結果を保存するフィールドを作成
  - 検索時には、クエリにフィールド名の指定を追加する。
  - ファセットサーチ(後述)の実現

フィールド=フィールド名＋文字列

情報検索の主要な技術

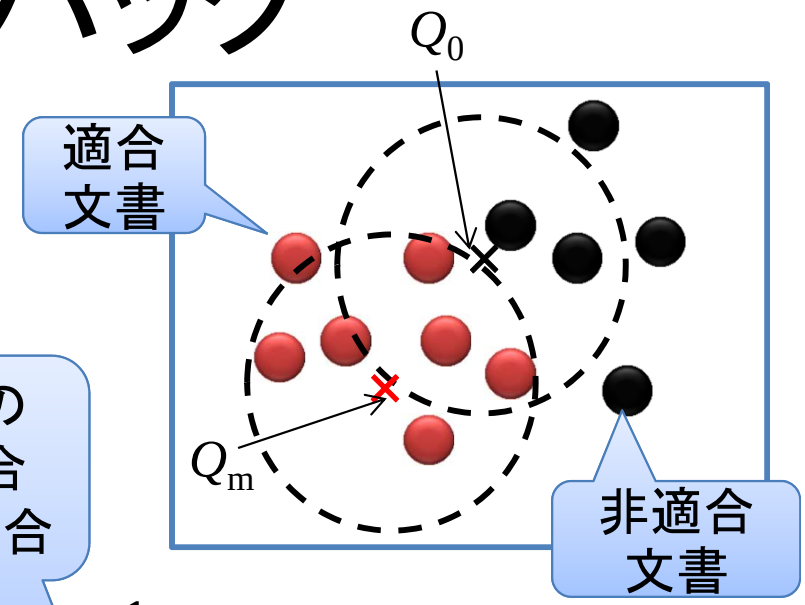
- 関連性フィードバック(適合性フィードバック) (Relevance feedback)
  - 疑似関連性フィードバック(Pseudo relevance feedback)
- 質問拡張 (Query expansion)

情報編纂と関連して

- ファセットサーチ (Faceted search)
  - 探索的検索(Exploratory search)の一種
    - 閲覧(Browse)と検索(Search)

関連性フィードバック

- Rocchioの手法



更新された
クエリ
ベクトル

元の
クエリ
ベクトル

既知の
適合
文書集合

既知の
非適合
文書集合

$$\overrightarrow{Q_m} = \alpha \overrightarrow{Q_0} + \beta \frac{1}{|R|} \sum_{\overrightarrow{D_r} \in R} \overrightarrow{D_r} - \gamma \frac{1}{|N|} \sum_{\overrightarrow{D_n} \in N} \overrightarrow{D_n}$$

- 適合文書、非適合文書がいくつか既知である必要
- 疑似関連性フィードバック
  - 検索結果の上位数件を適合文書だと仮定してフィードバック手法を適用

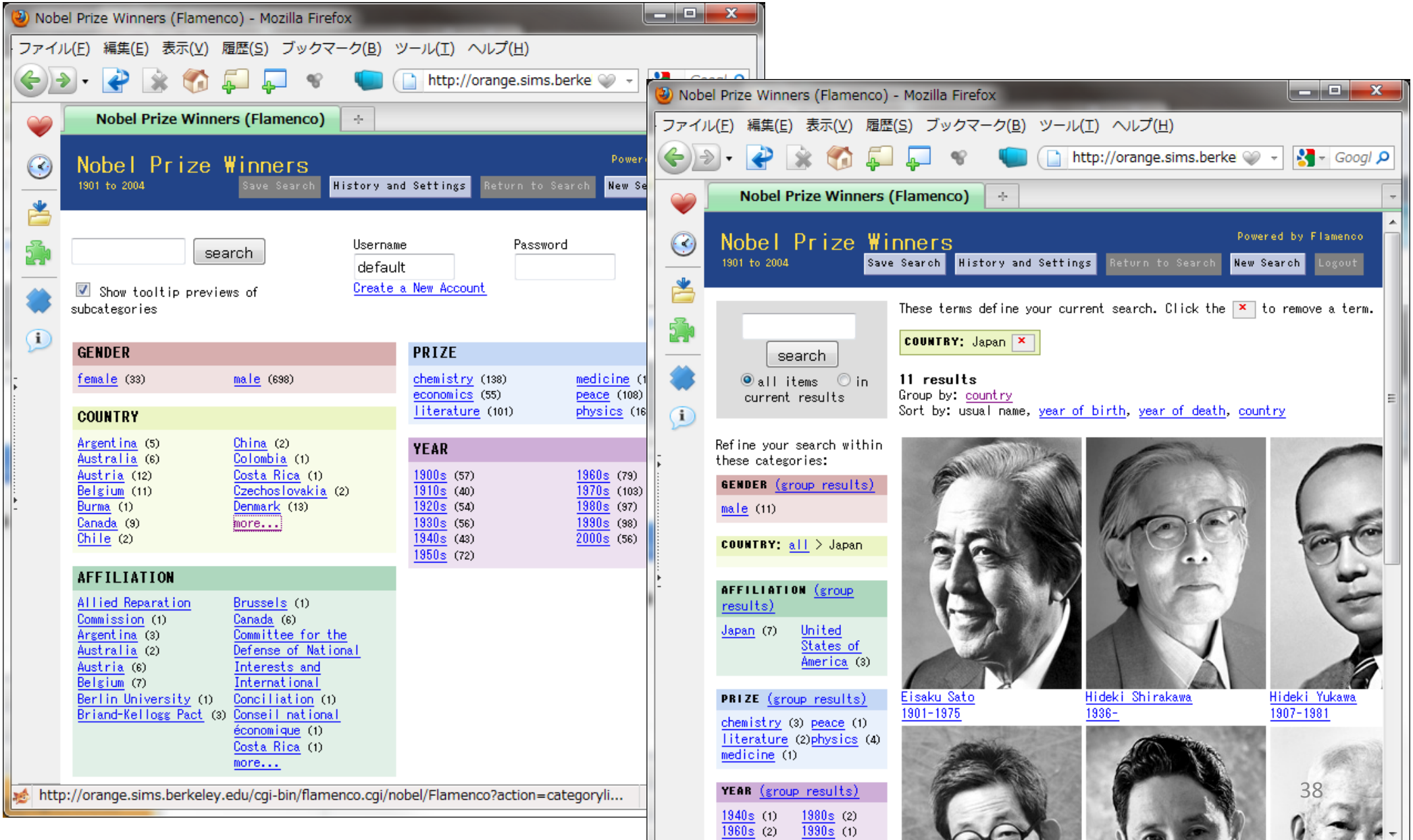
質問拡張

- 比較
  - 関連性フィードバック:「文書」に関する追加情報を与え、クエリを改訂
  - 質問拡張:「語」や「句」に関する追加情報を与え、クエリを改訂
- 例
  - Q=本 ⇒ Q=本 or ほん or 書籍 or ブック
  - Web検索エンジンの「示唆」機能
    - Googleサジェスト
- 方法
  - 人手作成のシソーラス
  - 大域的解析(静的、文書集合全体)
    - 自動抽出したシソーラス: 統計的共起情報
    - クエリログマイニング
  - 局所的解析(動的)
    - 検索結果文書集合の解析

ファセットサーチ

- ファセット:「側面」
  - 文書が複数の「側面」を持つと考え、それぞれの側面で絞り込みを行う。
  - それぞれの側面は、独立した体系をもつ。
- 情報検索に情報抽出結果を融合する一つの手法(森の私見)
  - 抽出された情報に基づく探索的検索を実現
- Flamenco Search: ファセットサーチの例
  - UC Berkelyのプロジェクト
  - http://flamenco.berkeley.edu/

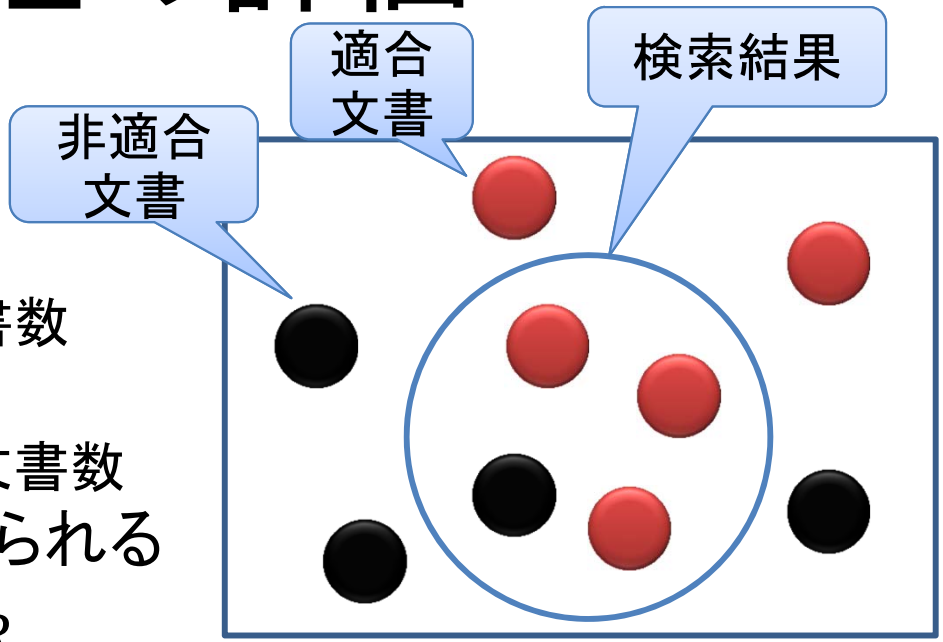
Flamenco searchのデモ「ノーベル賞受賞者」の例



情報検索過程の評価

- 順位が無い検索結果の評価
  - 適合率(Precision, P)
    - 検索された適合文書数/検索文書数
  - 再現率(Recall, R)
    - 検索された適合文書数/全適合文書数
  - F値(F-measure): $\beta=1$がよく用いられる

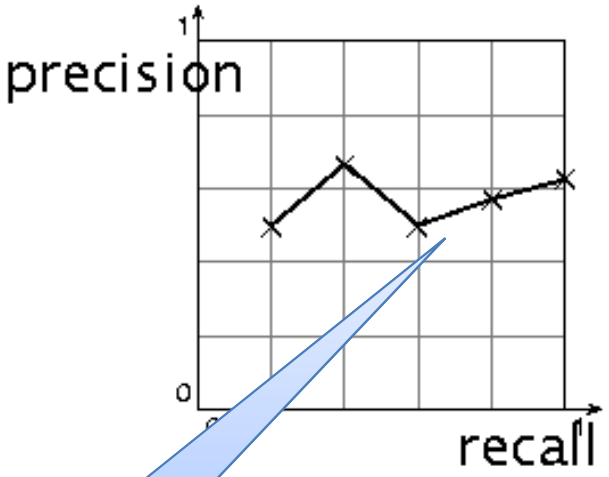
$$F=\frac{1}{\alpha\frac{1}{P}+(1-\alpha)\frac{1}{R}}=\frac{(\beta^2+1)PR}{\beta^2P+R}$$

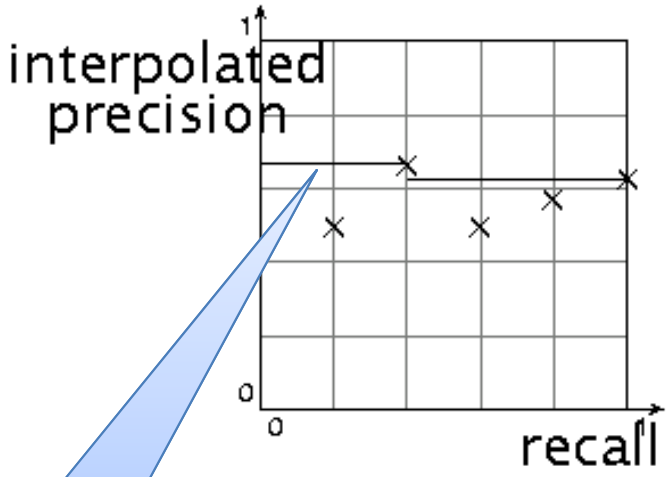


P=3/4, R=3/5

- 順位のある検索結果の評価
  - 平均適合率(Mean average precision: MAP)
    - 11点平均適合率
      - 再現率が0.0～1.0の間で0.1刻みに適合率を求め平均
      - 補間適合率(interpolated precision)を用いる
  - MRR (Mean reciprocal rank)
    - 最上位の適合文書の順位の逆数を得点とし、その得点の平均値

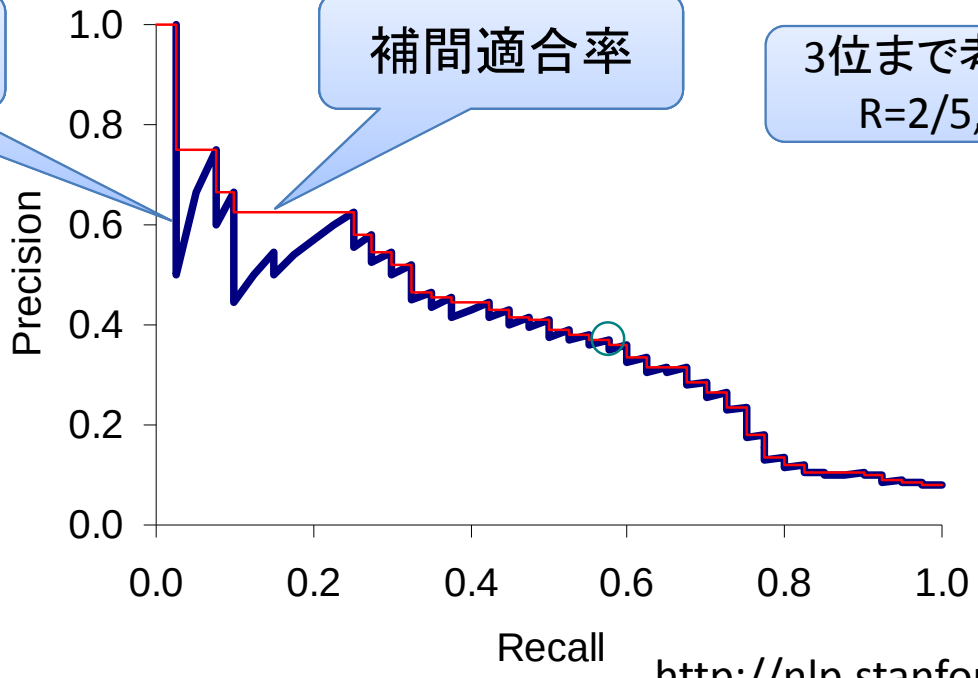
再現率-適合率曲線と補間適合率





観測された適合率

補間適合率



3位まで考慮すると
R=2/5，P=2/3

4位まで考慮すると
R=3/5，P=3/4

順位	文書	適合
1	D3	○
2	D10	×
3	D2	○
4	D5	○
5	D6	×
6	D7	×
:		
総適合文書数	5	

http://nlp.stanford.edu/IR-book/newslides.html より引用