┼文字列処理アルゴリズムから製品開発へ

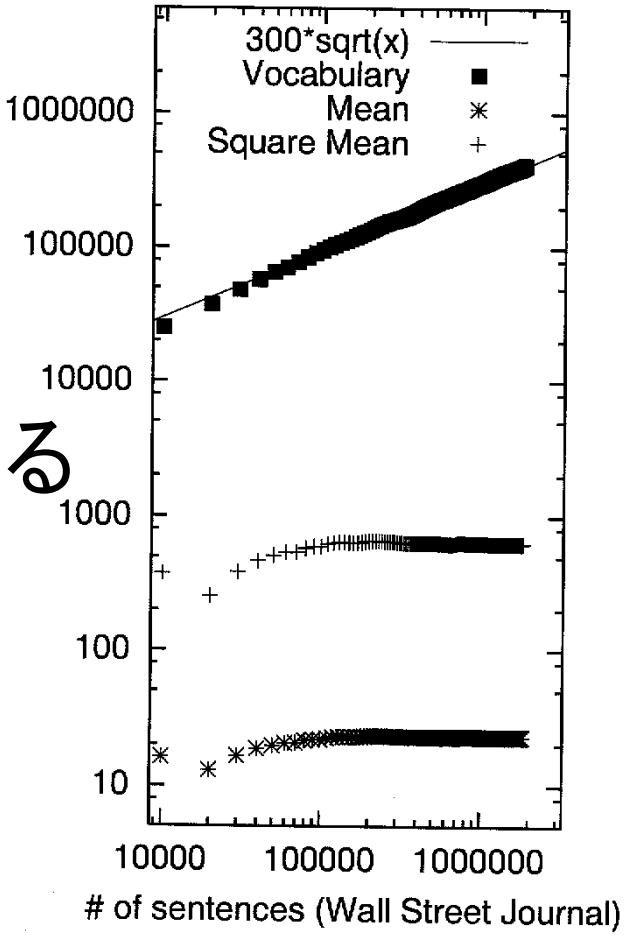
- 相関分析
- 文字列頻度計数／キーワード抽出
- 関連語発見
- 類似情報検索
- 製品化への適用
- 社会へのインパクト　<?>

問題1

大きなデータの関連ある単語の組み
をもれなくもとめてください
（相関の抽出）

分析対象：大語彙コーパス

- 語彙は増加しつづける
  - 新しい語が生まれつづける
  - 10万語／100万記事
- １記事は長くならない
- 関連する語を漏れなく求める



相関の分析

- 関係の強いものは、出現パターンが似ている

	北朝鮮	拉致	核開発	金総書記
記事１	○	×	○	○
記事２	×	×	×	×
記事３	○	○	×	○

相関の高いペアを求める

- 関係を推定するには

  ⇒　列ごとのパターンの相関を測定する

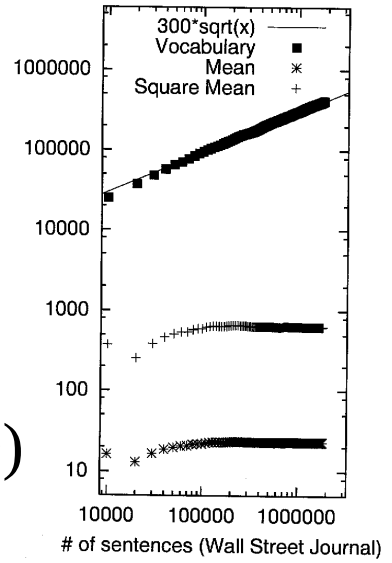
	北朝鮮	拉致	核開発	金総書記
記事１	○	×	○	○
記事２	×	×	×	×
記事３	○	○	×	○

相関の高いペアを全部求める

- ナイーブな方法
  - ペアのすべてについて、相関を計算する
  - メモリ使用量 $O(l^2)$
  - 計算時間 $O(l^2 \times N)$

l : 語彙の総数 (> 100,000)

N レコードの総数 (>1,000,000)



- 多項オーダだが
- 計算は実際的でない

相関を求める表

- 多くの関係は下記の表から求まる

# of document	*y*	not *y*
x	$df_a(x, y)$	$df_b(x, y)$
not *x*	$df_c(x, y)$	$df_d(x, y)$

$df_a(x, y)$：単語xと単語yの両方を含む文書数

表の要素にかかわる性質

- 必要な表は N とdf_a

$$df(x) = df_a(x,x)$$
$$df_b(x,y) = df_a(x,x) - df_a(x,y)$$
$$df_c(x,y) = df_a(y,y) - df_a(x,y)$$
$$df_d(x,y) = N - df_a(x,x) - df_a(y,y) + df_a(x,y)$$

N: ドキュメントの総数.

$df_a(x, y)$: 単語xと単語yの両方を含む文書数

効果的なアルゴリズム（既知）

- レコードごとに
  - $df_a(x, y)$ と Nの加算をする。
- ☆$df_a(x, y) > 0$の (x, y) ごとに
  - df_a, df_b, df_c, df_d を用いて相関を計算する。
- 相関の高い順にソートする
- 上位から高いものを取り出す

O(N log(N))ただし，１記事の長さに上限を仮定

$df_a(x, y)$：単語xと単語yの両方を含む文書数
☆：解の高速な第一次試験

実行時間

- ボトム：単純法，ベース：一次試験＋単純法

ドキュメント数	ボトム	ベース	メモリ
10	6.2	2.4	0.1
20	15.2	3.6	0.1
30	35.0	6.1	0.1
50	289.7	36.2	0.3
100	1561.0	107.0	0.5
200	11845.7	496.1	1.2
300	23225.0	818.9	1.4

大規模コーパスを処理すると

- 問題：大規模コーパスでは、$df_a(x, y)$ だけでも主記憶に保持するのは苦しい、
- 問題：外部記憶のランダムアクセスでは速度低下が著しい。
- アルゴリズムの方針はそのままで、ランダムアクセスをしないように工夫する。

$df_a(x, y)$：単語xと単語yの両方を含む文書数

外部記憶による実行結果

- 速度は半分だが、大規模な分析ができる。
- 実装には、プログラムのテクニックが必要

ドキュメント数	メモリ処理	ファイル処理
1000	3.4	7.7
3000	8.0	17.6
10000	29.6	63.8
30000	131.1	295.0
100000	実行不能	614.8
300000	実行不能	1565.6
732035	実行不能	4034.4

実際に分析すると。。
どの関数が効果的？

- どの関数が効果的か？
- 他に良い関数はないのか？

  下記では，df_a, df_b, df_c,df_dをa,b,c,dと記述

$$\cos(\vec{F},\vec{T}) = \frac{a}{\sqrt{(a+b)(a+c)}} \qquad I(x_1;y_1) = \log\frac{an}{(a+b)(a+c)}$$

$$S_d(\vec{F},\vec{T}) = \frac{2a}{(a+b)+(a+c)}$$

$$conf(Y \mid X) = \frac{a}{a+c} \qquad \gamma_{DE} = \frac{ad-bc}{\sqrt{(a+b)(a+c)(b+d)(c+d)}}$$

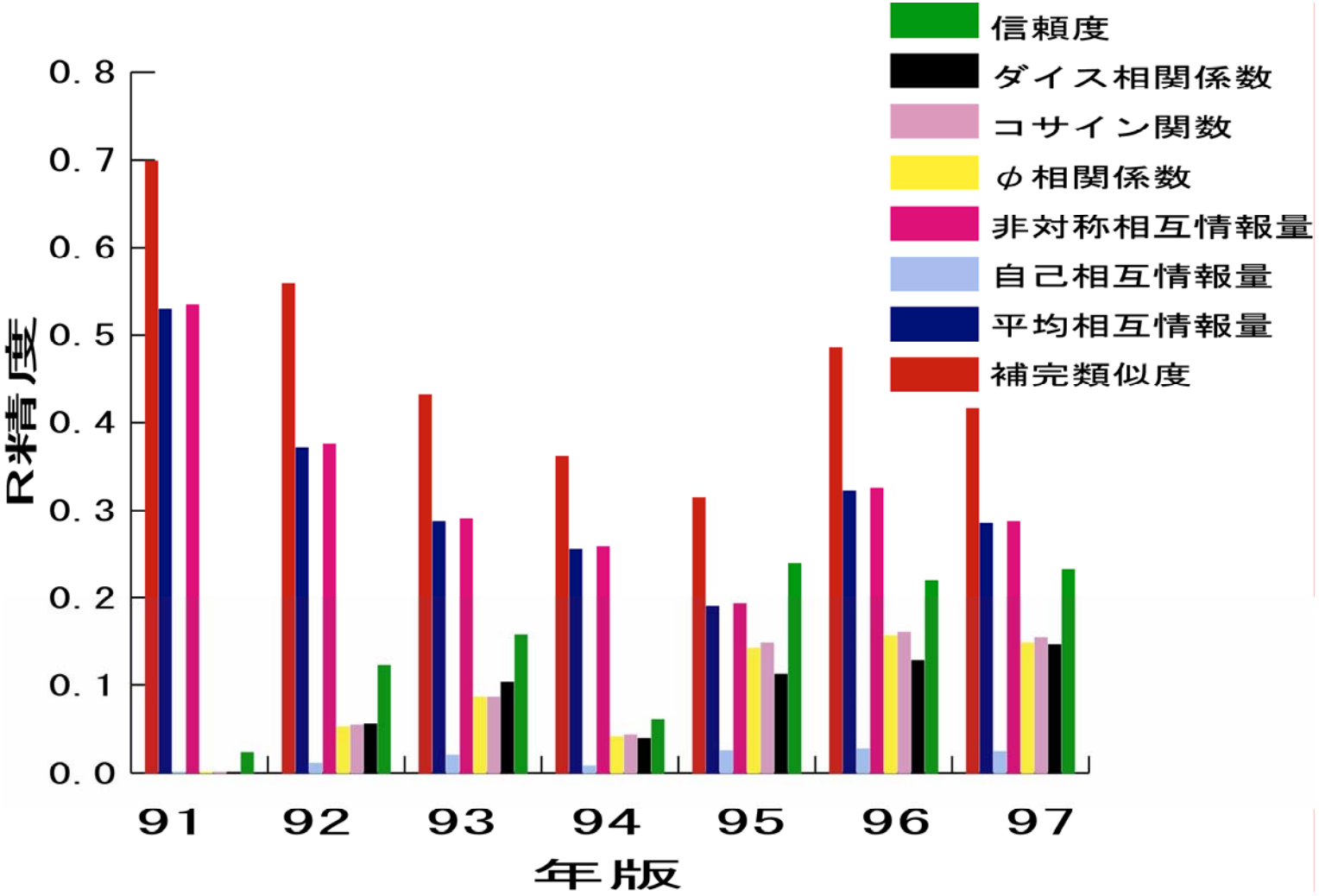
実データを使った調査

- 効果的な関数
- 山本英子、梅村恭司コーパス中の一対多関係を推定する問題における類似尺度自然言語処理　Vol. 9 No.2, pp45−75

下記では，df_a, df_b, df_c,df_dをa,b,c,dと記述

$$S_C(\vec{F},\vec{T}) = \frac{ad - bc}{\sqrt{(a+c)(b+d)}}$$

新聞記事での県名と市名との関係の実験
正解のわかった関係での検証



問題２

大規模なデータのすべての文字列から，
重要そうな文字列を選び出してください
（キーワードの抽出）

繰り返していることの分析

$\alpha = \frac{df(x)}{N}$ ： 文字列 x を含む文書の確率

$\beta = \frac{df2(x)}{df(x)}$ ： 反復度

N ： 全ドキュメント数

$df(x)$ ： 文字列x を含むドキュメントの数

$df2(x)$ ： 文字列x を2回以上含むドキュメントの数

反復度

「ある一つの単語をあるドキュメントが持つという条件で，同じ単語を二つ以上そのドキュメントが持つ確率」

着眼点：重要なことは繰り返す

$$\alpha = \frac{df(x)}{N} \quad : \quad \text{文字列 } x \text{ を含む文書の確率}$$

$$\beta = \frac{df2(x)}{df(x)} \quad : \quad \text{反復度}$$

「ある一つの単語をあるドキュメントが持つという条件で，同じ単語を二つ以上そのドキュメントが持つ確率」

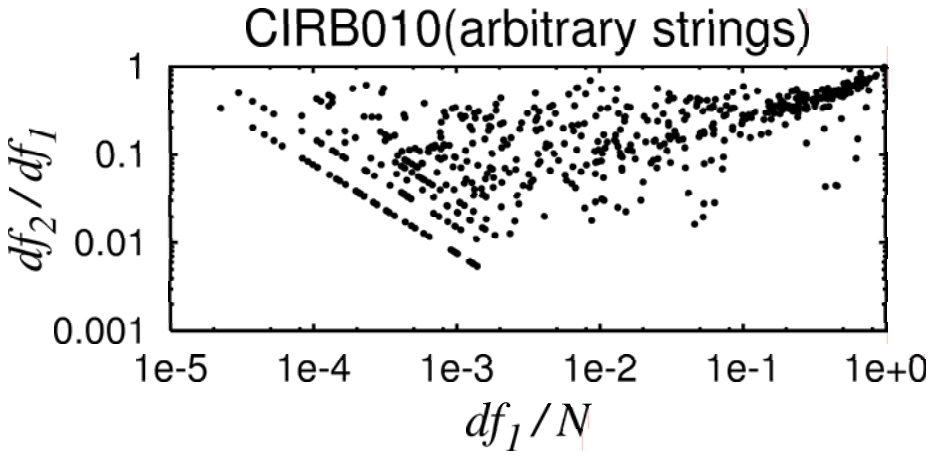
ドキュメントの表現したい主題にかかわる文字列は、ドキュメント集合中で反復度が大きいことが期待できる。

全文字列を対象とした分析

計算量：長さNのすべての部分文字列はN(N-1)/2 個
それぞれについて、文書頻度を求めたい。

実現した方法　前処理：O(N log(N)), 文字列ごとに:O(log(N))

x	df(x)	df2(x)
ロ	124696	79894
ロボ	3672	2413
ロボッ	3320	2237
ロボット	3320	2237
ロボットに	577	96
ロボットにつ	30	1
ロボットについ	30	1
ロボットについて	30	1



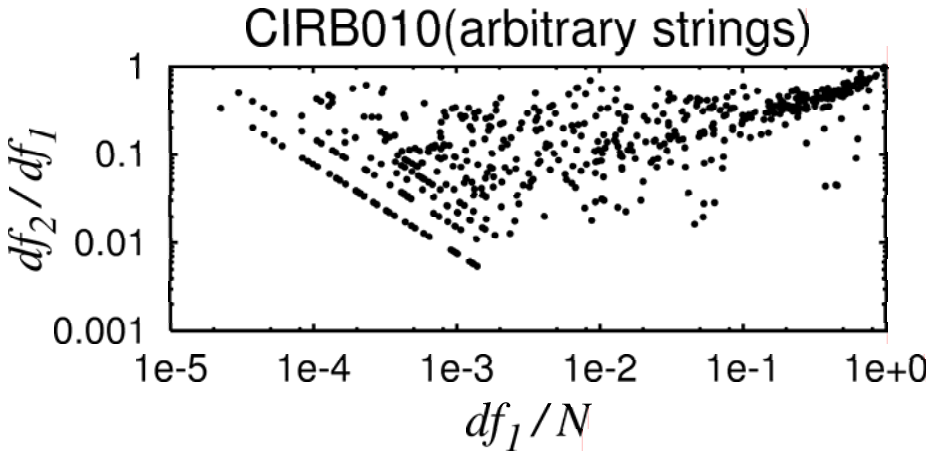
df2（x）：文字列xを2回以上含む文書数

アプローチ

あらかじめ表をつくっておいて、二分探索をする

実現した方法　　前処理：O(N log(N))，文字列ごとに:O(log(N))

x	df(x)	df2(x)
ロ	124696	79894
ロボ	3672	2413
ロボッ	3320	2237
ロボット	3320	2237
ロボットに	577	96
ロボットにつ	30	1
ロボットについ	30	1
ロボットについて	30	1



df2（x）：文字列xを2回以上含む文書数

表の大きさ

単純な方法だとN自乗の大きさの表

実現した方法　前処理：O(N log(N))，文字列ごとに:O(log(N))

x	df(x)	df2(x)
ロ	124696	79894
ロボ	3672	2413
ロボッ	3320	2237
ロボット	3320	2237
ロボットに	577	96
ロボットにつ	30	1
ロボットについ	30	1
ロボットについて	30	1

Nの自乗の表は実現不可能

同じ統計量をもつ文字列をまとめて
一つのクラスとして扱う。

df2（x）：文字列xを2回以上含む文書数

前処理の手間

単純な方法だと、クラス数・ドキュメント数の計算

実現した方法　前処理：O(N log(N))，文字列ごとに:O(log(N))

x	df(x)	df2(x)
ロ	124696	79894
ロボ	3672	2413
ロボッ	3320	2237
ロボット	3320	2237
ロボットに	577	96
ロボットにつ	30	1
ロボットについ	30	1
ロボットについて	30	1

←「ロボ」ではじまる文字列［ロボッ」、「ロボの」から寄せ集めて数を求める

df2（x）：文字列xを2回以上含む文書数

アルゴリズム

- Suffix Treeを利用した知られたアルゴリズムのSuffix Arrayでの実装
  - 実際上のメモリの桁違いの節約
- 文書中での出現集中を計測できるような統計量df2への適用
  - 梅村，真田：文字列をk回以上含む文書数の計数アルゴリズム，平成14年10月 自然言語処理第9巻第5号（43頁-70頁）
  - 注：より優れた実装あり：未踏2005年（岡野原）

実装したプログラムのメモリ負荷

ドキュメント数	10000	31623	100000	332918
グループの数	78K	230K	6816K	24018K
前処理	80M	234M	626M	2366M
分析処理	40M	116M	333M	1175M

単位：バイト

メモリの使用量はO(N)

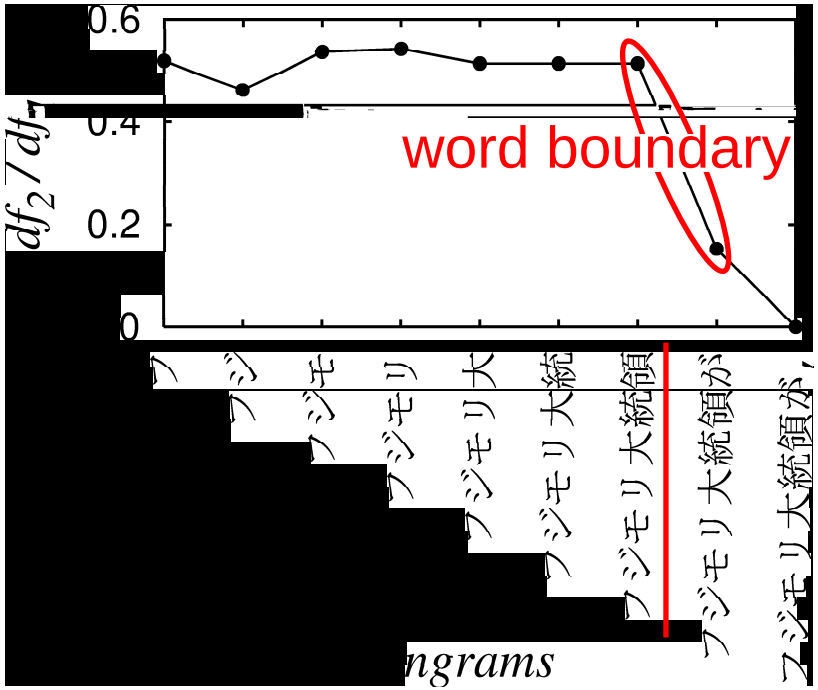
キーワード内部での振る舞い

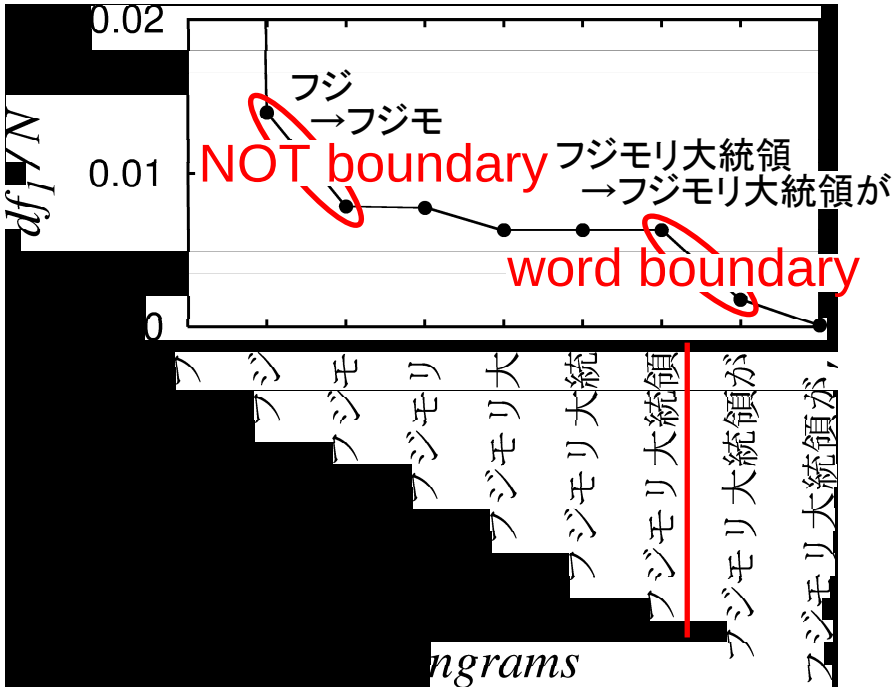
x	df(x)	df2(x)	α (= df/N)	β (= df2/df)
ロ	124696	79894	3.75E-01	6.41E-01
ロボ	3672	2413	1.10E-02	6.57E-01
ロボッ	3320	2237	9.97E-03	6.74E-01
ロボット	3319	2237	9.97E-03	6.74E-01
ロボットに	577	96	1.73E-03	1.66E-01
ロボットにつ	30	1	9.01E-05	3.33E-02
ロボットについ	30	1	9.01E-05	3.33E-02
ロボットについて	30	1	9.01E-05	3.33E-02

語の境界の内部でβ が一定

df2(x) : 文字列xを2回以上含む文書数

頻度と反復度の違い

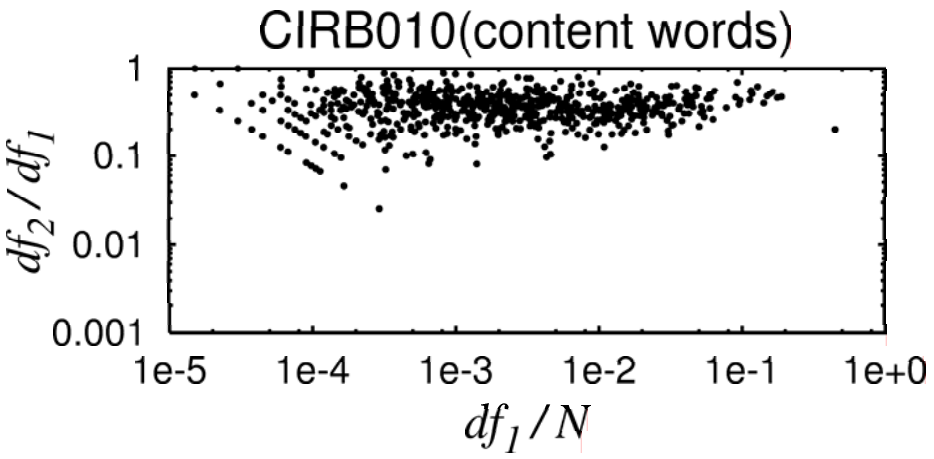


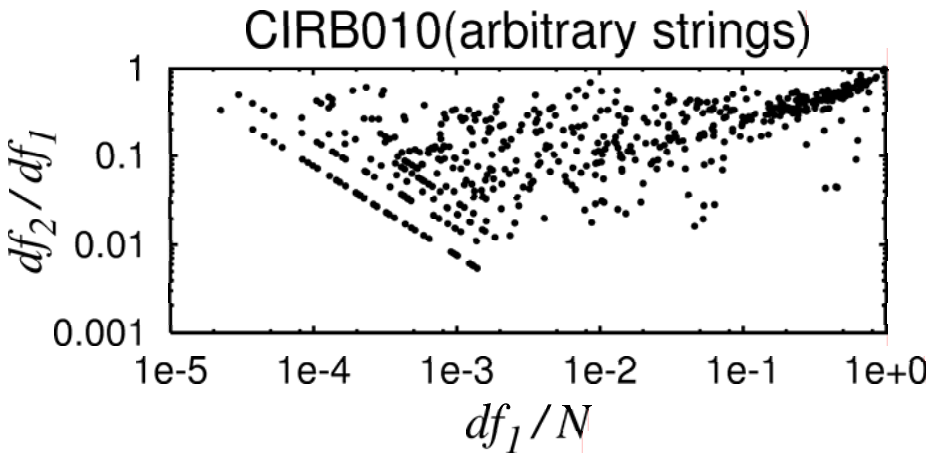


語の境界を検出しやすいのは反復度

df2(x) : 文字列xを2回以上含む文書数

キーワードと任意文字列の反復度





人間が選んだキーワード　　ランダムに選んだ文字列

観測：反復度に、キーワードを選択できる情報がある

df2（x）：文字列xを2回以上含む文書数

キーワードの抽出例1

論文アブストラクトの検索要求

自律移動ロボット自体の設計、開発、評価などが総合的に書かれた文献、または、自律移動ロボットにおける部分的なシステム（経路制御、物体認識など）の設計について書かれた文献が検索要求を満たす。自律移動はするがロボットではないものの設計、開発に関する論文も部分的に検索要求を満たす。自律分散システムなどの自律移動ロボットを応用したシステム、自律移動しないロボットに関する文献は検索要求を満たさない。

新聞記事のキーワード

日本語新聞からのキーワード

フジモリ大統領　三井海上火災　脳死判定
アメリカンフットボール　水道メーター
日本大使公邸占拠事件　アイルランド
Ｏ・Ｊ・シンプソン氏　カシオ計算機

コーパスの分野に依存しない。
（新聞でも、技術文献でも動作する）

略語・固有名詞を抽出

切り出し精度

正しい抽出語 (85.8[%]=97/113)	誤った抽出語 (14.2[%]=16/113)
PAC学習	エージェント	iアルゴリズム
クラスタリング	データベース	lient
データマイニング	ニューロン	イ論
ニューラルネットワーク	ロボット	ラルネットワーク
慣用表現	機械翻訳	経路制御	人工ニュー
構文解析	自動索引	自律	複合名
自律移動	自律移動ロボット	複合名詞の
自律分散	対話システム	報資源

問題３

大規模なデータの中で，同じように
に使われる文字列のペアをもれなく
もとめてください

（関連語の抽出）

関連語対の発見アルゴリズム

下記の組を辞書を使わずにテキストから選ぶアルゴリズム

- 表記の揺れを含む単語 「サーバ」と「サーバー」
- 省略形の単語 「メール」と「電子メール」
- 表記が異なる単語 「ウィルス」と「ウイルス」
- 文字種が異なる単語 「タンパク質」と「蛋白質」
- 文字コードが異なる単語 「靭性」と「靭性」

関連語対判定の
ヒューリスティックス

－辞書を用いない条件判定.

- 関連語を「前後に同じような文字列がある単語の組、つまり、テキスト中で同じように使用される単語の組」と定義する.

ヒューリスィックスの定式化

- **x,y:文字列. a,b:判定される単語. α:閾値.**
- **tf(z):文字列zの総出現頻度.**
- **df(z):文字列zが出現するテキスト数.**
- **N:テキストの総数.**
- **xay,xby:それぞれa,bの前後にx,yを結合した文字列**

$$IDF(z) = -\log(df(z)/N) \text{のとき},$$

$$score(a,b) = \sum_{x,y} tf(xay) \cdot IDF(xay) \cdot tf(xby) \cdot IDF(xby),$$

$$Relevants = \left\{(a,b) \mid \exists x \exists y (tf(xay) > 1 \wedge tf(xby) > 1) \wedge score(a,b) > \alpha \right\}$$

高いScoreを与える組：関連語
しかし，計算できるだろうか

$$score(a,b) = \sum_{x,y} tf(xay) \cdot IDF(xay) \cdot tf(xby) \cdot IDF(xby),$$

- **xay,xby:それぞれa,bの前後にx,yを結合した文字列**
- **IDF(z): -log(df(z)/N)**

– 表記の揺れを含む単語 「サーバ」と「サーバー」
– 省略形の単語 「メール」と「電子メール」
– 表記が異なる単語 「ウィルス」と「ウイルス」
– 文字種が異なる単語 「タンパク質」と「蛋白質」
– 文字コードが異なる単語 「靭性」と「靭性」

関連語対候補の一次選別

- 着眼点：前後の関係の総数はNを超えない
- O（N）の一次選別を行う

「少なくとも一つは、同じように使われた例がある。」

$$Bi(\alpha)=\left\{xy \mid \frac{P(xy)}{P(x)P(y)}>\alpha\right\},$$

$$F(c)=\{x \mid xc \in Bi(\alpha)\}, B(c)=\{y \mid cy \in Bi(\alpha)\}\text{のとき},$$

$$BF(a)=\{B(x) \mid x \in F(a)\}, FB(a)=\{F(y) \mid y \in B(a)\},$$

$$Candidates=\{(a,b) \mid b \in FB(a) \cap BF(a) \wedge a \neq b\}$$

関連度計算プログラムの速度

- 一組あたり1ミリ秒以下で判定する.
  - Suffix Arrayを用いた、全文字列の索引技術
  - 地道なプログラムのチューニング
  - 統計量、tf(文字列), df(文字列)のライブラリ

$$score(a,b) = \sum_{x,y} tf(xay) \cdot IDF(xay) \cdot tf(xby) \cdot IDF(xby),$$

•xay,xby:それぞれa,bの前後にx,yを結合した文字列

•IDF(z): -log(df(z)/N)

問題４

大規模の情報から，文字の挿入や削除があることを考慮して，与えられた文字列と似ている文字列をさがしてください

（類似情報検索）

DPマッチングによる情報検索

- 表記の揺れに対応する
  - 「語」の重みをどうするか？

    sim1：全文字‼均一，sim2：文字重み

    sim3:文字列重み
  - 「速度を」どうするか？
    - 一致をとる場所の間引き，

      FDP20: 20個の文字バイグラムによるDPマッチング

DPマッチングによる情報検索性能

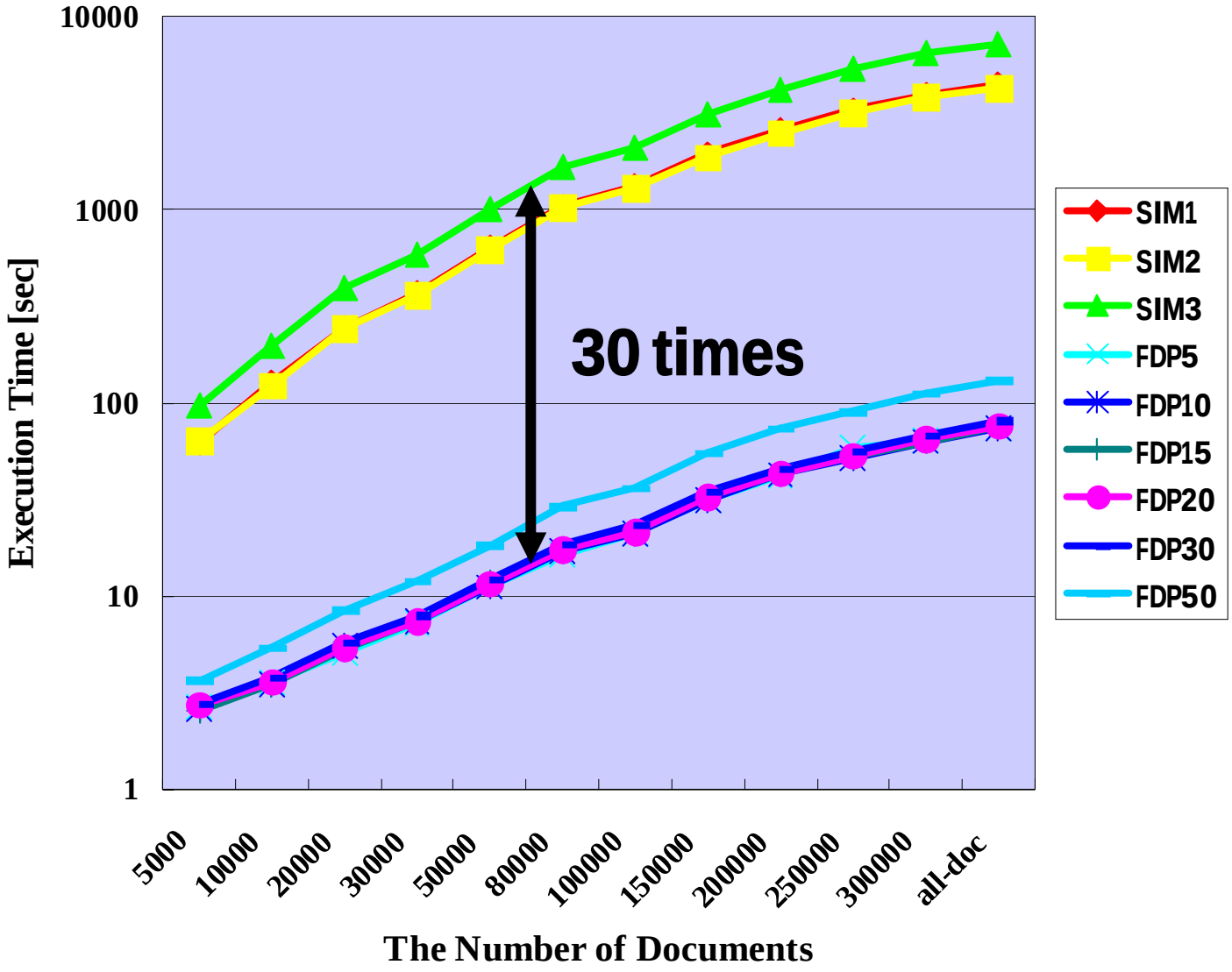
Topic01-30

Method	11 pt.	R-prec
SIM1	.1349	.1790
SIM2	.1948	.2296
SIM3	.2691	.3024
FDP5	.2547	.2649
FDP10	.2948	.3089
FDP15	.3109	.3446
FDP20	.3207	.3574
FDP30	.3176	.3421
FDP50	.3131	.3377
FDP500	.3172	.3419

Topic31-83

Method	11 pt.	R-prec
SIM1	.0545	.0845
SIM2	.1245	.1596
SIM3	.1807	.2083
FDP5	.1277	.1505
FDP10	.1766	.2031
FDP15	.2144	.2280
FDP20	.2398	.2621
FDP30	.2353	.2485
FDP50	.2354	.2488
FDP500	.2350	.2477

DPマッチングによる情報検索速度



類似情報検索エンジン

QuickSolution®

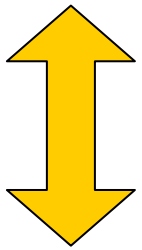
― 共同研究成果の商品化事例 ―

住友電工情報システム（株）

開発体制

豊橋技術科学大学　梅村研究室

：　コア・アルゴリズム開発



定期的な打合せ
アルゴリズム開発・評価依頼
特許の共同出願

住友電工／住友電工情報システム

- TUT成果のJava化、評価、製品への取り込み
- 拡張機能・応用機能の単独開発
- 特許出願

QuickSolutionの類似検索機能

ドキュメント群の中から自然文で入力された質問文と類似するドキュメントを検索する技術で、質問文と完全一致しなくても検索が可能です。

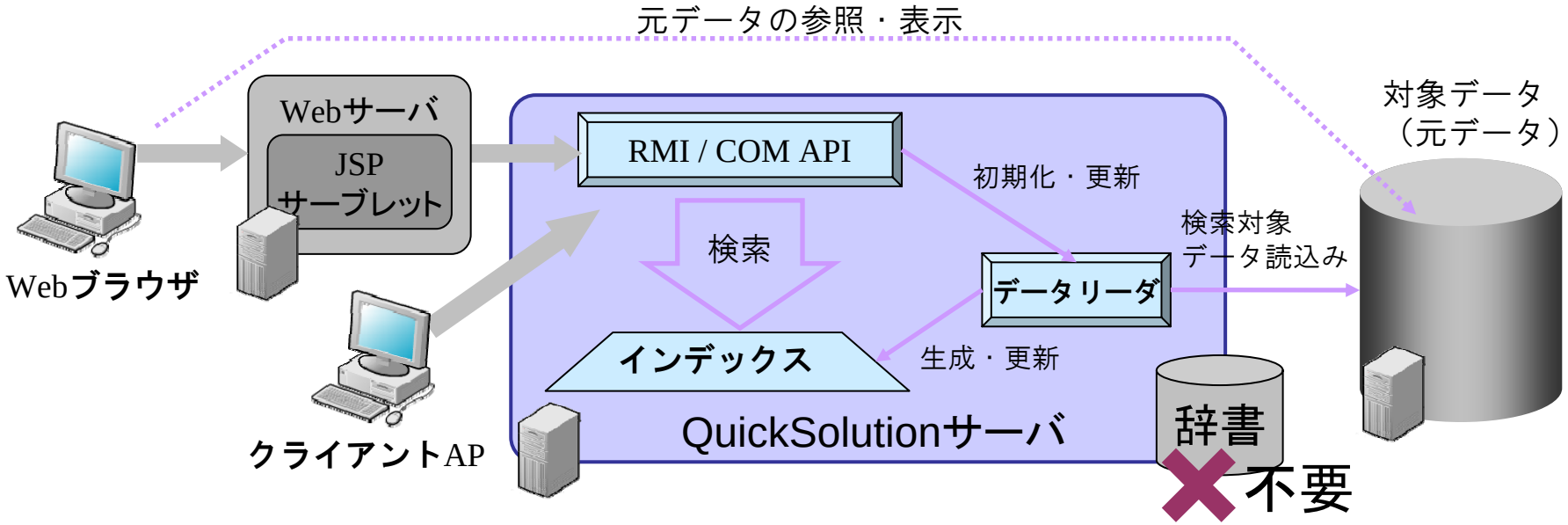
【検索例】

質問文：「デジタルカメラの画像をプリンタで印刷する方法は？」

類似度	検索結果
89%	デジタルカメラ画像をプリンターで印刷するには？
76%	デジカメ画像をプリンターで印刷するには？
67%	デジタルカメラの写真をプリンタで出力する方法について知りたい。
50%	プリンタでデジタルカメラ画像を印刷するにはどうすればよいでしょうか？
40%	デジタルカメラの使用方法について教えてください。

「*デジタルカメラ*」と「*デジカメ*」、「*プリンタ*」と「*プリンター*」等、表記の揺れを吸収。

QuickSolutionの概念図



- 高速検索処理用にインデックスを作成
- インデックスを使用して高速かつ高度な検索を実施
- 辞書が不要

QuickSolutionのアルゴリズム

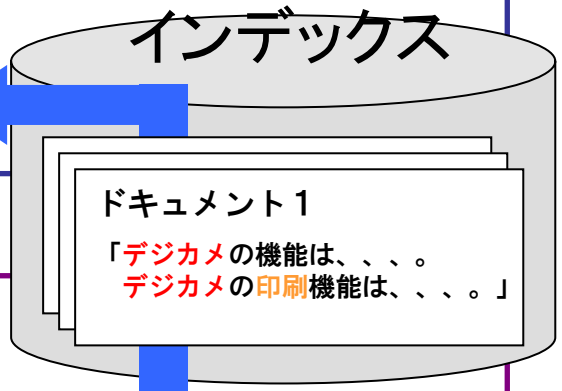
辞書、日本語処理を使用せず、純統計処理！

ステップ１：部分文字列の選別

質問文から出現頻度をもとに検索に有効な部分文字列を高速選別

- 検索精度を維持して検索処理を高速化
- 質問文が長文であっても大丈夫

（例）質問文：「デジタルカメラの画像をプリンタで印刷する方法は？」
⇒ 「デジタルカメラ」「画像」「プリンタ」「印刷」
＋「デジ」「ジタ」「タル」「ルカ」「カメ」「メラ」・・・



ステップ２：類似度の算出

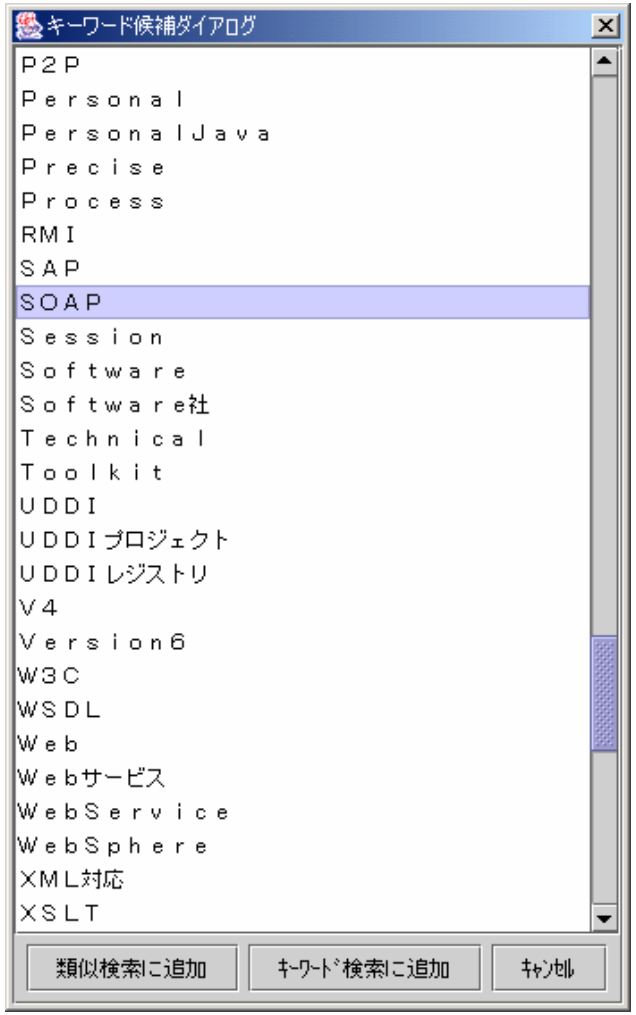
各部分文字列の出現頻度および出現集中度を考慮した重みを加算

- 出現頻度：小（出現するドキュメント数は限られる）
- 出現集中度：大（同じドキュメントに繰り返し出現）

（例）ドキュメント１の類似度
＝「デジ」＋「カメ」の重み（２回出現を考慮）＋「印刷」の重み

キーワード抽出と関連語抽出

ソースデータからキーワードを抽出

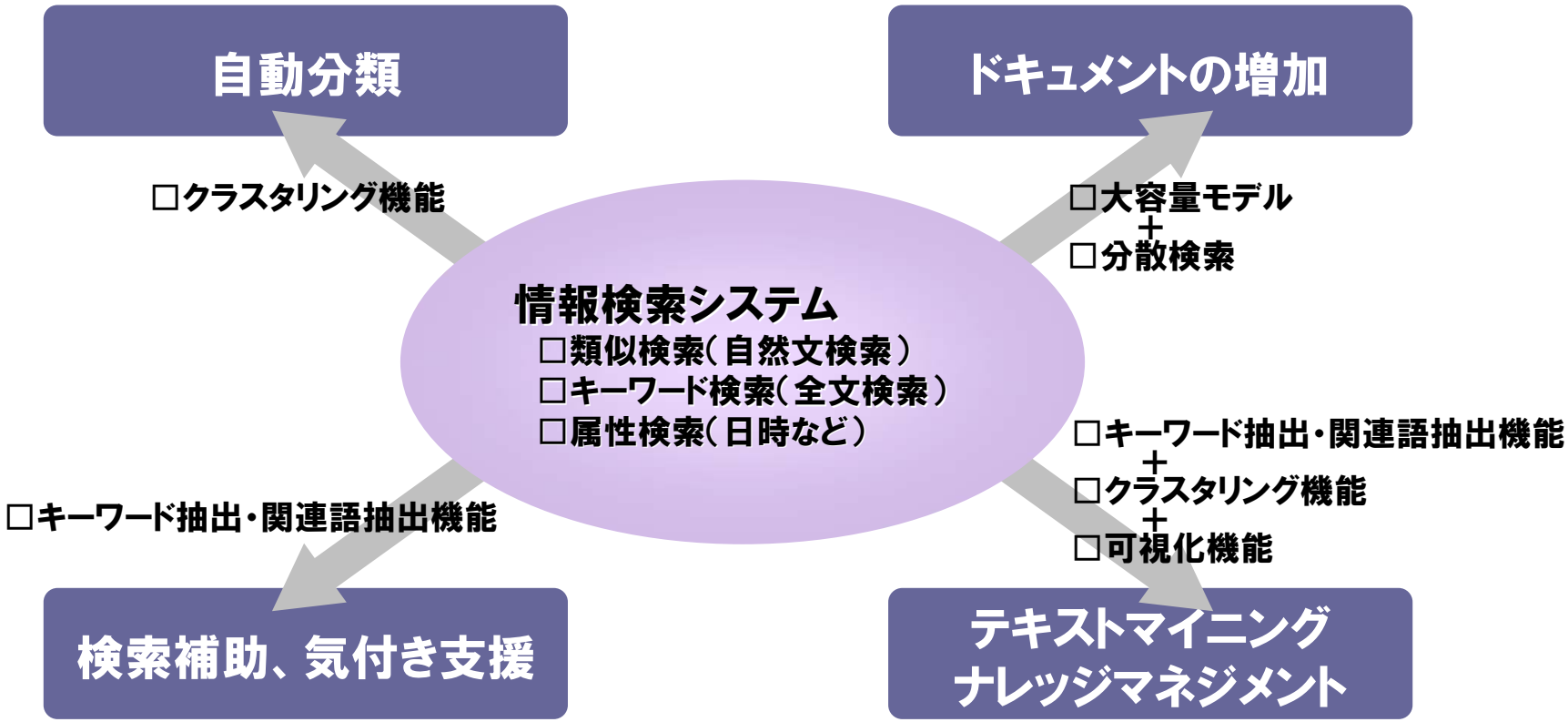


キーワード抽出結果を用いて関連語を自動抽出

キーワードを中心に関連語を抽出

QuickSolution® を用いた検索システムの拡張性

QuickSolutionの各種機能やオプションを利用して、検索のみならず様々なアプリケーションに拡張することが可能です。



＊上記の拡張には、別途、検索エンジン本体のアップグレードやオプション追加、システム開発費が必要となります。

活用分野･業務内容･導入目的など	業種･企業名･製品名	対象ドキュメントの種類
イントラネットのナレッジポータル	QuickSolution Portal 図研様 情報共有システム NEC様 BusinessPortal他	各種データソース XML-DB、文書ファイル 文書ファイル、RDB、グループウェア
Webサイト内検索	楽天トラベル様 音楽情報サイト、住友電工他	RDB、XML-DB、HTML、PDF等
メールアーカイブ／アクセスログ等の検索	エアー様 WISE Audit他	メール本文、添付ファイル、ログ等
FAQシステムへの適用	N1様 FAQ管理システム N2様 FAQ管理システム	RDB、文書ファイル
テキストマイニングツールへの適用	NRI様 TRUE TELLERのイントラネットキット･オプション	RDB、文書ファイル
データベースの全文検索	三井物産様 XML-DB NeoCore	XML-DB、文書ファイル
文書管理システム／ワークフロー	楽々Document/楽々WorkFlow	RDB、文書ファイル

楽天トラベル様：観光情報検索サイト

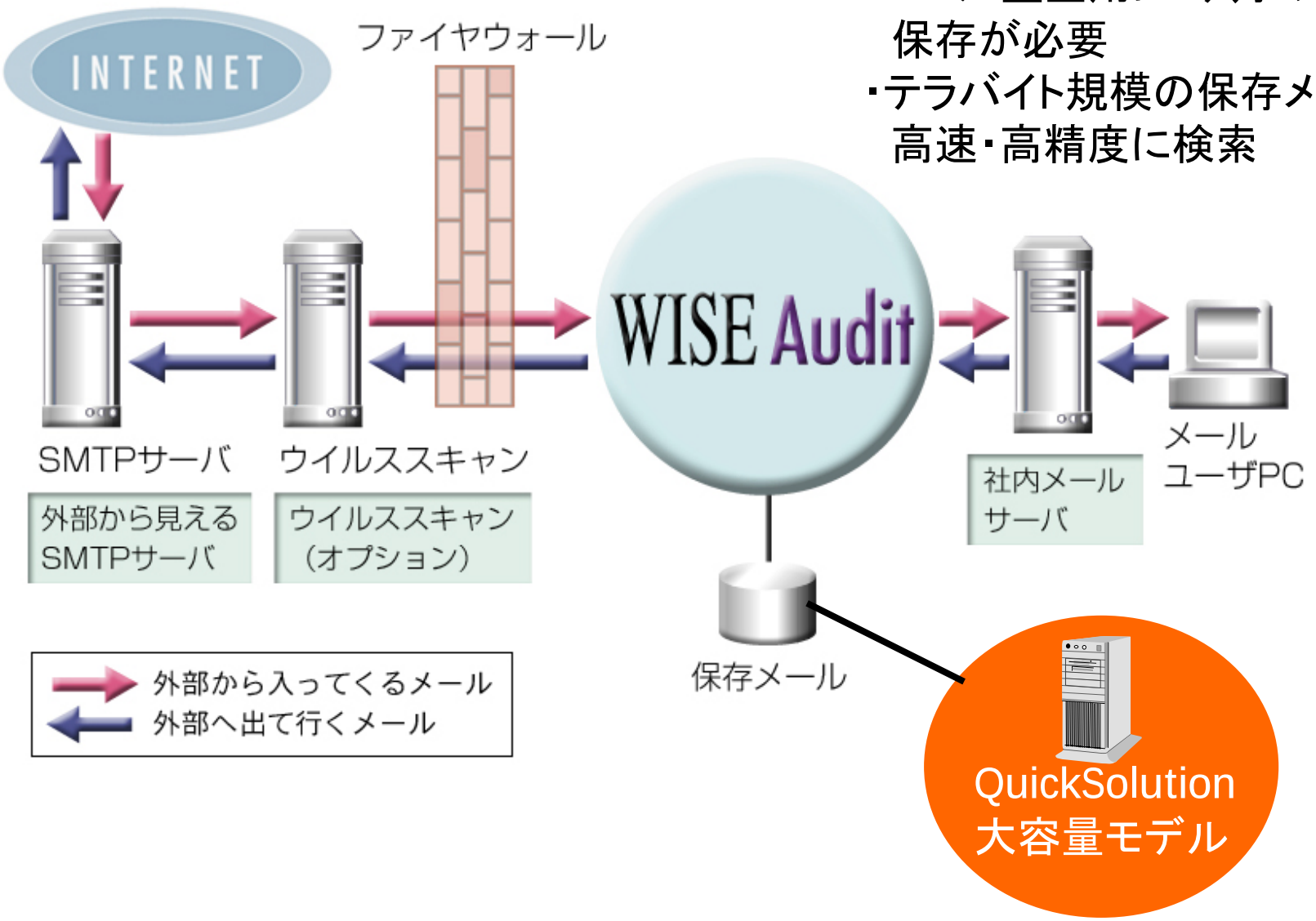
- 地名・施設名等
  固有名詞の検索

- 国内最大級の
  旅行予約サイト
  会員数：333万人
（2004/1/30）

- 24時間365日サービス

メール監査システム WISE Audit

・メール監査用に3ヶ月のメール保存が必要
・テラバイト規模の保存メールを高速・高精度に検索

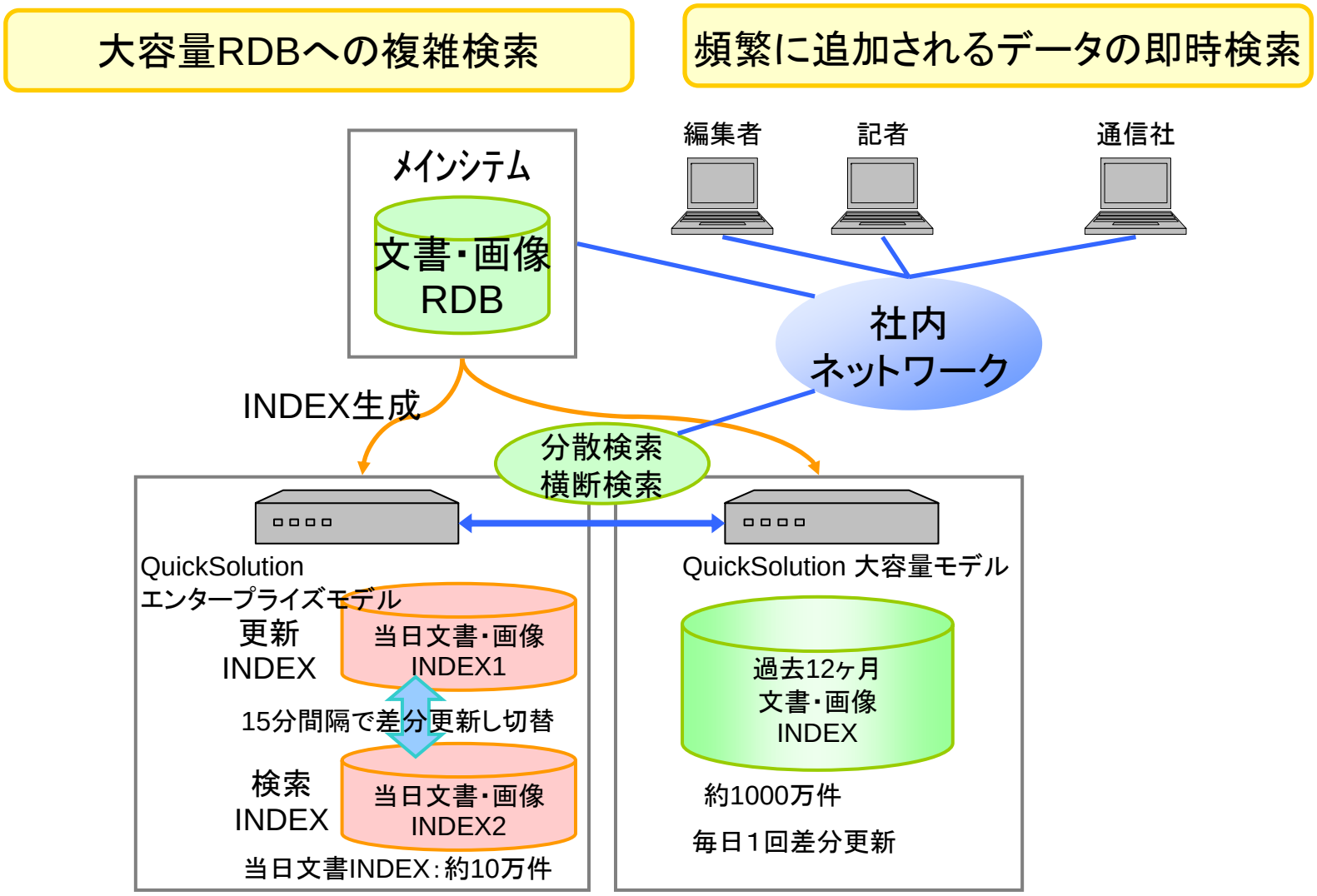


三井物産様：XML-DB NeoCore XMSへの組込み

NeoCore XML Management System

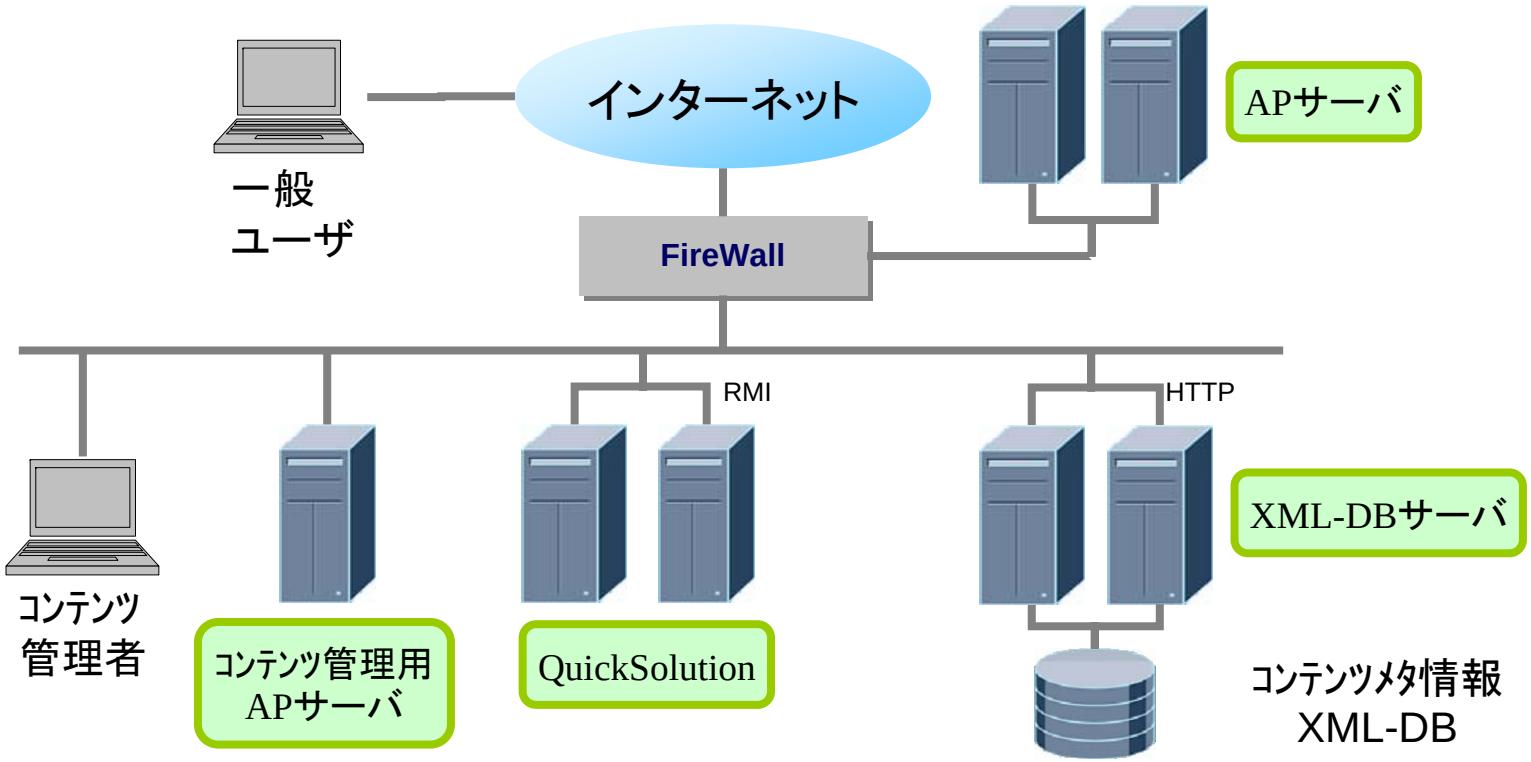
- NeoCoreに格納された大容量データを高速に全文検索、類似検索
- ファイル等を統合的に検索可能
- キーワード抽出等のデータ分析も可能

Case Study 1 : 膨大な記事を保有する報道A社



Case Study 2 : 大規模コンテンツを保有する音楽系B社

・類似検索および属性検索の両方を組み合わせて活用
・APIを用いてリアルタイム検索を実現
　属性検索インデックス：リアルタイム更新
　類似検索（全文検索）インデックス：「更新データを含めた検索機能」を使用
・一般ユーザ向け検索画面に加えて管理者向け専用検索画面を提供



製品化

- 速度が命
  - 情報検索の学会とは価値感が異なる
- 辞書を使わないのが命
  - 他社製品との明確な差別化できる
- ユーザへのカスタマイズが命
  - 欠点は一つもあってはならない

研究した技術は
役に立っているか？

- 見たことのある機能の桁違いの高速化の実現
  - DPマッチングによる情報検索
  - テラバイトクラスに対応
- 見たことのない機能の実現
  - 辞書を使わないで，キーワードを取り出す

研究した技術の
社会インパクト？

- 多くのひとに買ってもらえる技術ではない

＜でも＞

- 一部の人には，高額な対価を支払ってもらえるもの