# PLSIを用いたウェブ検索結果の要約

原島 純　　黒橋 禎夫
京都大学大学院情報学研究科
{harashima,kuro}@nlp.kuee.kyoto-u.ac.jp

## 1　はじめに

現在ウェブには膨大な情報が蓄積されており，その量は日々増加している．このような膨大な情報の中から求める情報を効率よく取得するためには，検索エンジンの利用が必要不可欠である．しかし既存の検索エンジンは一般に，検索結果をリスト形式で提示するのみにとどまっている．このようなリスト形式の提示結果は誘導型のクエリに対しては有効であるが，調査型のクエリに対しては満足な結果を得るのが難しい．一方検索結果とともに，クエリに関するキーワードを提示するシステムも提案されている [1][2]．しかしキーワードは所詮単語や句であり，これらを単に列挙しただけでは，クエリとキーワードの関係や，キーワードとキーワードの関係まで把握することはできない．そこで本研究では検索結果に複数文書要約を適用することで，クエリに関する要約を生成し，情報を自然文で提示するシステムを構築することを目標とする．

検索結果に対して複数文書要約を適用するのは本研究が初めてではない．文書自動要約に関する代表的な文献でも，複数文書要約の適用先の一例として検索結果が挙げられている [3][4]．検索結果に対して複数文書要約を適用するためには，まず検索結果をトピックごとにクラスタリングする必要がある．これは，検索結果中にクエリに関する複数のトピックが混在しており，そのままでは複数文書要約を行うのに適さない状態だからである．先行研究として Radev らは，文書ベクトルの類似度を算出することで検索結果をトピックごとにクラスタリングし，生成された各文書クラスタについて重要文抽出に基づく複数文書要約を行うシステムを提案している [5]．また村上らも，新聞記事を検索対象として，同様のシステムを提案している [6]．

先行研究を参考に，提案システムでもまず検索結果をトピックごとにクラスタリングする．そして生成された各文書クラスタについて，重要文抽出に基づく複数文書要約を行う．ここで提案システムを構築する上で 2 つの問題がある．

**問題 1. どのように検索結果をクラスタリングするか**
先行研究では検索結果をトピックごとにハードクラスタリングしている．しかし実際には一つの文書が一つのトピックだけに帰属するとは限らない．そのため検索結果のクラスタリングはソフトクラスタリングにすべきである．

**問題 2. どのように重要文を抽出するか** 重要文抽出に基づく複数文書要約では，抽出する重要文間の冗長性を考慮する必要がある．そのため，抽出済みの重要文との類似度が低い文を抽出する，といった工夫を採るのが一般的である [7]．一方提案システムのように複数のトピックごとに重要文を抽出する場合は，トピック内だけでなく，トピック間の重要文の冗長性も考慮する必要がある．しかしこの問題に対処する方法は自明ではない．

そこで本研究では PLSI [8] を用いてこれらの問題に対処する．すなわち，PLSI を用いて各トピックに対する各文書の帰属度を推定し，これを用いて検索結果をソフトクラスタリングする．同様に PLSI を用いて各トピックに対する各キーワードの帰属度を推定し，これを用いてトピック間の重要文の冗長性を削減する．

## 2　提案システムのアルゴリズム

図 1 に，提案システムのアルゴリズムの概要を示す．以下ではアルゴリズムの各ステップについて詳説する．

### 2.1　検索結果の取得

ユーザからクエリが入力されると，提案システムはまず，検索エンジン TSUBAKI [9] の API を用いて，クエリに対する検索結果を取得する．TSUBAKI は次世代検索研究をサポートするために構築された検索エンジンであり，日本語ウェブ文書 1 億件を検索対象としている．提案システムでは，TSUBAKI のランキング上位 1000 件を検索結果として取得する．ただし，ほとんど画像のみで構成されている文書など，文章量が

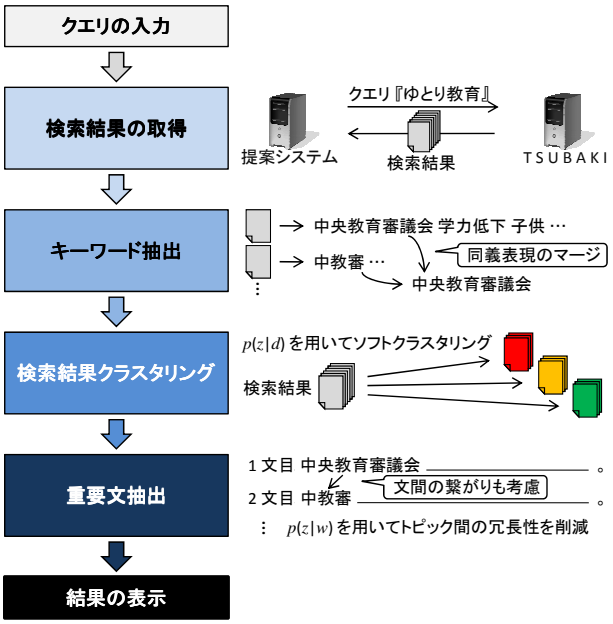


図 1: アルゴリズムの概要

乏しい文書はフィルタリングする．以降フィルタリング後の文書集合を $D$ とする．

## 2.2 キーワード抽出

馬場らの手法 [2] を用いて，$D$ からクエリに関するキーワードを抽出する．具体的には，まず各文書からクエリに関連する文を最大 15 文抽出する．次に各文から，複合名詞や括弧で囲まれた文字列をすべて抽出する．さらに抽出した文字列について，同義表現のマージ，表記揺れの解消等を行う．最後に残った各文字列についてクエリとの関連度を算出し，上位 100 個をクエリに関するキーワードとする．以降抽出されたキーワード集合を $W$ とする．

## 2.3 検索結果クラスタリング

PLSI を用いて $D$ をトピックごとにクラスタリングする．PLSI では，トピック $z$ のもとで文書 $d$ と単語 $w$ の生起を独立とし，$d$ と $w$ の同時確率 $p(d,w)$ を次のように表す．

$$p(d,w) = \sum_z p(z)\, p(d|z)\, p(w|z)$$

そして次式で算出される対数尤度関数 $L$ を最大とする $p(z), p(d|z), p(w|z)$ を，EM アルゴリズムを用いて推定する．

$$L = \sum_d \sum_w n(d,w) \log p(d,w)$$

ただし $n(d,w)$ は $d$ 中に $w$ が出現した頻度を表す．また E ステップと M ステップの式は以下の通りである．

**E ステップ**

$$p(z|d,w) = \frac{p(z)\, p(d|z)\, p(w|z)}{\sum_{z'} p(z')\, p(d|z')\, p(w|z')}$$

**M ステップ**

$$p(z) = \frac{\sum_d \sum_w n(d,w)\, p(z|d,w)}{\sum_d \sum_w n(d,w)}$$

$$p(d|z) = \frac{\sum_w n(d,w)\, p(z|d,w)}{\sum_{d'} \sum_w n(d',w)\, p(z|d',w)}$$

$$p(w|z) = \frac{\sum_d n(d,w)\, p(z|d,w)}{\sum_d \sum_{w'} n(d,w')\, p(z|d,w')}$$

提案システムでは $D$ と $W$ を入力として PLSI を実行し，$p(z), p(d|z), p(w|z)$ を推定する．ただし $d \in D, w \in W$ とする．各パラメータの初期値として，$p(z)$ には $1/K$，$p(d|z), p(w|z)$ にはランダムな値を与える．ただし $K$ はトピック数とする．また EM アルゴリズムの終了条件は，$L$ のゲインが閾値 $\alpha$ 以下になるまで，とする．現在は $L$ が十分収束する値として，$\alpha = 1$ としている．

PLSI を実行するためにはさらに，予め $K$ を与えておく必要がある．しかし，検索結果中に含まれるクエリに関するトピックの数を前もって知ることはできない．そこで提案システムでは複数の $K$ の値について PLSI を実行し，各 $K$ に対する AIC を算出することで $K$ を決定する．具体的には $K = 3$〜5 の 3 つの値について PLSI を実行し，次式で算出される AIC が最小となる $K$ を採用する．

$$AIC = -2L + 2K(|D| + |W|)$$

同時に，採用した $K$ における $p(z), p(d|z), p(w|z)$ の値を，PLSI の結果として採用する．

次に PLSI によって推定された $p(z), p(d|z)$ を用いて $p(z|d)$ を算出する．

$$p(z|d) = \frac{p(d|z)\, p(z)}{\sum_{z'} p(d|z')}$$

$p(z|d)$ は，トピック $z$ に対する文書 $d$ の帰属度を表す．そして各トピックについて，帰属度 $p(z|d)$ が閾値 $\beta$ 以上の文書を収集し，文書クラスタを生成する．現在は $\beta = 1/K$ としている．複数のトピックについて閾値以上の帰属度を持つ文書は，各トピックに割り当てる．

## 2.4 重要文抽出

2.3 節で生成された各文書クラスタについて，文中に含まれるキーワードの重要度と，文と文の繋がりを

表 1: $c\_score(w, s, S_z)$ の値

| | $S_z$ 中に含まれる | $S_z$ 中に含まれない |
|---|---|---|
| $s$ の主題 | 2 | -2 |
| $s$ の主題領域 | 0 | -1 |
| それ以外 | 0 | 1 |

※本研究では，係助詞「は」が付属する表現を主題，主題に係る領域を主題領域と呼ぶ．

考慮して，重要文を抽出する．トピック $z$ に対する文 $s$ の重要度 $s\_score(z, s)$ は，次式を用いて算出する．

$$s\_score(z, s) = \sum_{w \in W_s} w\_score(z, w)\ c\_score(w, s, S_z)$$

ただし $W_s$ は $s$ 中に含まれるキーワードの集合を表す．

$w\_score(z, w)$ は，トピック $z$ に対するキーワード $w$ の重要度を表す．ここで $w\_score(z, w)$ にどのような尺度を用いるか，という問題がある．まず考えられるのは，2.3 節の PLSI によって推定された $p(w|z)$（トピック $z$ におけるキーワード $w$ の生起確率）である．しかしクエリに関して一般的なキーワードは，複数のトピックにおいて高い $p(w|z)$ を持つ．それゆえ $w\_score(z, w)$ として $p(w|z)$ を用いると，各トピックについてこれらのキーワードを含む文が重要文として抽出され，トピック間の重要文に冗長性が生じてしまう．そこで提案システムでは $w\_score(z, w)$ として $p(z|w)$（トピック $z$ に対するキーワード $w$ の帰属度）を用いる．$p(z|w)$ は，PLSI によって推定された $p(z), p(w|z)$ を用いて算出する．

$$p(z|w) = \frac{p(w|z)\ p(z)}{\sum_{z'} p(w|z')}$$

$w\_score(z, w)$ として $p(z|w)$ を用いることで，各トピックに特徴的なキーワードに高い重要度を与えることができる．その結果，各トピックに特徴的な文を重要文として抽出しやすくなり，トピック間の重要文の冗長性を削減することができる．

$c\_score(w, s, S_z)$ は，抽出済みの重要文集合 $S_z$ が与えられた時の，文 $s$ におけるキーワード $w$ の重要度を表す．この値は，$s$ における $w$ の出現位置と，$w$ が $S_z$ 中に含まれるか否か，によって決定する．表 1 に $c\_score(w, s, S_z)$ の現在の値を示す．例えば $w$ が $s$ の主題で，$S_z$ 中に含まれていれば，$c\_score(w, s, S_z)$ の値は 2 とする．このように $S_z$ 中に含まれるキーワードを主題に持つ文の重要度を高くすることで，抽出済みの重要文と繋がりのある文を抽出しやすくする．また，例えば $w$ が $s$ の主題でも主題領域でもなく，$S_z$ 中に含まれていなければ，$c\_score(w, s, S_z)$ の値は 1 とする．このように $S_z$ 中に含まれないキーワードを持つ文の重要度を高くすることで，トピック内における重要文間の冗長性を削減する．


図 2: 提案システムの実行例（クエリ『ゆとり教育』）

トピック $z$ について抽出する重要文の数 $num(z)$ は，$p(z)$ に従って次の通り決定する．

$$num(z) = \begin{cases} \lfloor N * p(z) \rfloor & (p(z) \geq 0.2) \\ \lfloor N * 0.2 \rfloor & (p(z) < 0.2) \end{cases}$$

トピックの生起確率 $p(z)$ が大きいほど，多くの重要文を抽出する．ただし $p(z)$ がどれだけ小さくても，最低 $\lfloor N * 0.2 \rfloor$ 文は抽出する．また $N$ は各トピックについて抽出する重要文の合計を制御する値である．提案システムは実行結果をウェブブラウザ上に出力する．ウェブブラウザ上で，全トピックに対する要約を一目で俯瞰できる文の数が 10 文程度であったため，現在は $N = 10$ としている．

以上 2.1 節〜2.4 節の各ステップを経て，検索結果からクエリに関する要約を生成する．提案システムの実行例を図 2 に示す．

## 3 評価実験

まずアルゴリズムの有効性を調査した．提案システムにおいて特徴的な処理は，$p(z|d)$ を用いた検索結果クラスタリングと，$p(z|w)$ を用いたトピック間の重要文の冗長性削減である．しかしこれらを直接評価するのは難しい．これは，1000 文書からなる検索結果から，正解となるクラスタリング結果，及び正解となる要約を作成するのに，高いコストがかかるからである．そこで科研「情報爆発 IT 基盤」で行った共通ユーザ評価にて，アンケートによる間接的な評価を行った．具体的には，理系の大学生や IT 系の会社に勤める社会人を中心として 49 人の被験者に提案システムを利用させ，次の点を 5 段階で評価させた．

**Q1. 各トピックの内容はまとまっているか** 本来分かれるべきトピックがマージされていないか．逆にマージされるべきトピックが分かれていないか．

表 2: Q1 に対するアンケート結果

| 選択項目 | 人 (%) |
|---|---|
| 1. まとまっている | 16 (32.7) |
| 2. ある程度まとまっている | 21 (42.9) |
| 3. どちらとも言えない | 7 (14.3) |
| 4. あまりまとまっていない | 3 (6.1) |
| 5. まとまっていない | 2 (4.1) |

表 3: Q2 に対するアンケート結果

| 選択項目 | 人 (%) |
|---|---|
| 1. 要約提示の方が有効 | 20 (40.8) |
| 2. どちらかと言うと要約提示の方が有効 | 14 (28.6) |
| 3. どちらとも言えない | 5 (10.2) |
| 4. どちらかと言うとキーワード提示の方が有効 | 6 (12.2) |
| 5. キーワード提示の方が有効 | 4 (8.2) |

次にクエリに関する情報を収集する上で，提案システムがどの程度有効かを調査した．具体的には，提案システムの通常の実行結果と，各トピックについてキーワードだけを提示した結果を比較させ，次の点を 5 段階で評価させた．

**Q2. 要約提示とキーワード提示のどちらが有効か**
クエリに関する情報を収集する上で，どちらの提示結果が有効だったか．

表 2・表 3 に，各質問に対するアンケート結果を示す．表 2 を見ると，75% 以上の被験者が「まとまっている」もしくは「ある程度まとまっている」と回答している．このことから，$p(z|d)$ を用いた検索結果クラスタリング，及び $p(z|w)$ を用いたトピック間の重要文の冗長性削減が，うまく機能していることが分かる．一方，低評価のアンケート結果を調査したところ，マージされるべきトピックが分かれている，という意見が複数存在した．これは，AIC が $K$ を多く見積もる傾向がある，というところに一因があると思われる．今後は $K$ を決定するために他の尺度を用いることも検討する必要がある．

また表 3 を見ると，約 70% の被験者が「要約提示の方が有効」もしくは「どちらかというと要約提示の方が有効」と回答している．これにより，クエリに関する情報を収集する上で，提案システムがある程度有効であると言える．

さらに Q1・Q2 とは別に，提案システムに対する意見を募った．その結果，重要文の出典がわからないためその情報が信頼できない，という意見が複数寄せられた．これは，新聞記事等の信頼できる文書を対象とした要約システムと違い，ウェブ文書を対象とする提案システムならではの問題と言える．これらの意見を参考に，今後は各重要文が抽出された文書のタイプ（e.g. ニュースサイト，ブログ，…）を判別・提示する必要があると思われる．同時に，判別した文書タイプを利用して，信頼性を考慮した重要文抽出の方法も検討する必要がある．

# 4 おわりに

本研究では検索結果に複数文書要約を適用することでクエリに関する要約を生成し，情報を自然文で提示するシステムを構築した．提案システムでは PLSI を用いて各トピックに対する各文書の帰属度を推定し，これを用いて検索結果をソフトクラスタリングする．同様に PLSI を用いて各トピックに対する各キーワードの帰属度を推定し，これを用いてトピック間の重要文の冗長性を削減する．アンケートによる評価実験の結果，アルゴリズムがうまく機能していることが分かった．またクエリに関する情報を収集する上で，提案システムがある程度有効であることが分かった．

今後の課題として，3 節で述べたように，他のモデル選択尺度の利用や，文書の信頼性の判断・利用が挙げられる．また文書単位の信頼性だけでなく，文体・モダリティを手掛かりとして，文単位の信頼性も利用することを検討している．

# 参考文献


[1] Clusty. http://clusty.com/.

[2] 馬場康夫, 新里圭司, 柴田知秀, 黒橋禎夫. キーワード蒸留型クラスタリングによる大規模ウェブ情報の俯瞰. 情報処理学会論文誌, Vol. 50, No. 4, pp. 1399–1409, 2009.

[3] Mani Inderjeet. *Automatic Summarization*. John Benjamins Publishing Company, 2001.

[4] 奥村学, 難波英嗣. テキスト自動要約. オーム社, 2004.

[5] Dragomir R. Radev and Weiguo Fan. Automatic summarization of search engine hit lists. In *Proc. of ACL 2000 Workshop on Recent advances in NLP and IR*, pp. 1361–1374, 2000.

[6] 村上浩司, 野畑周, 関根聡, 井佐原均. 新聞記事を対象にした，検索，分類，複数文書要約システム ELIOT システム．言語処理学会 第 10 回年次大会発表論文集, pp. 143–146, 2004.

[7] J. Goldstein, V. Mittal, J. Carbonell, and M. Kantrowitz. Multi-Document Summarization By Sentence Extraction. In *Proc. of ANLP/NAACL 2000 Workshop on Automatic Summarization*, pp. 40–48, 2000.

[8] Thomas Hofmann. Probabilistic Latent Semantic Indexing. In *Proc. of SIGIR 1999*, pp. 50–57, 1999.

[9] Keiji Shinzato, Tomohide Shibata, Daisuke Kawahara, Chikara Hashimoto, and Sadao Kurohashi. TSUBAKI: An Open Search Engine Infrastructure for Developing New Information Access Methodology. In *Proc. of IJCNLP 2008*, pp. 189–196, 2008.