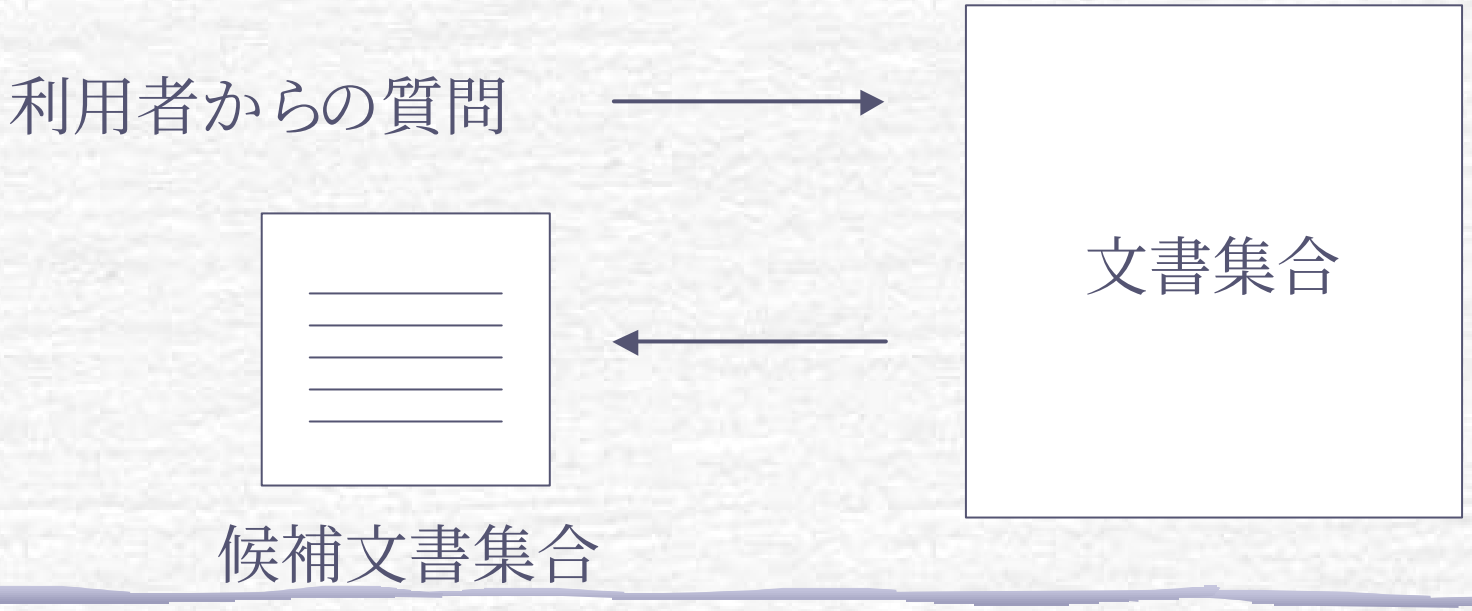
2000年11月30日

情報検索と言語処理

奈良先端科学技術大学院大学
情報科学研究科
松本裕治

情報検索

- 目的 :利用者が手に入れたい情報を含む文書を、大量の文書群の中から探し出す技術



質問に対する前処理 (1)

- 質問文の形式
  - 単語の集合
    - そのまま検索語として用いる
    - 複合語をより短い単語に分割する
    - 同義語あるいは類義語を検索語として含める (質問拡張)
  - 自然言語文による質問
    - 分かち書き (形態素解析) により単語を抽出
    - 質問のタイプ (ほしい情報の種類) と検索語の抽出
    - 同義語/類義語による拡張
  - 単語を用いた論理式
    - 単語の集合に AND, OR などを用いて論理式表現する

質問に対する前処理 (2)

- 質問文の拡張 (query expansion)
  - 単語の拡張
    - シソーラスに基づいて検索語に同義語を追加
    - 例：「会社」という検索語に対して、「企業」「商会」などを追加する
    - 曖昧性の解消を考慮することが重要
      - 「現金」に対して「金」を同義語として追加するならば、それは、「お金」としての「金」であるべきで、貴金属としての「金」であっては困る
  - 同義語による拡張についての2つの考え方
    - 質問/文書にある語の中で同じ意味の語を統一した同義語でおきかえる
    - 質問中の語を同義語によって拡張する

文書集合の取り扱い方

- 文書中で検索に用いる部分
  - 一部（例えばタイトルのみ）/ すべて
- 文書中で用いる情報
  - すべての文字列（すべて/ある固定長）
  - すべての単語
    - 事前に形態素解析を行う必要がある
  - 限定された単語/文字列
    - 索引語と呼ばれる
    - どのようにして索引語を決定するかが問題

文書集合に対する前処理

- 索引付けの対象の認定
  - 書誌情報（文書の著者・タイトル・出版年など）
  - 文書全体
- 索引語の認定
  - 索引語の選択
    - 不要語の削除
    - 重要語の認定、重要語がもつべき性質
- 索引語抽出のための言語処理
  - 単語への分かち書き
  - 形態素解析

日本語の形態素解析

形態素解析の機能

- 単語の分かち書き
- 活用語の語尾処理
- 品詞の同定

形態素解析システム「茶筌」の出力

茶筌は日本語を形態素解析する。

茶筌	チャセン	名詞-一般	
は	ハ	助詞-係助詞	
日本語	ニホンゴ	名詞-一般	
を	ヲ	助詞-格助詞-一般	
形態素	ケイタイソ	名詞-一般	
解析	カイセキ	名詞-サ変接続	
する	スル	動詞-自立 サ変・スル	基本形
。	。	記号-句点	

EOS

英語の解析例

単語の認定 •原形への復元 •品詞の同定

While John was in U.S.A., he often went to New York.

While	While	IN	
John	John	NNP	
was	be	VBD	
in	in	IN	
U.S.A.	U.S.A.	NNP	
,	,	,	
he	he	PRP	
often	often	RB	
went	go	VBD	
to	to	TO	
New York		New York	NNP
.	.	.	

文書の解析と単語集合の抽出

- 結核予防「BCG」でエイズワクチン開発　国立予防研など、サルで実験開始へ

結核予防ワクチンであるＢＣＧに、日本人とタイ人に特徴的なエイズ・ウイルス（ＨＩＶ）の遺伝子の一部を組み込んだエイズワクチンを、国立予防衛生研究所と味の素中央研究所のグループが開発、マウス実験などで免疫力を高める効果を確認した。近く国内で初めて、サルを使った感染予防実験を開始する。アジアを中心に広く途上国で使える可能性がある。

114 :全形態素数	3 だ	2 する	1 途上
7 を	2 国立	2 が	1 的
5 予防	2 結核	2 ある	1 中心
5 で	2 開発	2 BCG	1 中央
3 実験	2 開始	1 力	1 組み込む
3 ワクチン	2 サル	1 免疫	1 性
3 エイズ	2 など	1 味の素	1 人
3 の	2 と	1 日本人	1 初めて
3 に	2 た	1 特徴	1 使える

索引語の選択

- 不要語：一般に助詞、助動詞、あまりに一般的すぎる名詞や動詞、英語の前置詞や冠詞など
- このような単語の集合を事前に決めておき、一覧として集めたものを不要語リスト(stop word list)という。
  - このように事前に決められた不要語は索引語の候補に入れない
- 英語の不要語リストの例(SMART system(Salton & McGill 1983)):

  a, able, about, above, according, across, actually, after, afterwards,

  … b, be, because, became, become, becoming, been , before,

  … cannot, can’t, cause, causes, certain, certainly, changes,

索引語の選択基準

- ある程度以上の出現頻度がある
- あまりにも多く出現しすぎない
- 文書の集合に対して均質すぎる現れ方をしない

検索語の重要度を決める2つの尺度

1. 索引語の頻度 (term frequency)
2. 索引語の分布の偏り(inverse document frequency)

1. 索引語の頻度 (term frequency)

- 文書に現れる頻度

$$w_t^d = tf(t,d)$$

文書 d に 語 t が出現する頻度

- 文書の長さに依存しないように頻度を正規化

$$w_t^d = \frac{tf(t,d)}{\sum_{s \in d} tf(s,d)}$$

語の出現を単純な頻度ではなく、
文書中の出現の比率として」評価

2. 索引語の分布の偏り

- IDF (inverse document frequency)

  たとえ高頻度の語であっても、どの文書にも現れるような語は文書を特徴付ける語にはならない

  ➡ 文書集合の中で偏って現れる語の方が、文書を特徴付ける

$$idf(t) = \log \frac{N}{df(t)} + 1$$

N :全文書の数

$df(t)$:単語 t が出現する文書の数

語の重要度の計算例

文書が20万の記事からなるとする（N=200000)

単語 a が200の記事に出現

単語 b が50000の記事に出現するとき、

$$idf(a) = \log \frac{200000}{200} + 1 = \log 1000 + 1 \approx 11$$

$$idf(b) = \log \frac{200000}{50000} + 1 = \log 4 + 1 \approx 3$$

文書d中の語tの重要度は、*tf* と *idf* の積で評価することが多い

$$tf(t,d) \cdot idf(t)$$

索引語の候補抽出のための処理

- 単語に対する処理
  - 活用語尾の処理 (stemming)
    - documents ⟶ document
    - Documents ⟶ document
    - studies ⟶ study
    - writing ⟶ write
    - × king ⟶ k
    - ？ dumping ⟶ dump
    - 行く、行った、行かない、行けば ⟶ 行く
    - する、した、すれば、せよ ⟶ する

索引語の候補抽出処理 (2)

- 用語の認定 :特に複合語の処理
  - ばらばらの単語の集合をもとに検索するのではなく、意味のある複合語は一つにまとめて検索すべき
  - 複合語の区切り方の例
    - シドニーオリンピック大会
      → シドニー / オリンピック / 大会
      → シドニー / オリンピック大会
      → シドニーオリンピック / 大会
      → シドニーオリンピック大会
      → シドニー / オリンピック / シドニーオリンピック

文書検索のためのインデックス付け

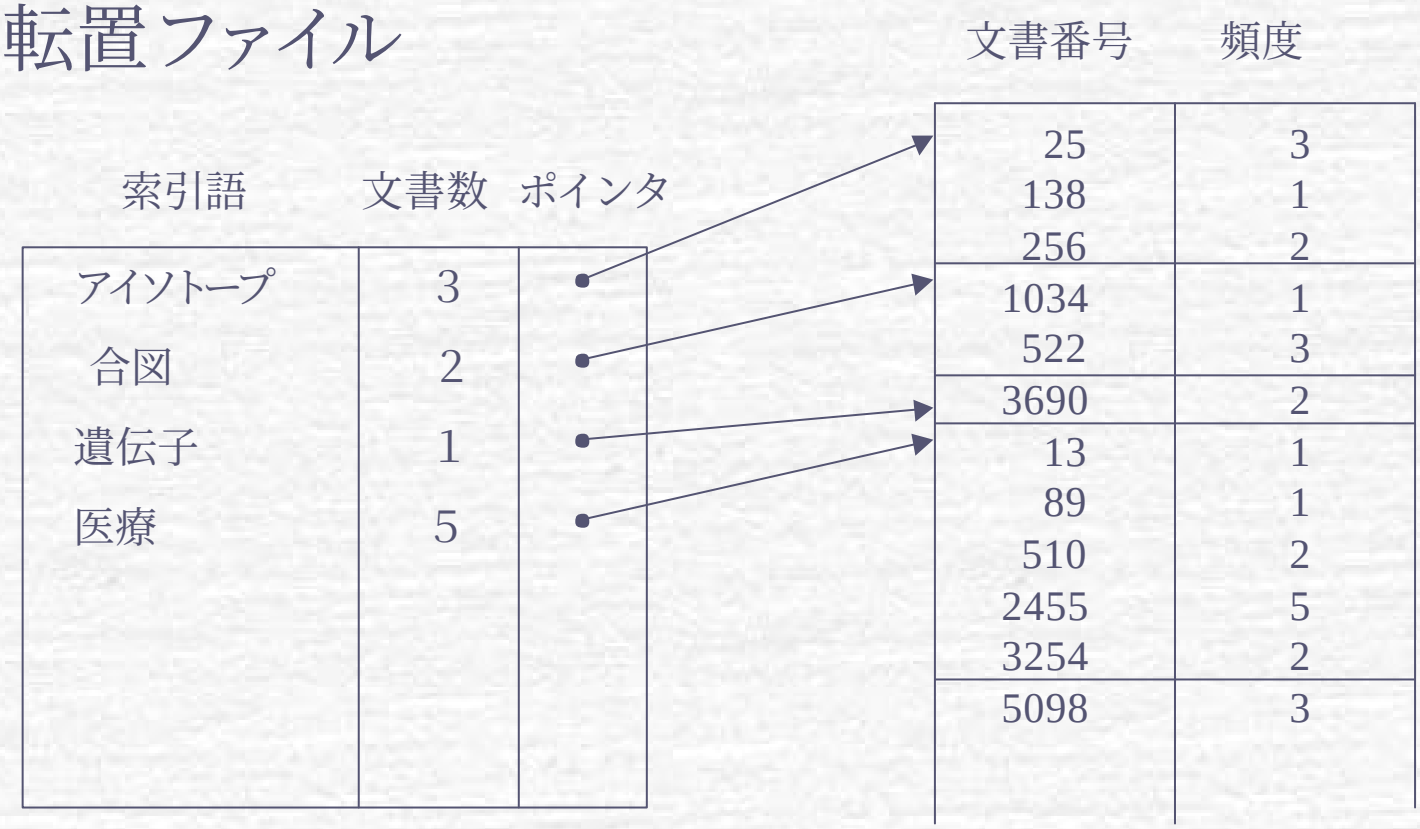
- 文書中に現れる索引語を高速に検索する手法
  - 通常、大量の文書を検索対象にするので、検索は高速でなければならない
  - ある特定の索引語が現れるすべての文書を一括して検索できることが好ましい
  - 索引語が現れている場所を事前に何らかの方法で求めておき、その情報を記録することをインデックス付け(indexing)という
- インデックス付けのための代表的な手法
  - 転置ファイル
  - Suffix Array

索引語へのインデックス付けの方法

- 転置ファイル (inverted file)
  - 個々の索引語に対して、それが出現する文書の数と、それぞれの文書番号の一覧を集めておく
  - 索引語は、ハッシュ法や 2分探索などを用いて高速に検索が可能
- Suffix Array
  - 文書中の検索語のすべての位置に対してポインタをおく
  - それぞれのポインタが、自分が指している文書の場所から始まる文字列を現すとみなして、ポインタを辞書順にソートする
  - 検索したい文字列に対して、ソートされたポインタを 2分探索すれば、検索文字列を接頭辞とするすべての出現位置を見つけることができる

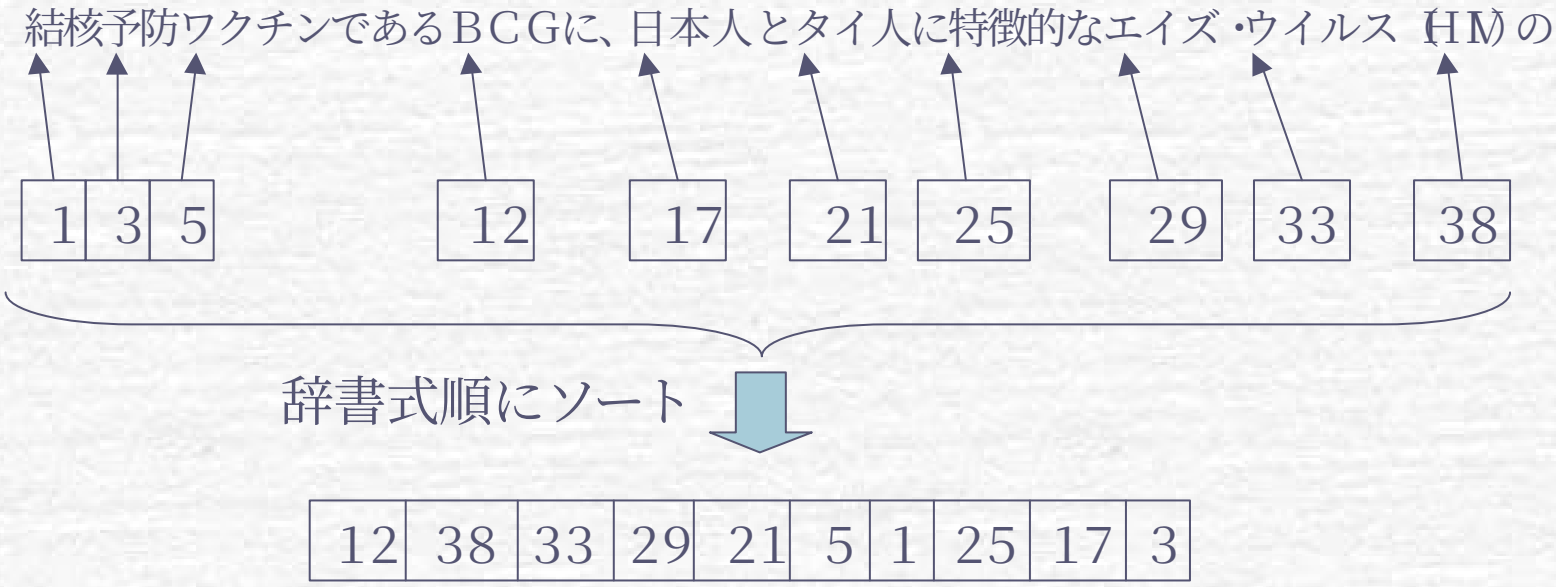
文書検索のためのインデックス付け(1)

- 転置ファイル



文書検索のためのインデックス付け(2)

- Suffix Array



検索語に対して、このポインタ列を2分探索する

様々な検索モデル

- ブーリアンモデル
  - 索引語の論理式（and, or など）によって質問文を表現
- ベクトル空間モデル
  - 質問文、文書を索引語のベクトルとして表現し、ベクトルの類似性によって、文書を順序付ける
- 確率モデル
- ファジィモデル
- ネットワークモデル

その他

- 関連性フィードバック

ベクトル空間モデル

- 文書とそれに出現する単語の対応行列を考える

$$
\begin{array}{c} \\ t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \end{array}
\begin{array}{c}
\begin{array}{ccccc} d_1 & d_2 & d_3 & d_4 & d_5 \end{array} \\
\begin{bmatrix} 0 & 2 & 3 & 0 & 1 \\ 1 & 1 & 0 & 3 & 2 \\ 0 & 2 & 1 & 2 & 0 \\ 0 & 2 & 4 & 0 & 3 \\ 2 & 0 & 1 & 0 & 2 \\ 1 & 0 & 2 & 1 & 1 \end{bmatrix}
\end{array}
$$

索引語→　　←文書

  - 各列は、索引語が現れる文書と出現数を表す（転置ファイルと同じ情報）
  - 列ベクトルは、それぞれの文書に現れる索引語の出現数よりなるベクトル

ベクトル空間モデル (2)

- ベクトル空間モデルでは、文書をそれに出現する単語のベクトル考える
- 文書の近さ（類似度）をベクトルの類似度で表す
  - 2つのベクトルがなす角度を類似度とみなす
  - 角度の尺度としてベクトルのなす余弦（コサイン）の値を用いる

$$\cos q = \frac{\vec{x} \cdot \vec{y}}{|\vec{x}| \times |\vec{y}|} = \frac{\sum_{i=1}^{n} x_i \times y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \times \sqrt{\sum_{i=1}^{n} y_i^2}}$$

- ベクトルの要素としては、単語の出現数(tf)ではなくtf・idf値を用いることも多い

まとめ

- 情報検索の基本的技術と言語処理の関連
  - 索引語の選択、質問・文書の表現
  - 言語処理：形態素解析
  - 索引語に対するインデックス付け
  - 情報検索モデル
- その他の言語処理技術
  - 係り受け解析：近年、統計的学習により高精度の解析が可能になりつつある
  - 語義の曖昧性の解消
  - 語の意味的類似性の自動獲得、シソーラスの自動構築