自然言語処理
第12回：
自然言語処理の応用：
情報構造化と検索

David Ramamonjisoa

目次

- 情報検索とは
- テキストに対する自働索引語付け
- 全文検索（ブーリアン検索モデル）
- ベクトル空間法
- 情報構造化
- テキストの構造化とハイパーテキスト
- まとめ

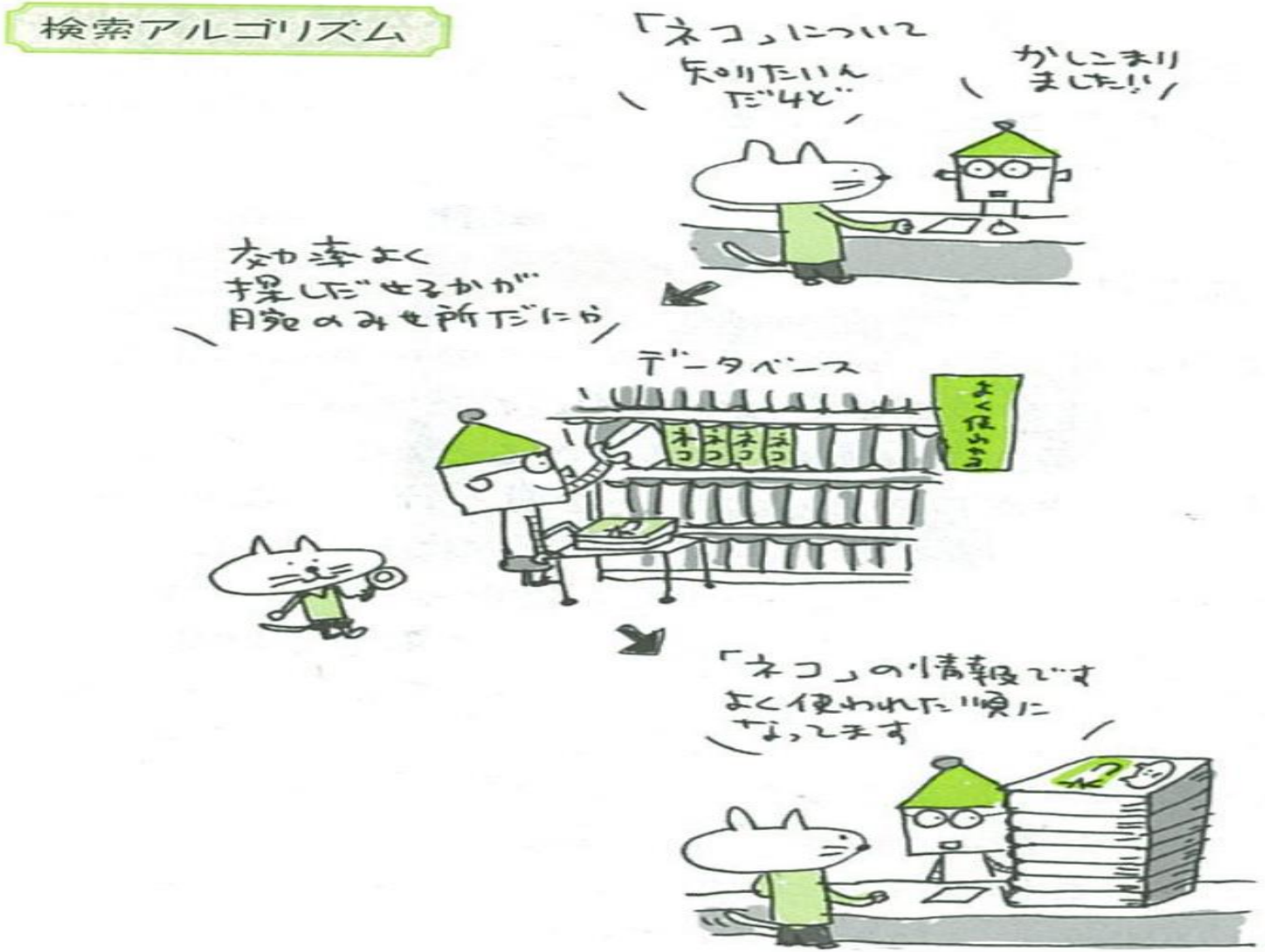
情報検索とは

- 情報検索とは(What is Information Retrieval?)

- 大量のデータの中からユーザの要求を満たす情報を見つけ出すことである(Finding relevant information in large collections of data)
- 検索対象となる情報として、主に文書となる（文書集合(document collection)）

ユーザからの情報要求は、自然言語の文や索引語(キーワード)で与えられるものである（検索質問(query)）

検索アルゴリズム



絵でわかる人工知能（三宅）

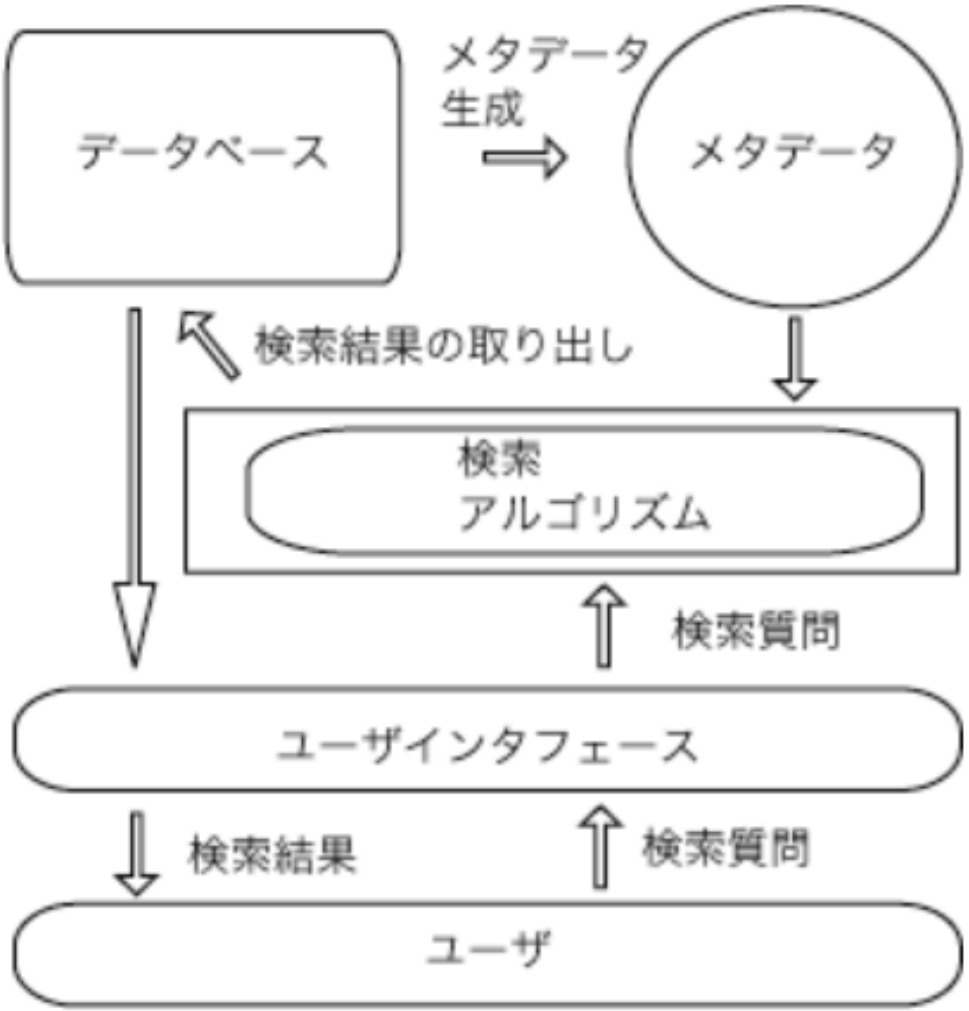
文書検索 (Document Retrieval)

- 図書館のオンラインカタログ検索(Online library catalogs (OPAC))
- 検索エンジンサイト(Internet search engines, such as AltaVista, Google, Goo, Nifty)
- 画像、映像、動画、音楽なども文書検索
- 専用システム(Specialized systems (aka vendors)):
  - MEDLINE (医療データベース　medical articles)
  - Lexis-Nexis (法律、ビジネス、などlegal, business, academic, . . .)
  - ReaD (日本の研究者データベース)

全文検索とディレクトリ検索

- ディレクトリ検索の代用サイト(Popular Web Directories):
  - Yahoo!, Open Directory Project (dmoz)
- ユーザが正しいディレクトリを見つける。(The user has to 'guess' the 'right' directories to find the information)
  - ウェブサイトはカテゴリに分類され、カテゴリは担当のエディタによって管理されている。
- 情報検索は膨大な情報の中から必要な情報を見つけ出す。
  - ユーザが自由に必要な情報だけを収集出来る

情報検索の全体図

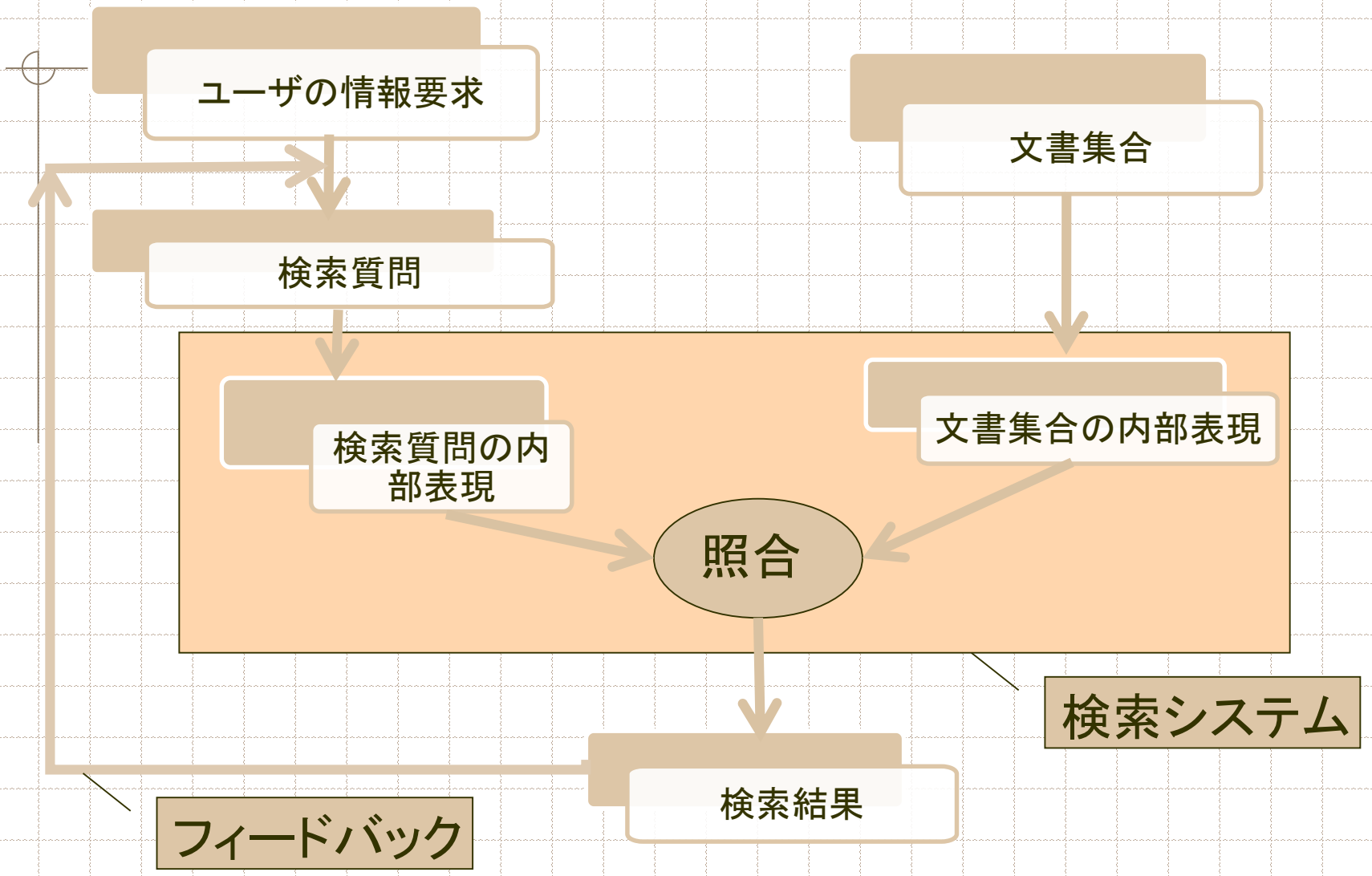


Source:Wikipedia

情報検索システムは主に以下に挙げる要素によって構成されている。

- **データベース**
- **検索対象のデータ**
- **メタデータ（索引語）**
- **ユーザインタフェース**
- **検索アルゴリズム**

情報検索のモデル

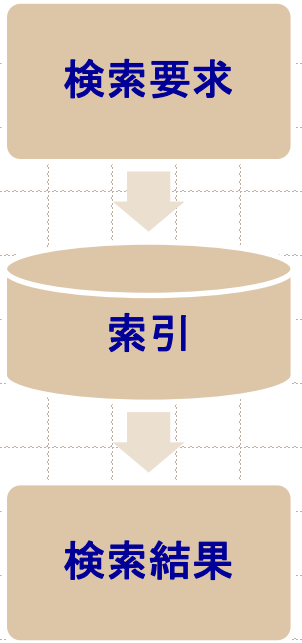

ユーザの情報要求
文書集合
検索質問
検索質問の内部表現
文書集合の内部表現
照合
検索システム
検索結果
フィードバック

情報検索モデルの分類

検索モデル	文書の内部表現	質問の内部表現	照合方法
ベクトル空間モデル	索引語の重みベクトル	索引語の重みベクトル	ベクトル間の類以度計算
ブーリアンモデル	特徴ベクトル・転置ファイル	特徴ベクトル/単語の論理式	論理演算＋遂次検索
半無限文字列に基づくモデル	パトリシア・トライ	文字列	木の探索
文字列照合	なし	なし	文字列同士の比較

内容型検索のモデル vs. 全文検索

- 内容型検索
  - 特徴付ける単語(Bag-Of-Words)
  - 類以度あるいは距離の定義
  - 例：ベクトル空間モデル、確率モデル、ネットワークモデル
- 全文検索
  - ■遂次検索:1文字づつ照合する
  - ■索引検索（前処理では索引を作成する：単語の位置を記録する）



検索モデルの要素

- ユーザ:
  - エキスパート vs. 初心者(Search expert (e.g., librarian) vs. non-expert)
  - ユーザのバックグラウンド
  - 詳しい　vs. 大まか (In-depth searching vs. 'just-wanna-get-an-idea' searching)
- 文書:
  - 複数言語(Different languages)
  - 半構造化文書 vs 生文書 (Semi-structured (e.g. HTML or XML) vs. plain)

文書の表現 (Document Representation)

- メタ記述(Meta-descriptions)
  - フィールド情報(Field information (author, title, date))
  - キーワード(Key words)
    - 既定義(Predefined)
    - 手作業で抽出される(Manually extracted (by author/editor))
- 内容：自動で判別する(Content: automatically identifying what the document is about)

文書の表現 (Document Representation)

索引作成	手動	自動
単語制御	既定分類方法	文章分類技術
自由文	既定分類方法	文章検索エンジン技術

手動 vs. 自動 の索引作成

- 手動の利点:

+ 人間の判断で行う
+ 単語制御の探索は効率的である。

- 手動の不利点:

– 時間がかかる
– 専門家しかできない仕事。コストが高い。
–分類の内容はたまに矛盾になる

自動内容作成表現

- 自然言語理解技術を利用する。
- スーパーコンピュータのみが出来る方法。
- 曖昧さが高い
- 文書は単純に単語の集合になる（単語箱）

自然言語処理による単語箱の作成

- アルファベット順で並べ
- 文法の情報は無くす
- トークン化
- 大文字と小文字の統一化
- ステム化、レッマ化(Stemming or lemmatization)
  - 形態素の情報を無くす
  - 例:‘agreements’ から
    ‘agreement’ (レッマ化)
    または ‘agree’ (ステム化)

単語箱"Bag of Words(BoW)"の例

Scientists have found compelling new evidence of possible ancient microscopic life on Mars, derived from magnetic crystals in a meteorite that fell to Earth from the red planet, NASA announced on Monday.

Bag-of-words=[a, ancient, announced, compelling, crystals, derived, earth, evidence, fell, found, from (2×), have, in, life, magnetic,mars, meteorite, microscopic, monday, nasa, new, of, on (2×), planet, possible, red, scientists, that, the, to]

ブーリアン検索(Boolean Retrieval)

- ブーリアン演算子: AND (NEAR), OR, NOT
- ブーリアン演算子の意味:
  - t1 AND t2 = {d | t1 ∈ r(d)} ∩ {d | t2 ∈ r(d)}

t1 と t2　が含まれている文書

  - t1 OR t2 = {d | t1 ∈ r(d)} ∪ {d | t2 ∈ r(d)}

t1またはt2を含まれている文書

  - NOT t1 = {d | t1 not in r(d)}

t1が含まれていない文書

ブーリアン検索の利点と不利点

- 利点:

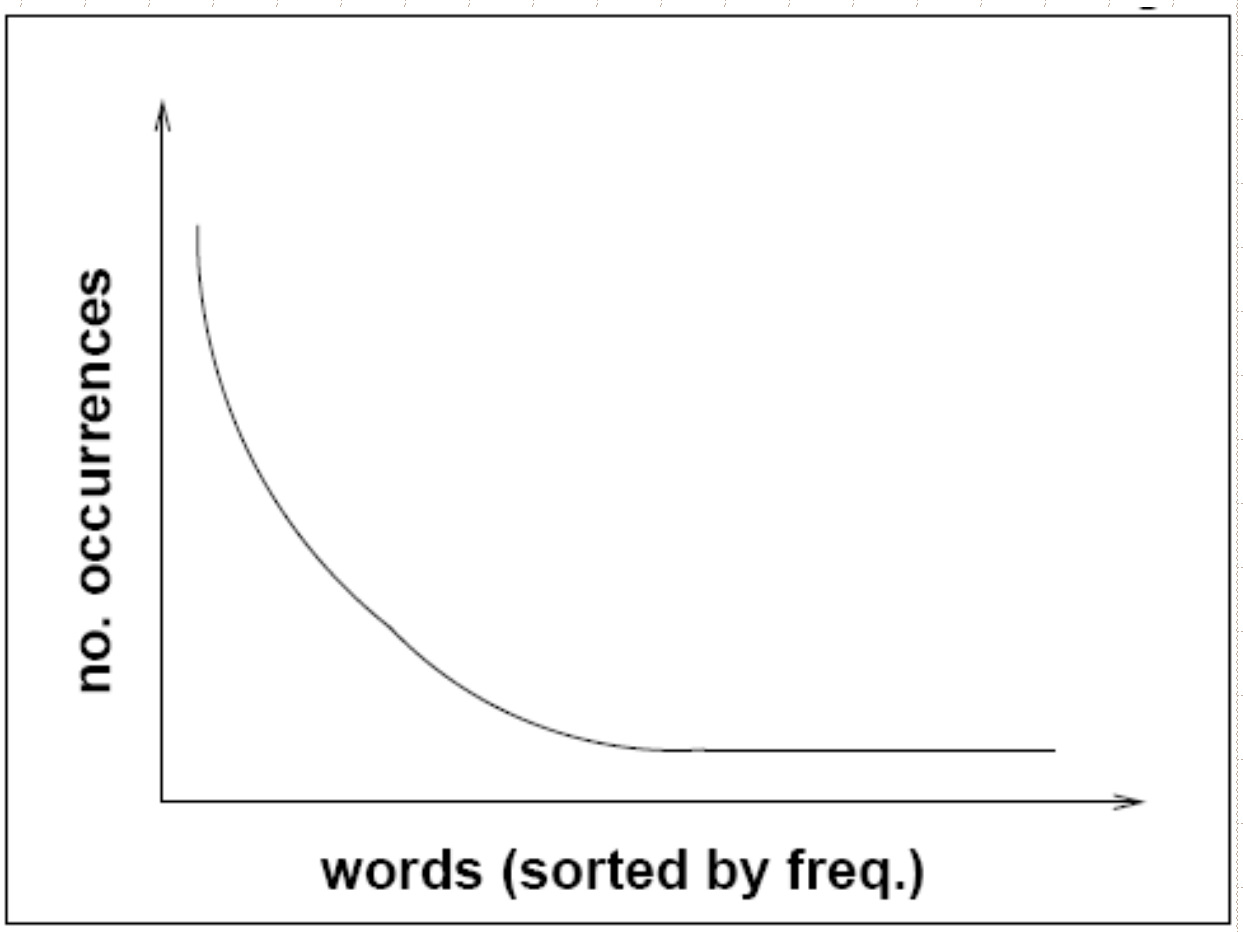
+ 単純な仕組
+ 質問文による結果が異なる

- 不利点:

– 専門家でないと公式の作成は難しいである
– 抽出した文書は順番なし
– 厳密照合は少数か沢山の文書を得られる

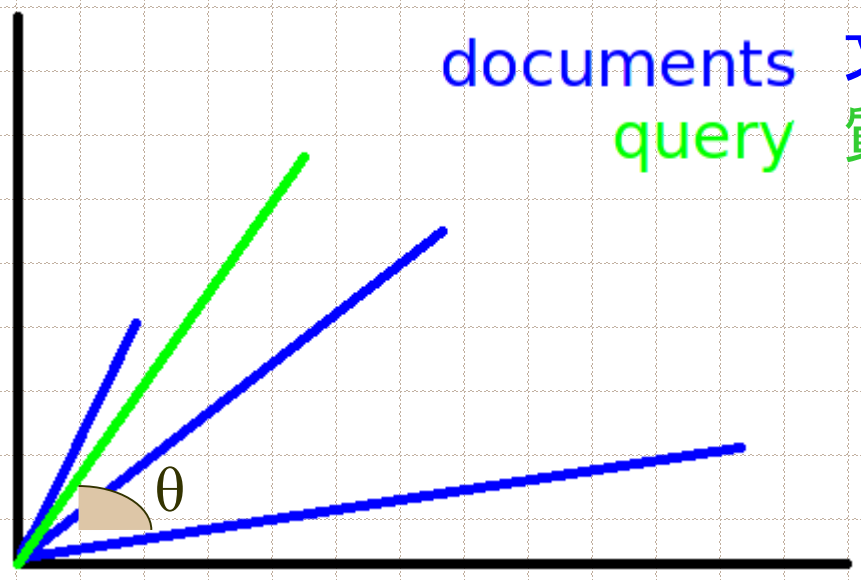
ジップ法(Zipf's Law)

- 少数の単語のみは頻度高い、残りは頻度低い



ベクトル空間モデル

- ユーザが自由文を質問文に入力できる
- 返す文書はランク付けている
- 最良なマッチングである（完全ではない）
- 全てが多次元空間でベクトルとなる



ベクトル空間の表現

- 文書が索引語の列
- 索引語は文書のベクトル(D)
- 同様に質問文も索引語のベクトル(q)
- ベクトルの長さが1に正規化する

	t_1	t_2	t_3	t_4	...
d_1	1	0	0	1	...
d_2	0	1	0	1	...
d_3	0	0	1	1	...
d_4	1	1	1	0	...
⋮	⋮	⋮	⋮	⋮	⋮

ベクトル空間モデル

- ベクトル間の類以度の計算
  - コサイン尺度　v1=Dj, v2=q (2つのベクトルのなす角度)
  - 類似度はコサインが1のちかいになる

$$\cos\theta = \frac{\mathbf{v_1} \cdot \mathbf{v_2}}{\|\mathbf{v_1}\| \|\mathbf{v_2}\|}$$

ベクトル空間モデルの例題

索 引 語

w_1	Bioinformatics
w_2	Biology
w_3	Chemistry
w_4	Enzymes
w_5	Evolution
w_6	Genes
w_7	Genome(s)
w_8	Proteins

文 書

1 Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins
2 Proteins, Enzymes, Genes: The Interplay of Chemistry and Biology
3 Adaptive Evolution of Genes and Genomes
4 Advances in Genome Biology: Genes and Genomes
5 Bioinformatics and Genome Research
6 Data Analysis in Molecular Biology and Evolution

索引語・文書行列：

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(索引語の重み：
索引語頻度)

ベクトル空間モデルの例題

$$\cos(d_j, q) = \frac{d_j \bullet q}{\|d_j\| \times \|q\|}$$

$$\cos(d_j, q) = \frac{\sum_{i \in [1, |d_j|]} w_{ij} q_i}{\sqrt{\sum_{i \in [1, |d_j|]} w_{i1}^2 \times \sum_{i \in [1, |q|]} q_i^2}}$$

```
>>> def cosine(d,q):
        d_q=[d[i] * q[i] for i in range(len(d))]
        d_2=[d[i]*d[i] for i in range(len(d))]
        q_2=[q[i]*q[i] for i in range(len(q))]
        cos_d_q=(sum(d_q)/(sum(d_2)*sum(q_2))**.5)
        return cos_d_q

>>> d4=[0,1,0,0,0,1,2,0]
>>> print ("%.3f" % cosine(d4,q))
0.866
>>> print ("%.3f" % cosine(d1,q))
0.408
```

検索質問文： Genes and Genomes

検索質問ベクトル： $\boldsymbol{q} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$

類似度計算

$$\begin{aligned}
\cos(\boldsymbol{d}_1, \boldsymbol{q}) &= 0.408 \\
\cos(\boldsymbol{d}_2, \boldsymbol{q}) &= 0.316 \\
\cos(\boldsymbol{d}_3, \boldsymbol{q}) &= 0.816 \\
\cos(\boldsymbol{d}_4, \boldsymbol{q}) &= 0.866 \\
\cos(\boldsymbol{d}_5, \boldsymbol{q}) &= 0.5 \\
\cos(\boldsymbol{d}_6, \boldsymbol{q}) &= 0.0
\end{aligned}$$

検索結果

検索順位	文書	類似度
1	D_4	0.866
2	D_3	0.816
3	D_5	0.5
4	D_1	0.408
5	D_2	0.316

ベクトル空間モデルの正規化

- 文書の長さ(Document Length)
- 索引語の重み(Term Weights)

  tf.idf-score

  索引語頻度×文書頻度の逆数

索引語の重み付け

- n個の文書$D_1, D_2, \ldots, D_n$があり、これらの文書集合から全部でm個の索引語$w_1, w_2, \ldots, w_m$が抽出される。
- 重みd_{ij}は索引語w_iの文書D_jの特徴付ける
  - 局所的重み:l_{ij}
  - 大域的重み:g_i
  - 文書正規化係数:n_j
  - 索引語重み$d_{ij} = \frac{l_{ij} g_i}{n_j}$

重み付けの計算式

表 3.3　重み付けの計算式

局所的重み付け	2 進重み (binary weight)	$l_{ij} = \begin{cases} 1 & (f_{ij} > 0) \\ 0 & (f_{ij} = 0) \end{cases}$
	索引語頻度 (term frequency; TF)	$l_{ij} = f_{ij}$
	対数化索引語頻度 (logarithmic TF)	$l_{ij} = \log(1 + f_{ij})$
	拡大正規化索引語頻度 (augmented normalized TF)	$l_{ij} = \begin{cases} 0.5 + 0.5\dfrac{f_{ij}}{\max_k f_{kj}} & (f_{ij} > 0) \\ 0 & (f_{ij} = 0) \end{cases}$
大域的重み付け	重み付けなし	$g_i = 1$
	文書頻度の逆数 (inverse document frequency; IDF)	$g_i = \log \dfrac{n}{n_i}$
	確率的 IDF (probabilistic IDF)	$g_i = \log \dfrac{n - n_i}{n_i}$
	大域的頻度 IDF (global frequency IDF)	$g_i = \dfrac{F_i}{n_i}$
	エントロピー (entropy)	$g_i = 1 + \dfrac{1}{\log n} \sum_{j=1}^{n} \dfrac{f_{ij}}{F_i} \log \dfrac{f_{ij}}{F_i}$
文書正規化	正規化なし	$n_j = 1$
	コサイン正規化 (cosine normalization)	$n_j = \sqrt{\sum_{i=1}^{m} (l_{ij} g_i)^2}$
	ピボット正規化 (pivoted normalization)	$n_j = (1 - \text{slope}) \times \text{pivot} + \text{slope} \times l_j$

表 3.2　記号の説明

記号	名前	意　味
m	索引語数	文書集合全体にわたる索引語の総数
n	文書数	文書集合中の文書の総数
f_{ij}	索引語頻度	索引語 w_i の文書 D_j における出現頻度
F_i	大域的頻度	文書集合全体を通しての索引語 w_i の出現頻度
n_i	文書頻度	索引語 w_i を含む文書数

重み付けtfidf

- 索引語頻度×文書頻度の逆数
- 索引語重み, iは文書, jは索引語

$$(1)\ tfidf_{ij} = f_{ij} * \log(\frac{\mathrm{n}}{n_i})$$

$$(2)\ tfidf_{ij} = \frac{f_{ij}}{f_i} * \log(\frac{\mathrm{n}}{n_i})$$

f_iは文書iの中の索引語の数

文書例題でtfidf(1)行列の結果

```
print(tf_idf.transpose())
```

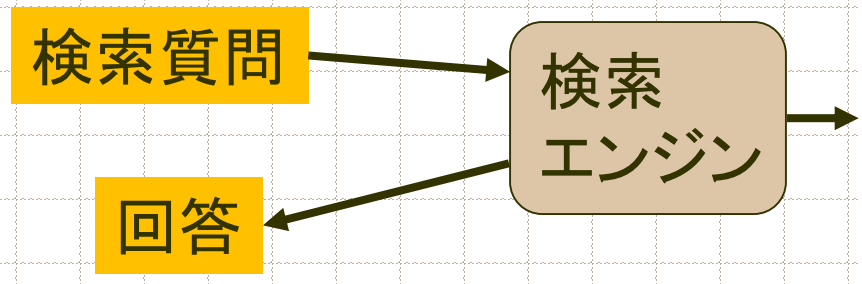
```
[[1.09861229 0.         0.         0.         1.09861229 0.        ]
 [0.         0.69314718 0.         1.79175947 0.         0.40546511]
 [0.         0.69314718 0.         0.         0.         0.        ]
 [0.         0.69314718 0.         0.         0.         0.        ]
 [0.         0.         1.79175947 0.         0.         0.40546511]
 [1.09861229 0.69314718 1.79175947 1.79175947 0.         0.        ]
 [0.         0.         1.79175947 3.58351894 1.09861229 0.        ]
 [1.09861229 0.69314718 0.         0.         0.         0.        ]]
```

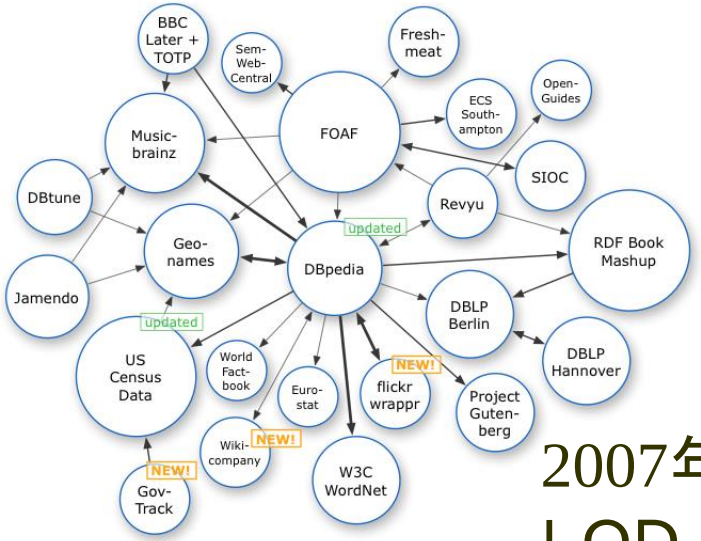
tfidf(genome,d4) = 3.58
tfidf(gene,d4)=1.79

tfidfの方が精度向上が分かる。

ナレッジグラフ(Knowledge Graph)

- ウエブ上から作られた知識ベース
- Linked Dataの応用一つ
- セマンティックウエブを用いて情報検索を行う
- 例：Google, Facebook, Microsoft　bingなど





2007年
LOD

まとめ

- 情報検索の基礎
- 情報検索モデル
- 内容型検索と全文検索
- 文書の表現
- 単語箱
- ブーリアン検索モデル
- ベクトル空間モデル
- セマンティックウエブ
- ナレッジグラフ