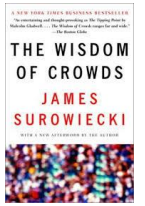
Webと集合知

集合知、群衆の英知、Web 2.0

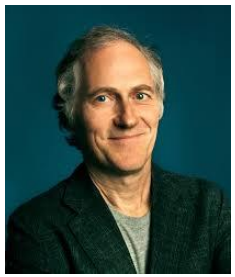
集合知

- **集合知**とは何か？　　中国語では集体智慧？
  - 2004年に**スロウィッキー**（James Michael Surowiecki）が **”The Wisdom of Crowds”**(群衆の英知)という著作を発表
    - 当初は集団的知性（Collective Inteligence）



ジェームス・ミカエル・
スロウィッキー
1967-

  - 2005年に**オライリー**（Tim O’Reilly）が **Web 2.0** という新しい概念を提唱
    - Webが従来のWeb（Web 1.0）と異なる7つの原則のうちの1つにスロウィッキーが提唱した集合知を活用

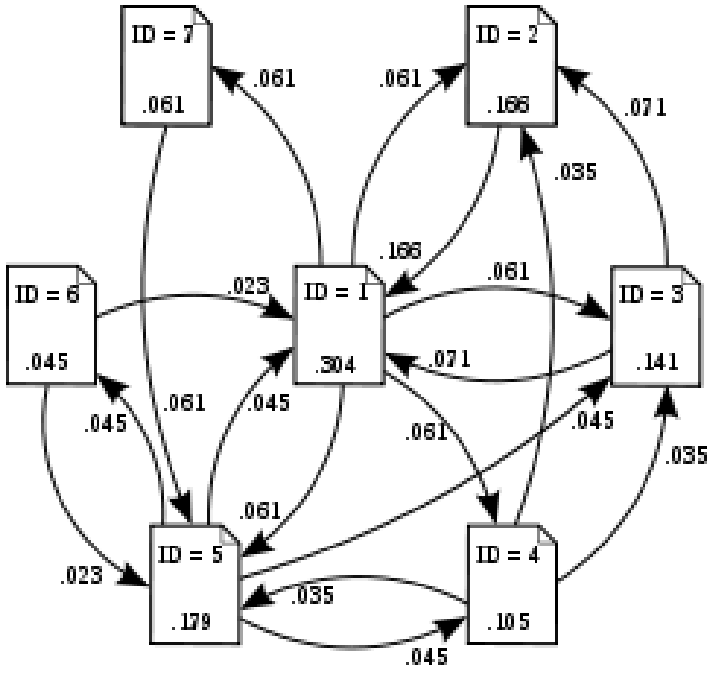


ティム・オライリー
1954-

２人の書籍・論文の影響は大きく、Web社会に大きなインパクトを与えた

集合知の典型的事例

- PageRank（Googleのランキングアルゴリズム）
  - リンクがより多く集まっているWebページはより重要である



スロウィッキーとオライリーの両者とも集合知の典型的事例としてPageRankをあげている

集合知とソーシャルコンピューティング

- 集合知とソーシャルコンピューティング
  - 情報科学の分野では、2000年代中頃から、集合知という用語が取りざたされてきた
  - **ソーシャルコンピューティング**の理論的背景となっている
    - 産業界ではIBM, HPがプロジェクトを立ち上げ
    - 学会では2009年にIEEE-CS（米国電気電子技術者協会-コンピュータ部会）が第１回ソーシャルコンピューティング国際会議（SocialCom-09）を開催

The Wisdom of Crowds（群衆の英知）

- スロウィッキーのいう群衆の英知
  - 一定の条件が整うと群衆は賢い判断を下せる
  - スロウィッキーは、群衆の英知の証としていくつかの事例を示した

(事例1) ゴールトンの実験

イギリス人学者ゴールトン（Francis Galton）の実験（1906年）

- 実験内容
  - 家畜見本市に出された雄牛の重量を群集が予測する
  - 実験の目的
    - 非常に優秀な人が少し、汎用な人がもう少し、それに多数の愚民の判断が混ざってしまうと、結論は愚かなものとなるだろうと予測　➔　的はずれな結果にる

- 結果
  - 多彩な群衆800人が予測した重量の平均値は、実際の値にほぼ近かった（群衆は賢かった）
    - 予測値平均：1197ポンド、実際の重量：1198ポンド

(事例2) チャレンジャー号の事故

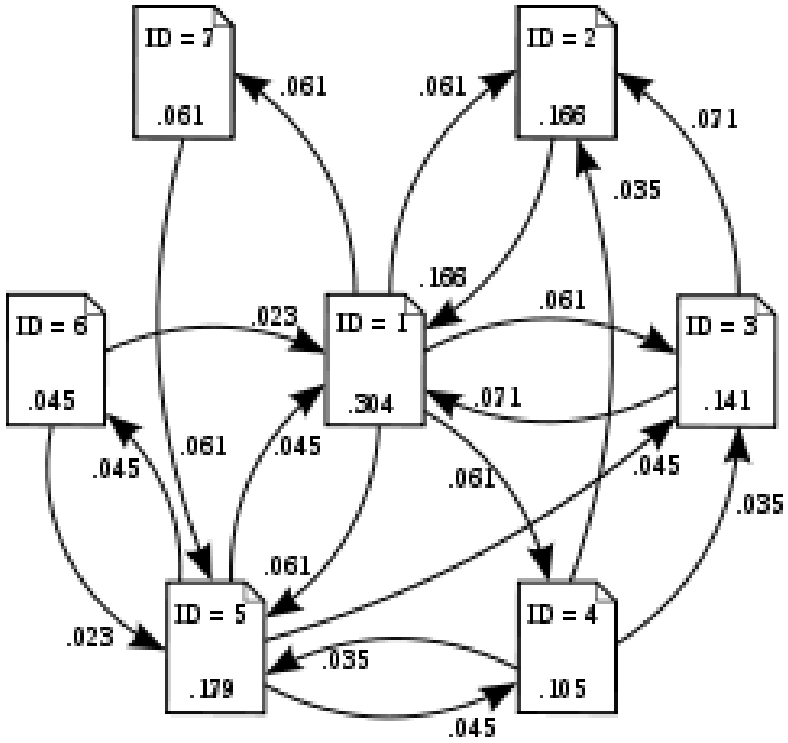
- スペースシャトル・チャレンジャー号の事故（1983年1月28日）
  - 打ち上げ73秒後に爆発し、乗組員7名全員が犠牲となった事故

- ニュース報道後の株価の変動
  - 事故の8分後の最初の報道から数分もしないうちにチャレンジャー号の発車に関わった主要企業4社の株の投売りが始まった

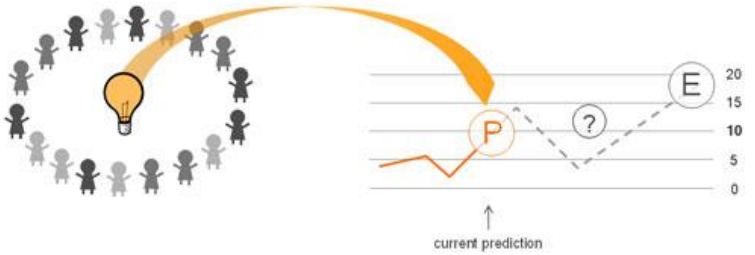
    - Rockwell International 社（シャトルとメインエンジン）
      - 下落幅：6%
    - Lockheed 社（地上支援）
      - 下落幅：5%
    - Martin Marieta 社（シャトルの外部燃料タンク）
      - 下落幅：3%
    - Morton Thiokol 社（固体燃料ブースター）
      - 下落幅：大きすぎてすぐに取引停止　➔　再開後の終値では12%下落
  - あらゆる検証をしても、事故の当日、Thikol社に責任があるというコメントは１つとして公開されていなかった
  - 爆発の6ヶ月後、チャレンジャー号大統領調査委員会は、原因はThiokol社にあったことを明らかにした
  - つまり、事故当日の株式市場は賢かった

(事例3) Google のランキングアルゴリズム(PageRank)

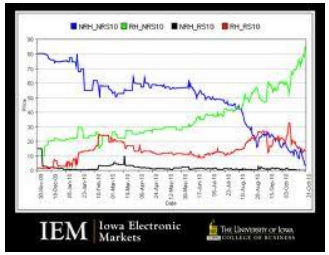
- 詳細は後ほど紹介



(事例4) 予測市場



- 予測市場
  - 将来予測をするための市場
    - 例えば、選挙で誰が当選するかを事前に予測して、それを商品とする市場
  - 有名な例に、アイオワ大学のIEM（Iowa Electronic Market）プロジェクト
    - このプロジェクトでは、あらゆる選挙結果を予測する市場を設けている
      - 選挙の当選者を予測する市場
      - 特定の候補者の最終得票率を予測する市場
    - 市場に参加する群衆の判断が、一部の権威ある調査機関の予測結果より優れていることを実証



(事例5) 株式市場

- 株式市場
  - 投資家が相互依存しながら、株価が「集合知」として機能する
    - 株で儲けたい投資家は、平均的な投資家がどう考えているかを予想して投資する。これが永遠と続く。

    - ただし、予測市場と違って結果に正解はない（現在の株価が適切かどうかは誰も証明できない）

  - みんなの意見が間違う可能性も同時に示唆
    - 株式バブル、大暴落

集合知の実験をしてみましょう！

（実験１）何個入っているでしょう？

<u>○ 個</u>

群衆が賢くあるための条件

- 集合知が適切に機能する条件
  - スロウィッキーは、集合知が適切に機能している事例に共通する性質として以下の4つを挙げている

1. 多様性 
2. 独立性 
3. 分散性 
4. 集約性 

１．多様性（diversity）

- 各参加者がそれぞれに独自の視点を持っていれば、総体として多くの候補解を列挙することができる
- 探索空間が狭い場合には、その探索空間内に適切な解が存在しない可能性がある

- 一番ふさわしい人をみつけることが良いというわけではない
  （例：最良なコンサルタント、最良なCEO ）
- 全体として多様な意見を持った参加者がいる方が多くの候補解を列挙できる

２．独立性（independence）

- 各参加者の持つ意見や提案が他の参加者の影響を受けないよう、各参加者の独立性が確保されている必要がある
- とくに小集団で議論を行う場合には、多様性が低いために偏った結論に集約される危険性がある

- 自ら考えることなく行う模倣は、時として問題を起こす可能性もある
  （例：バブル、情報カスケード）
- 個人が独立して意見を述べることことで、これらの問題を回避できる

３．分散性（decentralization）

- 問題を抽象化せず、各参加者が直接得られる情報に基づいて判断する必要がある
- 参加者ごとに得られる情報の種類は異なると予想されるが、多様性を維持するためにも、各参加者に共通する属性のみで判断すべきでない

- トップダウン的なアプローチよりも、分散したアプローチが優れている可能性もある（例：Linuxの開発、Open Source）

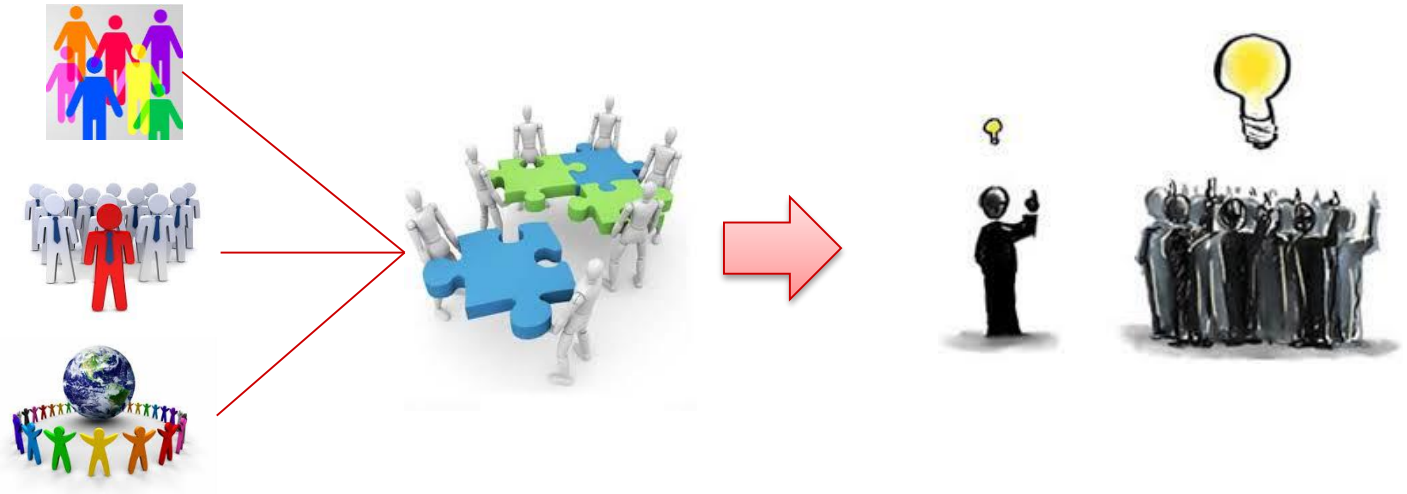
４．集約性（aggregation）

- 3点（多様性、独立性、分散性）の特性を生かして得られた知識を参加者全体で共有し、比較検討して最終的な結論を導く仕組みが必要である

- たさし、共通する原理は同じだとしても、現れてくる集約のメカニズムは事例ごとに異なってくる場合もある
  - 統計的データ
    - たくさん集めて集計してみると意外と当たっている（平均値、加重平均）
  - 複数人の知恵の集合
    - とても優秀な１人よりも、それなりに優秀な何人かの集合が複数の視点を上手に使って取り組む方が高い成果を生（Open Source）
    - 自分が知らないことも聞いてみれば誰かが答えてくれる（Q&Aコミュニティ）

集合知の実現

- 集合知（≠量）
  - 群衆が多ければよいというわけではない
  - 集合知の実現には、他の影響を受けない状態でのローカルな知識の生成メカニズムと、それらを集約するメカニズムの両方が必要

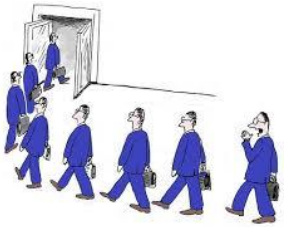


（補足）集団ゆえの危険性

- 集団思考・集団浅考（groupthink、グループシンク）
  - 集団で議論を行う場合に不合理あるいは危険な意思決定が容認されること、あるいはそれにつながる意思決定パターン

  - リスキーシフト
    - グループの意志が、より過激な方向にいってしまう傾向
    - （例）
      - 他のグループに対する過激な攻撃や敵意に発展する
      - リーダーへの絶対服従が起こり、反対意見は封殺される
      - メンバーは、より自分をアピールするために過激な意見を述べるようになる

  - コーシャスシフト
    - グループの意志が、何もしない現状維持的な方向にいってしまうこと
    - （例）
      - 新しいアイデアは採用されない
      - 「まあいいじゃないか、これまでうまくいってきたんだから」という雰囲気

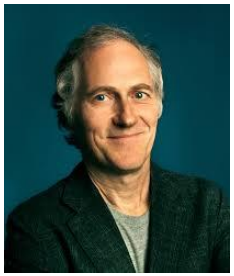


- 上記の研究は社会心理学で行われている
- 集合知と対比して集合愚と呼ばれることもある

Webと集合知

- Web 2.0
  - 2005年にオライリー（Tim O’Reilly）が提唱
  - 従来のWeb（Web 1.0）と異なる7つの原則をあげた

- Web 2.0 であるための7原則
  1. プラットフォームとしてのWeb
  2. 集合知の活用
  3. データは次なるIntel Inside
  4. ソフトウェアリリースサイクルの終焉
  5. 軽量なプログラミング
  6. 単一デバイスの枠を越えたソフトウェア
  7. リッチなユーザ経験



ティム・オライリー
1954-

Web 2.0 であるための7原則

1. プラットフォームとしてのWeb
   - これからのWebは、その上で様々なソフトウェアが走るための「プラットフォーム」として捉えるべき

2. 集合知の活用
   - ドットコムバブルで生き残った企業は、さまざまな形でスロフィッキーが主張する群衆の英知を活用している

3. データは次なる Intel Inside
   - データベース管理は、Web 2.0 企業の中核能力であり、Web 2.0 の重要なWebアプリケーションには必ずそれを支える専門のデータベースがあること
   - （例）Amazon.com

   - 「Intel Inside」とは
     - 当時、PCを購入すると、「intel inside」（このPCにはIntel社が製造したMPUが入っているという意味）の小さな商標が貼られていることが多々あったことから、これになぞらえて言った言葉

4. ソフトウェアリリースサイクルの終焉
   - モノではなくサービスとして提供されること
   - （例）Google社

5. 軽量なプログラミング
   - Web上のさまざまなアプリケーションが、大変軽量なインタフェースを介して、相互に連携が可能で、マッシュアップが容易に行えるようなWebサービスが提供されること
   - （例）REST

6. 単一デバイスの枠を越えたソフトウェア
   - 実用性が高く、単一デバイスの枠を越えた、Webをプラットフォームとするソフトウェアが次々と開発されて、それが大きな利益をもたらすこと
   - （例）スマートフォンやカーナビなどを、データの受信装置ではなくデータの発信装置として機能させることでリアルタイムでのトラフィックモニタリングするなど

7. リッチなユーザ経験
   - Webアプリケーションはユーザにパソコン並みのリッチなユーザ・インタフェースを通してリッチな経験をさせなければならない
     - （例）JavaScript、Ajax

Web 2.0 における集合知の活用

- 集合知を活用する企業
  - Google
    - PageRank という検索アルゴリズム
  - Amazon
    - ユーザレビュー、協調フィルタリング
  - Wikipedia
    - 誰でも記事を投稿し編集することができる
  - del.icio.us（現：delicious）
    - ソーシャルタギングによるコンテンツの分類

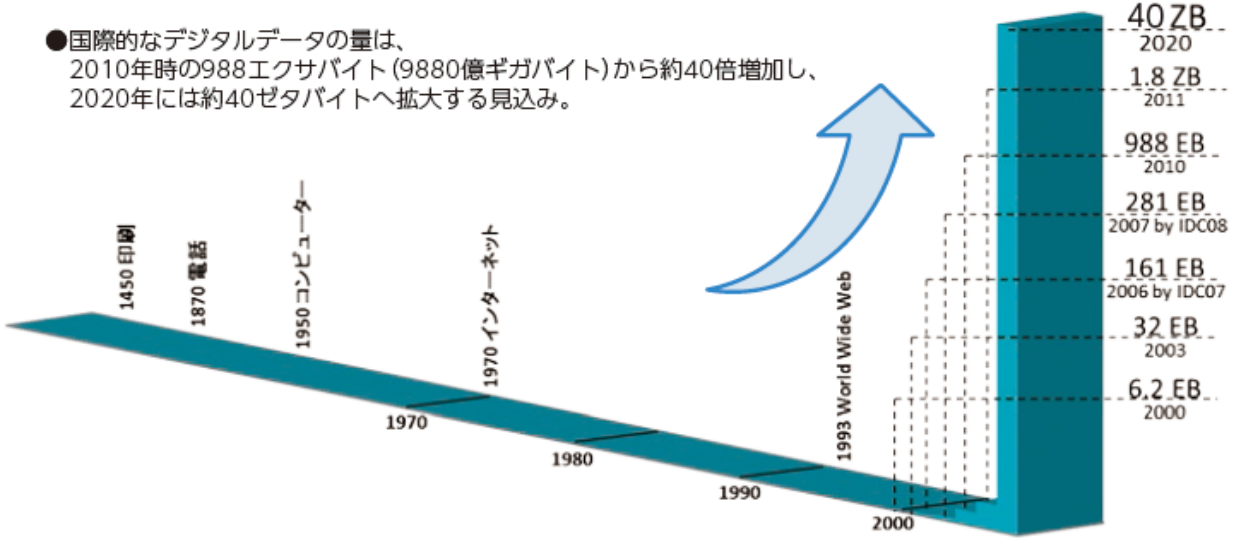
検索エンジン


Baidu百度
网址之家
www.hao123.com
土豆网
中关村在线
ZOL.com.cn
东方财富网
eastmoney.com
IT.com.cn
无忧考网
www.51Test.net
Gougou搜狗
迅雷在线
Google
SOSO搜搜
sina新浪網
腾讯网
QQ.com
tom.com 搜索
china.com
www.china.com
TOP100.cn
有道youdao
必应bing
網易NETEASE
www.163.com
188财富邮
114
.com
中搜
Sogou搜狗
YAHOO!中国雅虎
搜狐
SOHU.com
阿里巴巴
Alibaba.com
淘宝网
Taobao.com
Ku6.com
Chinaren

デジタルデータ量（Digital Data）の増加

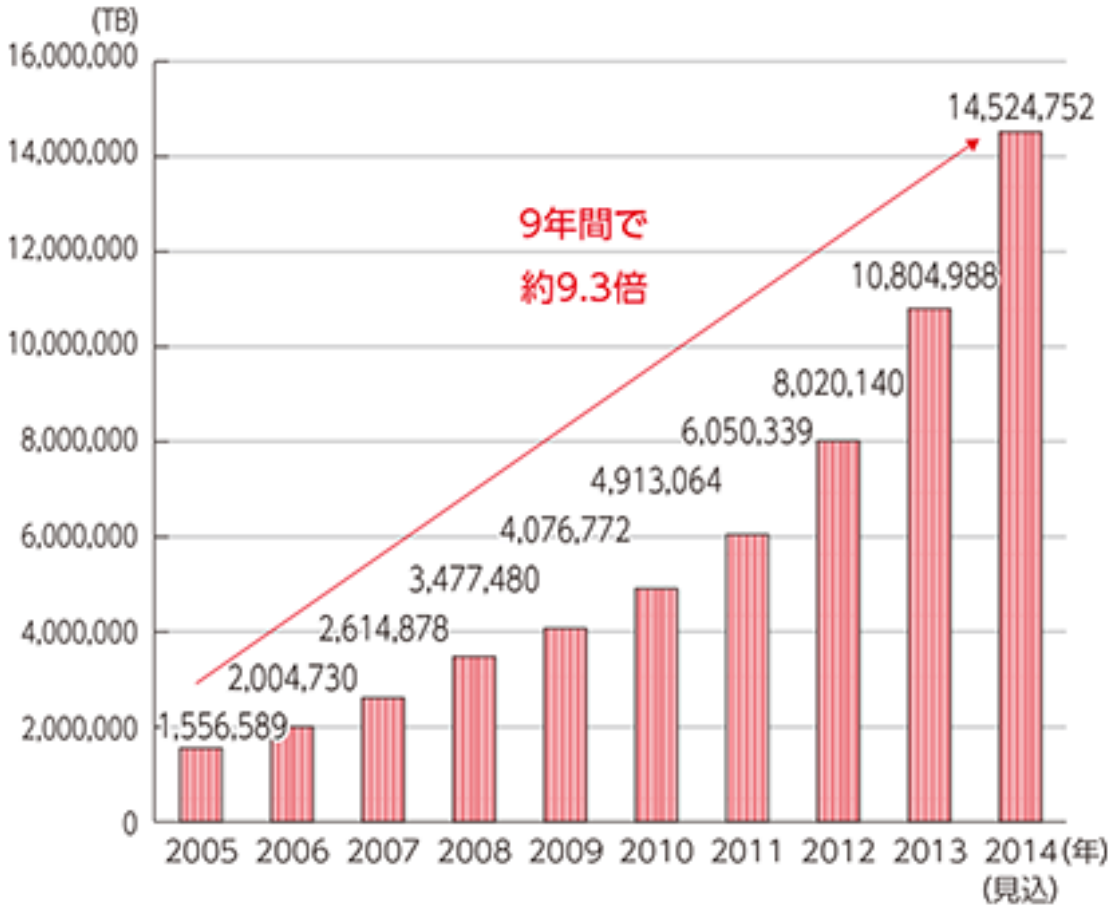
- 2011年
  - 約 1.8 ZB（ゼタバイト）
    （＝ 1,800,000,000,000 GB（ギガバイト））
- 2020年
  - 約 40 ZB（ゼタバイト）
    （＝40,000,000,000,000GB）に達すると予想

図表3-1-1-1 デジタルデータ量の増加予測



（出典）総務省「ICTコトづくり検討会議」報告書

データ流通量の推移（日本）

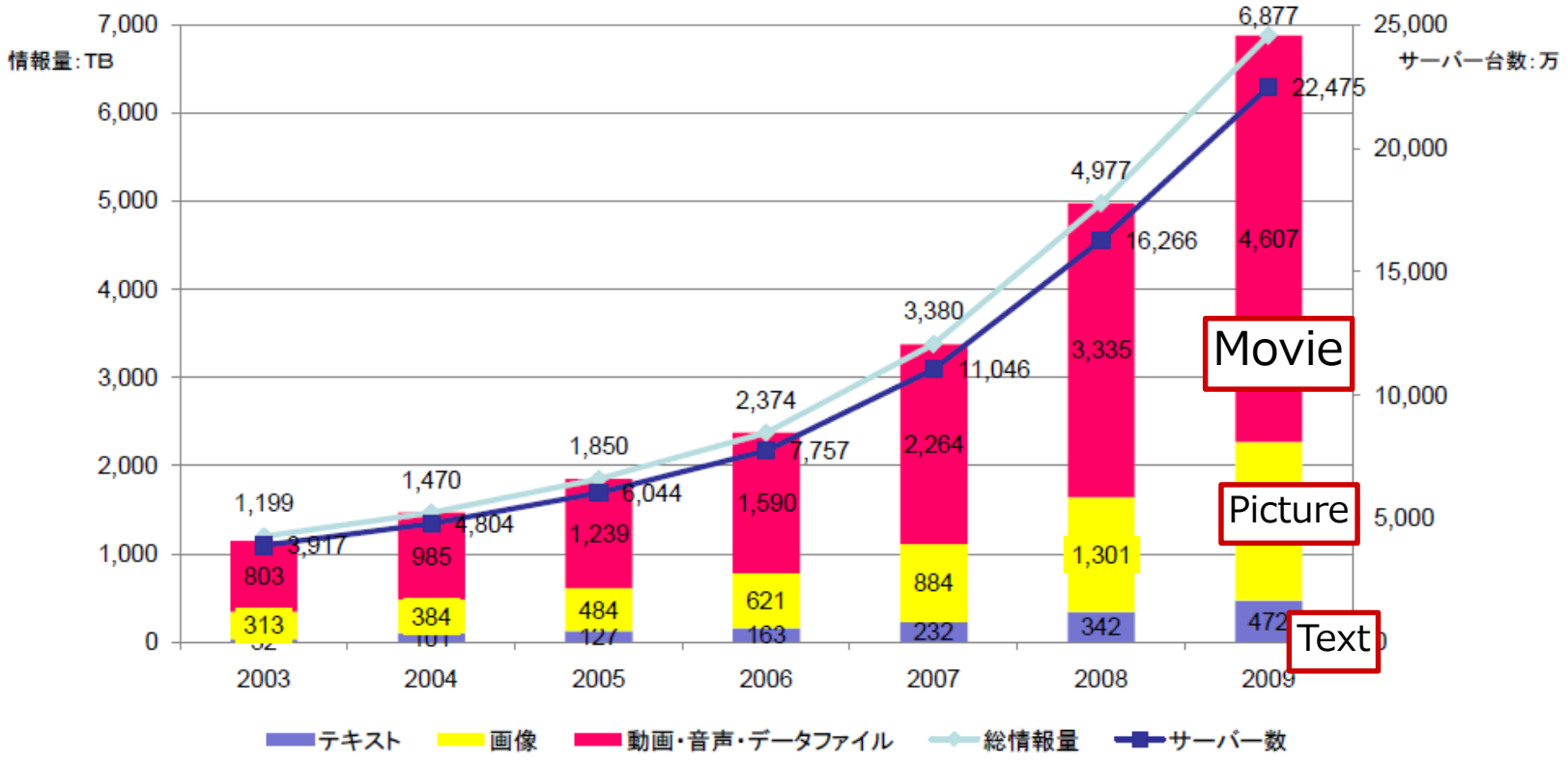


出典：27年度情報通信白書

- データ量が多いもの
  - 防犯遠隔監視カメラデータ(7.8EB)
  - センサーログデータ(3.2EB)
  - POSデータ(1.0EB)

インターネット上で検索可能な情報量(日本)

インターネット上における検索エンジンが検索可能な情報量は、ブロードバンドの普及により、動画、音声ファイル等（有料動画や会員制サイトのコンテンツを除く）の増加により、2009年までの5年間に6倍に増加。2009年1月時点における情報量は6,877TBと推計される。



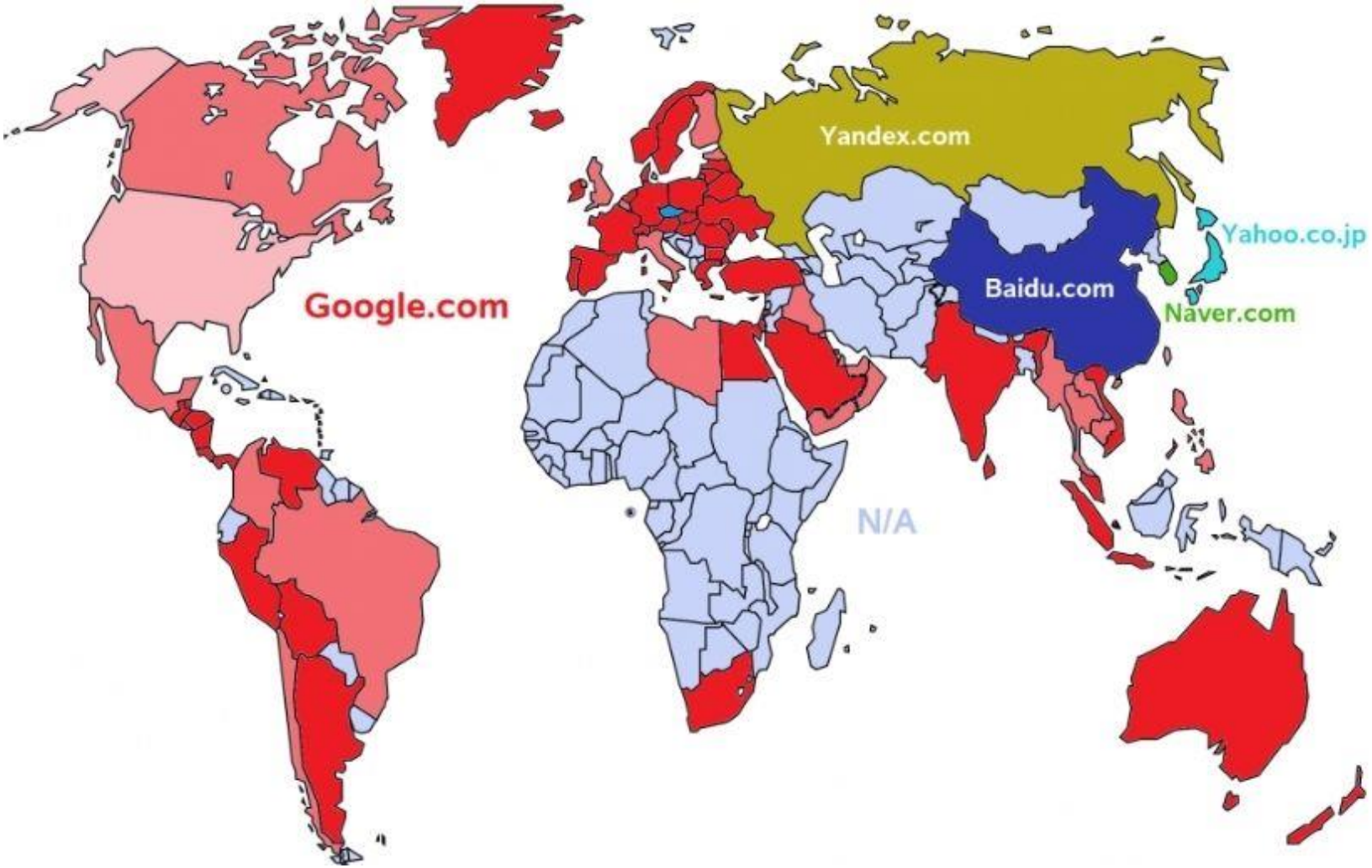
※ サーバ台数に対する情報量の算出式は総務省「WWWコンテンツ統計調査報告書」の1998年～2004年までのjpドメインの情報量の推移及び国内サーバーの相関系数を用い、jpドメインにおける情報量の増減傾向をインターネット全体の情報量の増減傾向に当てはめた。推計の根拠となる各年のwebサーバの台数は英国Netcraft社により公表されている数値を用いている。

インターネット検索エンジンの現状と市場規模等に関する調査　総務省情報通信政策研究所(H21)

検索エンジン（Search engin）とは？

- 指定したキーワード（Keyword）を含むドキュメントの一覧を出力するコンピュータプログラム

[参考] 世界の検索エンジンシェア



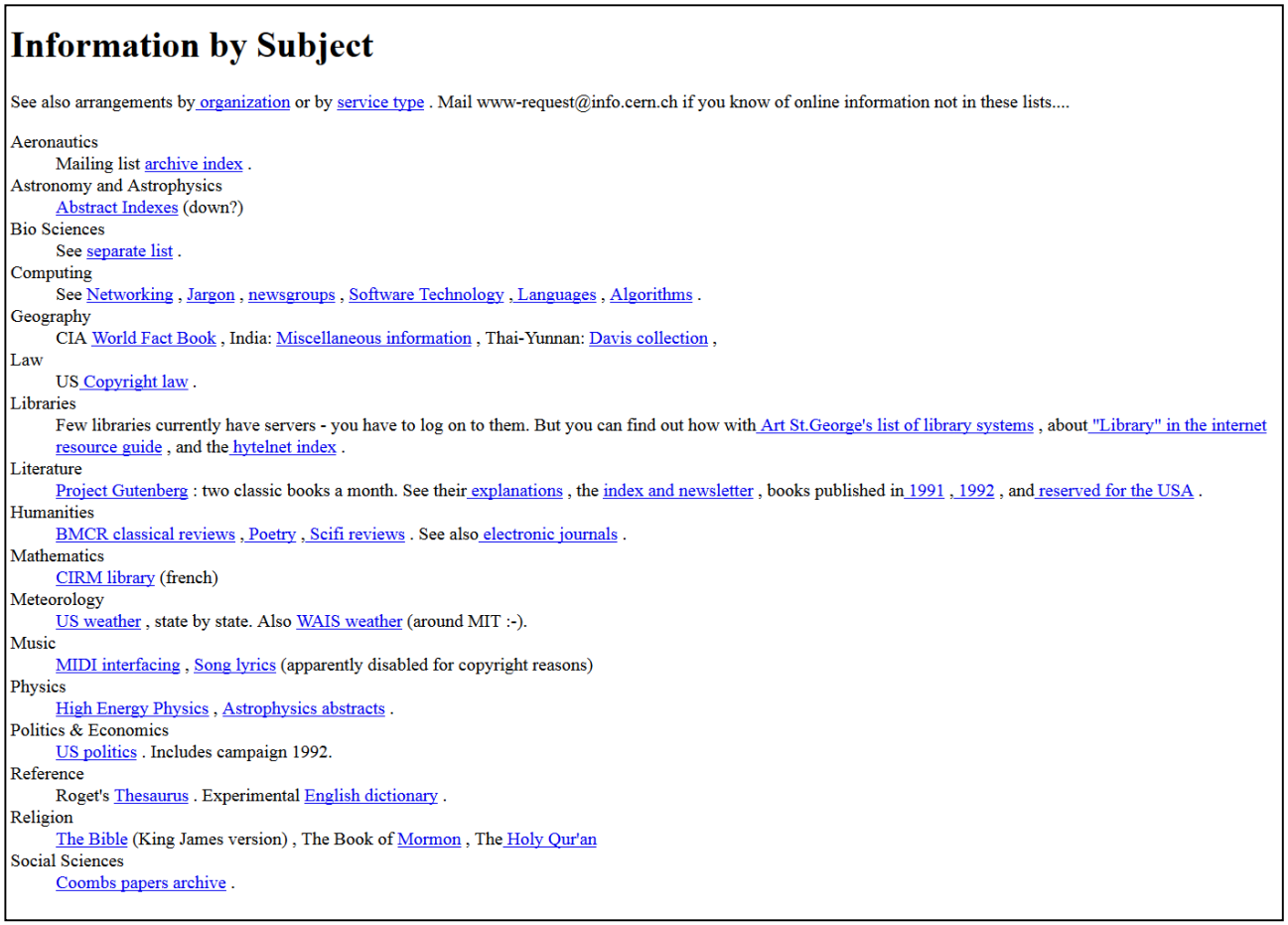
検索エンジンの難しさ

- データ面
  - データが広範囲に分散している
  - 頻繁に生成され更新されている
  - データ件数が膨大
  - 非構造で冗長
  - 非一様（メディアや形式も違う、言語も様々）

- 利用者面
  - 問合せ（Query）を記述するのが難しい
  - 検索結果が膨大で解釈できない

検索エンジンの歴史 -1

- 検索ディレクトリ　1992年11月　(世界初)



Information by Subject

See also arrangements by organization or by service type . Mail www-request@info.cern.ch if you know of online information not in these lists....

Aeronautics
Mailing list archive index .
Astronomy and Astrophysics
Abstract Indexes (down?)
Bio Sciences
See separate list .
Computing
See Networking , Jargon , newsgroups , Software Technology , Languages , Algorithms .
Geography
CIA World Fact Book , India: Miscellaneous information , Thai-Yunnan: Davis collection ,
Law
US Copyright law .
Libraries
Few libraries currently have servers - you have to log on to them. But you can find out how with Art St.George's list of library systems , about "Library" in the internet resource guide , and the hytelnet index .
Literature
Project Gutenberg : two classic books a month. See their explanations , the index and newsletter , books published in 1991 , 1992 , and reserved for the USA .
Humanities
BMCR classical reviews , Poetry , Scifi reviews . See also electronic journals .
Mathematics
CIRM library (french)
Meteorology
US weather , state by state. Also WAIS weather (around MIT :-).
Music
MIDI interfacing , Song lyrics (apparently disabled for copyright reasons)
Physics
High Energy Physics , Astrophysics abstracts .
Politics & Economics
US politics . Includes campaign 1992.
Reference
Roget's Thesaurus . Experimental English dictionary .
Religion
The Bible (King James version) , The Book of Mormon , The Holy Qur'an
Social Sciences
Coombs papers archive .

http://www.w3.org/History/19921103-hypertext/hypertext/DataSources/bySubject/Overview.html

検索エンジンの歴史 -2

- World Wide Web Worm Search Engine(1993)
- 最初の検索エンジン
- コロラド大学の Oliver McBryanが開発

http://www.grantcunningham.com/blog_files/351730-world-wide-web-worm.gif

検索エンジンの歴史 -3

- Yahoo (1994)
  - WWWディレクトリ
  - 人手で分類

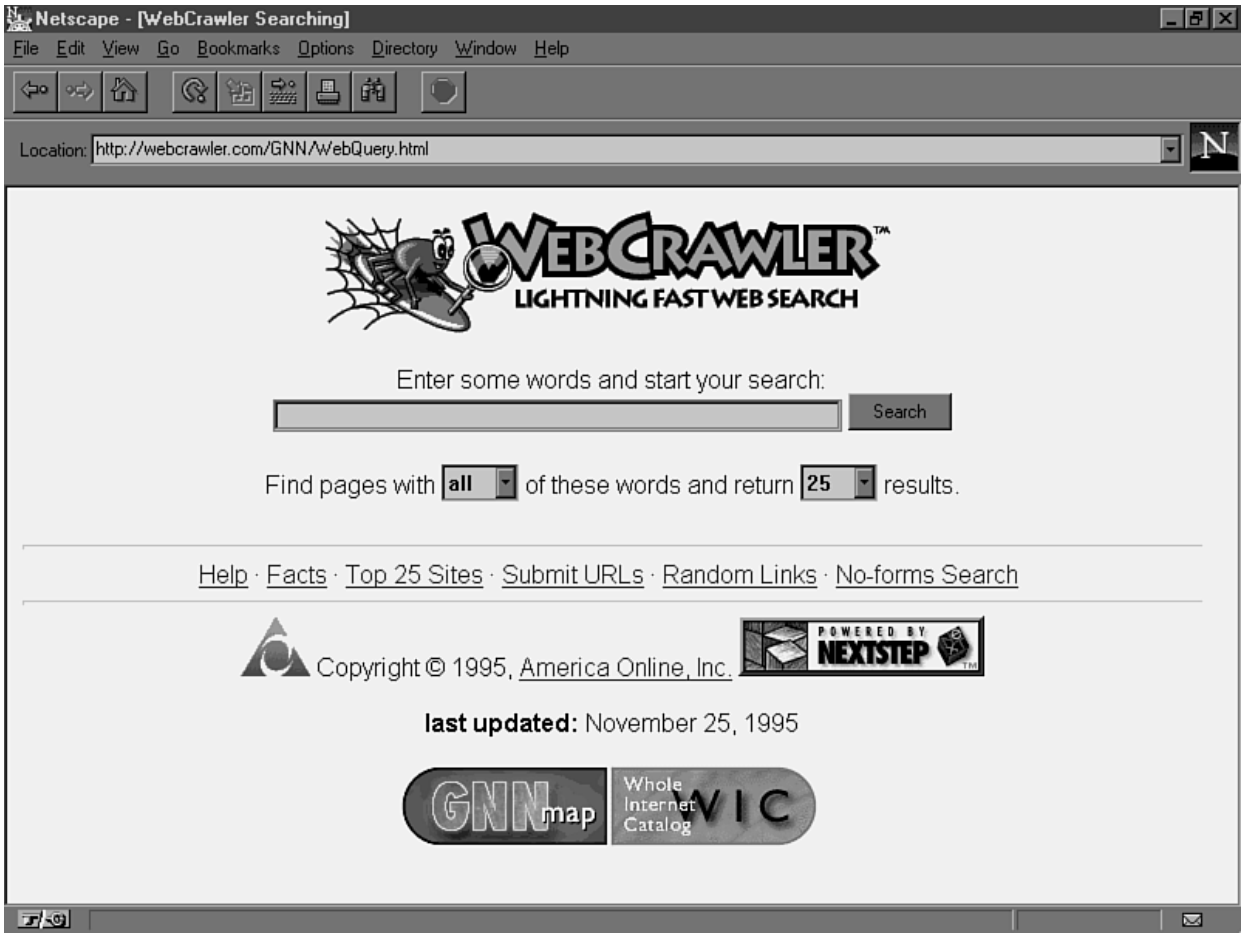
1996年10月の
Yahoo.com

https://web.archive.org/web/19961020022754/

検索エンジンの歴史 -4

- WebCrawler (1994)
  - 最初のテキストサーチエンジン

1995年12月の
WebCrawler

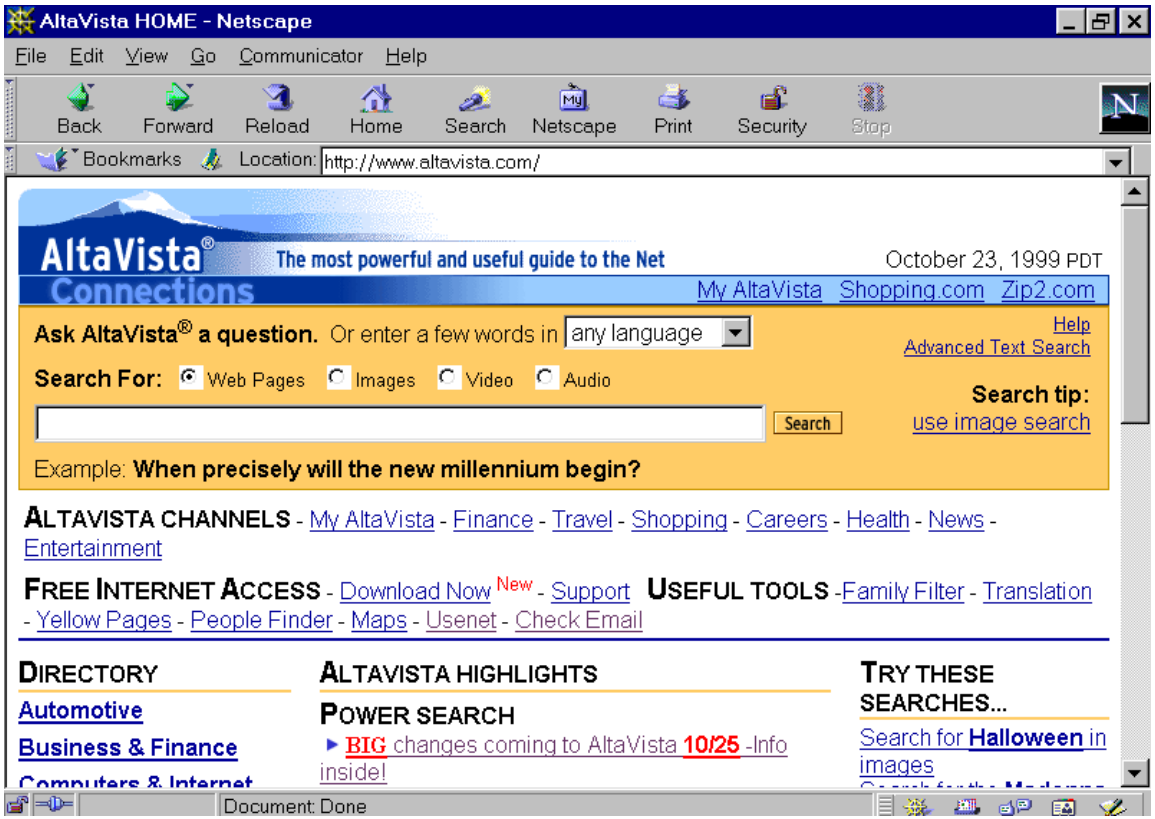


http://medialab.di.unipi.it/web/doc/tyw/15twp14.gif

検索エンジンの歴史 -5

- AltaVista (1995)
  - 自然言語での検索が可能
  - DEC研究者が立ち上げ

1999年ごろの
AltaVista



https://upload.wikimedia.org/wikipedia/en/d/d8/Altavista-1999.png

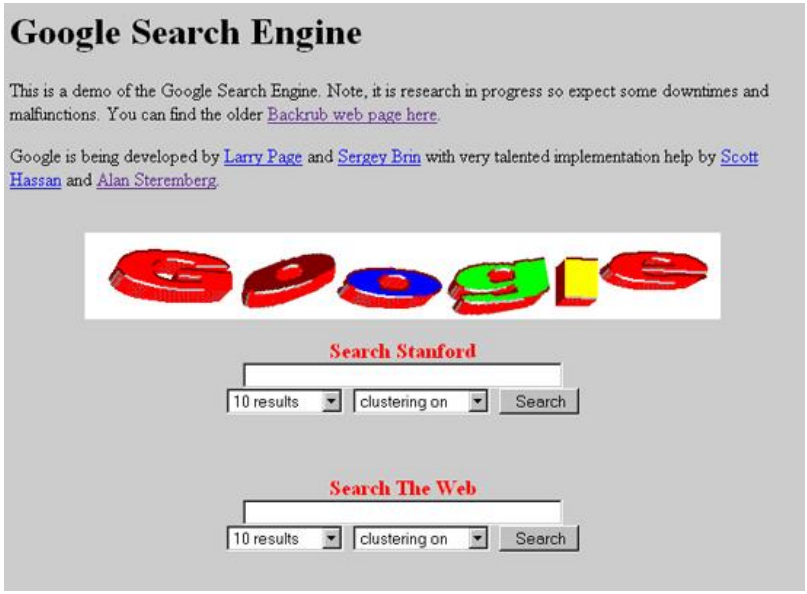
検索エンジンの歴史 -6

- Google (1997)
  - 1997年9月15日に開始
  - google.stanford.edu

1997年開始ごろのGoogle

http://amix.dk/uploads/google_1.png

1997年のdemoバージョン

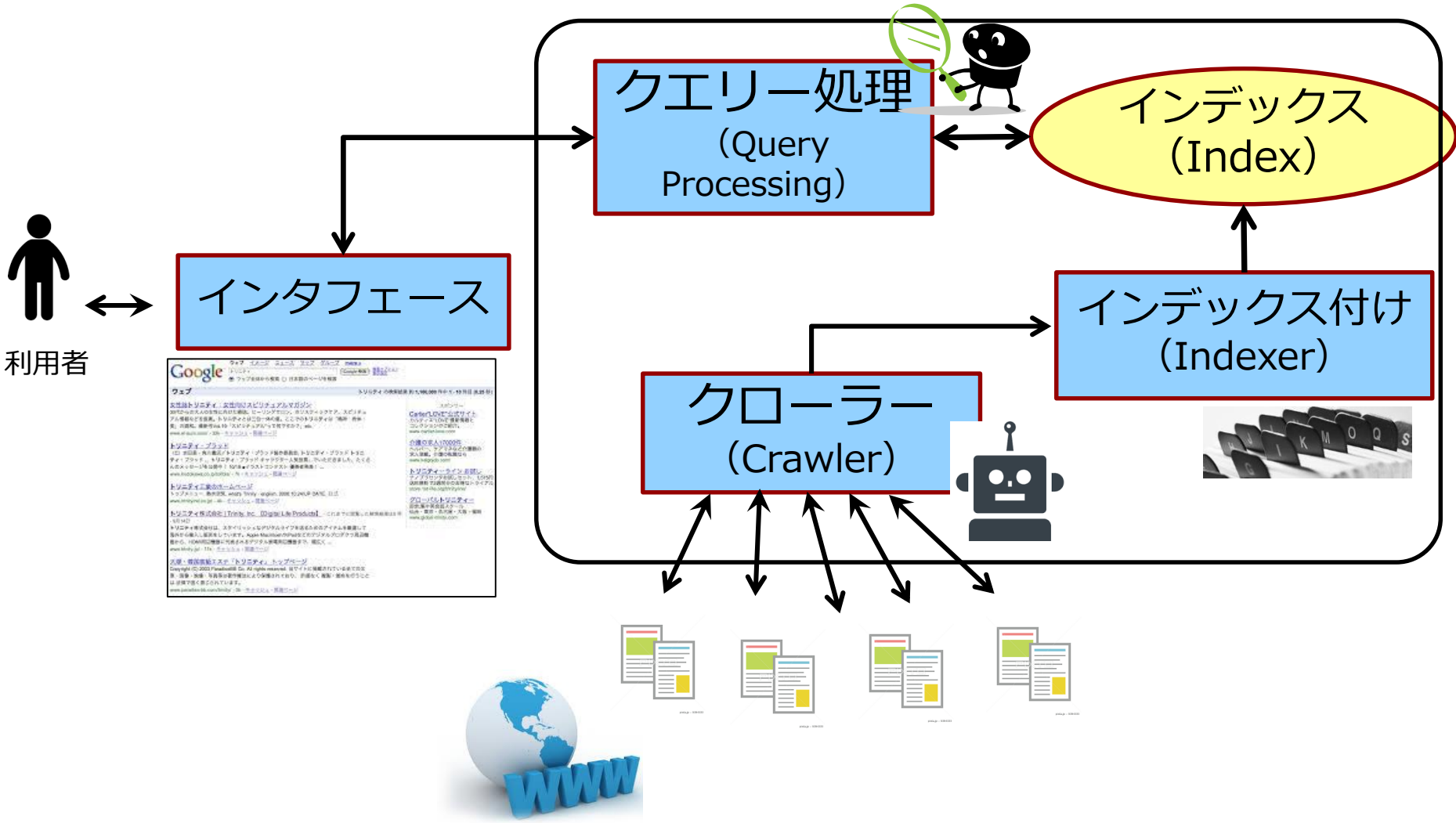


http://vendyxiao.com/wp-content/uploads/2012/10/google-1997.jpg

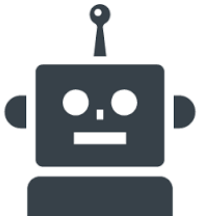
検索エンジンの歴史 -7

- Baidu (2001)
  - 2000年1月 - 北京大学の李彦宏と徐勇が北京中関村でBaidu, Inc.を創立。
  - 2001年8月 - サーチエンジンのベータ版である、Baiduを発表。

典型的な検索エンジンの構成（Crawler-Indexer architecture）



クローラー（Crawler）



- Webからページを自動的にダウンロードするプログラム
- ダウンロードしたページはインデックス付けと検索のために利用される

- 入力： 開始点となるシードページ(Seed page)の集合 S
- 処理：
  1. リンクを解析してダウンロードしていないページのURLを収集し、ダウンロードキューに格納する
  2. ダウンロードキューから新しいページを適切に選択し、そのページをダウンロードする。
  3. 1と２を与えられた終了条件を満たすまで繰り返す

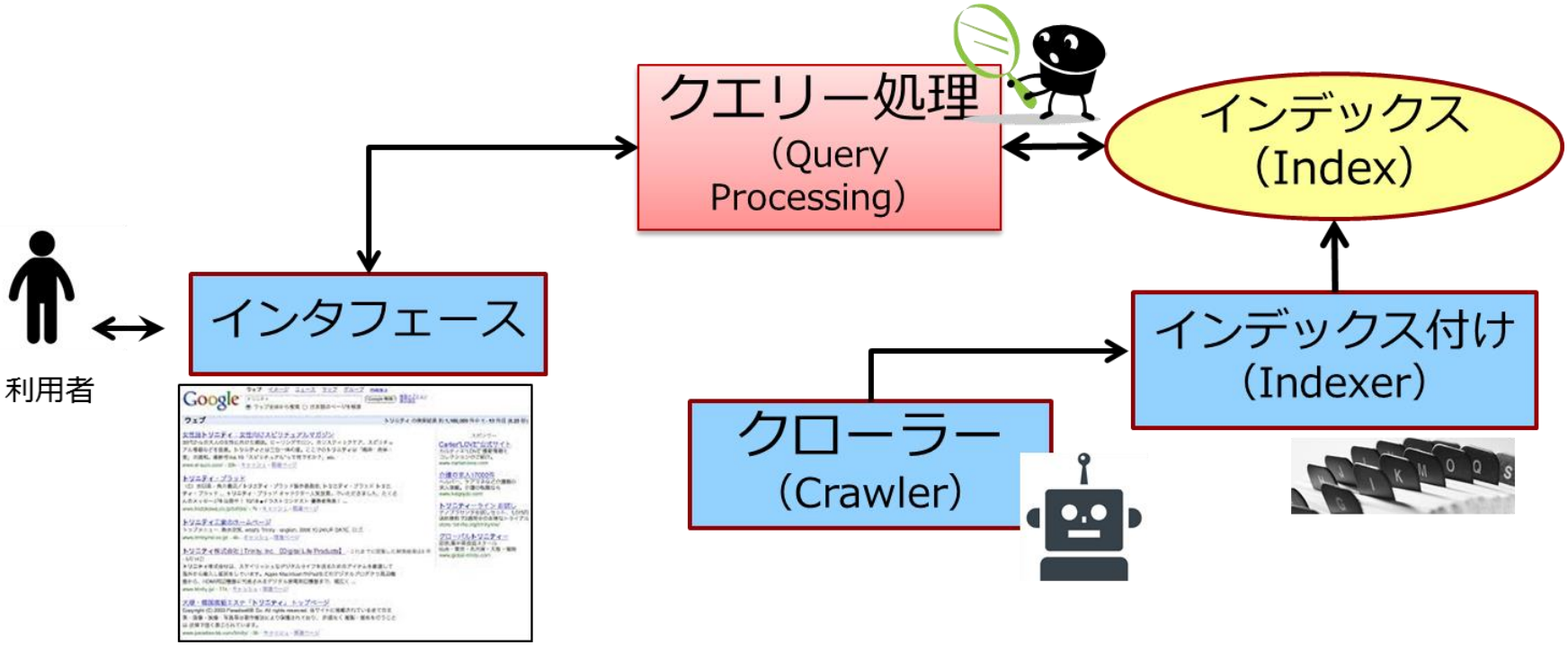
インデックス付け（Indexer）

- インデックスとは、格納されたデータをより早く検索したり、抽出できるように作られる索引データ
- クローラーが収集したWebページのデータを、データベースに（処理しやすいように整理された状態で）格納することをインデックス付け（Indexer）と呼ぶ

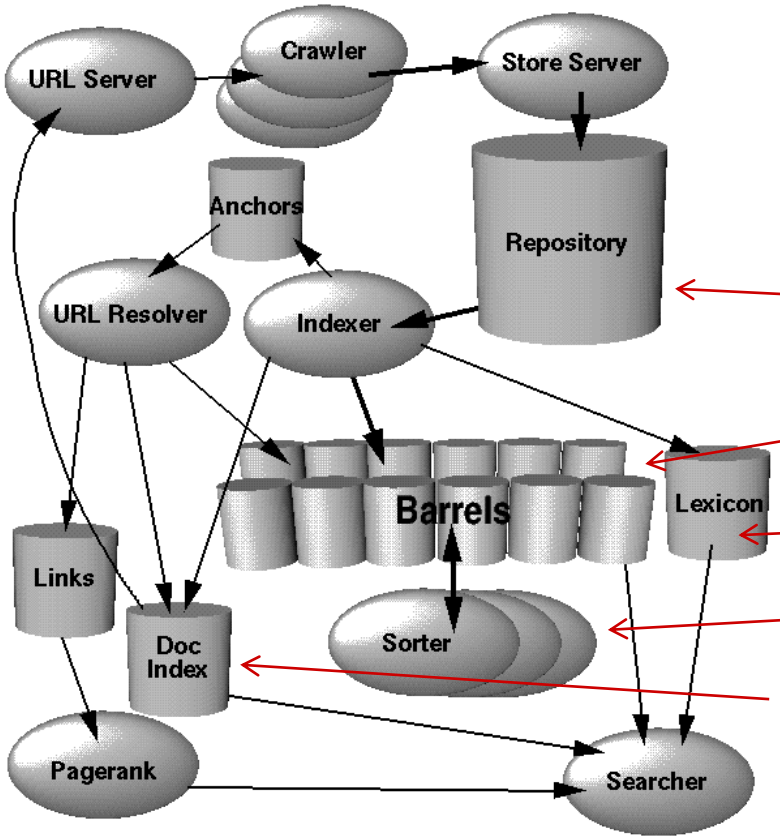
用語	Webページ
aardvak	3,117,3961
⋮	
aztec	3,15,19,101,673,1199
baby	3,31,56,94,909,11114,253791
⋮	
zymurgy	1159223

クエリー処理（Query processing）

- ユーザーからの検索キーワードを処理し、スコアリングををもとに検索順位を決め、結果をリスト表示する処理
  - 「要求受付」「集計」「結果表示」



初期のGoogleの構造



The Anatomy of a Large-Scale Hypertextual Web Search Engine
Sergey Brin and Lawrence Page

http://infolab.stanford.edu/~backrub/google.html

crawlしたwebページを圧縮して保管

docIDごとにhit(word出現回数)を保持する

単語辞書(1400万語)

wordIDからのサーチを可能とする

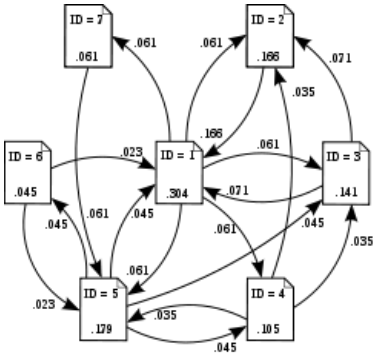
docIDごとにrepositoryへのリンク、統計情報等を保持する

検索エンジンの課題

- 検索できない情報の増加
  - SNSデータ(フロー型メディアのデータ)
  - Deep Web

PageRank

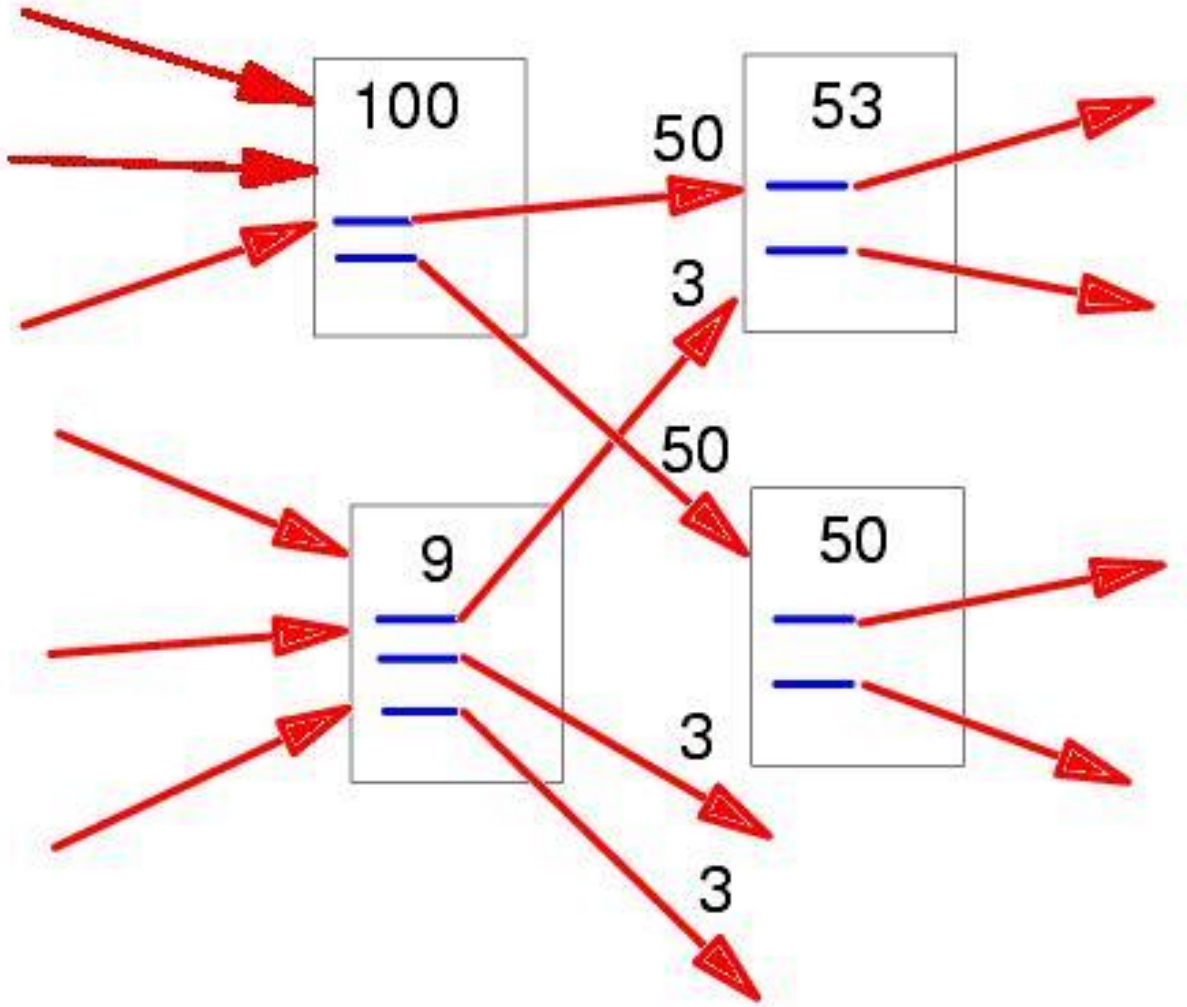

ウェブ
画像
Google
Googleアプリでサクサク検索


ID = 7
.061
ID = 2
.166
ID = 6
.045
ID = 1
.304
ID = 3
.141
ID = 5
.179
ID = 4
.105
.061
.061
.071
.035
.023
.166
.061
.071
.045
.061
.045
.061
.045
.035
.061
.023
.035
.045

Google社のウェブページランキングアルゴリズム（PageRank）

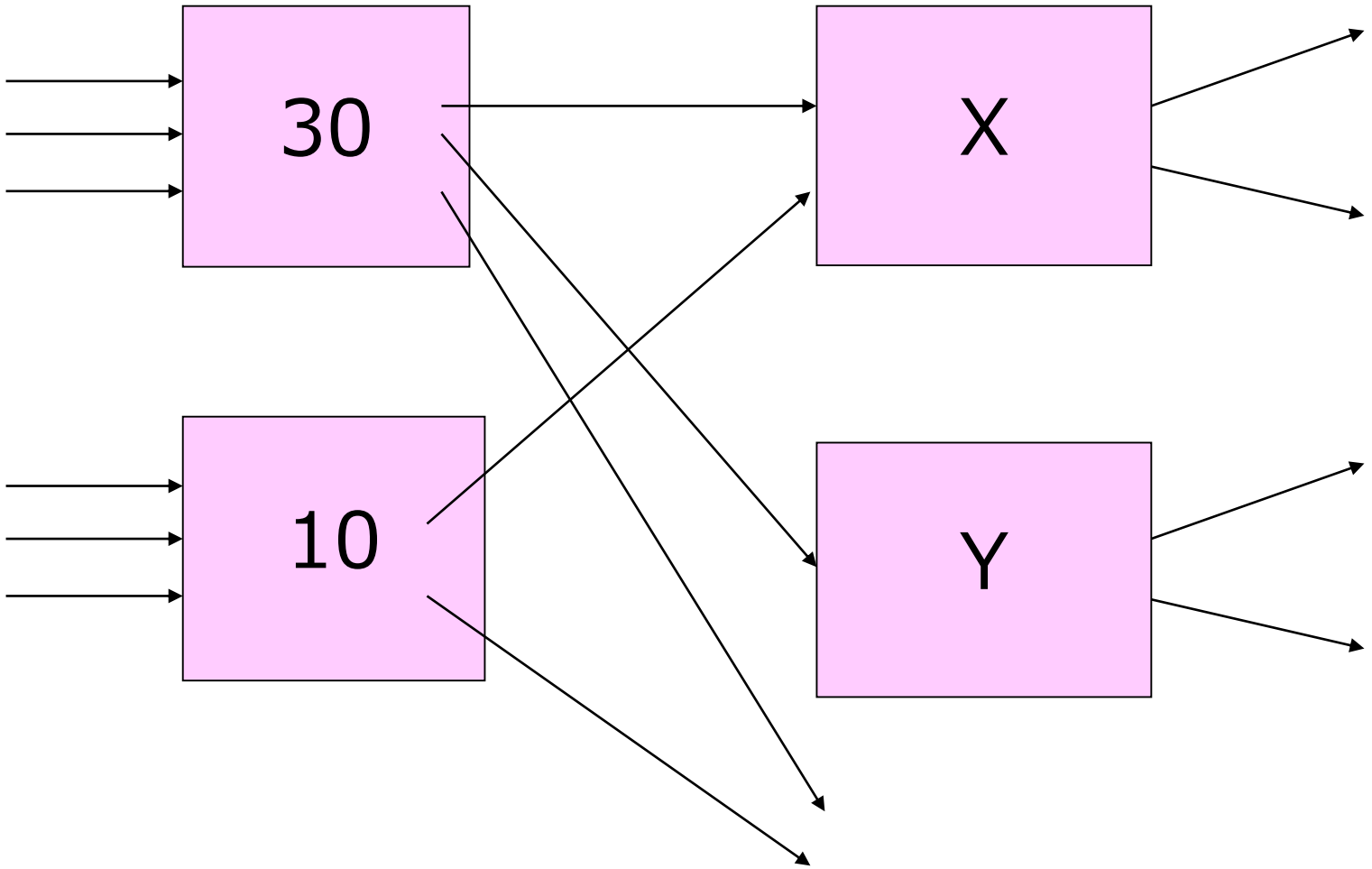
- PageRank
  - Google社が開発したウェブページの重要度の判定技術、および重要度の指標
  - 自社の検索エンジンに搭載
  - 「多くの良質なページからリンクされているページは、やはり良質なページである」 という再帰的な関係をもとに、ページの重要度を計算
    - あるページから別のページへのリンクを、リンクされたページへの「支持投票」とみなし、それにリンク元のページの重要度（そのページの被リンク数）の重みをつけて加算し、投票数によりそのページの重要性を判断している
  - ページの内容は影響しない
  - PageRank の基本論文
    - Lawrence Page, Sergey Brin, Rajeev Motwani, Terry Winograd, 'The PageRank Citation Ranking: Bringing Order to the Web', 1998

PageRank の概念



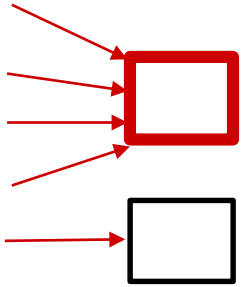
引用 : Page et al. (1998) Figure 2 'Simplified Page Calculation'

（確認問題）X,Yの重要度を計算してみましょう

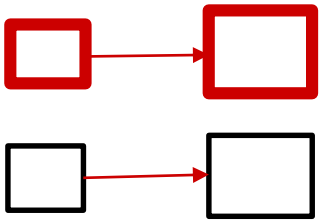


PageRank のポイント

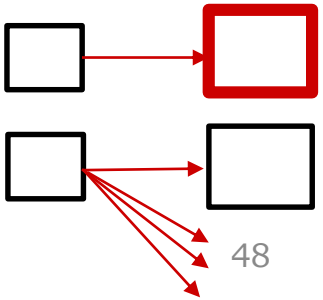
- 被リンク数
  - 多くのページからリンクが張られていればお薦めめ度は高くなる
    - 「たくさんリンクされる人気のあるページは、 きっと良いページであるに違いない」
    - リンクを張るという行為は、「このページを見るといい／このページは 役に立つ」という推薦行為を行っていることとみなせる



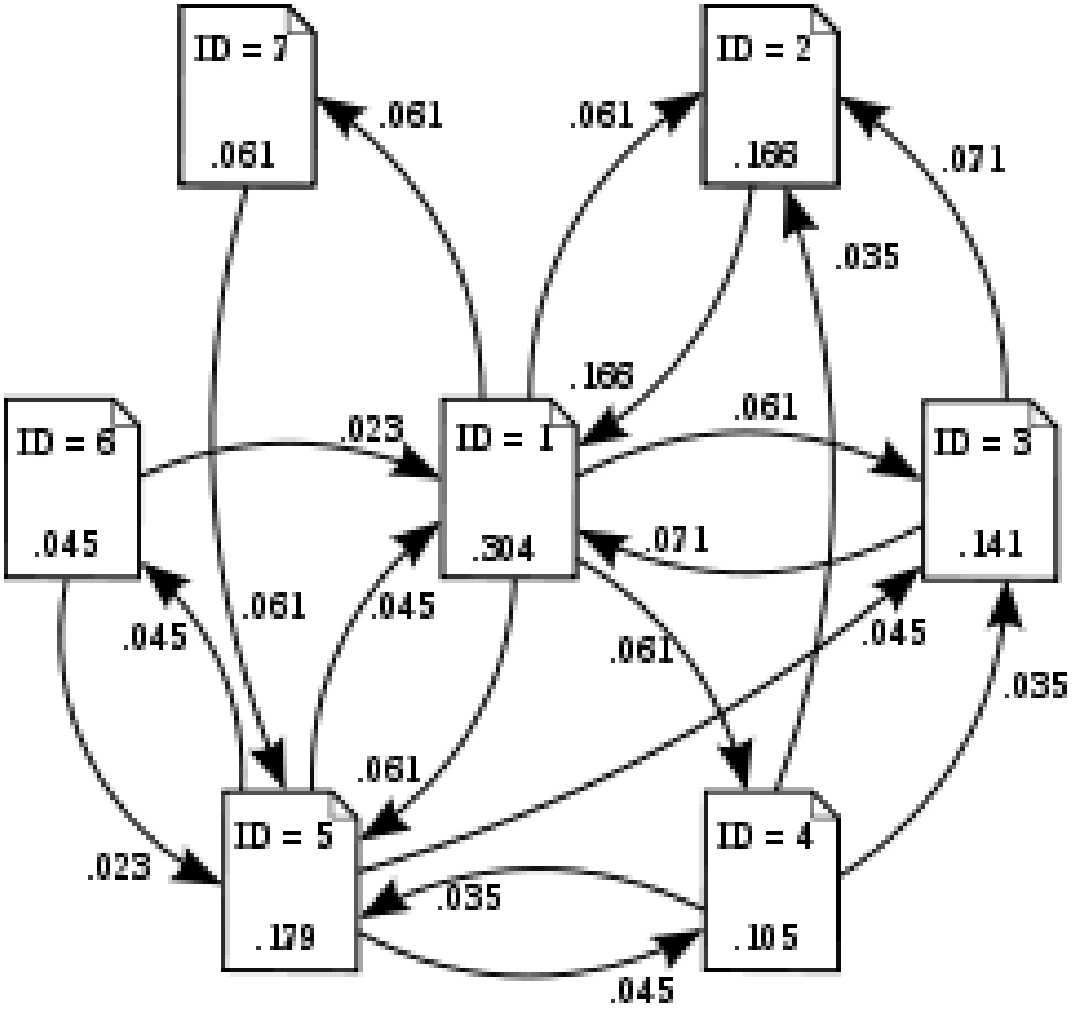
- おすすめ度の高いページからのリンクかどうか
  - 裏付けのある人気かどうか
  - （例）Yahoo! のような PageRank がとても高いサイトからリンク されると重要度は高い



- リンク元ページでのリンク数
  - 総リンク数が少ないページからのリンクは高く評価し、 総リンク数が多いページからのリンクは低く評価する
    - あれもこれもと関連のないところにリンクを張っているだけの単なるブックマーク的ページからのリンクはとんど価値がない



（例）PageRank の計算例



ランキングアルゴリズムの歴史

- Google 誕生前（~2000年）
  - カテゴリごとにウェブを分類・紹介するディレクトリ型検索が主流
  - サイトへ集客する手段としてディレクトリへの登録が推奨（例：Yahoo!）

- "SEO"の用語誕生からGoogle台頭時代（2000年代前半）

SEO（Search Engine Optimization）

- SEOとは
  - ある特定の検索エンジンを対象として検索結果でより上位に現れるようにウェブページを書き換えること、またはその技術のこと（検索エンジン最適化）

- SEOを行う理由
  - 検索エンジンサイトでのキーワード検索結果として、上位ページと下位ページでは、クリック率・誘導率にきわめて大きな乖離がある
  - 検索結果からの誘導そのものは、他の広告媒体などと異なりコストがかからないため、企業サイトにとって、極めて効果が高く重要なポジショニング
  - ランキングアルゴリズムを分析し、自社サイトの上位表示を目指すための修正・最適化を実施することで、これらを実現する風潮があらわれた

PageRank のメリット

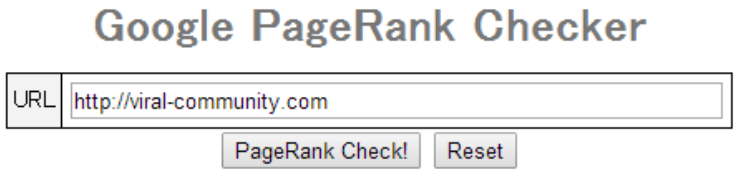
- 機械的に生成されたリンクの影響を受けにくい
  - 従来は、ページの重要度としてそのページの被リンク数だけを単純に用いることがあった
  - 従来の単純な SEOやSPAM手法（リンクを貼りまくる）は通用しない
  - に上がるが、そのためには内容の充実に努めなければならない

- ページの内容は影響しない
  - PageRank 自体はユーザが与えた検索式に与えた語句とは全く無関係な、すでに定まった量（検索語句を与えても一定の、文書固有のスコア量)
  - ページ自体のテキスト解析はしなくてよい

- PageRank を上げるための近道は基本的に無い
  - たとえば Yahoo! に登録依頼を出して通ればイッキ

[参考] PageRank を調べる

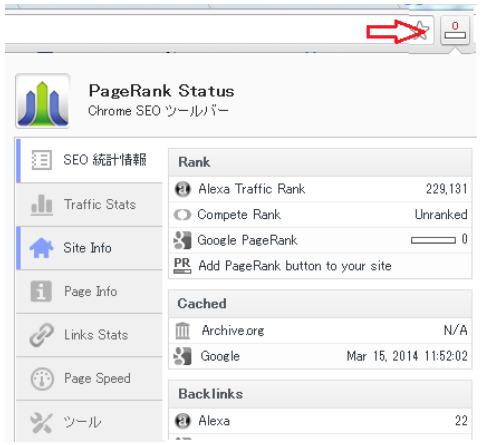
- Google PageRank Checker



調査結果

調査URL	http://viral-community.com
ページランク	PageRank 0 0
YAHOO! JAPAN カテゴリ Yahoo!Japan ヤフーカテゴリ	このサイトは登録されていません
あなたのサイトを大手ポータルに一括登録！ Goo・OCN・@nifty・エキサイト・ODNなど多数のポータルサイトに掲載 一覧	>>> クロスレコメンドに申し込む

- PageRank Status（Chrome拡張）



[参考] PageRank の値

- PageRank
  - 数ヶ月に一度更新。0-10の11ランク。

PageRank	位置づけ
0	評価なし（例えば、新設して間もないサイト）
1-3	標準サイト
4-6	人気サイト
7-9	ポータルサイトなど
10	Google, Government's Official Web Portal など

  - 参考
    - 大連理工大学(7)、立命館大学(7) 、東京大学(7)、慶應義塾大学(8)、北京大学(9) 、MIT(9)
    - Amazon (8)、Yahoo!(9)、Facebook(9)、Weibo(9)、Twitte(10)

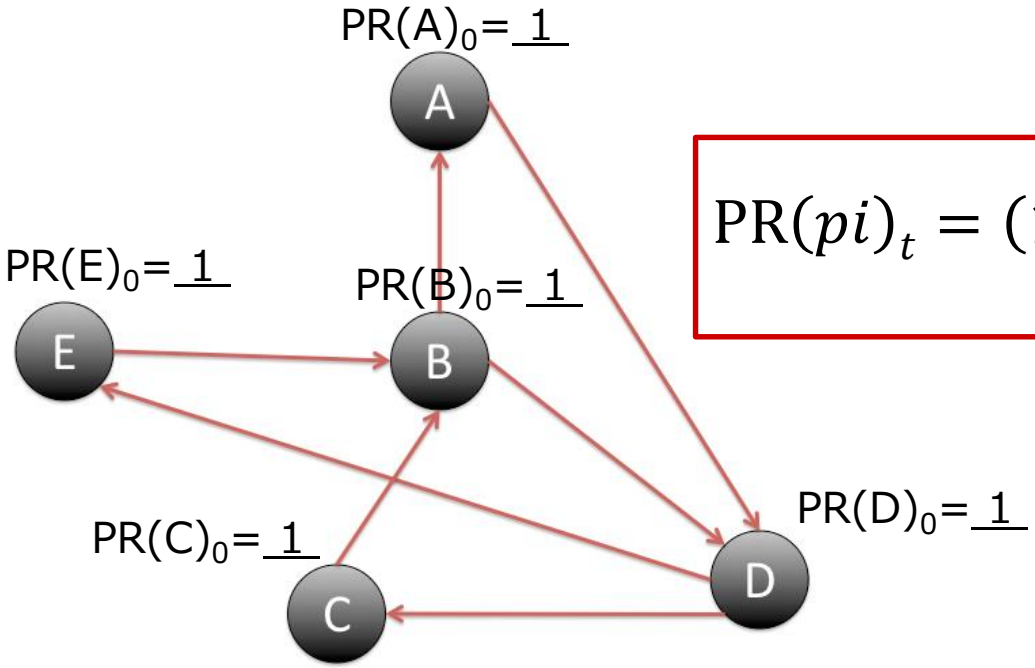
PageRank の計算

- 定義式

$$\mathrm{PR}(pi)_t = (1-d) + d \sum_{pj \in M(pi)} \frac{PR(pj)_{t-1}}{L(pj)}$$

- *PR(pi)* はページ *pi* の PageRank
- d は減退係数（damping factor）
  （論文では、通常 0.85 に設定すると明記されている）
- *M(pi)* はページ *pi* にリンクを張っているページの集合
- *L(pj)* はページ *pj* から他のページに張っているリンクの数
- *N* は文書数

PageRank を計算してみましょう（反復1回目：t=1）



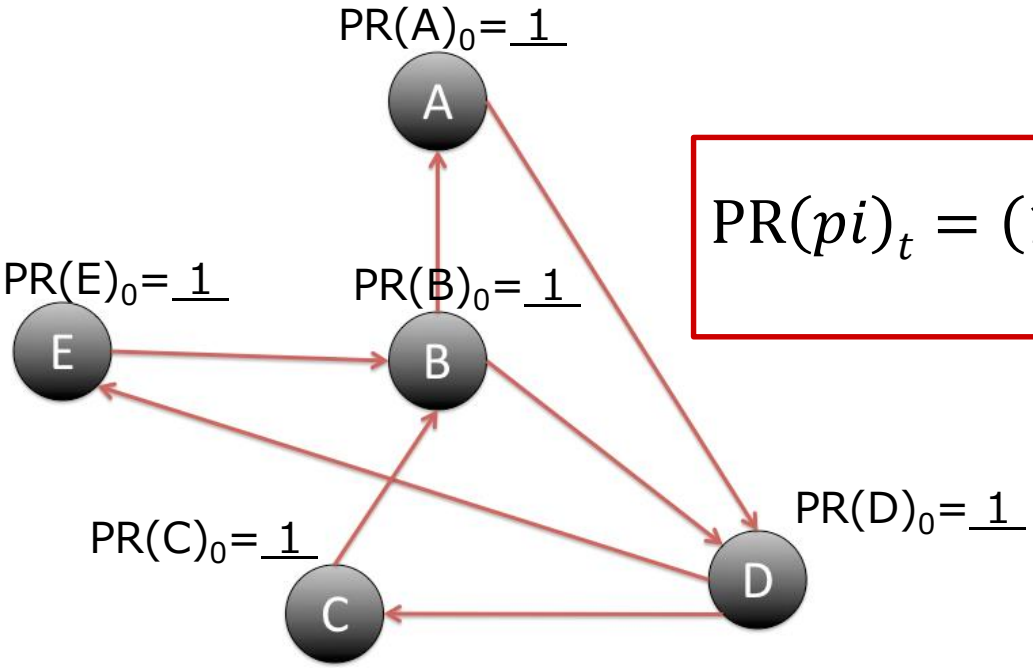
$$\mathrm{PR}(pi)_t = (1-d) + d \sum_{pj \in M(pi)} \frac{PR(pj)_{t-1}}{L(pj)}$$

PageRankの初期値($PR(x)_0$)は1とする

$PR(A)_1 = (1 - 0.85) + 0.85\{(PR(B)_0 / 2)\} = 0.455$

- *PR(pi)* はページ *pi* の PageRank
- d は減退係数（damping factor）
  （論文では、通常 0.85 に設定すると明記されている）
- *M(pi)* はページ *pi* にリンクを張っているページの集合
- *L(pj)* はページ *pj* から他のページに張っているリンクの数
- *N* は文書数

PageRank を計算してみましょう（反復1回目：t=1 ）



$$\mathrm{PR}(pi)_t = (1-d) + d \sum_{pj \in M(pi)} \frac{PR(pj)_{t-1}}{L(pj)}$$

$PR(A)_1 = (1 - 0.85) + 0.85\{(PR(B)_0 / 2)\} = 0.455$

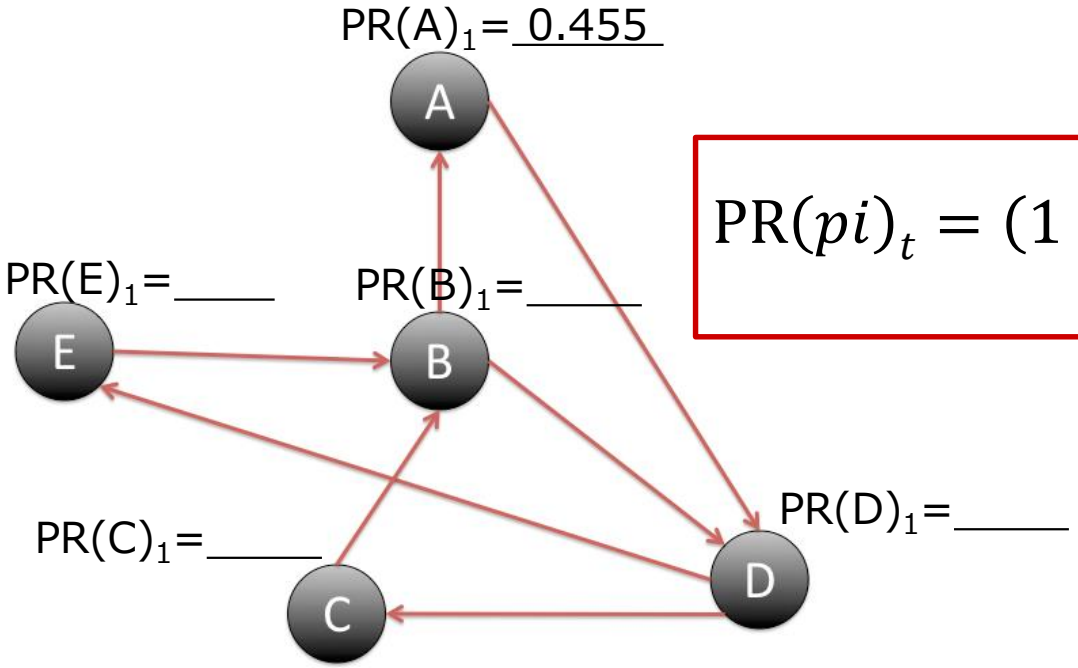
$PR(B)_1 = (1 - 0.85) + 0.85\{(PR(C)_0 / 1) + (PR(E)_0 / 1)\} =$ _____

$PR(C)_1 = (1 - 0.85) + 0.85\{(PR(D)_0 / 2)\} =$ _____

$PR(D)_1 = (1 - 0.85) + 0.85\{(PR(A)_0 / 1) + (PR(B)_0 / 2)\} =$ _____

$PR(E)_1 = (1 - 0.85) + 0.85\{(PR(D)_0 / 2)\} =$ _____

PageRank を計算してみましょう（反復1回目：t=2)



$$\mathrm{PR}(pi)_t = (1-d) + d \sum_{pj \in M(pi)} \frac{PR(pj)_{t-1}}{L(pj)}$$

$PR(A)_2$ = (1 - 0.85) + 0.85{($PR(B)_1$ / 2)} = _____

$PR(B)_2$ = (1 - 0.85) + 0.85{($PR(C)_1$ / 1) + ($PR(E)_1$ / 1)} = _____

$PR(C)_2$ = (1 - 0.85) + 0.85{($PR(D)_1$ / 2)} = _____

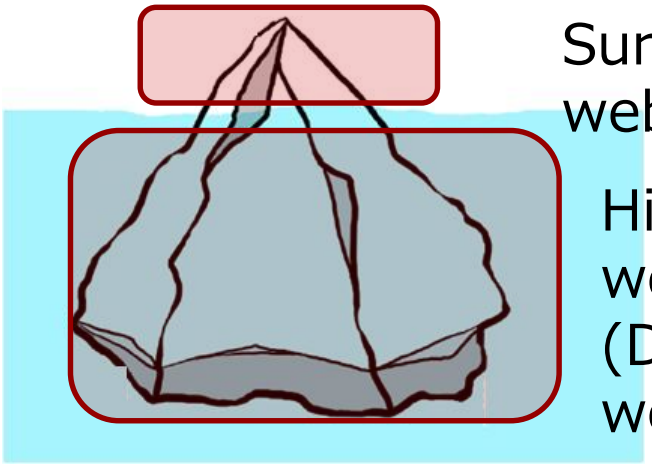
$PR(D)_2$ = (1 - 0.85) + 0.85{($PR(A)_1$ / 1) + ($PR(B)_1$ / 2)} = _____

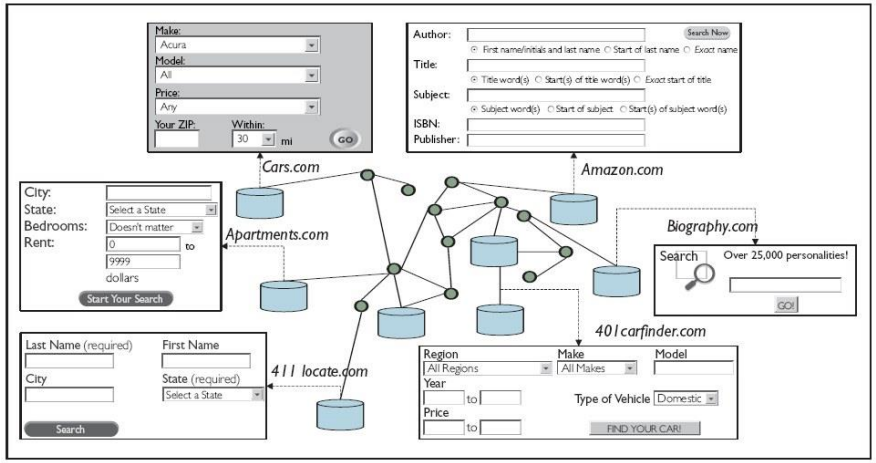
$PR(E)_2$ = (1 - 0.85) + 0.85{($PR(D)_1$ / 2)} = _____

検索エンジンの実際

Deep Web (1)

- Deep web とは
- 検索エンジンではアクセスできないWebのこと
- （アクセス可能なWebのことをSurface webともいう）
  2000年調査では、43,000サイト、7,500TB





Bin He, Mitesh Patel, Zhen Zhang, Kevin Chen-Chuan Chang
Communications of the ACM, Vol. 50 No. 5, Pages 94-1012007


Deep Web
jodacame.com
WordPress
Drupal
Joomla!
Google+
facebook
YouTube
twitter
Nivel 2
MEGA
TARINGA!
Nivel 3
The Pirate Bay
BitTorrent
Nivel 4
The Hidden Wiki
Nivel 5
Nivel 6

Hidden web (Deep Web) (2)

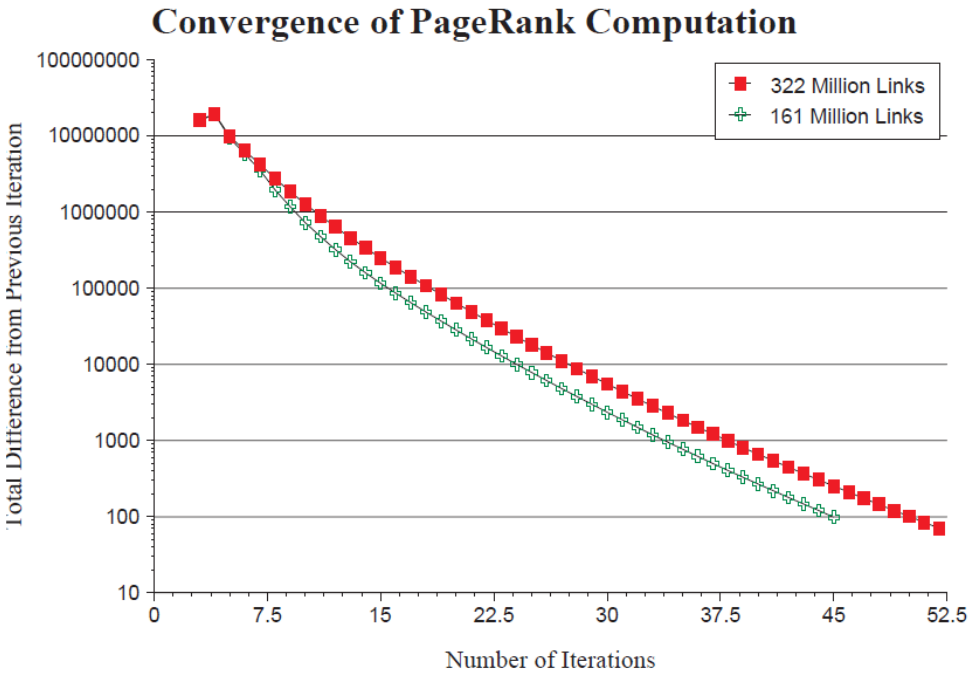
- Deep web はどうしてできるか
  - 利用者からの問合せに対応してページが動的に生成される
    - 大量の高品質な情報を持つ組織が保持する情報をWebで公開することで、Hidden webは大量となる
  - html形式でなく特殊な形式のページ
  - スクリプトで動作するページ
  - どこからもリンクされていないページ

大規模化(1)

一兆を越えるWebページについてPageRankを計算したり高速にWebページを検索することは可能であろうか

【PageRank計算効率化】

- 3億のリンク数だと50回程度の反復でPageRank計算は収束する
- 対数グラフに注意
- 通常の方法だと数時間の計算時間が必要
  ⇒現在では数分まで短縮可能

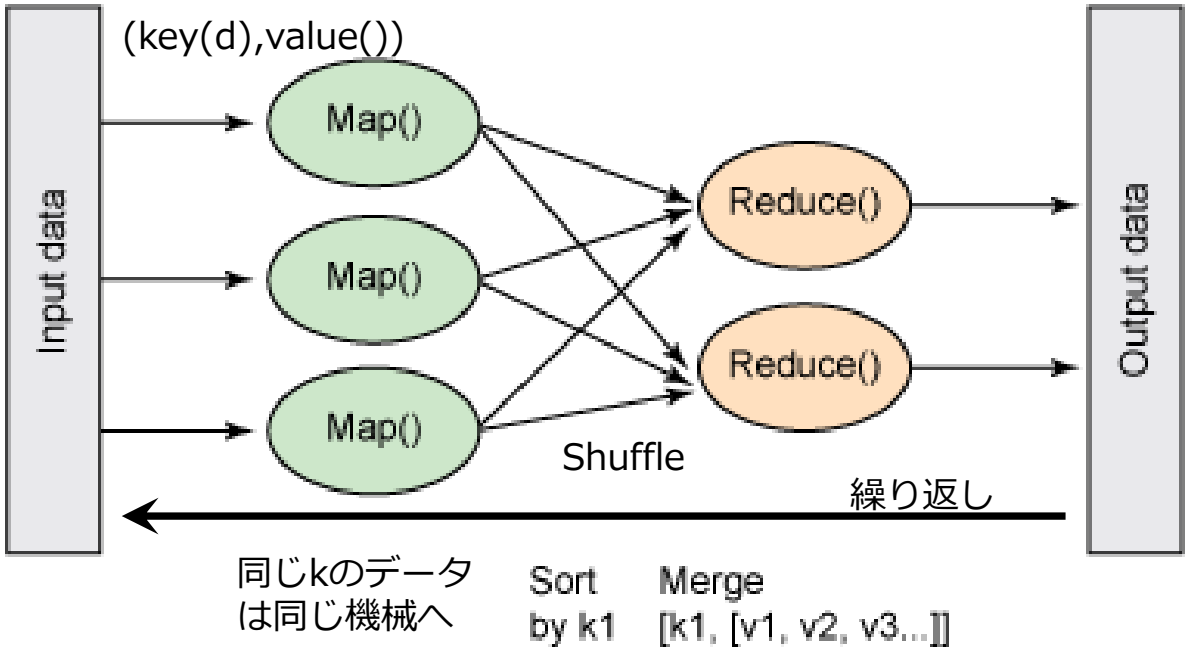


L Page: The pagerank citation ranking: Bringing order to the web. 1998

大規模化(2)

一兆を越えるWebページについてPageRankを計算したり高速にWebページを検索することは可能であろうか

•MapReduceの活用(スケール)(⇒Hadoop)



http://www.ibm.com/developerworks/jp/cloud/library/cl-openstack-deployhadoop/

PageRank P $P = \beta M + (1-\beta)B$

$$M = \begin{bmatrix} 0 & 1/2 & 0 & 0 \\ 1/3 & 0 & 1 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix}$$

Mapperへ渡すデータ
案１）垂直分割

$$M_1 = \begin{bmatrix} 0 \\ 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} \quad M_2 = \begin{bmatrix} 1/2 \\ 0 \\ 0 \\ 1/2 \end{bmatrix} \quad M_3 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad M_4 = \begin{bmatrix} 0 \\ 1/2 \\ 1/2 \\ 0 \end{bmatrix}$$

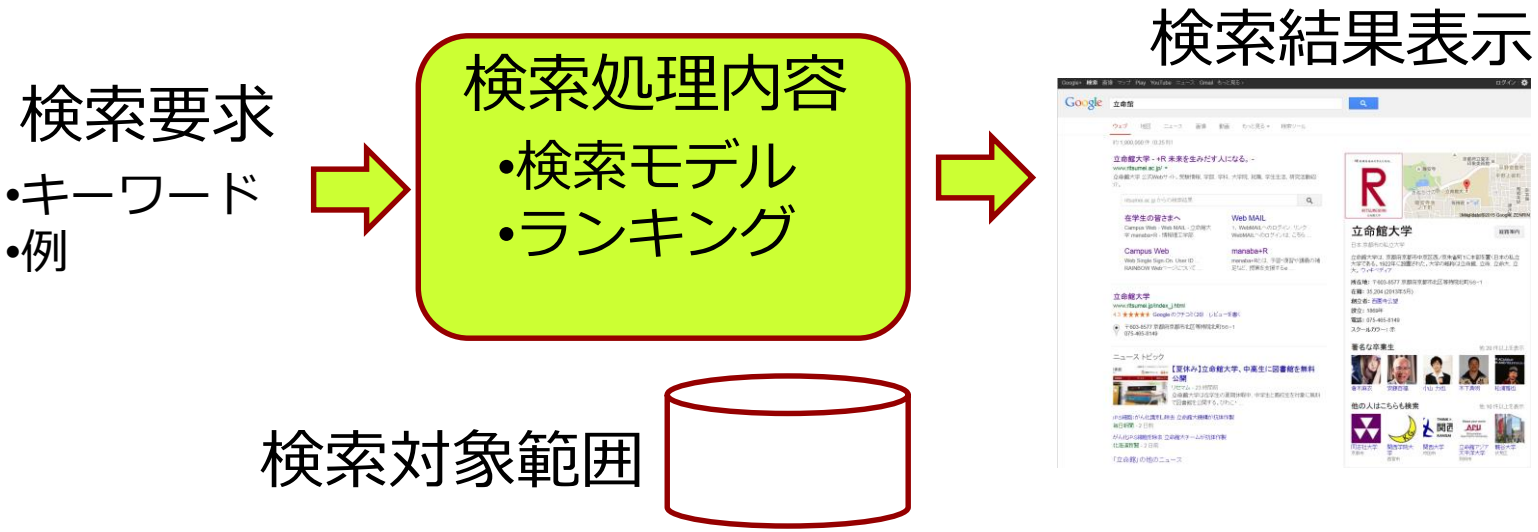
案２）ブロック行列

$$M_{1,1} = \begin{bmatrix} 0 & 1/2 \\ 1/3 & 0 \end{bmatrix} \quad M_{1,2} = \begin{bmatrix} 0 & 0 \\ 1 & 1/2 \end{bmatrix}$$

$$M_{2,1} = \begin{bmatrix} 1/3 & 0 \\ 1/3 & 1/2 \end{bmatrix} \quad M_{2,2} = \begin{bmatrix} 0 & 1/2 \\ 0 & 0 \end{bmatrix}$$

パーソナル化(1)

1. 検索対象範囲のパーソナル化
2. 検索処理内容のパーソナル化
3. 検索結果表示のパーソナル化



パーソナル化(2)

1. 検索対象範囲のパーソナル化（手動/自動）
   - URL/ドメイン限定（立命館、Yahoo)
   - 分野限定 (映画、ニュース・・）
   - メディア限定 (Web, 動画、画像・・・）
   - 期間限定
   - 言語・地域限定

パーソナル化(3)

2. 検索処理内容のパーソナル化（手動/自動)
   - キーワードの重みづけ
   - キーワードの付加
   - 検索モデル選択
   - ランキング方法選択

パーソナル化(4)

3. 検索結果表示のパーソナル化（手動/自動)
   - レイアウト
   - 件数
   - 表示内容（タイトル、スニペットsnippet)
   - 伝達方法(PC、携帯、メール）
   - など