分散システム

廣川佐千男

講義内容

1.HTML仕様
2.HTTPプロトコル
3.クローラー、検索エンジン
4.Webグラフ
5.Webマイニング

HTMLの変遷

HTML 2.0 (1995/09/21)

　HTMLの基本構造、見出し、フォーム

HTML 3.2 (1997/01/14)

　フォント属性、テーブルの追加

HTML 4.0 (1997/12/18)

　フレーム、スタイルシートに関する拡張

HTML 4.01 (1999/12/24)

　HTML 4.0の改良

HTML仕様

http://www.w3.org/TR/1999/REC-html401-19991224

HTML4. 01

1.About the HTML 4 Specification
2.Introduction to HTML 4
3.On SGML and HTML
4.Conformance: requirements and recommendations
5.HTML Document Representation - Character sets, character encodings, and entities
6.Basic HTML data types - Character data, colors, lengths, URIs, content types, etc.
7.The global structure of an HTML document - The HEAD and BODY of a document
8.Language information and text direction - International considerations for text
9.Text - Paragraphs, Lines, and Phrases
10.Lists - Unordered, Ordered, and Definition Lists
11.Tables
12.Links - Hypertext and Media-Independent Links
13.Objects, Images, and Applets
14.Style Sheets - Adding style to HTML documents

15.Alignment, font styles, and horizontal rules

HTML4. 01

16. Frames - Multi-view presentation of documents
17. Forms - User-input Forms: Text Fields, Buttons, Menus, and more
18. Scripts - Animated Documents and Smart Forms
19. SGML reference information for HTML - Formal definition of HTML and validation
20. SGML Declaration of HTML 4
21. Document Type Definition
22. Transitional Document Type Definition
23. Frameset Document Type Definition
24. Character entity references in HTML 4
25. Changes
25. Performance, Implementation, and Design Notes
27. References
28. Index of Elements
29. Index of Attributes
30. Index

構成

- **Sections 2 and 3: Introduction to HTML 4**
  - The introduction describes HTML's place in the scheme of the World Wide Web, provides a brief history of the development of HTML, highlights what can be done with HTML 4, and provides some HTML authoring tips.

構成

- **Sections 4 - 24: HTML 4 reference manual**
  - The bulk of the reference manual consists of the HTML language reference, which defines all elements and attributes of the language

This document has been organized by topic rather than by the grammar of HTML. Topics are grouped into three categories: **structure**, **presentation**, and **interactivity**. Although it is not easy to divide HTML constructs perfectly into these three categories, the model reflects the HTML Working Group's experience that separating a document's structure from its presentation produces more effective and maintainable documents.

WWWの3大特徴

2.1 What is the World Wide Web?

The *World Wide Web (Web)* is a network of information resources. The Web relies on three mechanisms to make these resources readily available to the widest possible audience:

1. A uniform naming scheme for locating resources on the Web (e.g., URIs).
2. Protocols, for access to named resources over the Web (e.g., HTTP).
3. Hypertext, for easy navigation among resources (e.g., HTML).

HTML：構造と表示の区別

2.4.1 Separate structure and presentation

HTML has its roots in SGML which has always been a language for the specification of structural markup. As HTML matures, more and more of its presentational elements and attributes are being replaced by other mechanisms, in particular style sheets. Experience has shown that separating the structure of a document from its presentational aspects reduces the cost of serving a wide range of platforms, media, etc., and facilitates document revisions.

データ構造としての文章

- HTMLは構造的文章の記述法
- 構造としてのHTML文書は要素 Elementから構成される
- Elementは開始タグ、内容、終了タグの3つで表現される。
- どのような要素がHTMLとしてあるか記述する定義が、DTD（document type definition）

  http://www.w3.org/TR/html4/strict.dtd

  (読み方は、SGML,XML

  DOCTYPE,ENTITY,ELEMENT,ATTRIBUTE)

Block-level Element and Inline Element

Generally, block-level elements may contain inline elements and other block-level elements. Generally, inline elements may contain only data and other inline elements. Inherent in this structural distinction is the idea that block elements create "larger" structures than inline elements.

Block-level Element and Inline Element

P / H1 / H2 / H3 / h4 / H5 / H6 / UL / OL / DIR / MENU / PRE / DL / DIV / CENTER / NOSCRIPT / NOFRAMES / BLOCKQUOTE / FORM / ISINDEX / HR / TABLE / FIELDSET / ADDRESS / MULTICOL

文字列 / TT / I / B / U / S / STRIKE / BIG / SMALL / EM / STRONG / DFN / CODE / SAMP / KBD / VAR / CITE / ABBR / ACRONYM / A / IMG / APPLET / OBJECT / FONT / BASEFONT / BR / SCRIPT / MAP / Q / SUB / SUP / SPAN / BDO / IFRAME / INPUT / SELECT / TEXTAREA / LABEL / BUTTON / BLINK / EMBED / LAYER / ILAYER / NOLAYER / NOBR / WBR / RUBY / RB / RP / RT / SPACER

HTML,HEAD,TITLE,BODY

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
   "http://www.w3.org/TR/html4/strict.dtd">
<HTML>
<HEAD>
<TITLE>A study of population dynamics</TITLE>
... other head elements...
</HEAD>
<BODY>
... document body...
</BODY>
</HTML>
```

Index of Elements

http://www.w3.org/TR/1999/REC-html401-19991224/index/elements.html

RFC (Request for Comment)

- インターネット上の技術文書
  - 事実上の標準規格
- 元の意味：コメントをください、意見募集
- 議論の場
  - IETF　Internet　draft
  - The Internet Engineering Task Force
- 入手
  - http://www.ietf.org/iesg/1rfc_index.txt
  - ftp://ftp.kyushu-u.ac.jp/pub/rfc/など

rfc-index.txt

- 索引（インデックス）のファイル
  - RFCのリスト

2616 Hypertext Transfer Protocol -- HTTP/1.1. R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee. June 1999. (Format: TXT=422317, PS=5529857, PDF=550558 bytes) (Obsoletes RFC2068) (Updated by RFC2817) (Status: DRAFT STANDARD)

番号　　状態　　文書の破棄・無効状態

proposed standard; Standard Track の第1段階
draft standard ; 標準化過程の第2段階
standard 標準化過程の最終段階
informational　広報
experimental　経験/試験的/実験的 ()
Best Current Practice … 現時点での最良の方法
historic … 歴史; 現在では使われていない技術

RFC関連ページ

- http://www.imasy.or.jp/~yotti/rfc-idx.html
- http://www.rfc-editor.org/rfcsearch.html
- http://rfc-jp.nic.ad.jp/

課題

- URI（RFC 2396）
- HTTP(RFC2616)

URI

- RFC2396:Uniform Resource Identifiers (URI): Generic Syntax
- A Uniform Resource Identifier (URI) is a compact string of characters for identifying an abstract or physical resource. This document defines the generic syntax of URI, including both absolute and relative forms, and guidelines for their use.

1.1 Overview of URL

Uniformの利点

- アクセス機構が異なる種類の資源でも、同一の形式で文脈で利用できる。
- 資源識別子に対する共通的解釈
- 新たな種類の資源についても記述可能
- さまざまな状況で識別子の利用が可能

1.1 Overview

Resource

- 識別できるものであれば何でもOK

  （e.g.,画像、サービス（今日の天気））

  network “retrievable”でなくてもOK

  （e.g.,人、会社、図書館の本）

- 概念対応であること

  ある時点で対応している内容である必要はない。

  時間とともに内容は変わっても資源としては同一とみなす。

1.1 Overview

Identifier

参照するための対象

特定の形式の文字列

1.2 URI,URL and URN

- URI = locator,name,両方
- URL　Uniform Resource Locator

  URIの一部。名前あるいは属性で資源を特定するのではなく、アクセス手段（例えばネットワーク上の場所）として資源を指定する方法
- URN

  固有の名詞を使って資源を特定する。永続的なラベリング。

  例：ISBN番号　1-23-456-7890 の本
  urn:ISBN:1-23-456-7890

1.3 Example URI

ftp://ftp.is.co.za/rfc/rfc1808.txt

-- ftp scheme for File Transfer Protocol services

gopher://spinaltap.micro.umn.edu/00/Weather/California/Los%20Angeles

-- gopher scheme for Gopher and Gopher+ Protocol services

http://www.math.uio.no/faq/compression-faq/part1.html

-- http scheme for Hypertext Transfer Protocol services

mailto:mduerst@ifi.unizh.ch

-- mailto scheme for electronic mail addresses

news:comp.infosystems.www.servers.unix

-- news scheme for USENET news groups and articles

telnet://melvyl.ucop.edu/

-- telnet scheme for interactive services via the TELNET Protocol

1.4 Hierarchical URI and Relative Forms

- 絶対識別子：独立に資源を参照

  例：http://www.yahoo.co.jp/index.html
- 相対識別子：差異を記述することで参照

  例：../../index.html
- URIスキームでは、階層を“/”で分割することで表記

レポート：相対表現は何の役に立つか？どんなときに役にたつか？

1.5. URI Transcribability

- A URI is a sequence of characters.
- A URI may be transcribed from a non-network source, and thus should consist of characters that are most likely to be able to be typed into a computer.
- A URI often needs to be remembered by people, and it is easier for people to remember a URI when it consists of meaningful components

- The goal of transcribability can be described by a simple scenario. Imagine two colleagues, Sam and Kim, sitting in a pub at an international conference and exchanging research ideas. Sam asks Kim for a location to get more information, so Kim writes the URI for the research site on a napkin. Upon returning home, Sam takes out the napkin and types the URI into a computer, which then retrieves the information to which Kim referred.

1.6. Syntax Notation and Common Elements

- layout form : a general description of the order of components and component separators

  <first>/<second>;<third>?<fourth>

- BNF-like grammar (AppendixA)

  alpha = lowalpha | upalpha

  lowalpha = "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" | "j" | "k" | "l" | "m" | "n" | "o" | "p" | "q" | "r" | "s" | "t" | "u" | "v" | "w" | "x" | "y" | "z"

2. URI Characters and Escape Sequences

```
uric          = reserved | unreserved | escaped
reserved      = ";" | "/" | "?" | ":" | "@" | "&" | "=" | "+" |
                "$" | ","
unreserved    = alphanum | mark
mark          = "-" | "_" | "." | "!" | "~" | "*" | "'" |
                "(" | ")"
escaped       = "%" hex hex
hex           = digit | "A" | "B" | "C" | "D" | "E" | "F" |
                        "a" | "b" | "c" | "d" | "e" | "f"
```

2.1 URI and non-ASCII characters

- URIでは文字とオクテット(an “octet” (an 8-bit byte))を区別
- A URI is represented as a sequence of characters
- A URI scheme may define a mapping from URI characters to octets

2.2. Reserved Characters

- URI成分の区切り記号のための予約語

reserved = ";" | "/" | "?" | ":" | "@" | "&" | "=" | "+" | "$" | ","

成分として使うときにはエスケープして使うこと

2.3. Unreserved Characters

- 区切り記号などの特殊な目的を持たない文字

unreserved = alphanum | mark

mark   = "-" | "_" | "." | "!" | "~" | "*" | "'" | "(" | ")"

2.4. Escape Sequences

escaped = "%" hex hex

hex = digit | "A" | "B" | "C" | "D" | "E" | "F" |
"a" | "b" | "c" | "d" | "e" | "f“

- e.g., "%20" US-ASCII space character.
- "%7e" is sometimes used instead of "~" in an http URL
- パーセント文字“%”をURIのデータと使うときには、

“%25”

2.4.3. Excluded US-ASCII Characters

- control = <US-ASCII coded characters 00-1F and 7F hexadecimal>
- space = <US-ASCII coded character 20 hexadecimal>
- delims = "<" | ">" | "#" | "%" | <">
- unwise = "{" | "}" | "|" | "¥" | "^" | "[" | "]" | "`“

  because gateways and other transport agents are known to sometimes modify such characters, or they are used as delimiters.
- これらを使うときには、エスケープすること

3. URI Syntactic Components

- <scheme>:<scheme-specific-part>
- 一般的形式
  <scheme>://<authority><path>?<query>

  （authority,path,query　部分がないこともある）

例 absoluteURI　= scheme ":" (hier_part | opaque_part)

hier_part　　= (net_path | abs_path) ["?" query]

net_path　　= "//" authority [abs_path]

abs_path　　= "/" path_segments

opaque_part　= uric_no_slash *uric

uric_no_slash = unreserved | escaped | ";" | "?" | ":" | "@" |
"&" | "=" | "+" | "$" | ","

3.1. Scheme Component

scheme = alpha *(alpha | digit | "+" | "-" | ".")

- the first component defining the semantics for the remainder of the URI string.
- 小文字英字、数字、＋、．、－

3.2. Authority Component

- name spaceを決めるもの
  - an Internet-based server
  - a scheme-specific
  - registry of naming authorities
- authority = server | reg_name
- The authority component is preceded by a double slash "//" and is terminated by the next slash "/", question-mark "?", or by the end of the URI. Within the authority component, the characters ";", ":", "@", "?", and "/" are reserved.

3.2.2. Server-based Naming Authority

server = [[userinfo "@"] hostport]

userinfo = *(unreserved | escaped |
";" | ":" | "&" | "=" | "+" | "$" | ",")

hostport = host [":" port]

host = hostname | IPv4address

hostname = *(domainlabel ".") toplabel ["."]

domainlabel = alphanum | alphanum *(alphanum | "-") alphanum

toplabel = alpha | alpha *(alphanum | "-") alphanum

IPv4address = 1*digit "." 1*digit "." 1*digit "." 1*digit

port = *digit

3.3. Path Component

path = [abs_path | opaque_part]

path_segments = segment *("/" segment)
segment = *pchar *(";" param)
param = *pchar

"/", ";", "=", and "?" are reserved

pchar = unreserved | escaped |
":" | "@" | "&" | "=" | "+" | "$" | ","

3.4. Query Component

query = *uric

- Within a query component, the characters ";", "/", "?", ":", "@", "&", "=", "+", ",", and "$" are reserved.

4. URI References

- absolute or relative

  URI-reference = [absoluteURI | relativeURI] ["#" fragment]

5. Relative URI References

- necessary for the long-term usability of embedded URI

relativeURI = (net_path | abs_path | rel_path)
["?" query]

rel_path = rel_segment [abs_path]

rel_segment = 1*(unreserved | escaped |
";" | "@" | "&" | "=" | "+" | "$" | ",")

“. ” カレントレベル

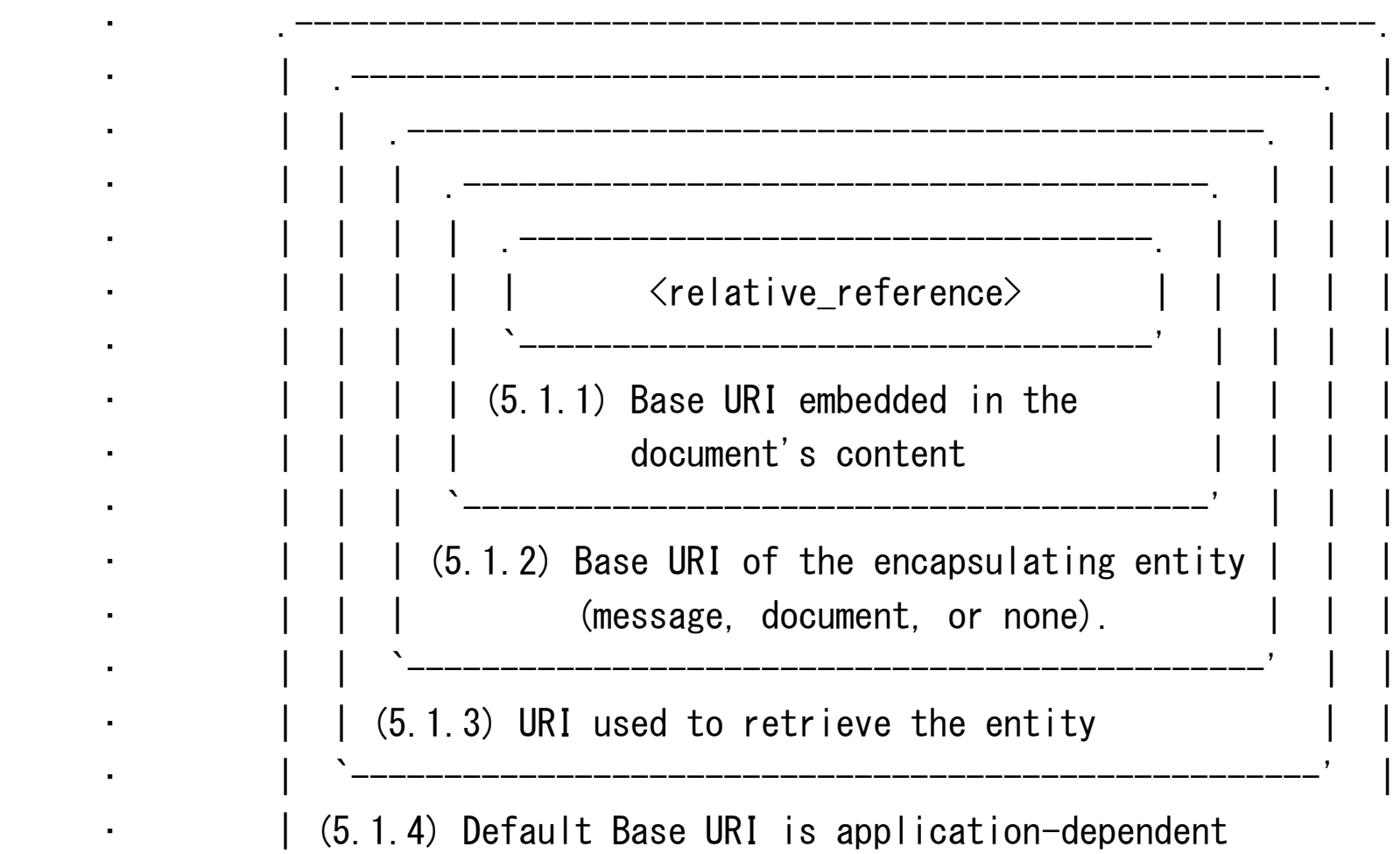
“. . ” 一つ上のレベル

a path segment which contains a colon character cannot be used as the first segment of a relative URI path (e.g., "this:that"), because it would be mistaken for a scheme name.

this:that　だめ

./this:that　OK

5.1. Establishing a Base URI



B. Parsing a URI Reference with a Regular Expression

```
^(([^:/?#]+):)?(//([^/?#]*))?([^?#]*)(¥?([^#]*))?(#(.*))?
 12            3  4          5       6  7        8 9
```

http://www.ics.uci.edu/pub/ietf/uri/#Related

$1 = http:
$2 = http
$3 = //www.ics.uci.edu
$4 = www.ics.uci.edu
$5 = /pub/ietf/uri/
$6 = <undefined>
$7 = <undefined>
$8 = #Related
$9 = Related

scheme = $2
authority = $4
path = $5
query = $7
fragment = $9

C. Examples of Resolving Relative URI References

- base URIがhttp://a/b/c/d;p?q　のとき、相対URIがどうなるか？

appendixC

D. Embedding the Base URI in HTML documents

```
<!doctype html public "-//IETF//DTD HTML//EN">
<HTML><HEAD>
<TITLE>An example HTML document</TITLE>
<BASE href="http://www.ics.uci.edu/Test/a/b/c">
</HEAD><BODY>
<A href="../x">a hypertext anchor</A> ...
</BODY></HTML>
```

1st Report

(Q1) Explain the importance of "relative forms" of URI.
In what situation the relative form is useful?

(Q2) What is "BNF-form"? That does "BN" stand for?

.

Hypertext Transfer Protocol (RFC2616)

The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. It is a generic, stateless, protocol which can be used for many tasks beyond its use for hypertext, such as name servers and distributed object management systems, through extension of its request methods, error codes and headers [47]. A feature of HTTP is the typing and negotiation of data representation, allowing systems to be built independently of the data being transferred.

HTTP has been in use by the World-Wide Web global information initiative since 1990. This specification defines the protocol referred to as "HTTP/1.1", and is an update to RFC 2068 [33].

3 Protocol Parameters
3.1 HTTP Version

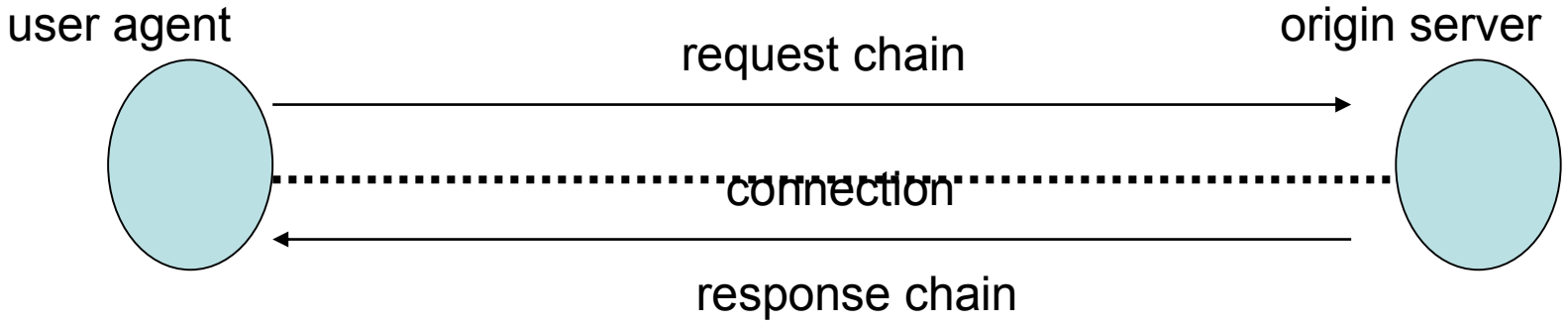
- <major>.<minor>
- 送信者側が、メッセージ形式や理解能力を示すもの。通信の特性を示すものではない。
- 番号の増加

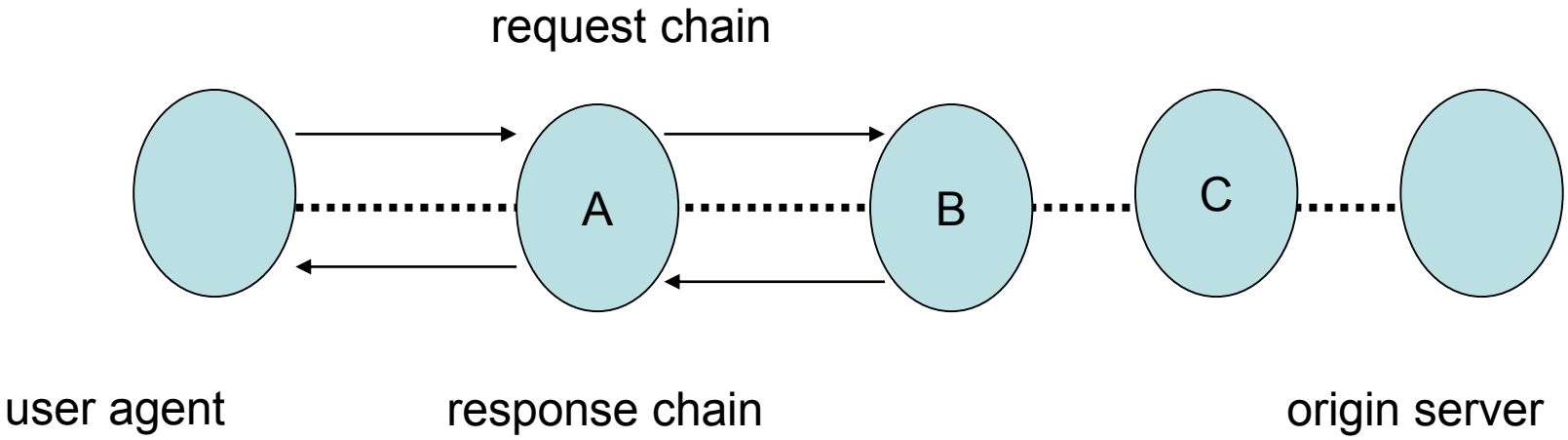
  <major>メッセージの形式が変更されたとき

  <minor>メッセージの意味への追加
- メッセージの最初のバージョンフィールドに表示

例）HTTP/2.4 ＜ HTTP/2.13 ＜HTTP/12.3

- proxyやgatewayが自分のバージョンより高いものを受けたときには、下げるか、エラーを返すか、トンネルしなければならない。





3.2.2 http URL

http_URL = "http:" "//" host [":" port]
[abs_path ["?" query]]

80

3.2.3 URI Comparison

- port番号が空、あるいは与えられていない
  ⇒デフォルト・ポート
- ホスト名の比較：大文字小文字無視
- スキーム名の比較：大文字小文字無視
- 絶対パスが空
  ⇒絶対パス “/” と同一視
- 例　下の3つは同じ

reserved,unsafe以外は
%HEX HEXと同じ

http://abc.com:80/~smith/home.html

http://ABC.com/%7Esmith/home.html

http://ABC.com:/%7esmith/home.html

3.3 Date/Time Formats

3つのフォーマット

Sun, 06 Nov 1994 08:49:37 GMT ..... preferred

Sunday, 06-Nov-94 08:49:37 GMT

Sun Nov 6 08:49:37 1994

Greenwich Mean Time

3.4 Character Sets

- 大文字小文字を区別しないtokenによって識別

  charset = token

- IANA Char Set registry (RFC1700)

  Internet Assigned Numbers Authority

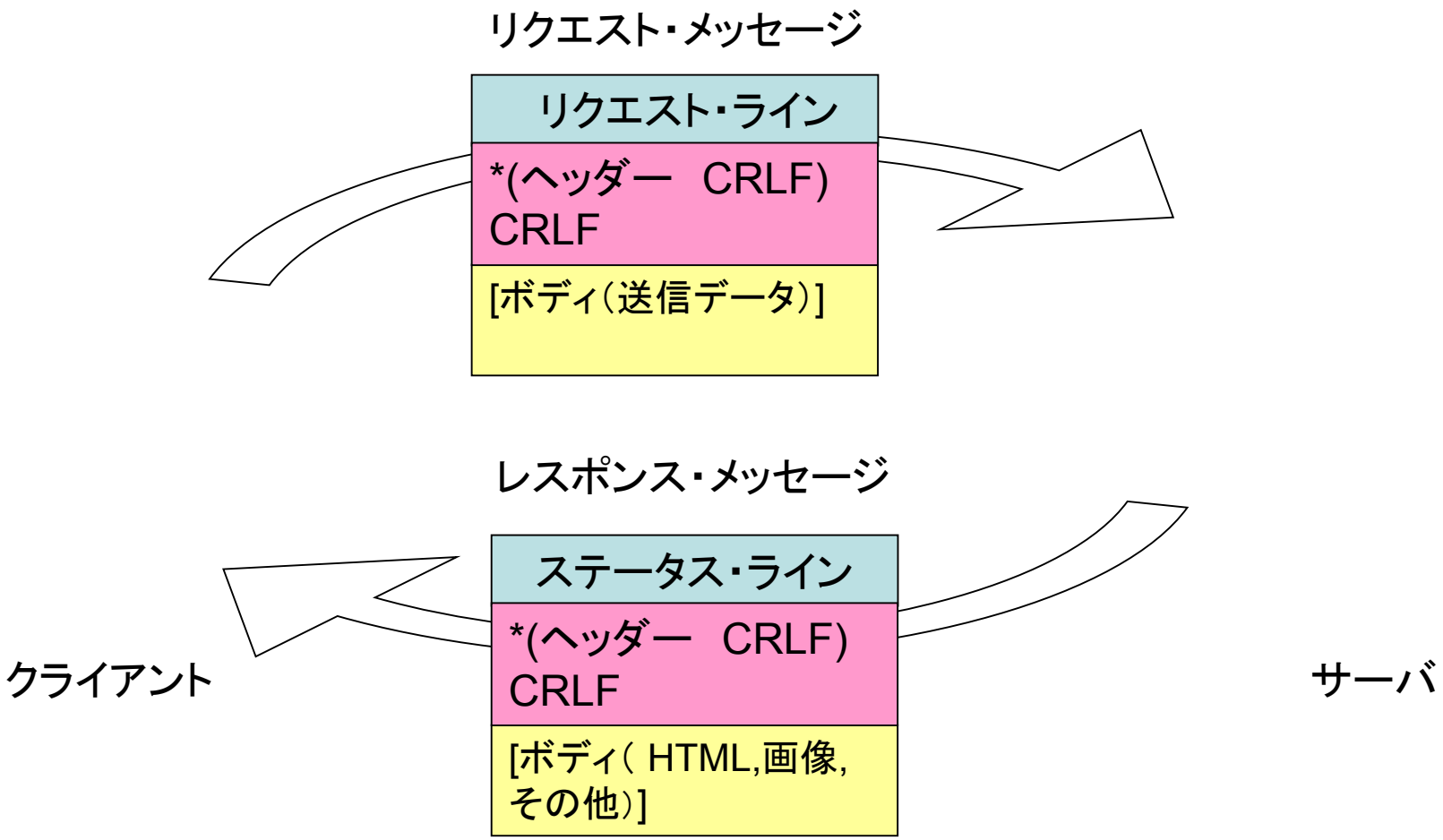
3.4.1 Missing Charset

- HTTP/1.0のソフトでcharsetパラメータに対応しないものもあるが、送信側のcharsetを尊重すべき

3.5 Content Codings

index

クライアント・サーバー間のメッセージ送受信を行うためのプロトコル



HTTP Message

HTTP-message = Request | Response

generic-message = start-line
*(message-header CRLF)
CRLF
[message-body]

start-line = Request-Line | Status-Line

message-header = field-name ":" [field-value]

続く行の先頭が空白かタブのときは複数行にまたがることができる

Request

```
Request  = Request-Line                ; Section 5.1
           *(( general-header          ; Section 4.5
            | request-header           ; Section 5.3
            | entity-header ) CRLF)    ; Section 7.1
           CRLF
           [ message-body ]            ; Section 4.3
```

Request-Line

Request-Line =

Method SP Request-URI SP HTTP-Version CRLF

Method =

- GET 指定したURIが示すリソースを取得、ボディにデータが含まれる
- HEAD 取得した結果レスポンスのヘッダーのみを取得
- POST 指定したサーバのコマンドに対し、データを転送。ボディにデータが含まれる
- OPTIONS 使用できるメソッドやオプションの一覧を取得

他にPUT、DELETE、TRACE、CONNECTなど

HTTP-Version “HTTP/1.1“, “HTTP/1.0”など

Request-line例

GET http://www.w3.org/pub/WWW/TheProject.html HTTP/1.1

GET /pub/WWW/TheProject.html HTTP/1.1
Host: www.w3.org

request-header

Accept 利用可能なアプリケーション・メディアタイプ
Accept-Encoding 利用可能なエンコーディング形式
Accept-Language 利用可能な言語コード
Authorization ログインに必要な認証情報
Expect
From利用ユーザーに固有なメールアドレス
Host リクエスト先サーバ名
If-Modified-Since Dateを指定する
if-Match 指定したエンティティタグに一致する場合のみ更新／取得
If-None-Match
If-Range
If-Unmodified-Since
Max-Forwards 経由できるプロキシの最大数
Proxy-Authorization プロキシにログインが必要な場合のための認証情報
Range 取得するデータのバイトレンジ。単位はバイト
Referer 直前にリンクされていたURL
TE 利用可能なエンコーディング形式（Transfer Coding方式）
User-Agent Webブラウザの固有情報

Response

```
Response      = Status-Line               ; Section 6.1
                *(( general-header        ; Section 4.5
                 | response-header        ; Section 6.2
                 | entity-header ) CRLF)  ; Section 7.1
                CRLF
                [ message-body ]          ; Section 7.2
```

Status-Line =

HTTP-Version SP Status-Code SP Reason-Phrase CRLF

Status-Code

1xx: Informational - Request received, continuing process

2xx: Success - The action was successfully received, understood, and accepted

3xx: Redirection - Further action must be taken in order to complete the request

4xx: Client Error - The request contains bad syntax or cannot be fulfilled

5xx: Server Error - The server failed to fulfill an apparently valid request

response-header

```
response-header = Accept-Ranges           ; Section 14.5
                | Age                     ; Section 14.6
                | ETag                    ; Section 14.19
                | Location                ; Section 14.30
                | Proxy-Authenticate      ; Section 14.33
                | Retry-After             ; Section 14.37
                | Server                  ; Section 14.38
                | Vary                    ; Section 14.44
                | WWW-Authenticate        ; Section 14.47
```

entity-header

```
entity-header  = Allow                    ; Section 14.7
               | Content-Encoding         ; Section 14.11
               | Content-Language         ; Section 14.12
               | Content-Length           ; Section 14.13
               | Content-Location         ; Section 14.14
               | Content-MD5              ; Section 14.15
               | Content-Range            ; Section 14.16
               | Content-Type             ; Section 14.17
               | Expires                  ; Section 14.21
               | Last-Modified            ; Section 14.29
```

telnet でHTTPサーバに接続

%telnet www.kyushu-u.ac.jp 80

Trying 133.5.1.2...

Connected to www.kyushu-u.ac.jp.

Escape character is '^]'.

HEAD / HTTP/1.0

HTTP/1.1 200 OK

Date: Mon, 05 Oct 2009 06:16:05 GMT

Server: Apache/2

X-Powered-By: PHP/4.3.9

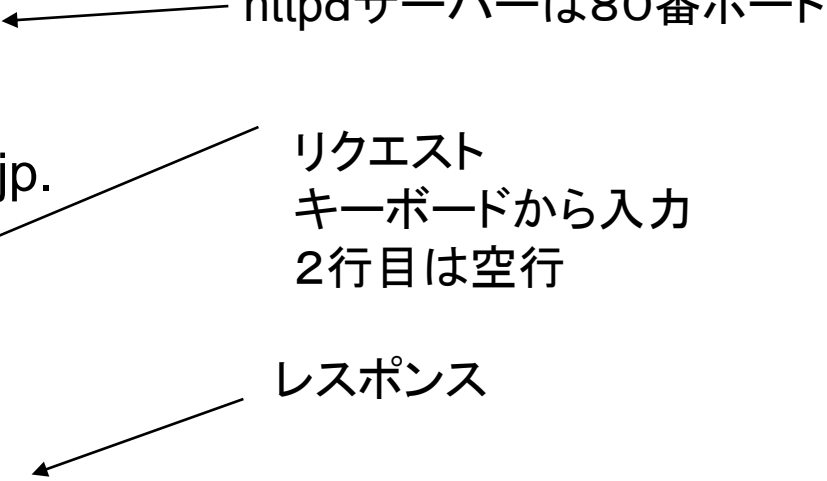
Cache-Control: no-cache

Pragma: no-cache

Expires: 0

Connection: close

Content-Type: text/html; charset=shift_jis



2nd Report

- Examine the name and version of HTTP server for 50 Web sites. Describe how you chose the sites.

  Hint:use “telnet” command and HTTP protocol.

Webロボット

Webロボットとは

- リンクをたどりWeb文書を自動的に収集するプログラム
- robot, crawler, spider, botということもある

- robotプログラムは一つの計算機上で稼動し他の複数の計算機にアクセスするが、プログラム自身が他のサイトに移って増殖するものではない。

検索エンジンと検索ロボット

- 検索エンジンの実施機関が独自にロボットを持っていることもあるが、他のロボットを利用しているとか、他のロボットが収集したデータを利用している場合もある。

（注１）浅井勇夫検索デスク

http://www.searchdesk.com/survey.htm

（注２）検索エンジンの相関表

http://www5d.biglobe.ne.jp/~hokugyo/search/h4110pp.htm

検索エンジン	検索ロボット	備考
Google	Google	
goo	inktome+goo	NTT
Fresheye	TOCC(inktome)	東芝
Alltheweb	FAST	
TOCC	inktome	三菱
NAVER	NAVER	韓国製
AAA!Cafe	?	和歌山大？
AltaVista	AltaVista	
Infoseek	Infoseek	

ロボット排除規約

- サイト管理者や情報提供者がロボットにどれだけ情報を提供するか規定する。
- 　ロボットプログラムが従わなければならない倫理的基準の　ガイドライン。
- http://www.robotstxt.org/wc/exclusion.html
- ２つの方法
  - robots.txt　　... サイト管理者
  - ロボットMATAタグ... ページ作成者

robots.txt

- http://.../robots.txtに記述
- サイト管理者がロボットが訪問できない場所を記述
- サイトに1つだけ。管理者しかできない。
- 1行に一つの指示

    エージェント指定または禁止対象指定（一つ）

A Standard for Robot Exclusion,Martijn Koster

http://www.robotstxt.org/wc/norobots.html

robots.txtの例

すべてのロボットにすべてのファイルを禁止

```
User-agent: *
Disallow: /
```

注意：正規表現が使える訳ではない

robots.txtを空としても同じ効果

すべてのロボットに特定の領域を禁止

```
User-agent: *
Disallow: /cgi-bin/
Disallow: /tmp/
Disallow: /private/
```

一つのロボットだけ禁止

```
User-agent: Badbot
Disallow: /
```

一つのロボットだけ許可

```
User-agent: WebCrawler
Disallow:
User-agent:  *
Disallow: /
```

ロボット・メタ・タグ

- Webページ作成者が、インデックス禁止、リンク解析禁止などをHTMLのMETAタグとして記述
- HTMLファイルのHEAD部分に記述

http://www.robotstxt.org/wc/exclusion.html#meta

ロボット・メタ・タグ記述例

```
<html>
<head>
<meta name="robots" content="noindex,nofollow">
<meta name="description" content="This
   page ....">
<title>...</title>
</head>
<body>
...
```

```
content    = all | none | directives
all        = "ALL"
none       = "NONE"
directives = directive ["," directives]
directive  = index | follow
index      = "INDEX" | "NOINDEX"
follow     = "FOLLOW" | "NOFOLLOW"
```

ロボット作成側が他に気を付けるべきこと

- User-Agentを指定
- Fromを指定
- Referrerを指定
- HEADを利用
- Acceptを指定
- 適切にアクセス
- 適度にアクセス

ページ提供側の2種類の要求

見られたくない

1. ロボット排除規約 … ロボットとの約束
2. アクセス制限…サーバーごとに異なる .htaccess

   httpサーバが管理するので、ロボットプログラムで 制御できない。

多くの人に伝えるため検索エンジンに登録してもらいたい

- 登録
- 訪問者数を上げる工夫（SEO Search Engine Optimization）... METAタグ

```
<meta http-equiv="Content-TYpe" Content="txt/html;charset=Shift_JIS">
<meta name="keywords" content="キーワード1,キーワード2">
<meta name="description" content="説明文">
```

- 無駄な例

```
<meta name="ROBOTS" content="INDEX,FOLLOW">
```

Quiz

- Google
  - How does Google make profits?
  - How do they make money?
- TV

```
Webロボット(最初に取得すべきURL){
変数 取得済URLリスト,未得済URLリスト;
     発見したURLリスト,次に取得すべきURLリスト;
     ファイルの内容,設定内容
  if(設定内容.アクセス許可(最初に取得すべきURL)){
  未取得リスト.add(最初に取得すべきURL);
  }
  while(未取得リスト.要素数>0){
    次に取得すべきURL = 未取得リスト.geturl();
    ファイル内容 = HTTP_GET(次に取得すべきURL);
    if (ファイル内容.metaタグよる制限){
      continue;
    }
    発見したURLリスト=URL抽出(ファイル内容);
    foreach(発見したURL in 発見したURLリスト){
      if (!((取得済URLリスト.isContained(発見したURL)||
           ((未取得リスト.isContained(発見したURL)))){
        if(設定内容.アクセス許可(発見したURL)){
          未取得リスト.add(発見したURL);
        }
      }
    }
  }
}
```

検索エンジン

- クローラー：Web文書収集
- 検索

  文書群の中から質問に適合する文書を見つけ出すこと

検索モデル

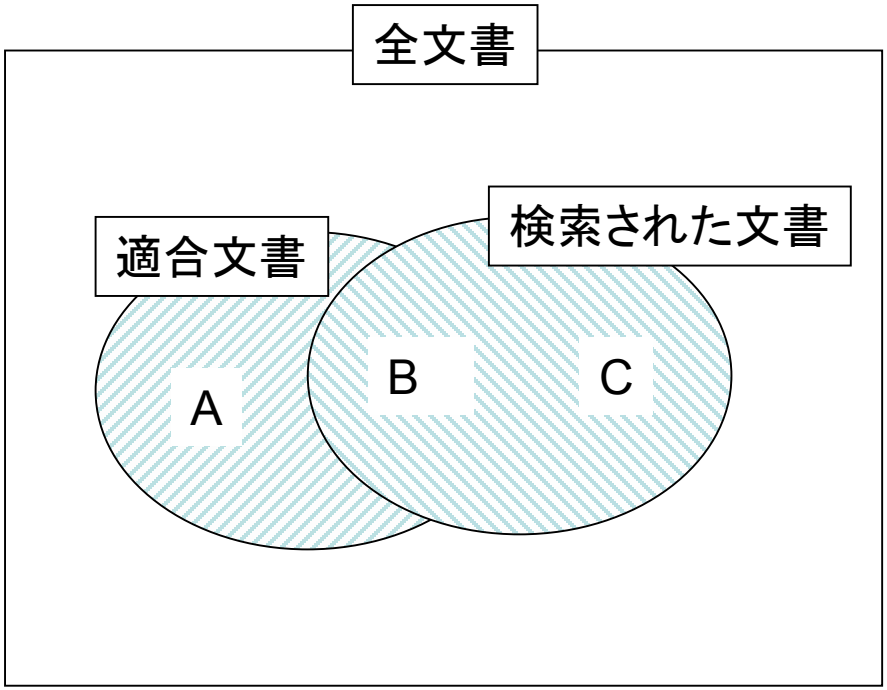
	文書の内部表現	質問の内部表現	文書と質問の照合
ベクトル空間モデル	索引語の重みベクトル	索引語の重みベクトル	ベクトルの類似度
ブーリアンモデル	特徴ベクトルと転置ファイル	特徴ベクトル、論理式	論理演算
全文検索			

検索システムの評価尺度

- 適合率(precision)P：
  検索で得られた文書の中で、検索質問に適合する文書の割合
  検索の正確さの評価
- 再現率(recall)R：
  検索対象となる文書全体で検索質問に適合する文書のうち、
  検索された文書の割合
  検索漏れの少なさに対する評価
- F尺度(F-measure)F：
  再現率R、適合率Pの調和平均（逆数の平均の逆数）
  F=1／(1/R+1/P)

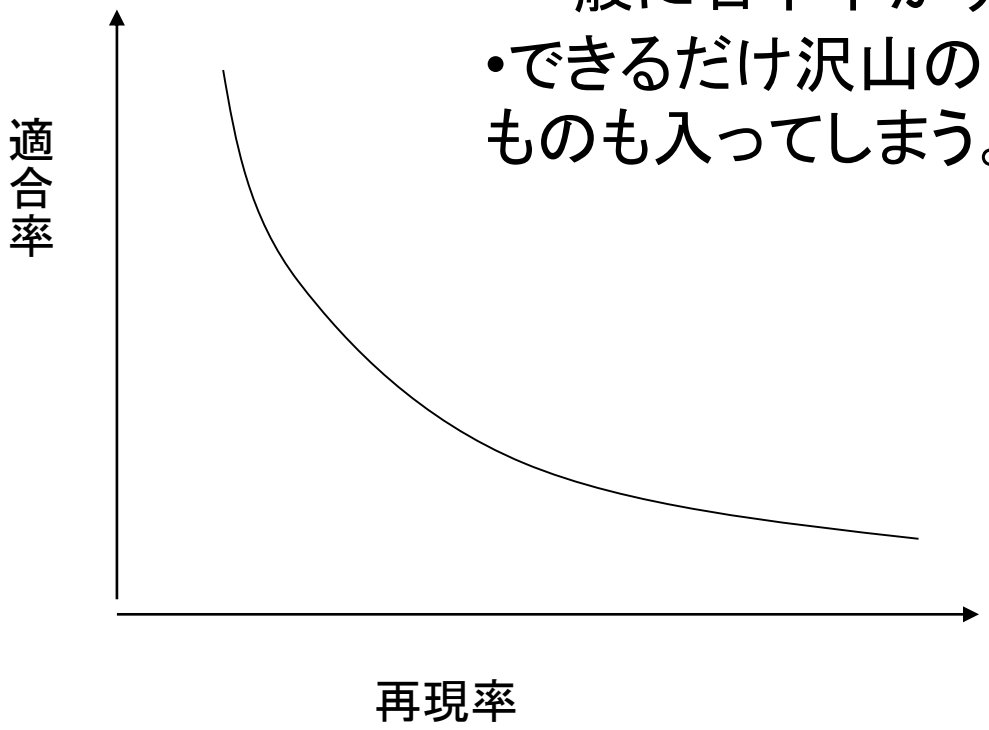
適合率、再現率

- 適合率(precision)　P = B/（B+C)
- 再現率(recall)　　R = B/（B+A）



再現率･適合率曲線

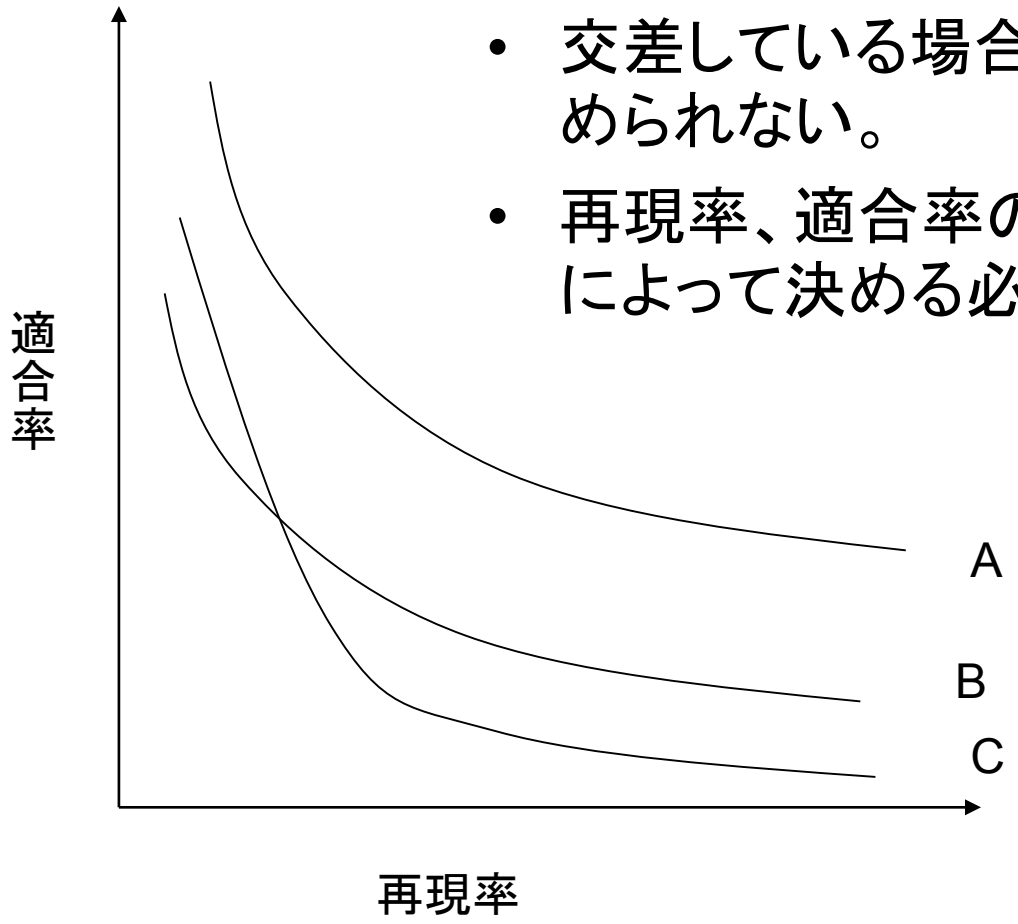
質問Qiに対する再現率Ri、適合率Piを(Ri,Pi)をプロットしたグラフ



- 一般に右下下がり
- できるだけ沢山の文書を返そうとすると誤ったものも入ってしまう。

再現率-適合率曲線

- 2つの検索システムについては、グラフが上にある方が優れている
- 交差している場合には、一概に優劣は決められない。
- 再現率、適合率のどちらを重要視するかによって決める必要がある。



ベクトル空間モデル

- 文書をその文書に現れるキーワード群で特徴づけるキーワードのベクトルとして表す

索引語 w1,w2,...,wm

1 Bioinformatics

2 Biology

3 Chemistry

4 Enzymes

5 Evolution

6 Genes

7 Genomes

8 Proteins

文書D1,D2,...,Dn

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

Term*Document Matrix

Term*Document Matrix n行m列
d(i,j) 単語wiの文書Dj中の出現回数
行:ターム
列:ドキュメント

Query :Genes and Genomes
w6, w7 -> (00000110)^t

類似度計算
文書も質問のn次元ベクトルとして表現し
類似度はコサイン
sim(dj,q)= dj*q/|dj|*|q|
= (sum dij*qi)/(sum dij*dij)(sum qi*qi)

課題:この質問に対する各ドキュメントの類似度を求めよ

	ドキュメント					
ターム	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

索引語

- 予め人手で決める方式
- 文書群から抽出する方式
  - キーワードの重要性を求め、重要なものだけを索引語とする。
  - キーワードの重み

    ブーリアンモデル：1,0

    ベクトル・モデル：0～1の間の実数

ベクトル・モデル

- TF Term Frequency
- DF Document Frequency
- IDF Inverse Document Frequency

単語の出現頻度
（TF　Term Frequency）

ターム　　t1,...,tm

ドキュメント D1,...,Dn

tf(i,j)

ドキュメントDj中におけるタームtiの出現頻度

「高い頻度で出現する単語はその文書に特徴的」と考える。

単語の出現頻度

出現頻度だけでは不十分

日本語...「は」という助詞はどんな文書でも高い頻度で出現するが、それぞれの文書の特徴ではない

英語　... the,is

単語の出現頻度

- 単語の出現頻度についての「Zipfの法則」
- 不要語リスト stop list

  SMARTシステム

  ftp://ftp.cs.cornel.edu/pub/smart/englis.stop
- ステミング stemming 単語の語幹への変換

  retrieves,retrieved,retrieving,retrieval-->retrieve

  Porter algorithm

  http://www.cs.jhu.edu/~weiss/stem.c

tf*idf

- df(i) ＝単語tiを含む文書の総数
- idf(i) : df(i)の逆数

  正規化　log(文書総数/df(i))+1

- tf*idf : i番目の単語の重要性

  w(i) = tf(i)*idf(i)

  dfが小さくtfが大きいときに大きい。

  その単語を含む文書が稀であり、その稀な文書中で頻繁に現れる。

ブーリアンモデル、転置ファイル法

- 転置ファイル法 inverted file indexing
  索引語⇒出現する文書番号ベクトル
- Query :Genes and Genomes

  Genes　w6 ->1 1 1 1 0 0

  Genomes w7 ->0 0 1 1 1 0

  論理積　　0 0 1 1 0 0 ... D3,D4

  D3 Adaptive Evolution of Genes and Genomes

  D4 Advances in Genome Biology:Genes and Genomes

Nグラム索引

- 特徴ベクトルの生成の問題点

  索引語の決定

  形態素解析、キーワード抽出、ステミング
- Nグラム索引
  - 予め索引を決めておくのではなく、Nを固定して任意のNグラムを索引語と考える。
  - 索引語が出現する文書番号と文書内での出現位置
  - ユニグラム、バイグラム、トライグラム

例:ユニグラム索引と検索

0 2 4 6 8 10121416182022 24

D:にわにはにわにわとりがいる

に	0,4,8,12
わ	2,10,14
は	6
と	16
り	18
が	20
い	22
る	24

検索語:「にわとり」
...「に」「わ」「と」「り」
に ⇒0,4,8,12
わ ⇒2,10,14
と ⇒16
り ⇒18

例:バイグラム索引と検索

0 2 4 6 8 10121416182022 24

D:にわにはにわにわとりがいる

にわ	0,8,12
わに	1,10
はに	4,6
わと	14
とり	16
りが	18
がい	20
いる	22

検索語:「にわとり」

…「にわ」「とり」

にわ ⇒0,8,12

とり ⇒16

Webマイニング
Webからの情報発見、収集、統合

- Web上のデータ… HTMLファイル
- 価値のある情報
  - 構造と表示の区別
  - HTMLファイルから構造情報とコンテンツの分離
- 大量にあることへの対応
- 多様性への対応

HTMLファイルの分析例:ラッパー

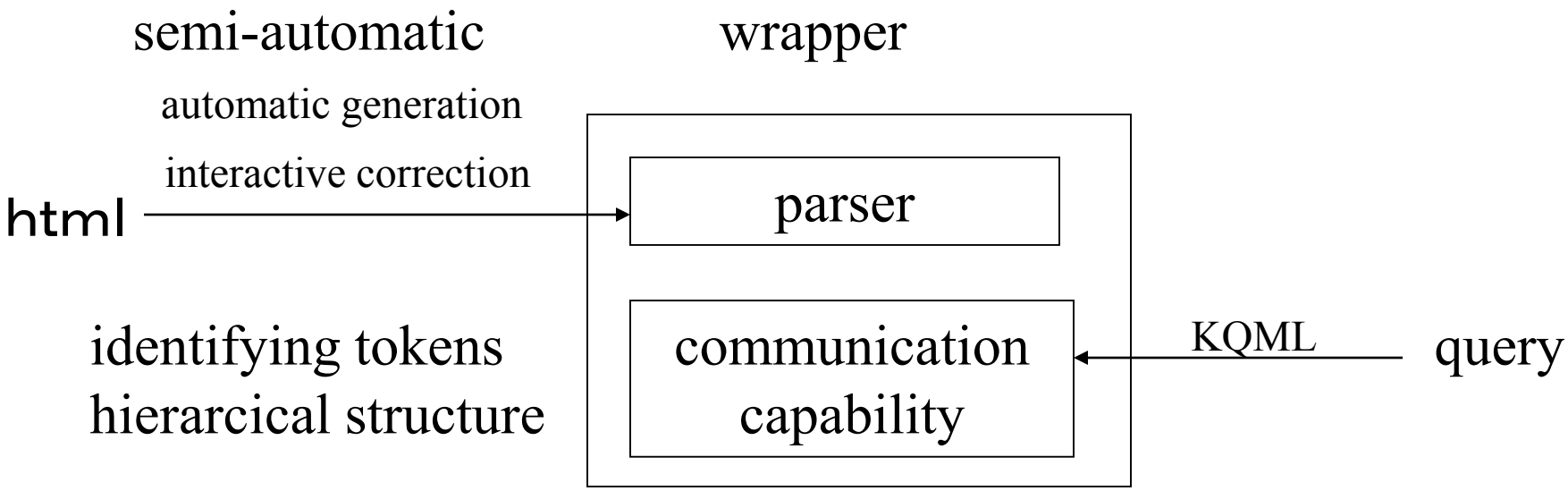
N. Ashish and C. Knoblock,
Wrapper Generation for Semi-structured Internet Sources,
ACM SIGMOD Record 26 (4), 8--15, 1997.

N. Kushmerick, D. Weld and R. Doorenbos,
Wrapper induction for information extraction,
IJCAI'97, 729--737, 1997.

P. Atzeni, G. Mecca,
Cut and Paste,
Proceedings of 16th ACM SIGMOD Symposion on
Principles of Database Systems, 144--153, 1997.

D. Embley, S. Jiang, Y.-K. Ng,
Record-boundary discovery in Web documents,
Proceedings of 1999 ACM SIGMOD International Conference on
Management of Data, 467--478, 1999.

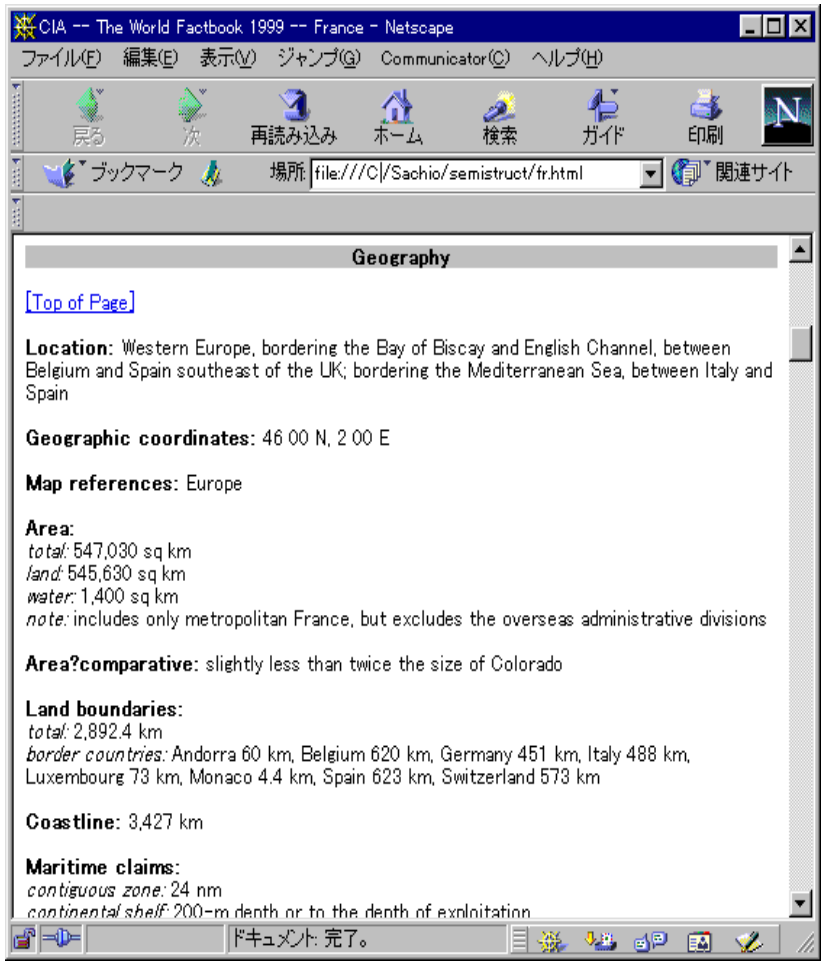
N. Ashish and C. Knoblock,
Wrapper Generation for Semi-structured Internet Sources,
ACM SIGMOD Record 26 (4), 8--15, 1997.



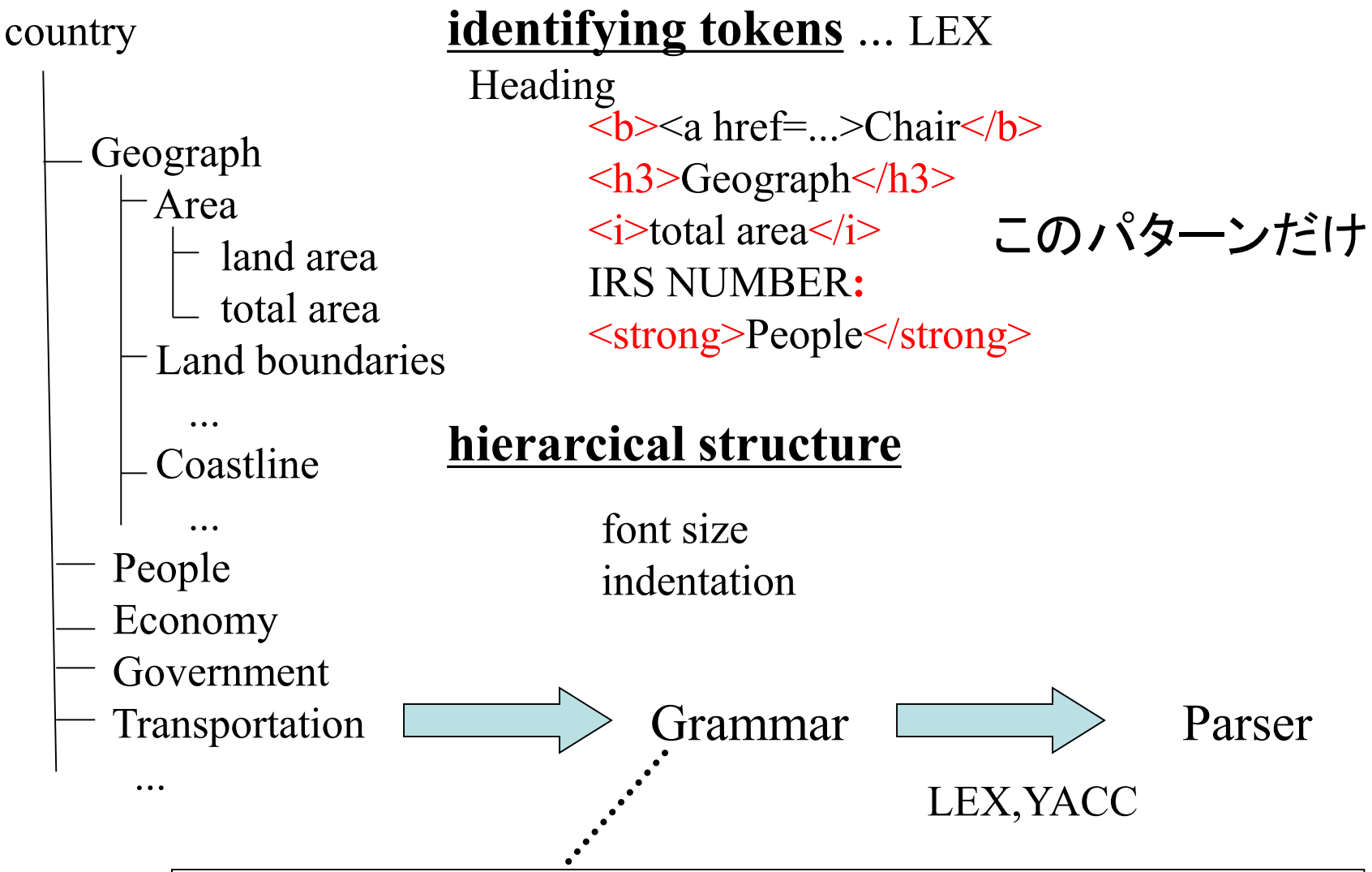
•Find the Land_boundaries and Area of France.
•Find the national_Product and Defense_Expenditures of all countries in Europe.

CIA World Fact Book

www.odci.gov/cia/publications/factbook/fr.html




```
<h3>France</h3>
<b>Location:</b>
Western Europe, bordering the Bay of ・・・<p>
<b>Geographic coordinates:</b> 46 00 N, 2 00 E <p>
<b>Map references:</b> Europe <p>
<b>Area:</b>
  <br><i>total:</i> 47,030 sq km
  <br><i>land:</i>545,630 sq km
  <br><i>water:</i>1,400 sq km
  <br><i>note:</i>
     includes only metropolitan France, ... <p>
<b>Area&amp;#151;comparative:</b>
  slightly less than twice the size of Colorado<p>
<b>Land boundaries:</b>
  <br><i>total:</i>2,892.4 km
  <br><i>border countries:</i>
    Andorra 60 km, Belgium 620 km, ...
```


country
Geograph
Area
land area
total area
Land boundaries
...
Coastline
...
People
Economy
Government
Transportation
...
identifying tokens ... LEX
Heading
<a href=...>Chair
<h3>Geograph</h3>
total area
IRS NUMBER:
People
このパターンだけ
hierarcical structure
font size
indentation
Grammar
Parser
LEX,YACC
CIApage -> Geograph People Government Economy Transportation
Geograph -> Location Map_references Area Land_boundaries coastline ...
Area -> total_area comparative_area

N. Kushmerick, D. Weld and R. Doorenbos,
Wrapper induction for information extraction,
IJCAI'97, 729--737, 1997.

relational data
  telelphone directoreis
  product catalog
  tabular layout

対象が限られている

HLRT
  a wrapper class
  finite-state automata
PAC larning

wrapper
  input: a page
  output: the set of tuples in the page

Congo	242
Egypt	20
Belize	501
Spain	34

```
<HTML><TITEL>Some Country Codes</TITLE>
<BODY><B>Some Country Codes</B><P>
<B>Congo </B><I>242</I><BR>
<B>Egypt </B><I> 20</I><BR>
<B>Belize</B><I>501</I><BR>
<B>Spain </B><I> 34</I><BR>
<HR><B>End</B></BODY></HTML>
```

wrapper

```
ExtractCCs(page P)
skip past fist occurence of <P> in P
while next <B> is before next <HR> in P
  for each (li,ri) in {(<B>,</B>), (<I>,</I>)}
    skip past next occurence of li in P
    extract attribute from P to next occurence of ri
return extracted tuples
```

Head

Tail

Left

Right

H L_1 R_1… L_k R_k T

国名 番号

HLRTで決まるWrapper

```
ExecuteHLRT((h,l1,r1,...,lk,rk,t),page P)
skip past first occurence of h in P
while next l1 is before next t in P
  foreach i (li,ri) in {(l1,r1),...,(lk,rk)}
    skip past next occurence of li in P
    extract attribute from P to next occurrence of ri
return extracted tuples
```

wrapper construction problem

(instance_1,lable_1)
instance:htmlファイル
label:そこに含まれる組
{(congo,242),(ggypt,20),(belize,501),(spain,34)}
(instance_2,lable_2)
(instance_3,lable_3)
(instance_3,lable_3)
...
学習目的
(h,(l1,r1),...,(lk,rk),t)

How many examples must a learner see to be confident that its hypothesis is good enough?

P. Atzeni, G. Mecca,
Cut and Paste,
Proceedings of 16th ACM SIGMOD Symposion on
Principles of Database Systems, 144--153, 1997.

Editor : a language for manupulating semi-structured documents

search : select regions

cut&paset: restructure

complete:

computable document restructuring can be expressed in Editor

subclass:

polynomial-time restructurings

ARANEUS project

database views over Web sites

```
search(HTMLPage,"<TITLE>*</TITLE>");
copy(HTMLPage);
paste(Title);
```

ドキュメントHTMLPageの中のタイトルのパターンを探し、
クリップボードへコピーし、
それを別のドキュメントTitleへ書き込む

```
search(HTMLPage,"<TITLE>*</TITLE>");
cut(HTMLPage);
replace(HTMLPage,e,"<TITLE>My Title</TITLE>");
```

ドキュメントHTMLPageの中のタイトルのパターンを探し、
クリップボードへカットし、
その場所で、e(空文字列)をMy Titleで置き換える

繰り返し

```
loop search(D,Pat)
  Body
end loop
```

```
loop search(HTMLPage,"<H1>*</H1>")
  copy(HTMLPage);
  paste(ToC);
end loop
```

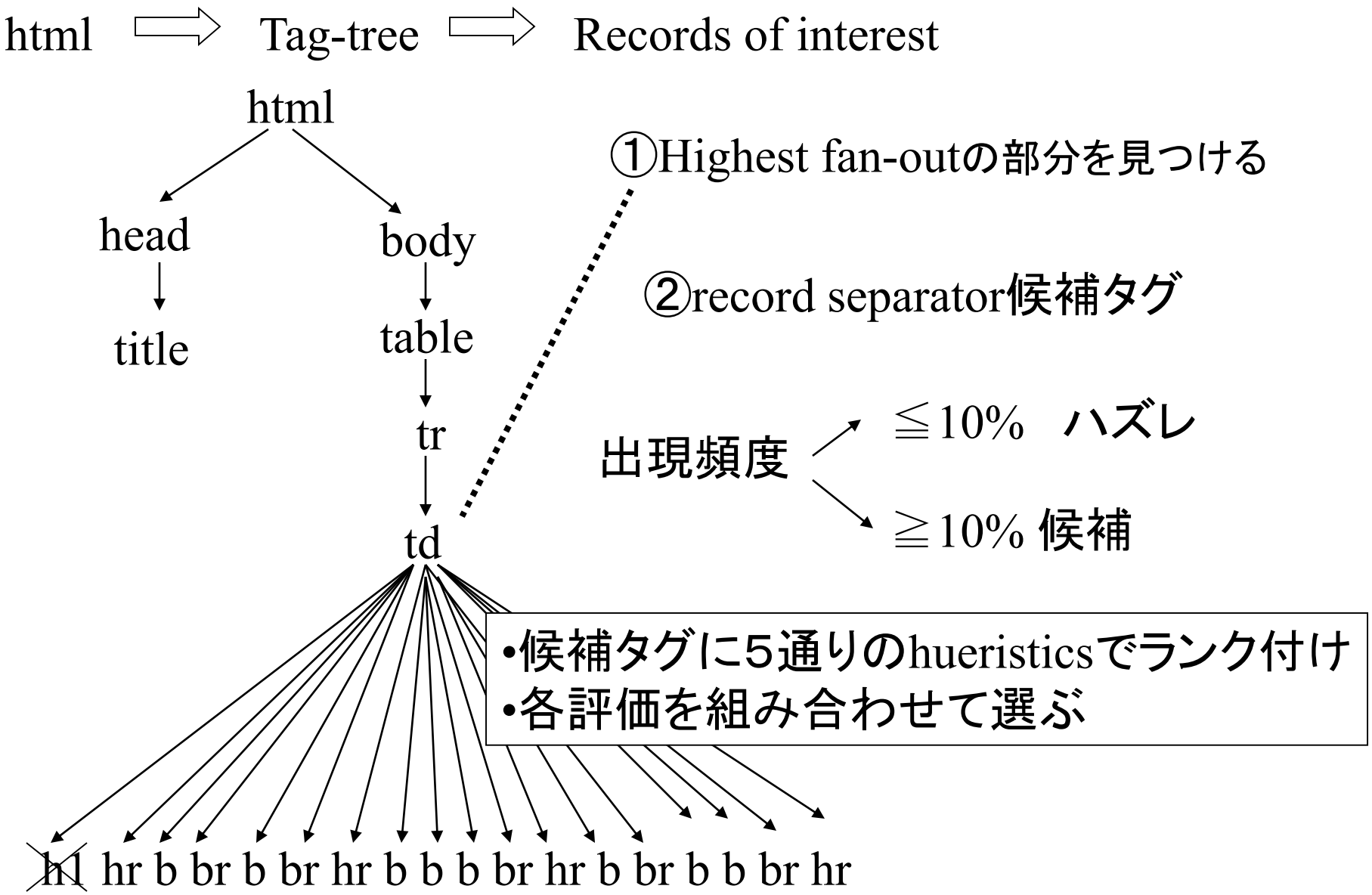
ドキュメントHTMLPageに現れるヘッダーを、
ToCに並べる

Title	Author
[A Boy Lighting a Candle with a Brand	Domenico Theotokopulus Called El Greco]
[A Group of Angels	Annibale carracci]
[A Group of Armigers	Michelangelo Buonarroti]
[A Landscape with the Good Samaritan	Henry met de Bles Ccalled Civetta]
[A Market with a Seller of Flowers	Arnout de Myuser]
...	...
[Bacchus	Annibale Carracci]
[Cebetis Thebani Tabula	Jan Sons]
...	...
[Wrath	Jacques de Backer]

```
replace(Table,e,"Title ¥t Author ¥n");
replace(Table,e,"----- ¥t ----- ¥n");
loop search (PaintList,"<LI>*");
    copy(PaintList);
    paste(Temp);
    replace(Table,e,"[");
    search(Temp,"<LI>*");
    cut(Temp);
    search(Temp,"*</A>");
    cut(Temp);
    paste(Table);
    replace(Table,"</A>","¥t");
    search(Temp,"*)");
    cut(Temp);
    paste(Table);
    replace(Table,"(",e);
    replace(Table,")","]¥n");
end loop;
```

D. Embley, S. Jiang, Y.-K. Ng,
Record-boundary discovery in Web documents,
Proceedings of 1999 ACM SIGMOD International Conference on
Management of Data, 467--478, 1999.

```
<html><head><title>Classifiedd</title></head>
<body ..>
<table<tr><td>
<h1 ..>Funeral Notices </h1>October 1,1998
<hr>
<b>Lemar K.Adamson</b><br>died on March 20,1998. ...
church...<b>BRING'S MEMORIAL CHAPEL</b>,...<br>
<hr>
Our beloved <b>Brian Fielding Frost</b>,age 41, passed ...
...
held at .. in the <b>Hover Stake Center</b>
<b>Wasatch Lawn Mortuary</b>
Wasatch Lawn Memorial Park.<br>
<hr>
<b>Leonard Kenneth Gunther</b><br>passed away on March 19,1998....
...at <b>HEATHER MORTUARY</b>,...
...at 11:00 a.m. at <b>HEATHER MORTUARY</b> on Thursday, March 19, 1998..<br>
<hr>
</td></tr></table>
All mateiral is copy righted.
</body>
</html>
```


html ⇨ Tag-tree ⇨ Records of interest
html
head
body
title
table
tr
td
①Highest fan-outの部分を見つける
②record separator候補タグ
出現頻度
≦10% ハズレ
≧10% 候補
•候補タグに5通りのhueristicsでランク付け
•各評価を組み合わせて選ぶ
h1 hr b br b br hr b b b br hr b br b b br hr

Heuristics

- HT (highest-count tags) 出現頻度でソート
- IT (identifiable “separator” tags) らしい

  hr,td,tr,a,table,p,br,,h4,h1,strong,b,i
- SD (standard deviation) 標準偏差

  各タグについて出現間隔の分布を考え、標準偏差が最小のタグを選ぶ

  根拠：複数回現れていて、同じ長さのレコードのはず
- RP (repeating-tag pattern)

  record boundaryは<x>|<y>の形

  その時には #<x> ≒#<x><y>のはず。そこで、

  ｜#<x><y> ー#<x>｜＋ ｜#<x><y> ー#<y>｜を各

  <x>,<y>について計算して最小のものを求める
- OM(ontology-matching)

  1つのレコード中に1回だけ現れるfieldがあるはず

  obituaryの例では、death dat,I.e., “died on”, “passed away”

  そんな record idenfying fieldの出現回数を求め、その回数

  と同じ出現回数のタグがseparator

SD (standard deviation) 標準偏差

タグ	間隔	回数
hr	5	3
b	1	3
	2	2
	3	3
br	2	1
	3	2
	4	0
	5	1

散らばっていない
これだ！

Combined Heuristics

- 初期実験 HT,IT,SD,RP,OMの正解率を求める
- Stanford Certianty Theory
  X,Yの正解率 C(X),C(Y)のとき
  combined ceitainty
    C(X)+C(Y)ー C(X)*C(Y)

OM hr:1,br:2,b:3
RP hr:1,br:2,b:3
SD hr:1,br:2,b:3
IT hr:1,br:2,b:3
HT b:1,br:1,hr:3

疑問：５通りも考えて頑張っているが、
ITだけでほとんど当たっている？

課題：whizbang

- どんな会社か
- 誰が設立したか、どんな関係者がいるか？
- その後、どうなっているか？どうなったか？