構造的言語処理による大規模ウェブ情報のクラスタリング

馬場 康夫 † 笹田 鉄郎 †† 新里 圭司 † 黒橋 禎夫 †

† 京都大学大学院情報学研究科 †† 京都大学工学部電気電子工学科
{banba, sasada, shinzato, kuro}@nlp.kuee.kyoto-u.ac.jp

1 はじめに

現在、ウェブには膨大な量の情報が氾濫しており、その中から求める情報を得るためには、検索エンジンの利用が不可欠である。しかしながら、Google に代表される既存の検索エンジンは、検索クエリと関連する文書をランキングしてリスト形式で提示するだけであり、利用者は自分の欲しい情報をリストの中から探し出さなければならない。また、検索結果には、数百、数千の文書が含まれるため、既存の検索エンジンを用いたのでは検索結果全体を瞬時に俯瞰することは難しい。そこで本研究では、検索エンジンより得られる結果を、係り受け関係などの構造的言語情報を手がかりに分類することで、効率的な情報アクセスおよび検索結果の鳥瞰図的把握を利用者に提供するシステム (クラスタリングシステム) の構築を目的とする。

クラスタリングシステムは、検索結果に含まれる大量の文書をクラスタ (文書の集合) という形で組織化して利用者へ提示するため、

1. 効率的な情報アクセス手段を利用者へ提供できる
2. 検索クエリの新しい側面を利用者へ気づかせることが期待できる

という利点を持つ。

本稿では、我々が開発している大規模ウェブ情報を対象としたクラスタリングシステムについて述べる。

2 関連研究

効率的な情報アクセスを利用者へ提供するために、様々なシステムが研究・開発されている。商用ではあるが、検索結果クラスタリングエンジンとしては、Clusty[1]が有名である。Clusty では、複数の検索エンジンに対して一斉に検索を行い (メタ検索)、その結果得られる検索結果をマージし、クラスタリングすることで、複数の検索エンジンより得られる検索結果に対する、効率的な情報アクセスを利用者に提供している。

一方で、検索結果クラスタリングの主要な研究としては、Scatter/Gather[1] システムを用いて検索結果を対話的にクラスタリングしていく Hearst ら [2] による研究や、STC (Suffix Tree Clustering) と呼ばれるアルゴリズムを用いて検索結果に含まれる文書を逐次的にクラスタリングする Zamin ら [3] の研究がある。

その他のシステムとしては、Hearst[4] が開発した Flamenco がある。Flamenco では、あらかじめ人手で作成されたファセットと呼ばれるドメイン固有のカテゴリ (例えば、「ノーベル賞受賞者」というドメインであれば、国や賞の名前など) を用いて、検索対象 (アインシュタインや湯川秀樹など) を分類しており、利用者はファセットを辿ることで、自分の欲する情報にアクセスできる。

3 大規模ウェブ情報クラスタリングシステム

システムの概要を図 1 に、クラスタリング結果の表示例を図 2 に示す。

3.1 検索クエリを含む文書の収集

既存の検索エンジンを利用して、検索クエリを含む文書を収集する。本システムでは検索エンジンとして TSUBAKI[2] を用いる。TSUBAKI では、日本語ウェブ文書 5000 万ページを検索対象としており、API[3] を利用して検索クエリを含む文書を取得可能である。TSUBAKI の特徴として、以下の二点が挙げられる。

1. API の利用回数や取得可能な文書数に制限がない
2. 標準フォーマット [5] 化されたウェブ文書を入手可能である

標準フォーマットとは、ウェブ文書から抽出された日本語文および、その構文解析結果が埋め込まれた XML

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

[2] http://tsubaki.ixnlp.nii.ac.jp/index.cgi
[3] http://tsubaki.ixnlp.nii.ac.jp/api.cgi

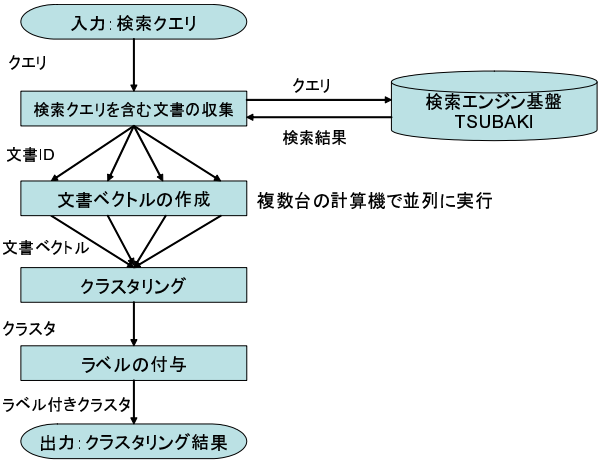

図 1: クラスタリングシステムの概要

図 2: クラスタリング結果の表示例

文書のことであり、TSUBAKI API を用いることで容易に得られる。

3.2 文書ベクトルの作成

TSUBAKI API を用いて収集された各ウェブ文書について、文書ベクトルを作成する。以下に文書ベクトル作成のための手順を示す。Step 1、2 は高速にクラスタリングを実行するために、複数台の計算機で並列に実行される。

Step 1: 重要文の抽出

Step 2: 重要文からのベクトル要素候補の抽出

Step 3: 関連表現集合を用いたベクトル要素候補のフィルタリング

Step 4: ベクトル要素の重みづけ

本システムでは、重要文と関連表現集合を用いることで、検索クエリと関連の強い表現だけを文書ベクトルの要素として抽出することを目指す。

以下では、各ステップについて述べる。

3.2.1 Step 1: ウェブ文書からの重要文抽出

Step 1 では、ウェブ文書中から検索クエリについての重要文を抽出する。具体的には、検索クエリ中の内容語をより多く含む文およびその周辺の文 (本システムでは前後 2 文) を重要文と仮定し、内容語を含む文には 1、前後 1 文には 0.5、前後 2 文に対しては 0.25 をそれぞれスコアとして与える。そして、各文についてスコアを計算した後、スコアの上位 N 文に文書のタイトルを加えた $N+1$ 文を重要文として抽出する。本システムでは、$N = 15$ としている。

3.2.2 Step 2: ベクトル要素候補の抽出

Step 2 では、Step 1 で抽出された重要文から、文書ベクトル要素の候補となる表現を抽出する。ベクトルの要素として、自立語に加えて、係り受け関係にある自立語のペアも考慮する。本システムでは、複合名詞中の連続する自立語同士も、係り受け関係として扱う。

以下では「子ども服をせんたくする」を例に、ベクトル要素候補を抽出する手順を示す。

1. 標準フォーマット化されたウェブ文書から KNP[4]の解析結果を抽出する。KNP の解析結果では、「子ども」に対しては「子供」が代表表記として与えられており、「せんたく」は曖昧性を持つため、「洗濯」と「選択」の 2 つが代表表記として与えられている。

2. KNP の解析結果よりベクトル要素候補を抽出する。本システムでは、ベクトル要素候補の持つ曖昧性を全て考慮しており、その出現頻度としては曖昧性の個数で割ったものを用いている。表 1 に、「子ども服をせんたくする」から生成されるベクトル要素候補と各要素の出現頻度を示す。

3. 抽出されたベクトル要素候補から、品詞が指示詞、形式名詞、副詞的名詞であるものを削除する。

本システムでは、係り受け関係を考慮することで、「子供服」のような複合名詞や、「服→洗濯」のような述語項構造もベクトル要素候補として自然に扱えている。

3.2.3 Step 3: 関連表現集合を用いたベクトル要素候補のフィルタリング

Step 3 では、Step 2 で抽出されたベクトル要素候補を洗練するため、検索クエリと関連の強い表現 (自

[4]http://nlp.kuee.kyoto-u.ac.jp/nl-resource/knp.html

表 1: 「子ども服をせんたくする」から抽出されるベクトル要素候補

ベクトル要素候補	出現頻度
子供	1
服	1
洗濯	0.5
選択	0.5
子供→服	1
服→洗濯	0.5
服→選択	0.5

立語または自立語同士の係り受け関係) の集合 (以下、**関連表現集合**と呼ぶ) を生成する。そして、関連表現集合に含まれていないベクトル要素候補を削除する。

関連表現集合の生成方法は以下のとおりである。まず、Step 2で抽出された全てのベクトル要素候補について、検索クエリとの関連の強さを表すスコアを計算する。検索クエリを Q、クラスタリング対象となる文書集合を D_Q、TSUBAKIが検索対象とする文書集合を D とした時、ベクトル要素候補 e と Q との関連の強さは以下の式で計算される。

$$dfidf(e) = ldf \times \log \frac{|D|}{gdf}$$

ここで ldf は文書集合 D_Q 中で表現 e を含む文書の数、gdf は文書集合 D 中で表現 e を含む文書の数である。$dfidf(e)$ の値は、クラスタリング対象となる文書にだけ現れやすく、他の文書には現れにくい表現 e ほど大きくなる。

全ベクトル要素候補についてスコアを計算後、スコアの上位 M 個を関連表現とし、関連表現集合を生成する。本システムでは $M = 1000$ としている。

3.2.4 Step 4: ベクトル要素の重みづけ

関連表現集合を用いてフィルタリングした結果、残ったベクトル要素候補に対して重みづけを行う。本システムでは、文書 d に対するベクトル要素 e の重み $w_{e,d}$ を、Okapi BM25 [6] を用いて計算する[5]。

$$w_{e,d} = \frac{3.0 \times fq}{2.0 + fq} \times \log \frac{|D| - gdf + 0.5}{gdf + 0.5}$$

ここで fq は文書 d 中での e の出現頻度である。

3.3 クラスタリング

3.3.1 群平均法を用いたクラスタリング

クラスタリング手法は、群平均法に代表される階層的手法と k-means 法に代表される非階層的手法の 2 種類に大別される。本システムがクラスタリングの対象とする文書集合は検索クエリについて検索した結果得られたものであるため、各文書は多かれ少なかれ類似していると考えられる。k-means 法ではクラスタの核となる文書を初期値として複数個与える必要があるが、類似した文書集合を適切に分類する核を与えるのは容易ではない。そこで本システムでは、群平均法により文書集合のクラスタリングを行う。

[5] パラメータとして $k_1 = 2.0,\ k_2 = 0,\ k_3 = 0,\ b = 0.75$ を用いている。

1. 任意の 2 文書 d_i, d_j 間の類似度 $sim_d(d_i, d_j)$ を、両文書ベクトルのコサイン類似度により求める。
2. 各文書 d_i を単一のクラスタ C_i に割り振る。
3. 任意の 2 クラスタ C_A, C_B 間の類似度 $sim(C_A, C_B)$ を以下の式により求める。

$$sim(C_A, C_B) = \frac{\sum_{d_i \in C_A} \sum_{d_j \in C_B} sim_d(d_i, d_j)}{|C_A||C_B|}$$

   ここで $|C|$ は、クラスタ C に属する文書数である。
4. クラスタ C_A, C_B の大きさによる補正をかけた類似度 sim′ を下式により求める。

$$sim'(C_A, C_B) = \frac{sim(C_A, C_B)}{|C_A| + |C_B|}$$

5. 補正類似度が最大のクラスタ同士を結合する。
6. 以下の終了条件 1、2 を共に満たす場合クラスタリングを終了する。それ以外の場合は手順 3 に戻る。

   条件 1: クラスタ「その他」に含まれないクラスタ数 N_C が $5 \leq N_C \leq 8$

   条件 2: クラスタ「その他」に属する文書数が、クラスタリングの対象となっている文書数の 20%以下

図 3: クラスタリングの手順

図 3 に、本システムのクラスタリング手順を示す。クラスタリング結果の閲覧性の良さを決定する大きなファクターの一つは、生成されるクラスタの大きさである。そのため本システムでは、クラスタ同士を結合する際、類似度に加えて両クラスタの大きさを考慮し、極端に大きなクラスタが生成されないようにしている (図 3 の手順 4)。また、小さなクラスタが数多く生成されるような状況も望ましくないため、小さなクラスタ (メンバ数がクラスタリング対象となっている文書数の 4%以下) はクラスタ「その他」としてまとめている。

3.3.2 サンプリングによるクラスタリングの高速化

クラスタリングの対象となる文書数を N としたとき、図 3 に挙げたクラスタリングアルゴリズムの計算量は $O(N^2 \log N)$ である。そのため、クラスタリング対象となる文書集合の規模が大きくなると、クラスタリングに要する時間的コストは 2 乗のオーダーで増加する。これは、検索エンジンより返される大量の検索結果をクラスタリングすることを考えている本システ

表 2: サンプリングの効果 (検索クエリ=「五十肩」, $N_s = 500$)

文書数 (N)	100	500	1000	2000	3000
サンプリングなし (秒)	10	31	83	305	679
サンプリングあり (秒)	10	31	39	64	84

ムにおいては大きな問題となる。

そこで本システムでは、最初に N_s 件だけ文書を抽出 (サンプリング) してクラスタリングを行い、その結果生成される各クラスタに対して、残りの文書 $(N - N_s)$ 件を付け足す方法をとった。これにより、N 件の文書をクラスタリングするのに要する計算量は $O(N_s^2 \log N_s + N)$ となり、文書数の増加に伴う計算時間の爆発を抑えることができる。

表 2 に、サンプリングをした場合としなかった場合でのクラスタリングに要する時間を示す。表より、サンプリングをすることでシステム全体の所要時間を大幅に改善できていることがわかる。

3.4 ラベルの付与

クラスタの内容を素早く把握するためには、各クラスタに適切なラベルを与えることが重要である。本システムでは、クラスタ C のラベルとして適切な表現は、

- クラスタ C に含まれている多くの文書に出現し、C 以外のクラスタに含まれている文書には現れにくい
- 3.2.4 節で計算した重みが大きい

と仮定し、クラスタ C に含まれる文書の文書ベクトルの各要素 e について以下のスコアを計算する。

$$label(e) = cdf \cdot \frac{cdf}{ldf} \cdot \frac{1}{|C|} \sum_{d \in C} w_{e,d}$$

ここで cdf は、クラスタ C 中で e を含む文書数、$w_{e,d}$ は 3.2.4 節で求めた文書 d に対する e の重みである。

基本的には、上記のスコアが最も大きい要素 e_{max} がクラスタのラベルとして付与される。しかし、より長い表現がラベルとして付与された方が、クラスタの内容を把握しやすいとの考えから、e_{max} に対して以下の処理を行う。

1. e_{max} が単語の場合、スコアの上位 L 件中に、e_{max} を含む係り受け関係があれば、スコアの最も高いものを新たに e_{max} とする
2. e_{max} が係り受け関係の場合、スコアの上位 L 件中に、e_{max} と共通の表現を親または子に持つ係り受け関係があれば、それらを連結する。例えば、「人工→衛星」と「衛星→スプートニク」からは「人工衛星スプートニク」が新しく生成される。

以上の操作により生成された表現 e_{max} をラベルとしてクラスタに付与する。本システムでは $L = 10$ としている。

4 考察および今後の課題

まず、「スプートニク」と「地球温暖化」を検索クエリとし、それぞれについて収集された検索結果をクラスタリングした結果について簡単に考察する。「スプートニク」については、小説について記述された文書および人工衛星について記述された文書からなるクラスタがそれぞれ生成され、ラベルも比較的良好な表現 (「スプートニク恋人」や「人工衛星スプートニク」) が付与されていることが確認できた。一方で「地球温暖化」については、「二酸化炭素」や「地球温暖だ化防止」などの表現をラベルとして持つクラスタが生成されたが、実際にクラスタに属する文書を見ると、必ずしもラベルと関連のある文書ばかりでなく、ラベルとあまり関係のない文書も含まれていた。

今後の課題としては、3 節で述べた個々の要素技術の評価および、クラスタリングシステムとしての全体的な評価を行うことが挙げられる。また、実用性の観点から、クラスタリングに要する時間の改善も今後の課題として挙げておく。

参考文献

[1] D. R. Cutting, J. O. Pedersen, D. Karger, and J W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the 15th Annual Int'l ACM SIGIR Conference on R&D in IR*, pp. 318–329, 1992.

[2] M. A. Hearst and J. O. Pedersen. Reexamining the cluster hypothesis: Scatter/gather on retrieval results. In *Proceedings of SIGIR-96, 19th ACM Int'l Conference on R&D in IR*, pp. 76–84, Zürich, CH, 1996.

[3] O. Zamir and O. Etzioni. Grouper: a dynamic clustering interface to Web search results. *Computer Networks (Amsterdam, Netherlands: 1999)*, Vol. 31, No. 11–16, pp. 1361–1374, 1999.

[4] M. A. Hearst. Clustering versus faceted categories for information exploration. *Commun. ACM*, Vol. 49, No. 4, pp. 59–61, 2006.

[5] 新里圭司, 橋本力, 河原大輔, 黒橋禎夫. 自然言語処理基盤としてのウェブ文書標準フォーマットの提案. 言語処理学会第 13 回年次大会論文集, 2007.

[6] S.E. Robertson and S. Walker. Okapi/Keenbow at TREC-8. *NIST Special Publication 500-246: The Eight Text REtrieval Conference (TREC-8)*, pp. 151–162, 1999.