計算機ソフトウェア
第十一回

電気電子工学科
黒橋禎夫

テキスト処理

- テキストの扱い
  - 文字コード、フォント、テキストの表現形式
- 文字列操作
  - 整列、探索、照合

文字コード

- 文字 a ⇔ 文字コード 01100001

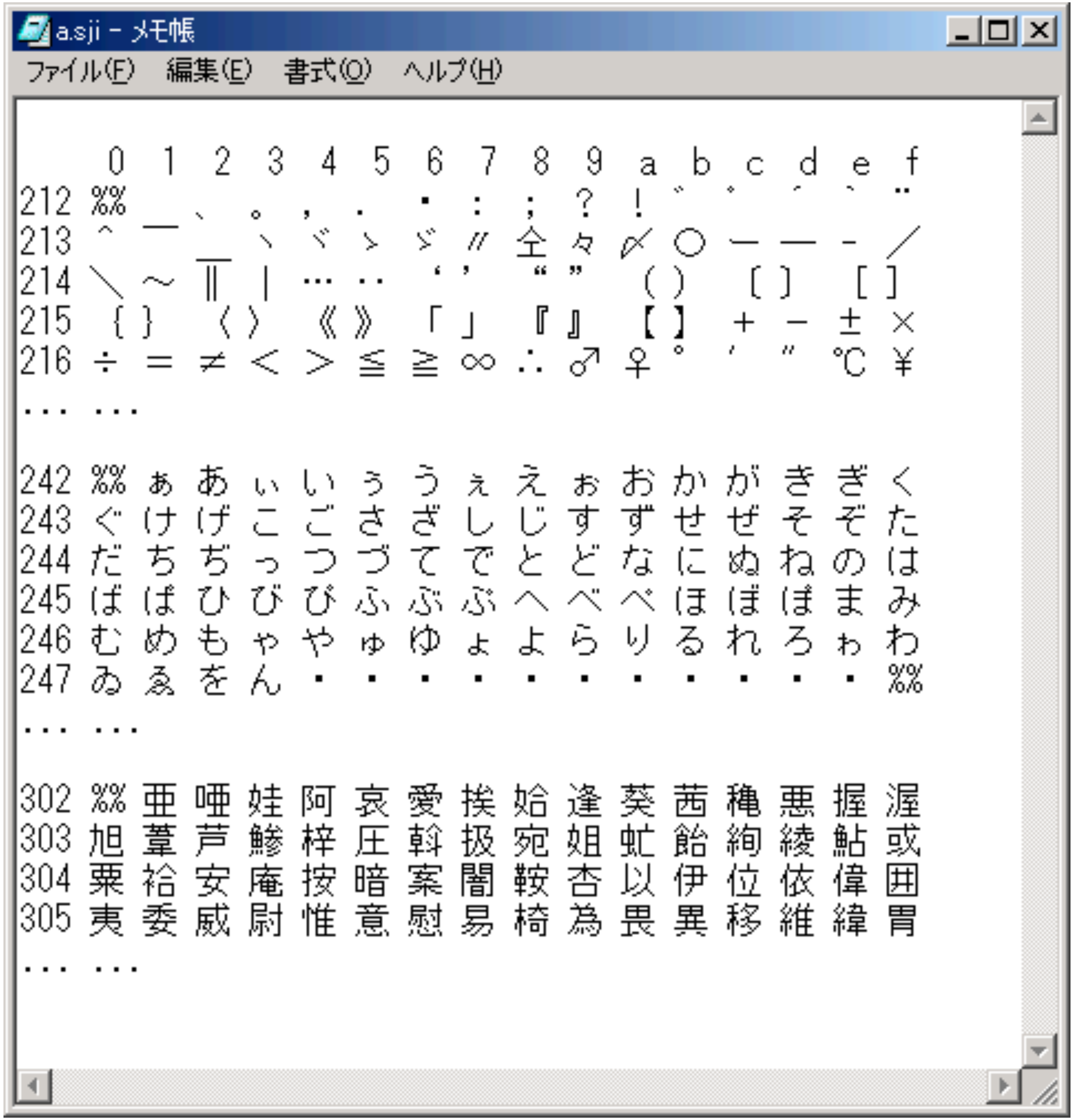
- 文字コードセット
  - ある特定の文字集合
  - ISO（国際標準化機構）、JIS（日本工業規格）などの規格

文字コードセット

- ASCIIコードセット（ISO8859－1）
  - 最上位ビットは0
  - 制御文字領域（00～20，7F）
- JISローマ字（JIS X0201 ローマ字）
  - 5C：￥、7E：‾
- JIS片仮名（JIS X0201 片仮名）

文字コードセット

- JIS漢字（JIS X0208）
  - 1978年、83年、90年、97年
  - 21～7Eの範囲（ASCII制御文字範囲を除く）
  - 第一水準、第二水準
- JIS補助漢字（JIS X0212）
  - 1990年



a.sji - メモ帳

ファイル(F) 編集(E) 書式(O) ヘルプ(H)

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
212	%%		、	。	，	．	・	：	；	？	！	゛	゜	´	｀	¨
213	＾	￣	＿	ヽ	ヾ	ゝ	ゞ	〃	仝	々	〆	〇	ー	―	‐	／
214	＼	～	∥	｜	…	‥	‘	’	“	”	（	）	〔	〕	［	］
215	｛	｝	〈	〉	《	》	「	」	『	』	【	】	＋	－	±	×
216	÷	＝	≠	＜	＞	≦	≧	∞	∴	♂	♀	°	′	″	℃	￥

・・・ ・・・

242	%%	ぁ	あ	ぃ	い	ぅ	う	ぇ	え	ぉ	お	か	が	き	ぎ	く
243	ぐ	け	げ	こ	ご	さ	ざ	し	じ	す	ず	せ	ぜ	そ	ぞ	た
244	だ	ち	ぢ	っ	つ	づ	て	で	と	ど	な	に	ぬ	ね	の	は
245	ば	ぱ	ひ	び	ぴ	ふ	ぶ	ぷ	へ	べ	ぺ	ほ	ぼ	ぽ	ま	み
246	む	め	も	ゃ	や	ゅ	ゆ	ょ	よ	ら	り	る	れ	ろ	ゎ	わ
247	ゐ	ゑ	を	ん	・	・	・	・	・	・	・	・	・	・	・	%%

・・・ ・・・

302	%%	亜	唖	娃	阿	哀	愛	挨	姶	逢	葵	茜	穐	悪	握	渥
303	旭	葦	芦	鯵	梓	圧	斡	扱	宛	姐	虻	飴	絢	綾	鮎	或
304	粟	袷	安	庵	按	暗	案	闇	鞍	杏	以	伊	位	依	偉	囲
305	夷	委	威	尉	惟	意	慰	易	椅	為	畏	異	移	維	緯	胃

・・・ ・・・

文字コード体系

- 複数の文字コードセットを混在
  - 3021:0！（JISローマ字）、亜（JIS漢字）
- JISコード
  - エスケープシーケンス
    - ASCIIコードセット　ESC　（　B
    - JISローマ字　　ESC　（　J
    - JIS漢字　　ESC　$　B
  - 最上位ビットが0→データ通信

文字コード体系

- 日本語EUCコード
  - もともとUNIX上での規格（Extended UNIX Code）
  - JISローマ字　0???????
  - JIS漢字　　1??????? 1???????
  - JIS片仮名　8E 1???????
  - JIS補助漢字 8F 1??????? 1???????

文字コード体系

- シフトJISコード（MS漢字コード）
  - マイクロソフトなどによって作成
  - JISローマ字 0???????
  - JIS片仮名　1???????
  - JIS漢字　　１バイト目を８１～９F、E０～EF
  - 問題点
    - 制御文字領域の使用
    - JIS補助漢字などの追加不可能

Unicode

- 1993年に国際標準化機構(ISO)でISO/IEC 10646の一部(UCS-2)として標準化された文字コード体系。
- すべての文字を16ビット(2バイト)で表現し、1つの文字コード体系で多国語処理を可能にしようとするもの。
- 2バイト表記では最大65536文字しか収録できないため、中国語・日本語・韓国語で同じ意味や同じルーツの漢字はすべて同じ文字とみなし、同じコードを割り当てる統合作業が行われている。
- 最初の規格が策定された後にハングル文字の追加や異体字表現方式の策定が行われ、部分的に3バイト以上を使用する体系に変化している。このため、現在はUnicode全体は4バイトで定義(UCS-4)されている。
- cf. UTF-8, UTF-16

UTF-8

0x00～0x7Fまでの古典的ASCII文字はそのまま、
0x7Fより大きい文字は0x80以上のバイト列に変換

0x00000000 – 0x0000007F:
0???????
0x00000080 – 0x000007FF:
110????? 10??????
0x00000800 – 0x0000FFFF:
1110???? 10?????? 10??????
0x00010000 – 0x001FFFFF:
11110??? 10?????? 10?????? 10??????
0x00200000 – 0x03FFFFFF:
111110?? 10?????? 10?????? 10?????? 10??????
0x04000000 – 0x7FFFFFFF:
1111110? 10?????? 10?????? 10?????? 10?????? 10??????

0110001 11100011 10000001 10000010
a あ

文字の表示

- ビットマップ
  - 白と黒の領域（ドット）の格子
- 解像度
  - dpi : １インチ(25.4mm)に何ドットか
  - ディスプレイ : 70～100dpi、
  - プリンタ : 240～1200dpi
- ポイント（1/72.27インチ　0.3mm）
  - 14ポイントの文字を72dpiのディスプレイに表示すると何ドット？

フォント

- 一組の字種に対して、文字の図形を統一的に指定したもの
  - ビットマップ・フォント
  - アウトライン・フォント

- デザイン（書体）、大きさ、傾きなどのバリエーション

テキストの表現形式

- イメージ情報
- 物理情報
  - ポイント、フォント
- 論理情報
  - タイトル、強調

SGML

- Standard Generalized Markup Language
- 論理情報の記述
- 1986年、ISO8879、1992年、JIS X4151
- アメリカの公的機関、業界団体の支援
- DTD (Document Type Definition)
- 物理情報への対応付けはDSSSLで

SGMLのDTD

```
<!ELEMENT chp - - (chp-title, intro, sec+)>
<!ELEMENT chp-title - - (#PCDATA)>
<!ATTLIST chp-title change (no|yes) yes>
<!ELEMENT intro - - (#PCDATA)>
<!ELEMENT sec - - (sec-title, p+)>
<!ELEMENT sec-title - - (#PCDATA)>
<!ELEMENT p - - (#PCDATA)>
```

SGMLのテキスト

```
<!DOCTYPE CHAPTER SYSTEM "CHAPTER.DTD">

<CHAPTER>
<chp><chp-title change="no">テキスト処理</chp-title>
<intro> 本章では. . . .  </intro>
<sec><sec-title>計算機内部での文字の扱い</sec-title>
<p>0か1. この区別を. . . . </p>
<p>0/1を区別する1単位を. . . . </p>
. . . .
</sec>
<sec><sec-title>文字の表示</sec-title>
<p>計算機内部で文字コードとして. . . . </p>
. . . .
</sec>
</chp>
</CHAPTER>
```

HTML

- HyperText Markup Language
- 1990年、CERN(欧州粒子物理研究所)
- 物理情報への対応付けはブラウザ
- 音声、画像データの挿入、リンク
- テキストの構造は正確に表現できない

- SGMLとHTMLの中間　→　XML

TeX

- D. E. Knuthによって開発
- 科学技術論文などで広く利用
- 論理情報を表現
- 物理情報へはスタイル・ファイル
- 数式の表示、フォント整備環境、章・節等の自動管理など

PostScript と PDF

- １９８５年、アドビ・システムズ社
- 物理情報
- アウトライン・フォント、カラーの扱いを含むプログラミング言語

- PDFは、バイナリ形式、サイズ小、編集可能、ページの区別

文字列操作

- コンピュータによるテキスト管理
  → 高速な検索

- 整列、探索、照合

辞書式順序

- 文字間の順序関係: 文字コード順
  - A < B < C < … < a < b < c …
- 二つの文字列の順序関係:
  はじめにあらわれた異なる文字間の順序
  - A < AA < AAA
  - ABC < ABD < ACA < aaa

文字列の整列

- クイックソート
  - 基準値を選び、それより小と大にわける
  - これを再帰的に繰り返す
- 基底法
  - 文字列に限られる
  - 整列対象の値が限られた範囲であれば、それぞれの出現回数を数えればよい
  - これを末尾の文字からくりかえす

文字列の探索

- 二分探索
- ハッシュ法（hashing）

- トライ法（trie）
- パトリシア木（patricia tree）

文字列の照合

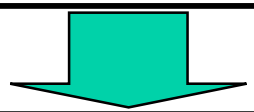
- 完全一致
  - ボイヤー・ムーア法
- 正規表現の照合
  - (T | t)ext : Text または text
  - コンピュータ(ー |)
  - (本当の)*話 : 本当の話、本当の本当の話…

文字列の照合

- 近似照合
  - desk desks, happy happierなどの語尾変化
  - center centreなどの異表記
  - 検索キーが不確かな場合
  - テキストに誤りがある場合

転置インデックス（索引）

文書1	言語、コンピュータ、問題
文書2	コンピュータ、問題
文書3	言語、問題、情報
文書4	問題、情報



言語	文書1、文書3
コンピュータ	文書1、文書2
問題	文書1、文書2、文書3、文書4
情報	文書2、文書3、文書4

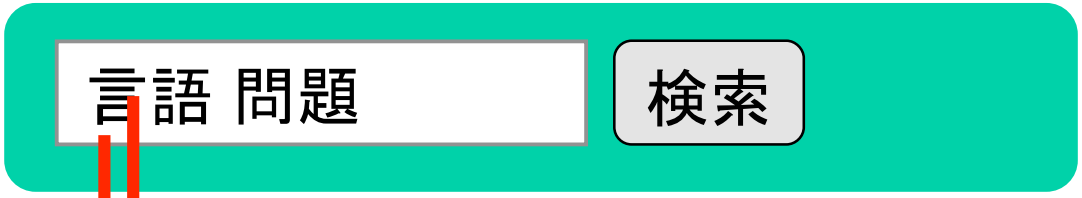
語の重要度（TF.IDF）

語の頻度（Term Frequency）

TF	文書1	文書2	文書3	文書4	IDF
言語	2	0	1	0	2
コンピュータ	1	1	0	0	2
問題	2	2	3	1	1
情報	0	1	2	1	1.3

全文書数 / 語の出現する文書数
（Inverse Document Frequency）

語の重要度（TF.IDF）



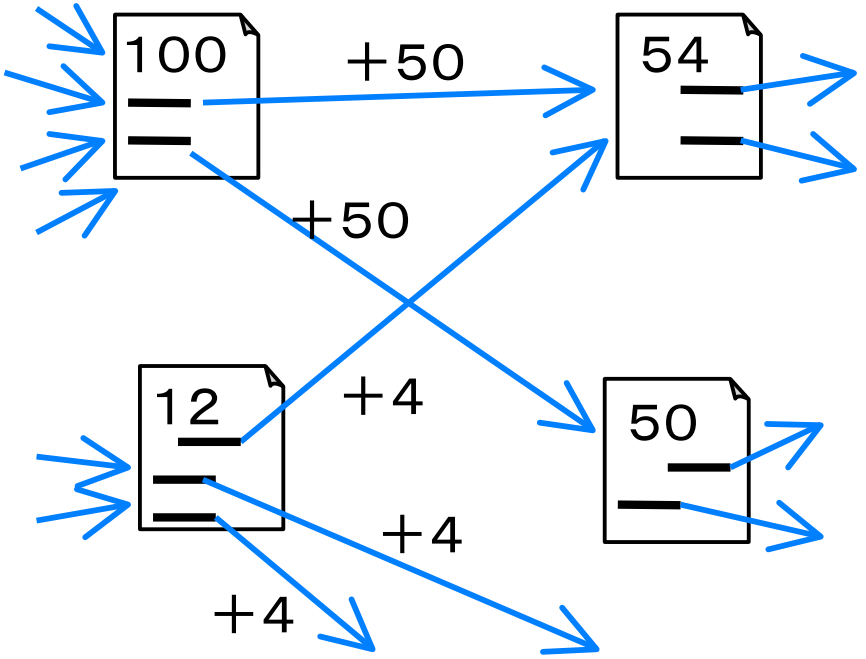
TF.IDF	文書1	文書2	文書3	文書4
言語	4	0	2	0
コンピュータ	2	2	0	0
問題	2	2	3	1
情報	0	1.3	2.6	1.3
	6	(2)	5	(1)

PageRank

- 「多くの良質なWebページから参照されている Webページは良質である」

$$\mathrm{R}(u) = \sum_{v \to u} \frac{\mathrm{R}(v)}{|\mathrm{B}v|}$$

$$\mathbf{R} = c\mathbf{A}\mathbf{R}$$



アンカーテキストの利用

- アンカーテキスト：リンクが張られた文字列
  例：京都大学
- アンカーテキストはリンク先テキストの一部とみなす
  - 特定のトピックに関連し、被参照数の大きいWebページが検索されやすい
  - リンク先に含まれない語句でも検索できる（例：“京大”）