特集 WWWを対象にした日本語研究

WWWの記事をその内容によって自動的に分類する

新納 浩幸

本稿はWWWの記事（文書）をその内容によって自動的に分類すること（文書クラスタリング）について、以下の四つの点から概説する。

1. それはどのような処理であり、何が難しいのか
2. どのような場面で、そのような処理が必要なのか
3. そこにはどのような技術が使われているのか
4. それは言語研究とどのような関わりがあるのか

1 文書クラスタリングとその困難性

文書クラスタリングとは多数の文書が与えられたときに、それらを内容によっていくつかのグループに分割する処理である。これはなじみのある処理であり、あらためて説明することはないのかもしれないが、その困難性を明らかにするために、もう一歩進めてその処理の内容を示したい。

まず「分類」と「識別」との違いに注意する必要がある。「分類」とはある対象群をなんらかの類似性に基づいて部分的な対象群に分割する処理であり、「識別」とはある対象に対して、予め用意されているラベル群の中からその対象に適したラベルを選択し、そのラベルを付与する処理である。例えば多数の文書（文書群）があるときに、文書内容の類似性からいくつかのグループに分割するのが「分類」であり、各文書に、例えばこれは「経済」に関する文書であるとか、「スポーツ」に関する文書であるなどのある種のラベル（この場合、カテゴリ）を与えるのが「識別」である。当然、「識別」を行うことができれば、ラベル別に文書群が「分類」できる。「分類」も想定したラベル別に文書群を分割しているようにも見えるため、「分類」と「識別」を区別しないことも多い。実際、専門用語上でも、文書の識別を「文書分類」と呼んでいる。またここでいう文書の分類は「文書クラスタリング」と呼ばれている。本稿で扱うのは「文書クラスタリング」である。

「分類」と「識別」は似ている処理だが、技術的には明確に異なる。それは「識別」の場合、予めラベル群が用意されているのに対して、「分類」ではラベル群が用意されていないからである。そしてこの点のために「分類」が非常に困難なタスクになっている。先の例で言えば、「識別」の場合、新聞記事の分類で使われるようなカテゴリ（「政治」「経済」「スポーツ」など）が予め用意されている。そのため「識別」の処理で行うことは、その文書がどのカテゴリに属する文書であるかを判定することである。つまり「識別」では、文書の内容を見る観点が固定されているが、「分類」ではそのような観点が固定されていない。観点が固定されていないため、「分類」には正解というものがない。どのような観点に注目するかで分割の結果は異なる。新聞記事の文書群が与えられたとしても、観点として「最近の話題」、「古い話題」などで分類することも可能であり、その場合「経済」や「スポーツ」などで分類した結果とは全く異なってしまう。

観点が固定されていないということは、いくつのグループに分割するかも固定されていないことを意味する。非常に細かく分類することも可能であるし、二、三の荒いグループに分割することもできる。いくつのグループに分割するかに対しても正解はない。

まとめると、文書クラスタリングには二つの困難な点がある。一つは観点が固定されていないという点と、もう一つは観点の粒度が固定されていない点である。これはコンピュータにとって困難であるということではなく、本質的な困難性であり、文書クラスタリングは人間にとっても困難な処理である。そのため文書クラスタリングをコンピュータを使って自動的に行うことは、ある意味不可能だし、ポジティブに言えば非常に挑戦的なタスクと言える。

2 文書クラスタリングの目的

情報は文書によって伝え、保存するのが基本である。

また近年、社会の電子化が進み、電子的な文書の量は増える一方である。このような背景から、なんらかの処理を行わなければならない多数の文書が存在するという状況は、ごくごく一般的である。

この「なんらかの処理」として次の二つが想像できる。一つは文書群の内容の全体像を把握する処理であり、もう一つは文書群の中から、自分の欲しい情報が記載されている文書を見つける処理である。前者はテキストマイニングに属する技術であり、後者は情報検索に属する技術である。どちらの技術に対しても文書クラスタリングが重要な要素技術となっている。

まずテキストマイニングの例を示す。

例えば上司から突然多数の書類をドッサと渡され、「この書類群を明日までにまとめなさい」と言われたらどうするだろうか。おそらく、ざっと斜め読みして、内容別に分類して、どのような構成でまとめるかを考えるであろう。この「内容別に分類する」処理が、文書クラスタリングである。実は、この処理によって全体をどのような観点で捉えるかが決まってしまい、まとめ方の大枠が決まってしまう。そのためある意味最も重要な処理と言える。つまり多数の文書の全体像を理解するための第一歩は、文書クラスタリングを行うことである。このため文書クラスタリングの技術が、人間の知的生産性を向上させるのに役立つことは間違いない。

もう少し現実的な例を挙げれば、旅行会社が行う自由記述のアンケート結果の分析がある。このようなアンケート結果の分析は、その後の旅行の企画に役立つはずである。分析を行うために、すべてのアンケート結果を読んで、どのような観点でまとめられるかを試行錯誤するのはかなり大変な作業である。最終的には人間がなんらかの発見を行わなければならないが、そのためのツールとして文書クラスタリングが利用できる。例えばアンケートの結果に対して文書クラスタリングを行い、いくつかの文書群に分割し、ある文書群は「予算」、ある文書群は「目的地」またある文書群は「食事」といった要求や不満が書かれていたとしたら、この分類結果はアンケートの結果をまとめることや何らかの企画を考えることに役立つはずである。

あるいは企業のコールセンターでは同じような質問が何度も繰り返し聞かれるので、質問応答集を作っておくことは作業の効率化に繋がる。質問応答集を作るリソースは個々の質問応答であるが、これを質問のタイプ毎に分類しておかないと、質問が来たときに質問応答集に答えがあるのかどうかが即座に判断できない。質問応答のセットを質問のタイプ毎に分類するためにも文書クラスタリングが重要となる。

先ほどの例の文書は、通常のテキストファイルが想定されている。本稿はWWW上にある文書という制約がついているが、この点は特に気にする必要はない。あるテキストファイルがWORDなどの文書から取り出されたものか、WWW上のHTML文書から取り出されたものかに本質的な違いはない。先ほどの例で言えば、アンケートや質問はWWW上の掲示版に記載されたものと考えればよいであろう。

WWW上にある文書という点が明確になる文書クラスタリングの応用例は、情報検索に関する技術である。これはクラスタリングサーチエンジンとして知られている。Googleなどの通常の検索エンジンは、入力されたキーワードを含むWWW上のページのリンクとそのスニペット（検索キーワードが含まれた抜き書き）を出力する。キーワードを含むページは多数あるが、Google独自のランキング手法により、検索目的に合ったページが上位に提示される。このランキングの優秀性がGoogleの一つのアドバンテージである。

ただし、このランキング法も完璧ではなく、必ずしも最上位のランクのページに自分の欲しい情報が書かれているわけではない。実際は上位の方からスニペットを利用して目的のページを探す作業が必要となる。特にキーワードが一般的な単語の場合、その単語を含むページ数は膨大であるため、そこから目的の文書を見つけることはかなり難しくなる。この問題に対して、情報検索の分野では様々な取り組みがなされている。クラスタリングサーチエンジンはその一つである。

クラスタリングサーチエンジンでは検索結果のWWW上のページの集合に対して文書クラスタリングを行い、分割された各グループにキーワードを与える。サーチエンジンの利用者は自分の欲する情報と最も関連がありそうなキーワードに対応するグループに限定して、さらに検索を進めることで効率的に所望のページにたどり着くことができる。

代表的なクラスタリングサーチエンジンはVivisimo社が運営しているclusty（http://clusty.jp/）である。二〇〇七年一一月三〇日にその日本語版が公開された（**図1**）。

例えば、このサイトで「アップル」で検索を行うと、**図2**のような検索結果が得られる。

図2の左側に「iPod」「apple」「ジャパン」などのキーワードがあるが、これが文書クラスタリングによって分割されたページのグループに与えられたキーワードである。

ここから利用者は「アップル」の何を調べたいのかを考えて、キーワードを選択する。例えば「Apple」を選択した場合、キーワード「Apple」を含むページが提示される。「Apple」の中の文書群を更にクラスタリングしたければ、キーワード「Apple」の左のフォルダーをクリックする。このとき**図3**のように「Apple」のサブカテゴリが表示される。このように検索対象を絞り込んでゆくことで、目的のページを探してゆくことができる。

クラスタリングサーチエンジンは一種の検索ナビゲーションである。検索ナビゲーションと言えば、ディレクトリ型の検索が代表的である。現在はGoogleのようなキーワードを入力するタイプの検索システムが主流であるが、インターネットの創世記には、ディレクトリ型の検索システムも存在し、それなりに利用者も多かった。

ディレクトリ型の検索システムは、予めWWW上のページをカテゴリ別に分類しておく。利用者はその分類に従って目的のページを探してゆく。あるカテゴリ内のページは更にそのサブカテゴリに分割されており、全体として木構造（注1）をなしている。このカテゴリ分けされたページの集合はウェブディレクトリと呼ばれる。

図4はYahoo!のウェブディレクトリの一部である。例えば「ビジネスと経済」のカテゴリを選ぶ。すると更にページがサブカテゴリに分類されている（**図5**）。このようにして徐々に対象のページを限定させてゆくことで目的のページにたどり着くのがディレクトリ型の検索である。

すぐに気がつくことであるが、ディレクトリ型の検索とクラスタリングサーチエンジンは類似している。違いはカテゴリへの分割が、予め行われているかどうかである。ディレクトリ型の検索ではカテゴリへの分割が予め行われており、クラスタリングサーチエンジンの場合は、オンデマンドでカテゴリへの分割を行っている。つまりウェブディレクトリを作成するために、文書クラスタリングが利用できることは明らかである。

ウェブディレクトリ中のカテゴリは様々な観点から分類されている。例えば、Yahoo!のウェブディレクトリで「漫画家」というカテゴリは、どのようなサブカテゴリに分類されているかご存じだろうか。答えは「あ」～「わ」、つまり漫画家の氏名の頭文字である。分類の観点を柔軟に考えないと、有用なウェブディレクトリは作成できない。観点を固定しない文書クラスタリングが、なんらかの助けになるはずである。

図1　クラスタリングサーチエンジン clusty

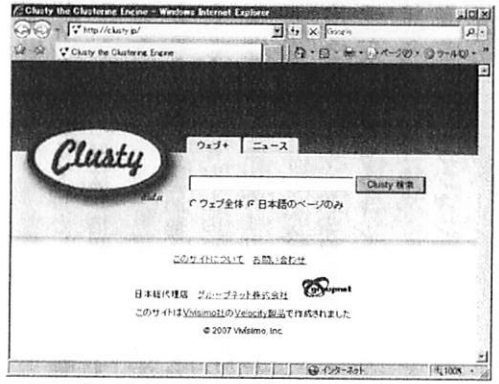

図2　「アップル」の検索結果

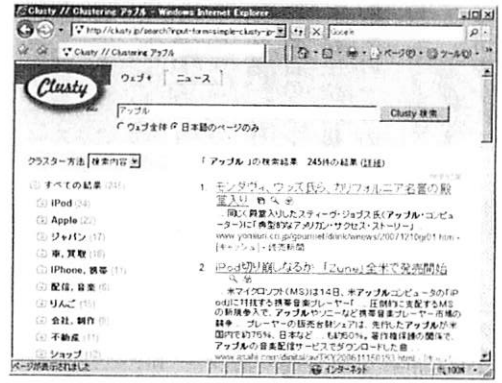

図3　サブカテゴリの表示

図4　Yahoo!のウェブディレクトリ

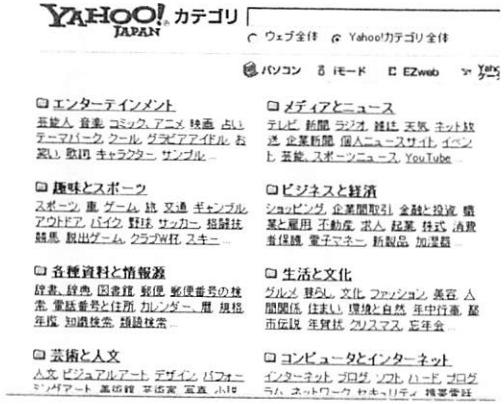

図5　「ビジネスと経済」のサブカテゴリ

3 文書クラスタリングの技術

「分類」という処理は知的作業の基本であり、古くから様々な学問分野でその手法が研究されている。現在は「クラスタリング」という名の下に、多変量解析の一手法として位置づけられている。

文書クラスタリングも分類対象のデータが文書というだけであり、多変量解析のクラスタリング手法を用いて実行することができる。ただし多変量解析のクラスタリング手法では、データをN次元ベクトルで表現しなくてはならない。概略、データをN次元空間上に配置し、距離の近いもの同士をまとめていくことでクラスタリングが行える。例えば古典的な階層的手法では、各データが一点からなるクラスタ群から出発し、クラスタ間の距離の近い物から順にクラスタを併合させてゆくことでクラスタリングを行う。

本稿ではクラスタリングのアルゴリズムの詳細は述べず、文書をどのようにN次元ベクトルで表現するかを説明する。現実のクラスタリングの処理で最も重要なこと、つまり、最もクラスタリング結果に影響するのは、クラスタリングのアルゴリズムの選択ではなく、現実のデータをN次元ベクトルに変換する方法である。

文書クラスタリングに限らず、文書処理では文書データをN次元ベクトルで表現することが一般的である。そこでは共通して、「bag of words」と呼ばれるモデルが使われる。これは文書を単語が詰め込まれた袋（集合）と見なす考え方である。

具体的な例を見てみる。簡単にするために、三つの文書が対象であり、各文書は以下のような一文からなっているとする。〝／〟は単語区切りを示す。

文書1 私／は／茨城／県／に／ある／茨城／大学／の／学生／です／。

文書2 私／は／茨城／県／の／日立／市／に／住ん／で／い／ます／。

文書3 日立／市／に／は／日立／の／工場／が／たくさん／あり／ます／。

今、名詞のみに注目すると、上記文書集合中に現れた単語は以下の八種類である。

(1)私、(2)茨城、(3)県、(4)大学、(5)学生、(6)日立、(7)市、(8)工場

それぞれ括弧に示した数値を次元に対応させると、文書1～3のベクトルは以下のようになる。

文書1 (1,2,1,1,1,0,0,0)

文書2 (1,1,1,0,0,1,1,0)

文書3 (0,0,0,0,0,2,1,1)

このベクトルの表現方法だと、文書中のすべての単語が同じ重みで評価されてしまう。しかし単語には文書の分野を推定できるような単語やどんな文書でも現れやすい一般的な単語が存在する。前者のような単語には重みを重く、後者のような単語には重みを軽くした方がその文書をよく表したベクトルが生成される。重みの付け方としてはTF*IDFという方法が標準的である。

今、文書d_i中の単語w_jに重みを付けることを考える。TFとはTerm Frequencyの略であり、文書d_i中の単語w_jの頻度のことである。これをf_{ij}とする。IDFとはInverse Document Frequencyの略であり、全文書数Nをw_j含む文書数n_jで割った値に対数を取ったものである。つまり$f_{ij}*\log(N/n_j)$であり、TF*IDFとは文書d_i中の単語w_jの重みを$f_{ij}*\log(N/n_j)$にすることである。TF*IDFを用いて重みをつけ、さらにベクトルの大きさを1に正規化すると、文書1～3のベクトルは以下のようになる。

文書1 (0.1193,0.2385,0.1193,0.3231,0.3231,0.0000,0.0000,0.0000)

文書2 (0.4932,0.4932,0.4932,0.0000,0.0000,0.4932,0.4932,0.0000)

文書3 (0.0000,0.0000,0.0000,0.0000,0.0000,0.3997,0.1999,0.5415)

文書間の類似度は、文書のベクトルの余弦尺度（コサイン（注2））で測ることが一般的である。上記の例であれば、文書1と文書2の類似度は0.2353、および文書2と文書3の類似度は0.2957となり、文書2は文書1よりも文書3に類似していると判定される。

4 言語研究との関わり

文書クラスタリングは、文書データをN次元ベクトルで表現してしまえば、後は単純に数値的な解析が進むだけである。そこには対象のベクトルが本来文書であったのか、画像であったのかなどは関係ない。そのため文書クラスタリングを単にクラスタリング手法の実験対象として扱うのであれば、言語との関わりは少ない。しかし前章で述べたとおり、現実のクラスタリング結果に最も影響を与えるのはデータからN次元ベクトルへの変換方法である。またクラスタリングで本質的に必要となるのはデータ間の類似度である。ここに文書クラスタリングと言語研究の関わりが存在する。つまり二つの文書の内

容の類似性を、どのような方法で数値化できるかである。

現状は「bag of words」のモデルを用いて文書をベクトル化し、ベクトル間の余弦尺度により類似度を定義するのが主流である。そしてそれをベースに様々な改良が試みられている状況である。代表的な改良として、単語を概念に汎化させ、概念を次元にしてN次元ベクトルを構成する方法がある。例えば「値段が高い」という文と「価格が安い」という文は似ているが、「値段」「高い」「価格」「安い」は全て異なる単語なので、単純に単語を次元に取れば、二つの文の類似度は0である。ここで「値段」と「価格」は同じ概念「Price」、「高い」と「安い」は同じ概念「HighOrLow」などに汎化し、この概念を次元に取ってベクトルを作成すれば、二つの文の類似度は1となる。単語を概念に汎化するには、既存のシソーラスが利用できる。概念の汎化に対応する処理を自動的に行う手法も存在するが、既存のシソーラスを使った方が精度は高いようである。つまりシソーラスを改善してゆくことが、文書クラスタリングの改善に繋がる。他にも係り受けの情報も加味してベクトルを作成する改良がある。これによって例えば「石で壊す方法」と「石を壊す方法」などが区別される。あるいはWWW上の文書には文書のメタ情報（リンクやキーワードなどの情報）も取り出せることがあるので、それらを加味してベクトルを作成する改良もある。ただしこれらの改良によって、現実的かつ実効性のある成果は出ていないように感じる。

個人的に言語研究に期待したいのは、「bag of words」を超えるモデルである。「bag of words」のモデルは非常に荒い。このモデルでは単語の現れる順序、共起する単語対、類似の単語群などが考慮されておらず、言語的な観点からすると何か違和感がある。

文書は文の集合であり、各文が全体的に様々な役割を担って、文書全体の内容が作られる。そうであれば、二つの文書間の類似性を定義するには、二つの文間の類似性をまず考える必要がある。現状は文の類似性さえ、はっきりした定義はない。文の類似性を考える場合、文の意味にまで立ち返る必要があるだろう。

結局、ありきたりの結論であるが、意味の研究が文書クラスタリングはもちろん自然言語処理システムの根幹になっている。

注

1 厳密にはサイクルが含まれる場合があるので、木構造ではない。

2 この場合各ベクトルの大きさが1なので、余弦はベクトル間の内積で求まる。

（しんのう・ひろゆき　茨城大学工学部準教授）