Web 検索結果におけるキーワード出現相関の可視化と対話的な質問変換

武藤　克弥

富山県立大学　電子・情報工学科

June 11, 2021

はじめに

背景

Web 検索において，ユーザは 1 つまたは複数のキーワードを考えて検索する．その際，検索結果 Web サイトの見出し (スニペット) を 1 つ 1 つ眺めて適切なサイトを探す必要がある．また同じ単語で違う意味のものが検索結果に並んでいたり，絞り込みでかえって検索結果が限定されてしまうという問題があった．

目的

- 検索結果から抽出した単語をグラフ上に可視化し，一目で分かるようにする
- グラフ上の操作で，Web サイトを探せるようにする

Web 検索の現状

(1) 検索結果の可視化

情報の探索簡略化のために，検索結果をグラフにして可視化する技術は多くあるが，グラフを対話的に操作することは考慮されていなかった．

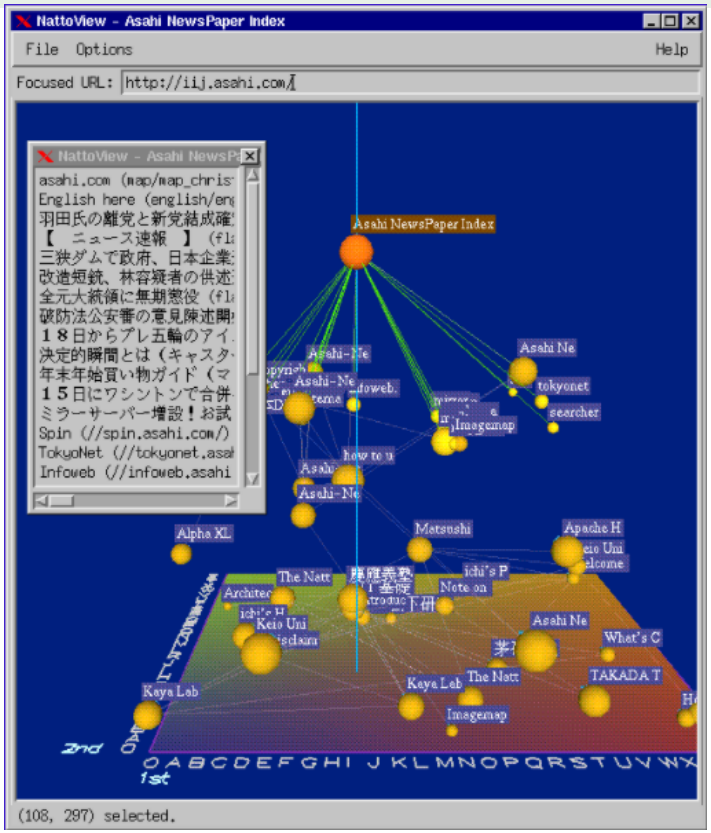


図 1: 可視化の例

Web 検索の現状 2

(2) 対話的な検索結果の修正

かつて存在した「121r」という検索結果を再ランキングするシステムでは，レーダチャート中の多次元の基準軸を操作して，再ランキングを行っていたが，基準軸を動的に変えられなかった．

例：オンライン会議用のホテル探し,「安い」軸と「高級感」軸
→「安い」軸の値を増やして再ランキング

提案システム

グラフ操作と動的に変わる基準軸で情報探索できるシステムの提案を行う

提案システムの概要

- 検索ワードを「クエリ」，検索結果のスニペットに出てくる重要語を「話題語」とする．
- 「話題語」と強い相関を示す語を「共起語」と定義する
- グラフ上には「クエリ」,「話題語」,「共起語」の３つを表示する

クエリ・スニペット

クエリ … 問い合わせ，要求の意味．検索ワードで検索結果のデータを問い合わせるところからきている

スニペット … 検索結果に表示されるサイトを要約した見出し文

図 2: スニペット

提案システムの概要２

グラフの概要

ノード … 単語が書いてある
クエリノード（赤）　　＝ 検索ワード
トピックノード (青)　＝ 話題語
緑色のノード ＝ 共起語
エッジ … クエリノードからトピックノードへの矢印

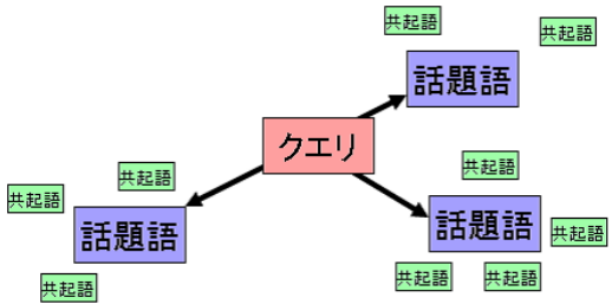


図 3: グラフのイメージ

提案システムの概要 3

提案システムの流れ

1. 検索ワードを入力 (1 キーワード)
2. Google の検索結果上位 100 件のスニペットを取得
3. 話題語と共起語を抽出 … I
4. 左側に抽出した単語の相関グラフ，右側に検索結果のスニペットを表示 (図 4)

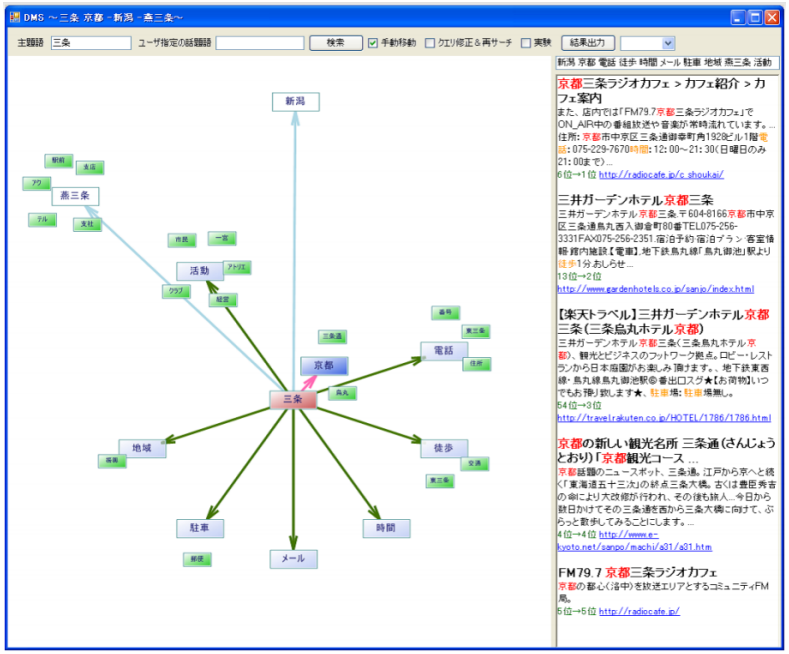


図 4: 実際のシステム図

提案システムの概要 3

提案システムの流れ

5. グラフ上のトピックノード (青) を操作 … II
6. 矢印の長さ (距離) に応じて再ランキング … III
7. 図 4 のグラフと検索結果が変化する
   →クエリノードに近い話題語が多く含まれるページが上位に来るようにランキングされる

I　話題語の抽出 (1)

話題語抽出アルゴリズム (前準備)

1. 上位 100 件のスニペットを形態素解析
   →抽出した単語を話題語候補とする

   ◎話題語候補になる単語： 名詞，その他 (数詞，非自立語，接頭語，接尾語以外の名詞)，未知語 (ひらがな，カタカナ，漢字になっているもの)
2. 話題語候補が含まれているスニペットの数 (DF 値) を計測

話題語抽出アルゴリズム

【話題語の定義】それを追加すると検索結果の話題を限定できる

0. 話題語候補 n 個に対して，DF 値の高い順に $T_i(1, 2, ..., n)$ という名前を付ける
   → T_1 から順に抽出確定語が 10 個たまるまで繰り返す
1. 100 個のスニペットを候補 T_1 の含む集合 P_1(正例)，含まない集合 N_1(負例) に分ける
2. 正例スニペットにおいて，話題語候補全ての DF 値を要素にしたベクトル $\overrightarrow{df_{P_1T}} = \{df_{P_1T_1}, df_{P_1T_2}, ..., df_{P_1T_n}\}$ を作る

   →負例スニペットも同様に $\overrightarrow{df_{N_1T}} = \{0, df_{N_1T_2}, ..., df_{N_1T_n}\}$
3. $\overrightarrow{df_{P_1T}}$ と $\overrightarrow{df_{N_1T}}$ のコサイン類似度を求める

$$cos\theta = \frac{\overrightarrow{df_{P_1T}} \cdot \overrightarrow{df_{N_1T}}}{|\overrightarrow{df_{P_1T}}||\overrightarrow{df_{N_1T}}|} \tag{1}$$

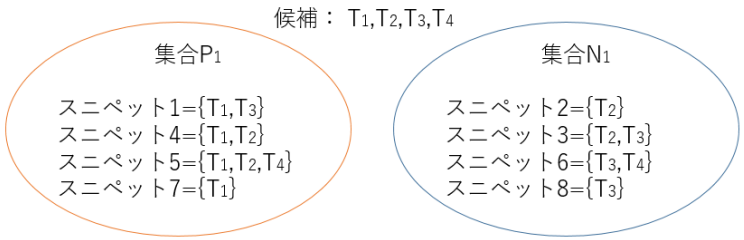
I　話題語の抽出 (3)

話題語抽出アルゴリズム 2

4. $cos\theta$ の値が $\theta <= 0.6$ となるような値 ($\overrightarrow{df_{P_1T}}$ と $\overrightarrow{df_{N_1T}}$ のなす角約 53 ℃以上　) なら T_1 を話題語としてトピックノードに追加

   →　T_i の入っている $\overrightarrow{df_{P_1T}}$ と入っていない $\overrightarrow{df_{N_1T}}$ の方向がかなり異なる　＝　話題語 T_i が入ることで話題の限定ができている

T_2 から T_n までトピックノードが 10 個になるまで 1〜4 を繰り返す



例 (上図)　$\overrightarrow{df_{P_1T}} = \{4, 2, 1, 1\}$,　$\overrightarrow{df_{N_1T}} = \{0, 1, 3, 2\}$

$$cos\theta_1 = \frac{4\cdot0+2\cdot1+1\cdot3+1\cdot2}{\sqrt{4^2+2^2+1^2+1^2}\sqrt{0^2+1^2+3^2+1^2}} = 0.39\cdots < 0.6$$

→追加

I　話題語に共起する共起語の抽出 (1)

共起後抽出アルゴリズム

【共起語の定義】話題語を追加すると増える傾向にある語

0. 共起語候補 C_i は話題語候補 T_i と同じものを用いる (T_1 から T_n まで繰り返す)
1. 100 個のスニペットを候補 T_1 を含む集合 P_1(正例)，含まない集合 N_1(負例) に分ける
2. T_1 を含む正例スニペットの総数を n_P，負例の総数を n_N として数える
3. T_1 に共起する共起語候補 $C_i(i = 1, ..., n-1)$ に対して，正例中の C_i の DF 値 df_{PC_i}，負例中の df_{NC_i} を求める
4. $n_P, n_N, df_{PC_1}, df_{NC_1}$ を用いて C_i の出現頻度に関するカイ 2 乗検定を行う (C_1 から C_n まで順番に行う)　$(n = n_P + n_N)$

I　話題語に共起する共起語の抽出 (2)

共起後抽出アルゴリズム 2

5. 帰無仮説「正例と負例では C_i の出現頻度は等しい」と仮定してカイ二乗値 S_i を計算

$$S_i = \frac{n\{df_{PC_i}(n_N - df_{NC_i}) - (n_P - df_{PC_i})df_{NC_i}\}^2}{n_P n_N (df_{PC_i} + df_{NC_i})\{n - (df_{PC_i} + df_{NC_i})\}} \quad (2)$$

6. 閾値以上で帰無仮説が棄却されれば，出現頻度が異なることが分かる ($=C_i$ が T_1 に共起されている傾向がある)
7. トピックノード T_1 のまわりに緑の共起語ノードを追加する (T_2 以降も同様に検証)

	C_i を含む	C_i を含まない	合計
正例	df_{PC_i}	$n_P - df_{PC_i}$	n_P
負例	df_{NC_i}	$n_N - df_{NC_i}$	n_N
合計	$df_{PC_i} + df_{NC_i}$	$n - (df_{PC_i} + df_{NC_i})$	n

Table 1: 検定に用いる分割表

II　トピックノードの操作 (1)

トピックノード操作による検索結果の修正

- トピックノードをクエリノードに近づけたり遠ざけたりすることで，AND と NOT の検索ワードを変更したり，検索結果の再ランキングが行われる
  →後から相関グラフと検索結果ページを修正することができる

AND 検索と NOT 検索

- AND 検索 … 検索窓に「富山 大学」のように半角スペースを入れて検索する方法
  →キーワードの絞り込みができるが，つなぎすぎるとかえって限定されたページしか出てこないことも
- NOT 検索 … 「大学 -文系」のようにマイナスを付けることで，その単語を除外して検索する
  → (OR 検索 …「富山 | 大学」でどちらか片方だけ含まれているサイトでもヒットさせる)

II　トピックノードの操作 (2)

トピックノード操作による検索結果の修正 2

- AND 検索 = 赤色のエッジ (矢印)　：クエリノードに一番近づけたとき
- NOT 検索 = 水色のエッジ　：かなり遠ざけたとき
- 緑エッジ　：再ランキングの対象

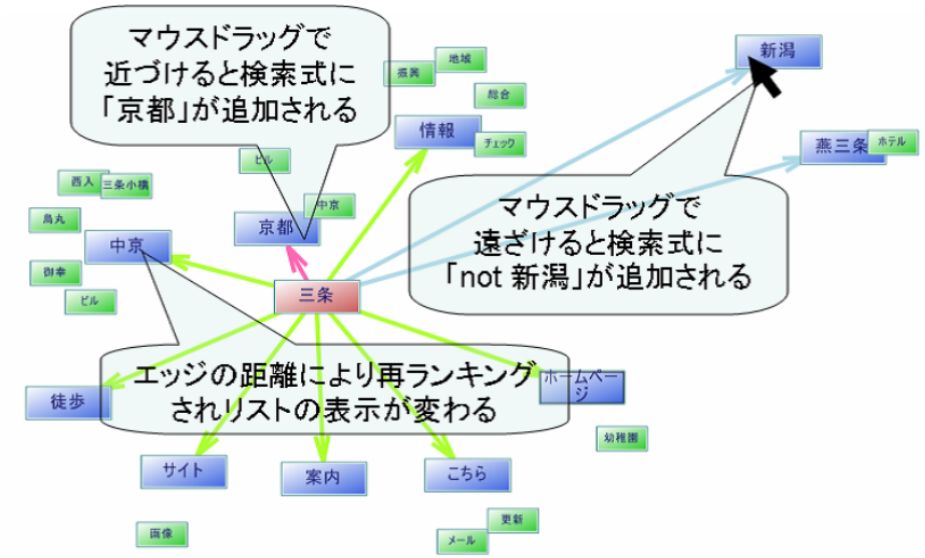


図 5: 検索結果の修正の様子

III　再ランキング手法 (1)

再ランキング手法

S_j：スニペットのスコア
d_i：クエリノードと各トピックノード (話題語)T_i との距離
$x_{ji} = 1$ (スニペット s_j が T_i を含むとき)
$x_{ji} = 0$ (含まないとき)

$$S_j = \sum_{i=1}^{n} \left(\frac{x_{ij}}{d_i} - \theta \right) \tag{3}$$

θ：閾値 $(= 0.003)$

- T_i を多く含むスニペットほど，クエリと距離の近い T_i を持つスニペットほどスコアが高くなる．
- スコアの大きいスニペットが上位に来るように再ランキングされる
  →それによってトピックノードも変化する

検証と考察 (1)

抽出した話題語・共起語の精度検証

- 比較対象として cos 類似度での足きりを行わず，DF 値の高さのみで抽出したものを用意
- クエリを「東西線」として検索

提案手法		DF 値のみ	
話題語	コサイン類似度	話題語	コサイン類似度
東京	0.552	東京	0.552
メトロ	0.564	メトロ	0.564
情報	0.550	地下鉄	0.618
不動産	0.542	情報	0.550
マンション	0.493	検索	0.622
路線	0.523	沿線	0.634
アパート	0.489	不動産	0.542
住宅	0.509	マンション	0.493
市営	0.567	路線	0.523
物件	0.573	アパート	0.489

図 6: 「東西線」の話題語抽出結果

話題語	話題語に共起する語
東京	メトロ
メトロ	東京
情報	アパート 検索 土地 不動産 マンション
不動産	アパート 住宅 情報 土地 マンション
マンション	アパート 住宅 情報 土地 不動産
路線	駅名 時刻
アパート	一戸建て 住宅 情報 土地 不動産
住宅	アパート 一戸建て おまかせ 土地 マンション
市営	札幌
駅名	御池 烏丸 クリック 時刻 選択

図 7: 話題語から共起した語

検証と考察 (2)

検索結果の修正検証

- 「京都」の「東西線」についての情報が欲しいユーザがトピックノードを操作
- 「東京」ノードを遠ざけて「東西線 not 東京」のクエリで検証

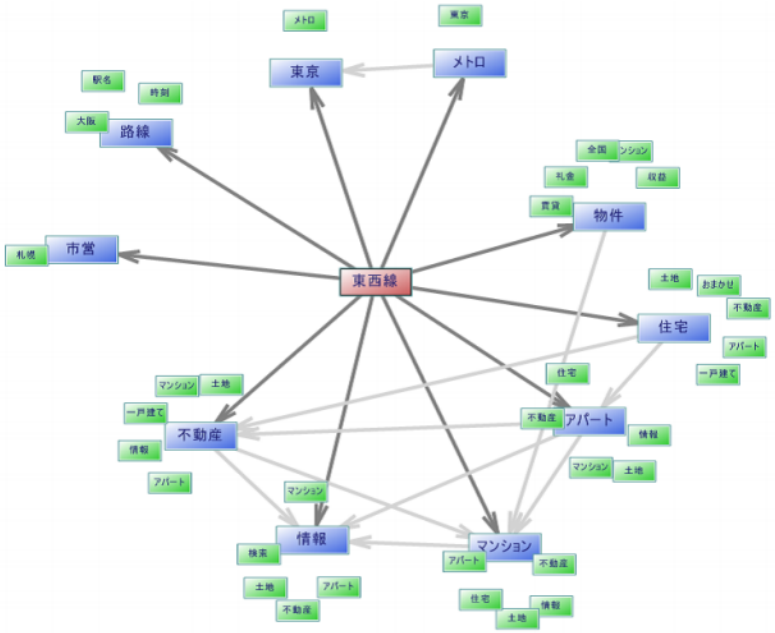


図 8: 「東西線の結果」

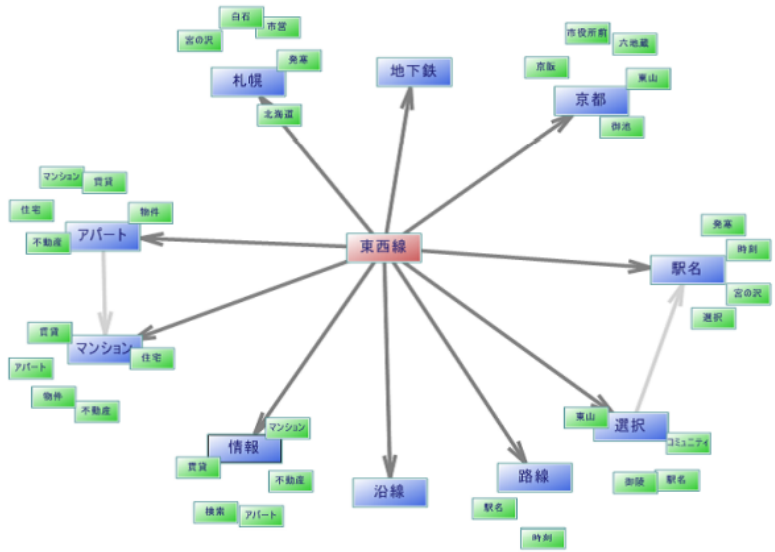


図 9: 「東西線 not 東京」の結果

検証と考察 (3)

トピックノード操作による検索結果の修正 2

図 9 の「京都」ノードを近づけて再ランキングすると，上位 10 件のほとんどが京都の「東西線」に関する検索結果だった．

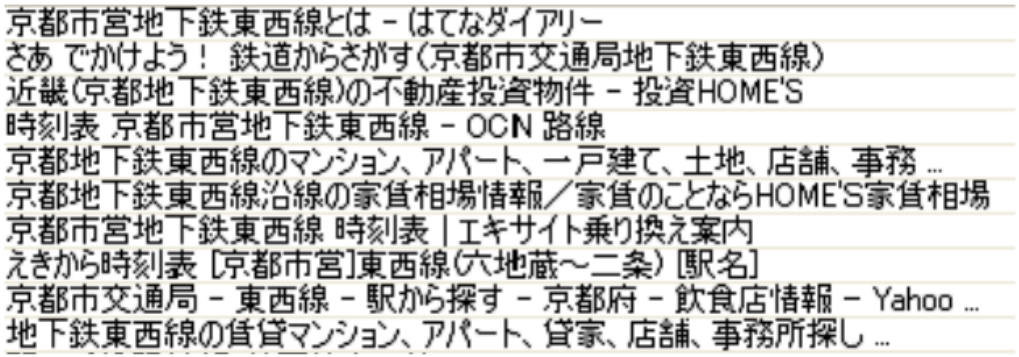


図 10: 検索結果の修正の様子

システムの拡張

「銀閣寺 金閣寺 清水寺」のような 3 つ以上含むようなクエリに対しても可視化を行った
→共通する話題語が共有されるようになり，複数のキーワードの共通点を見られるようになった．

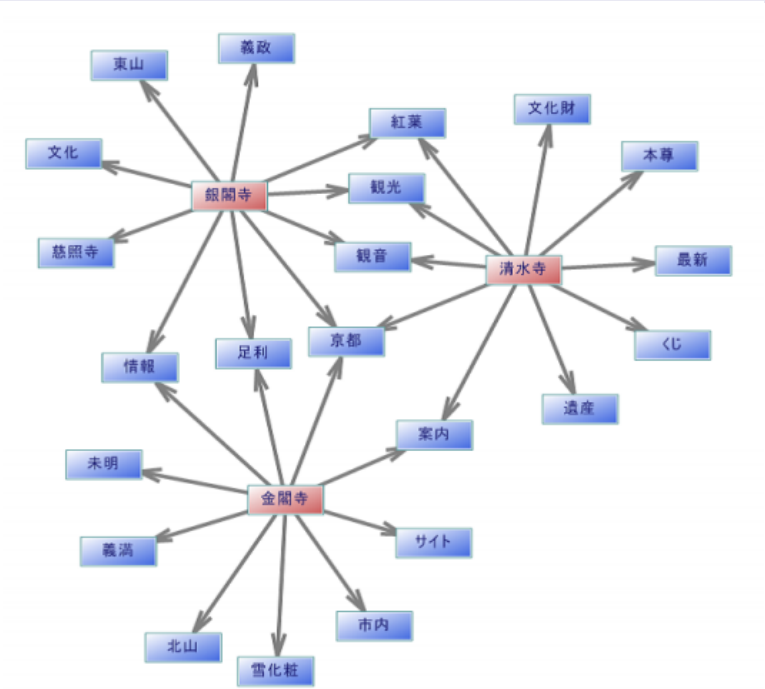


図 11: 拡張システムの結果

今後の課題

まとめ

- 検索結果をグラフ一つで可視化できるようにし，グラフ操作で Web サイトを探せることが確認できた

今後の課題

- 話題語候補の拡張： 商品のレビューなどに含まれる形容詞や観光地で何をするかの動詞を話題語にする
- 話題語抽出アルゴリズムの向上： まだまだ DF 値の高さに左右されていた
  →出現頻度の高さだけでなく，もっと意味のある話題を提供できるように改良