情報処理学会論文誌　データベース　Vol.5　No.3　51–63 (Sep. 2012)

Wikipediaのカテゴリグラフ解析による語句の確率的分類とその応用

白川 真澄[1,a)]　中山 浩太郎[2,b)]　原 隆浩[1,c)]　西尾 章治郎[1,d)]

受付日 2012年3月20日, 採録日 2012年6月6日

概要：語句をカテゴリ（トピック）に分類した概念辞書は，文書分類をはじめ様々なアプリケーションの基盤リソースとして必要とされている．代表的な概念辞書である WordNet は一般語を網羅的に定義しているが，固有名詞や専門用語，新語はあまり網羅されていない．一方，大規模 Web 百科事典である Wikipedia はそのような語句を数多く定義しており，また，語句を分類するためのカテゴリ構造を有している．しかし，Wikipedia のカテゴリ構造は，複数の親やループを許容するネットワーク構造であるため，ある語句がどのカテゴリに属しているかを判別するのは難しい．そこで本研究では，グラフ理論に基づいて Wikipedia のカテゴリネットワークを解析し，確率的に語句を分類する手法を提案する．また，語句の確率的分類の結果を教師データとし，ナイーブベイズによる文書分類を行う．Web 検索のスニペットを代表的な 8 カテゴリに分類するタスク，および科学に関するニュースのスニペットを 8 つの領域に分類するタスクにおいて評価を行い，提案手法の有効性を確認した．

キーワード：文書分類，ページランク，グラフカーネル

Probabilistic Term Classification by Analyzing Wikipedia Category Graph and Its Application

Masumi Shirakawa[1,a)]　Kotaro Nakayama[2,b)]　Takahiro Hara[1,c)]　Shojiro Nishio[1,d)]

Received: March 20, 2012, Accepted: June 6, 2012

Abstract: Taxonomies, which classify terms into categories (topics), are required as fundamental resources for many applications including text classification. WordNet is one of the representative taxonomies and defines general terms, though it defines few named entities, specific terms and new words. On the other hand, Wikipedia, a large-scale free online encyclopedia, defines such terms and classifies them into a variety of categories. However, because the category structure is a network that allows multiple parents and loops, it is hard to determine whether a term belongs to a category or not. In this paper, we propose a method to probabilistically classify terms by analyzing Wikipedia category network based on graph theories. We also propose a text classification method using the result of probabilistic term classification and Naive Bayes. In the experiments on both Web snippet dataset and science news dataset, we confirmed the effectiveness of our method for classifying texts into several categories.

Keywords: document classification, PageRank, graph kernel

[1] 大阪大学大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University, Suita, Osaka 565–0871, Japan
[2] 東京大学知の構造化センター
The Center for Knowledge Structuring, The University of Tokyo, Bunkyo, Tokyo 113–8656, Japan
[a)] shirakawa.masumi@ist.osaka-u.ac.jp
[b)] nakayama@cks.u-tokyo.ac.jp
[c)] hara@ist.osaka-u.ac.jp
[d)] nishio@ist.osaka-u.ac.jp

1. はじめに

語句をカテゴリ（トピック）に分類した概念辞書は，テキスト分類をはじめ様々なアプリケーションの基盤リソースとして必要とされている．WordNet [2] は一般語を分類した概念辞書であり，語句の上位下位関係を定義している．しかし，WordNet は固有名詞や専門用語，新語など

をあまり定義していないことがデメリットとしてあげられる．また，WordNet では語句の上位概念（“Lion” に対して “Mammal,” “Animal” など）は定義されているが，語句のトピックによる分類（“Lion” に対して “Nature” など）を行っていないため，本研究が目的とするトピックによる分類には適していないと考えられる．一方，大規模 Web 百科事典である Wikipedia では，固有名詞や専門用語，新語などを多数定義しており，それらの語句は上位概念だけでなく上位のトピックにも分類されているため，テキストを様々なトピックに分類するための外部知識として非常に優れている．本研究では，このような例に対して正しく語句をトピックに分類するため，Wikipedia のカテゴリ構造をそのまま入力として用いている．しかし，Wikipedia で定義されている任意のカテゴリに語句を分類することは難しい．これは，Wikipedia のカテゴリ構造が複数の親やループを許容するネットワーク構造をなしているためである．このような Wikipedia のカテゴリ構造の性質により，ある語句から親カテゴリをたどっていくと，まったく関係のないカテゴリに到達することが頻繁に起こりうる．たとえば，動物の “Lion” についての記事から親カテゴリをたどっていくと，“Lions,” “Panthera,” “Pantherinae,” “Felids,” “Cats,” “Domesticated animals,” “Agriculture” とあまり関係のないカテゴリに到達できる．さらに親カテゴリをたどれば “Humans,” “Economics,” “Education” などのカテゴリにも到達可能である．このように，Wikipedia では親カテゴリをたどることで 1 つの記事から様々な種類のカテゴリに到達できるため，単純に親カテゴリをたどる手法によって語句を分類することは難しい．

そこで本研究では，Wikipedia の記事を確率的に分類することを考える．つまり，カテゴリに属するか否かではなく，どの程度の確率で属するかという数値として表現する．Wikipedia のカテゴリ構造では，ある記事から親カテゴリをランダムにたどっていったとき，そのパス上でより確実に出現するカテゴリに対して，より強く所属していると考えられる．そこで，親カテゴリをたどる際に確率的にスコアを割り当て，より大きいスコアを持つカテゴリに強く所属すると見なす．これは，隣接ノードのいずれかに等確率で遷移するランダムウォークを用いて表現できる．提案手法では，あらかじめ指定した複数のカテゴリ（基底カテゴリ）に対し，ある語句から親カテゴリをたどったとき，それらのカテゴリに到達する確率を，ランダムウォークにより算出する．また，親カテゴリを再帰的にたどるという処理は計算量が大きいため，行列を利用した数値解析による手法を用いてグラフカーネルを構築し，計算の効率化を図る．具体的には，Wikipedia の各カテゴリをノードとしたグラフについて親カテゴリへの遷移確率行列を作成し，基底カテゴリを意図的にシンクノード（スコアを吸収するノード）として，各シンクにどの程度スコアが流れるかをべき乗法を用いて算出する．本手法は PageRank [4] の計算方法と似ているが，対象とする行列が既約ではない（もちろん原始的でもない）ことや，最終的に導出するものが各ノードのスコアではなくカーネルの役割を果たす行列であることから，べき乗法を収束させるための工夫が必要である．本研究では，グラフカーネルを構築するためのべき乗法の収束性と計算方法について明らかにする．

また，提案手法の応用として，Wikipedia で定義されている語句についての確率的な分類結果を用いて，テキスト（スニペット）の分類を行う．テキスト分類では一般的に，教師データを作成し，ナイーブベイズ（NB）[1] やサポートベクタマシン（SVM）[22] などの機械学習手法を用いる．最近では，Wikipedia を用いたテキスト分類に関する研究が行われているが，Wikipedia のカテゴリをそのまま用いるのではなく，教師あり学習の素性として用いている．一方，提案手法では，Wikipedia のカテゴリ構造をそのままテキストの分類に利用することが可能である．すなわち，語句を確率的に分類することにより，ナイーブベイズといった確率的な文書分類手法を適用できる．これにより，カテゴリを代表する名前を Wikipedia のカテゴリの中から指定するだけで，自動的にテキストを分類できる．

以下，2 章で関連研究について述べ，3 章で提案手法について詳述する．4 章で提案手法を用いたテキスト分類について説明し，5 章でテキスト分類における評価実験について説明する．最後に 6 章で本研究のまとめと今後の課題について述べる．

2. 関連研究

2.1 Wikipedia マイニング

Wikipedia を知識抽出の対象とする研究（Wikipedia マイニング）は 2006 年に注目を集め，以降急速に研究対象としての認知度が高まっていった．Wikipedia は，Wiki をベースにした大規模 Web 百科事典であり，誰でも Web ブラウザを通じて記事内容を変更できることが大きな特徴である．そのため，幅広い分野について，一般的なエンティティから新しいエンティティに至るまで記事が網羅されており，記事（エンティティ）数は，最も多い英語版で 380 万，日本語版で 78 万である（2012 年 1 月時点）．また，Wikipedia は，記事の網羅性や即時性だけでなく，カテゴリ構造，密な記事間リンク，言語リンク，質の高いアンカーテキスト，URL による語義の一意性など，知識抽出のコーパスとして有利な性質を数多く持っている [7]．加えて，Wikipedia の全データがオンラインで無償公開[*1] されていることも，Wikipedia マイニングに関する研究が急速に発展した要因の 1 つと考えられる．

Wikipedia のカテゴリ構造は，Wikipedia マイニングに

[*1] http://dumps.wikimedia.org/

関する研究において重要な性質であり，関連度計算 [17] や関係抽出 [8], [11], [19] など，様々な情報の抽出に用いられている．Wikipedia のカテゴリ構造を用いた文書分類（トピック推定）は Schönhofen [14] や Syed ら [21]，Phan ら [10] によって行われているが，Wikipedia で定義されているカテゴリをそのまま分類に用いるのでなく，教師あり学習の素性として利用している．本研究では，Wikipedia のカテゴリ構造をそのまま文書分類に利用できるような手法を提案している．また，隅田ら [20] は Wikipedia のカテゴリ構造や記事のテキストから語句の上位概念（“Bill Gates” に対して “CEO” や “Human” など）を抽出している．しかし，本研究が目的とするトピックによる分類においては，上位下位関係のみでは不十分である．たとえば，“Bill Gates” という語句はトピックの 1 つとして “Computing” に強く属していると考えられるが，“Bill Gates” から上位概念をたどっても “Computing” にたどりつくことはなく，結果として “Bill Gates” を “Computing” に分類できない．本研究では，このような例に対して正しく語句をトピックに分類するため，Wikipedia のカテゴリ構造をそのまま入力として用いている．

Wikipedia にランダムウォークを適用した例としては，WikiWalk [23] があげられる．WikiWalk では，Wikipedia の記事およびカテゴリをノードとしたグラフに対して PageRank を適用し，関連度を計算している．本研究でもランダムウォークを用いているが，単純な関連度ではなく，指定したトピックへの所属の度合いを算出している点で目的が異なる．加えて，本研究では Wikipedia のカテゴリグラフから所属確率を算出するために PageRank を拡張している．これは，対象とするカテゴリグラフから抽出した遷移確率行列が PageRank の収束条件を満たしていないことと，収束条件を満たすための一般的な方法が所属の度合いを算出するのにあまり適していないことに起因する．具体的には，べき乗法（後述）を用いて PageRank を収束させるためには遷移確率行列が原始行列（primitive matrix）となるよう修正する必要があり [4]，一般的な PageRank では意図的にある確率でランダムにグラフ中の別のノードに遷移（テレポート）させることでこれを解決している．本研究は，カテゴリ（トピック）への所属確率という，より関係性の明確な情報を得ることが目的であるため，上記の方法で収束条件を達成しようとした場合，得られる所属確率に多くのノイズ情報が含まれることになる．そのため，本研究ではカテゴリへの所属確率の計算に適した拡張を行い，収束条件の達成を図る．

2.2　ナイーブベイズによるテキスト分類

テキスト分類あるいは文書分類に関する研究については，これまで非常に多くの研究が行われてきた．文書分類とは，あらかじめ設定したカテゴリに対し，入力となる文書がどのカテゴリに属するかを決定するものであり，文書集合をいくつかのまとまりに分ける文書クラスタリングとは異なる．現時点において最も実用的な文書分類アルゴリズムの 1 つとして，ナイーブベイズ（NB）[1] があげられる．

ナイーブベイズでは，テキスト中に含まれる語句が互いに独立に発生したものであるというナイーブ（単純）な仮定を置き，それらの語句が出現したときのテキストのトピックへの所属確率を，ベイズの定理により求める．ナイーブベイズはシンプルでありながら高速に動作（学習時間が短い）し，精度も高いため，実用的な文書分類手法として一般に認識されている．また，教師データの削減や精度向上のため，ナイーブベイズの拡張として様々な手法 [9], [18] が提案されている．教師データが十分にある場合，ナイーブベイズは初期のシンプルな実装でも十分高い性能を発揮し，また，同じくシンプルな Complement NB [12] は実際にはてなブックマーク[*2]のエントリを分類するのに用いられている．一方で，教師データを用いずにテキスト分類を行う手法はあまり成功事例がないというのが実情である．

3.　提案手法

本研究では，Wikipedia をグラフ理論に基づいて解析することにより，既存の概念辞書ではあまり定義されていないような固有名詞や専門用語，新語を確率的にカテゴリに分類することを目指す．また，4 章では提案手法の応用として，ナイーブベイズによるテキスト分類を行う．以下ではまず，Wikipedia のカテゴリ構造について説明する．その後，提案手法のアプローチと具体的な計算方法について詳述する．

3.1　Wikipedia のカテゴリ構造

Wikipedia では，基本的に各記事（エンティティ）に対して 1 つ以上のカテゴリ（親カテゴリ）が割り当てられている．また，カテゴリにも同様に親カテゴリが割り当てられており，カテゴリ構造をなしている．この親カテゴリは，該当の記事あるいはカテゴリが所属すると思われるカテゴリであり，上位下位関係や全体部分関係を表すこともあれば，トピックや関連を表す場合もある．そのため，ある記事から親カテゴリをたどっていくと，ほとんど関係のないカテゴリに到達することが頻繁に起こりうる．たとえば，動物の “Lion” についての記事から親カテゴリをたどっていくと，“Humans,” “Economics,” “Education” などのあまり関係のないカテゴリに到達できる．これは，親カテゴリとして登録されるカテゴリが上位下位関係や全体部分関係だけでなく，様々な関係を表しているためである．このようなカテゴリに対する緩い制約により，Wikipedia のカテゴリ構造は**図 1** のような複数の親やループを許容する

[*2] http://b.hatena.ne.jp/

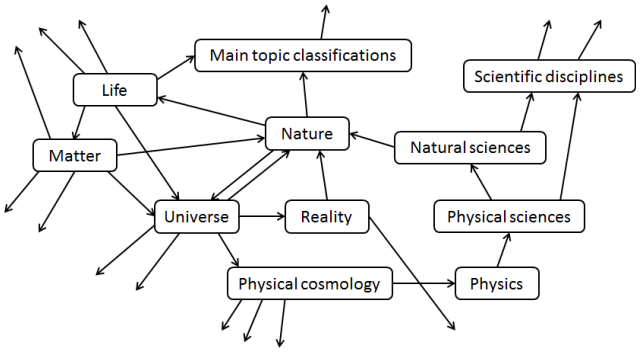

図 1　Wikipedia のカテゴリ構造の例
Fig. 1　An example of Wikipedia category structure.

ネットワーク構造となっている．なお，図 1 はすべてカテゴリであり，Wikipedia の各記事はこのようなカテゴリ構造において 1 つ以上のカテゴリに属している．このような構造のため，あるエンティティがどのカテゴリに属しているかという情報を，単純に親カテゴリや子カテゴリをたどるだけでは抽出できない．

3.2　ランダムウォークによる語句の確率的分類

前節で述べたように，Wikipedia のカテゴリ構造はネットワーク構造であるため，ある記事に対し，指定したカテゴリ（基底カテゴリ）に属するか否かを判断することが困難である．そこで本研究では，カテゴリに属するか否かではなく，どの程度の確率で属するかという数値として表現する．Wikipedia のカテゴリ構造では，ある記事から親カテゴリをたどるとき，そのパス上で出現しやすいカテゴリに対してより強く所属していると考えられる．この考え方に基づき，親カテゴリをたどるときに確率的にスコアを割り当て，より大きいスコアを持つカテゴリに強く所属すると見なす．これは，隣接ノードのいずれかに等確率で遷移するモデルであるランダムウォークを用いて表現できる．提案手法では，カテゴリをノード，親カテゴリへのリンクを有向リンクとしたグラフに対して，ランダムウォークを適用する．十分な時間が経過した後のランダムウォークによるスコアは，あるノードから出発したときに，そのカテゴリに到達する確率を表す．この確率を所属確率として用いる．

提案手法のアプローチについて，例を**図 2** に示す．図 2 では，記事 A から確率優先探索（確率が同じ場合はノード番号順）により各カテゴリへの所属確率を算出している．まず，記事 A は親カテゴリを 2 つ持っているため，それぞれのカテゴリ（カテゴリ 8，9）への遷移確率をそれぞれ $\frac{1}{2}$ とする．カテゴリ 8 は親カテゴリを 1 つしか持たないため，カテゴリ 3 への遷移確率をそのまま $\frac{1}{2}$，また，カテゴリ 9 は親カテゴリを 2 つ持っているため，カテゴリ 5，6 への遷移確率をそれぞれ $\frac{1}{4}$ とする．このような処理を繰り返すことにより，基底カテゴリへの所属確率を算出する．な

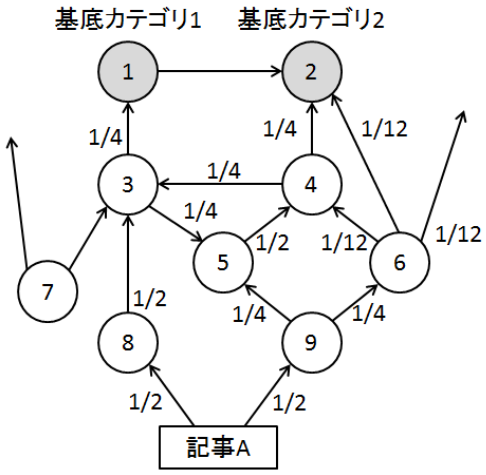

図 2　ランダムウォークによる記事 A の基底カテゴリへの所属確率計算
Fig. 2　Calculation of the probability that an article A belongs to each base category by random walk.

お，ここでは基底カテゴリに到達するか，ループの発生を検知した場合，親カテゴリの探索を中止している．すべてのカテゴリについて親カテゴリの探索が終了した，または中止された場合に処理を終了する．

基本的には，このようにある記事からスタートし，親カテゴリをたどることで基底カテゴリへの所属確率を算出するが，親カテゴリをたどるにつれて指数関数的に計算量が大きくなることや，ループに対する計算方法など，実際的な問題が発生する．そこで提案手法では，次節に示すように，Wikipedia のカテゴリネットワークに対してグラフ理論に基づく解析を行い，グラフカーネルを構築する．

3.3　グラフ理論に基づくカテゴリグラフカーネルの構築

本研究では，Wikipedia のカテゴリネットワークにおいて，ランダムウォークに基づく所属確率を効率的に算出するために，カテゴリグラフカーネルを構築する手法を提案する．カテゴリグラフカーネルとは，ある記事の親カテゴリを確率ベクトルとして表現したとき，そのベクトルとの内積計算によって基底カテゴリへの所属確率を算出可能な行列（あるいはベクトル群）である．すなわち，ある記事の親カテゴリの系列を入力とすると，カテゴリグラフカーネルによって基底カテゴリの系列と所属確率が出力される．なお，グラフカーネルには von Neuman カーネル [3] をはじめとして様々なものがあるが，本研究で提案するグラフカーネルは，ランダムウォークに基づく定常状態での遷移確率を表すシンプルなものである．カテゴリグラフカーネルは，基底カテゴリを祖先カテゴリ（親カテゴリをたどることで到達可能なカテゴリ）として持つすべてのカテゴリについて，各基底カテゴリへの所属確率をあらかじめ計算したものである．ここで前節と同様に問題となるのは，どうやって基底カテゴリへの所属確率を効率的に計算するか

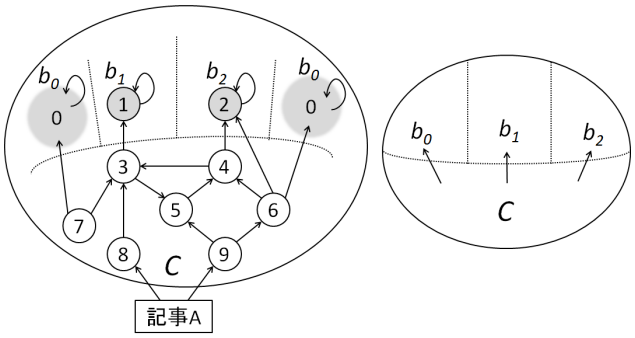

図 3　図 2 のグラフに対する基底カテゴリ $b_i \in B$ およびそれらを祖先に持つカテゴリ集合 C の再帰・非再帰の関係

Fig. 3　Relation of the recursion/non-recursion between base categories $b_i \in B$ and their descendant category set C.

という点である．以下ではカテゴリグラフの遷移確率行列を用いた手法について説明する．

まず，Wikipedia のカテゴリの中から分類に用いるカテゴリをユーザが選択（ここでは m 個選択したとする）し，基底カテゴリ $b_i \in B$（$i = 1, \cdots, m$）とする．次に，基底カテゴリのいずれかを祖先カテゴリとして持つカテゴリをすべて収集し，それらのカテゴリの集合を C とする．そして B および C について，親カテゴリへのリンクを有向リンクとしてランダムウォークに基づく遷移確率行列 $\mathbf{M}$ を作成する．ここで提案手法では，基底カテゴリに対する遷移確率として，自身に確率 1 で遷移するよう設定する．また，簡単のため，$c \notin C$ なるカテゴリ c をすべて 1 つのカテゴリとして扱い，それらの集合体として基底カテゴリ b_0 を追加する．これにより，すべての基底カテゴリ $b_i \in B$（$i = 0, \cdots, m$）はシンク（ランダムウォークにおけるスコアを吸収するノード）として機能し，基底カテゴリ以外のすべてのカテゴリ $c \in C$ は，必ず 1 つ以上の基底カテゴリへのパスを持つようになる．**図 3** は，図 2 のグラフについて B と C の再帰・非再帰の関係を表している[*3]．カテゴリ $c \in C$ から 1 度でも基底カテゴリのいずれかに遷移すると，以降その基底カテゴリに滞在することになる．つまり，どのカテゴリからスタートしても，定常状態ではいずれかの基底カテゴリに遷移した状態となっている．すなわち，定常状態における遷移行列 $\lim_{\alpha\to\infty} \mathbf{M}^\alpha$ を計算すれば，全カテゴリ $c \in C$ について，基底カテゴリ b_i への遷移確率 $P(b_i|c)$ を算出できる（$\sum_{b_i \in B} P(b_i|c) = 1$ を満たす）．そこで，提案手法ではべき乗法を用いて定常状態における遷移行列を導出する．なお，PageRank [4] でもべき乗法を用いて定常状態における各ノードのスコアを算出しているが，提案手法では，対象とする行列が既約ではない（もちろん原始的でもない）ことや，最終的に導出すべきものが行列である点で大きく異なる．以下では，べき乗法による収束の保証と初期ベクトルの設定方法について述べる．

[*3] C のカテゴリ間はそれぞれ相互に遷移できる関係ではないことに注意する．

3.3.1　べき乗法による収束の保証

べき乗法とは，絶対値最大の固有値と固有ベクトルを求める数値解法の 1 つである [13]．また，絶対値最大固有値が重解であるときは，それらの固有ベクトル群からなるベクトルが入力に応じて得られる．ここでは，遷移確率行列 $\mathbf{M}$ の定常状態を表す式において，カテゴリグラフカーネルを表すベクトル群 X が絶対値最大固有値の固有ベクトルとして出現することを証明し，べき乗法を用いてカテゴリグラフカーネルを導出できることを示す．

遷移確率行列 $\mathbf{M}$ は，以下のように基底カテゴリの部分とそれ以外のカテゴリの部分に分けられる．

$$\mathbf{M} = \begin{bmatrix} \mathbf{I}_{|B|} & \mathbf{0} \\ \mathbf{M_I} & \mathbf{M_0} \end{bmatrix} \tag{1}$$

$\mathbf{I}_{|B|}$ は $|B| \times |B|$ の単位行列，$\mathbf{0}$ は $|B| \times |C|$ のゼロ行列であり，基底カテゴリが自身にのみ遷移することを表している．$|B|$, $|C|$ はそれぞれの集合の要素数であり，$|B| = m+1$ である．また，$\mathbf{M_I}$ および $\mathbf{M_0}$ はそれぞれ $|C| \times |B|$, $|C| \times |C|$ の行列であり，カテゴリ $c \in C$ の親カテゴリへの遷移確率を表している．次に，$\lim_{\alpha\to\infty} \mathbf{M}^\alpha$ は以下の形の行列となる．

$$\lim_{\alpha\to\infty} \mathbf{M}^\alpha = \begin{bmatrix} \mathbf{I}_{|B|} & \mathbf{0} \\ \mathbf{M}_\infty & \mathbf{0} \end{bmatrix} \tag{2}$$

$|C| \times |B|$ の行列である $\mathbf{M}_\infty$ は，カテゴリ $c \in C$ が各基底カテゴリにそれぞれどの程度の確率で遷移するかを表しており，本研究で算出すべきカテゴリグラフカーネルの主要部分である．α が十分に大きいとき，$\mathbf{M}^\alpha$ は定常状態となり，以下の等式が成り立つ．

$$\mathbf{M}\mathbf{M}^\alpha = \mathbf{M}^\alpha \tag{3}$$

ここで，以下のような行列 $\mathbf{X}$ を考えると，$\mathbf{X}$ は各カテゴリが最終的にどの基底カテゴリに遷移するかを表したカテゴリグラフカーネルとなる．

$$\mathbf{X} = \begin{bmatrix} \mathbf{I}_{|B|} \\ \mathbf{M}_\infty \end{bmatrix} \tag{4}$$

$\mathbf{X}$ を用いると，式 (3) から以下の式が導かれる．

$$\mathbf{M}\mathbf{X} = \mathbf{X} \tag{5}$$

上式の $\mathbf{X}$ は遷移確率行列 $\mathbf{M}$ に対する固有値 1 の固有ベクトルに似た形をしているが，$\mathbf{X}$ はベクトルではなく，$m+1$ 個の線形独立なベクトルからなる行列であることに注意する．この $\mathbf{X}$ が何を意味しているのかを明らかにするため，特性方程式 $|\mathbf{M} - \lambda\mathbf{I}_{|B|+|C|}| = 0$ を用いて固有値 λ を算出する．

$$\begin{aligned} |\mathbf{M} - \lambda\mathbf{I}_{|B|+|C|}| &= \begin{vmatrix} (1-\lambda)\mathbf{I}_{|B|} & \mathbf{0} \\ \mathbf{M_I} & \mathbf{M_0} - \lambda\mathbf{I}_{|C|} \end{vmatrix} \\ &= (1-\lambda)^{|B|} |\mathbf{M_0} - \lambda\mathbf{I}_{|C|}| \\ &= (1-\lambda)^{m+1} |\mathbf{M_0} - \lambda\mathbf{I}_{|C|}| = 0 \end{aligned} \tag{6}$$

なお，$\mathbf{I}_{|B|+|C|}$ は $(|B|+|C|)\times(|B|+|C|)$ の単位行列，$\mathbf{I}_{|C|}$ は $|C|\times|C|$ の単位行列である．

ここで $\mathbf{M_0}$ について，$\lim_{\alpha\to\infty}\mathbf{M_0}^{\alpha}=\mathbf{0}$ に収束することから，$\mathbf{M_0}$ の固有値 λ はすべて $|\lambda|<1$ を満たす．したがって，$\mathbf{M}$ の絶対値最大固有値は 1 であり，かつ $m+1$ 個の重解である．以上より，$\mathbf{X}$ は最大固有値 1 に対する $m+1$ 個の独立なベクトルから合成される固有ベクトル（一般固有空間）を表していることが分かる．このことから，$\mathbf{X}$ はべき乗法を用いて求められる．

3.3.2　べき乗法によるカテゴリグラフカーネルの導出方法

前項で示したとおり，カテゴリグラフカーネル $\mathbf{X}$ はべき乗法を用いて算出できる．そこで，$\mathbf{X}$ に収束させるため，行列 $\mathbf{Y}$ を以下のように定義する．

$$\mathbf{Y}=\begin{bmatrix}\mathbf{I}_{|B|}\\ \mathbf{M}'\end{bmatrix} \tag{7}$$

$\mathbf{M}'$ は $\mathbf{M}_\infty$ と同じ $|C|\times|B|$ の行列で，任意の値を持つ．

この $\mathbf{Y}$ を $m+1$ 個の初期ベクトルとし，$\mathbf{Y}\leftarrow\mathbf{MY}$ の更新式を繰り返すことにより，$\mathbf{Y}$ は以下のような形の行列に収束する．

$$\mathbf{Y}=\begin{bmatrix}\mathbf{I}_{|B|}\\ \mathbf{M}'_\infty\end{bmatrix} \tag{8}$$

この行列が $\mathbf{M}$ の最大固有値 1 に対する一般固有空間に内包されることから，$\mathbf{M}'_\infty=\mathbf{M}_\infty$ であり，$\mathbf{Y}$ は $\mathbf{X}$ に収束していることが分かる．これにより，得られた行列 $\mathbf{X}$ をカテゴリグラフカーネルとして，ある記事の親カテゴリの列とその確率から，基底カテゴリの列とその確率に変換できる．

3.3.3　カテゴリグラフカーネルの導出アルゴリズム

カテゴリグラフカーネルを導出するためのアルゴリズムは非常にシンプルに記述できる．先ほどは簡単のため，$c\notin C$ なるカテゴリ c をすべて 1 つのカテゴリとして扱い，それらの集合体として基底カテゴリ b_0 を追加していたが，実際には b_0 への所属確率は意味をなさないため，b_0 を追加しなくても問題ない[*4]．Wikipedia のカテゴリ集合を C_{all}，カテゴリ数を N とし，以下のアルゴリズムによりカテゴリグラフカーネルを構築する．

(1) Wikipedia のカテゴリ構造から，親カテゴリへのリンクを遷移確率行列 $\mathbf{M}$ として抽出する．すなわち，$c_i\in C_{all}$ $(i=1,\cdots,N)$ に対して c_i の親カテゴリの集合を A_i，親カテゴリ数を $|A_i|$ とすると，i 行 j 列目の要素 p_{ij} について，$c_j\in A_i$ のとき $p_{ij}=\frac{1}{|A_i|}$，$c_j\notin A_i$ のとき $p_{ij}=0$ に設定する．親カテゴリを持たないカテゴリについては，すべてのカテゴリに対して遷移確率を 0 として設定する．

(2) 基底カテゴリとして選択する m 種類のカテゴリ $c_k\in C_{all}$ $(k=k_1,\cdots,k_m)$ に対し，自身に確率 1 で遷移するよう $\mathbf{M}$ を再設定する．すなわち，k 行 x 列目の要素 p_{kx} について，$k=x$ のとき $p_{kx}=1$，$k\neq x$ のとき $p_{kx}=0$ とする．

(3) $\mathbf{M}$ の k 列目 $(k=k_1,\cdots,k_m)$ のみをベクトルとして取り出して，それらの列ベクトルを合わせた行列 $\mathbf{Y}$ を初期行列とし，$\mathbf{Y}$ が十分に収束するまで $\mathbf{Y}\leftarrow\mathbf{MY}$ を繰り返す．

あらかじめ遷移確率行列 $\mathbf{M}$ を抽出しておけば，基底カテゴリの選択に対して (2)，(3) の処理を行うだけでカテゴリグラフカーネルを構築できる．PageRank と同様に，$\mathbf{Y}\leftarrow\mathbf{MY}$ は数十回程度で収束する．実際に 5 章で使用する Web データセットに対してカテゴリグラフカーネルを構築したときの収束の様子を**図 4** に示す．図 4 より，初めのうちは基底カテゴリに近いカテゴリについてのみベクトルの値が変化し，反復回数が増えるにつれて徐々に末端のカテゴリについてもベクトルの値が変化していることが予測できる．その後，値の変化するベクトルの数が収束していき，それぞれ 25 回目でいずれかの要素が 0.01，35 回目で 0.001，45 回目で 0.0001 以上変化するベクトルの数が 0 となっている．

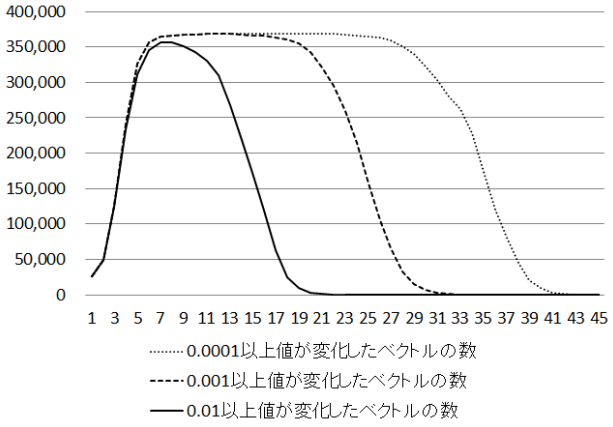

図 4　べき乗法の反復回数と値の変化したベクトルの数

Fig. 4　The number of iterations of power iteration method and the number of vectors whose values change.

3.4　語句の確率的分類の出力例

提案手法により Wikipedia の記事（エンティティ）を確率的に分類した例を**表 1** に示す．なお，ここでは 5 章で使用する Web データセットに対する基底カテゴリを用いており，べき乗法による反復回数を 50 回としている．出力例から，主観的にはある程度精度良く所属確率を算出できており，Wikipedia のカテゴリ構造を用いた確率的な語句の分類が機能していることを確認した．また，複数のカテゴリに属すると考えられるエンティティについても，それら複数のカテゴリへの所属確率が得られていることを確認した．

[*4] $c\notin C$ について余分に計算するコストと，あらかじめ $c\in C$ を選出するコストのトレードオフとなるが，筆者らの経験的に後者のほうが計算時間が大きかった．

表 1　語句の確率的分類の例
Table 1　An example of probabilistic term classification.

基底カテゴリ	Business	Computers	Culture Arts Entertainment	Education Science	Engineering	Health	Politics Society	Sports
Goldman Sachs	0.258	0.000	0.027	0.111	0.002	0.002	0.088	0.016
Subprime lending	0.575	0.000	0.026	0.139	0.001	0.001	0.035	0.000
Twitter	0.024	0.104	0.061	0.167	0.026	0.000	0.122	0.000
Microsoft Windows	0.013	0.232	0.006	0.041	0.017	0.000	0.008	0.000
Kabuki	0.018	0.000	0.307	0.244	0.001	0.001	0.112	0.000
Lady Gaga	0.037	0.000	0.180	0.140	0.001	0.001	0.116	0.000
Magnetism	0.020	0.000	0.004	0.681	0.024	0.000	0.010	0.000
Stanford University	0.035	0.000	0.016	0.215	0.002	0.003	0.101	0.033
Derrick	0.234	0.000	0.004	0.168	0.338	0.000	0.004	0.000
Dehydration	0.036	0.000	0.079	0.222	0.000	0.256	0.111	0.000
AIDS	0.045	0.000	0.050	0.247	0.001	0.162	0.147	0.000
Anarchism	0.254	0.000	0.139	0.252	0.000	0.000	0.322	0.000
Barack Obama	0.048	0.000	0.056	0.190	0.002	0.001	0.190	0.001
Football	0.016	0.000	0.105	0.145	0.000	0.000	0.060	0.500
Koji Murofushi	0.034	0.000	0.025	0.139	0.002	0.006	0.112	0.182
Edubuntu	0.051	0.493	0.069	0.246	0.092	0.000	0.048	0.000
Bibio	0.060	0.000	0.342	0.306	0.003	0.001	0.288	0.000
Kaikai Kiki	0.136	0.000	0.244	0.300	0.001	0.001	0.318	0.000
Tricuspid valve stenosis	0.069	0.001	0.060	0.320	0.001	0.283	0.266	0.000
S&P Global 1200	0.686	0.002	0.034	0.178	0.007	0.000	0.094	0.000
Knattleikr	0.022	0.000	0.119	0.295	0.002	0.002	0.093	0.468

基底カテゴリ別にみると，どのようなエンティティに対しても多少の確率を出力するカテゴリ（“Culture,” “Society” など）と，関連のないエンティティに対してはいっさい確率を出力しないカテゴリ（“Computers,” “Engineering” など）に大別できることが分かる．これは，Wikipedia のカテゴリ構造において，ネットワークに近い形をしている部分と，DAG 構造[*5]に近い形をしている部分があることに由来すると考えられる．ネットワーク構造に位置するカテゴリほどノイズとして所属確率を出力しやすい傾向があるため，所属確率が最大となる基底カテゴリが直感とは一致しないケースも存在する．たとえば，“Twitter” は “Computers” に最も強く属していると考えられるが，実際には “Society” への所属確率が最大となっている．このようなケースに対応するためには，カテゴリ間の意味的なつながりを考慮した拡張が必要となる．

また，知名度の低い記事についてみても，ある程度正しく分類できていることが分かる．“Edubuntu”（教育向けの Linux のディストリビューション），“Bibio”（イギリスの音楽家），“Kaikai Kiki”（日本のアーティスト集団），“Tricuspid valve stenosis”（心臓弁膜症の一種），“S&P Global 1200”（株価指数の 1 つ），“Knattleikr”（アイスランドのバイキングの間で行われているスポーツ）などの記事は記述が少なく，英語圏ではかなり知名度が低いエンティティであると考えられるが，知名度の高い記事と同様に提案手法がうまく機能している．これは，提案手法が記事の内容ではなくカテゴリ構造を用いており，記事に対して正しくカテゴリが付与されていれば記事の充実度にあまり影響を受けないためである．

なお，提案手法では，基底カテゴリに到達できない記事については確率値を計算することはできないが，このような記事はいずれの基底カテゴリにも属さない特殊な記事であると見なす．Zesch らの調査では（ドイツ語版）Wikipedia のカテゴリ構造の最大連結成分は全カテゴリの 99.8%を占めており [24]，一般的なカテゴリどうしはほとんど連結していると考えられる．また，Wikipedia のカテゴリ構造自体に明らかな誤りがあり，その結果正しい確率値を割り当てられないケースに対しては，意味を考慮せずグラフ解析を行う提案手法では対応できない．このような問題に対応するためには，カテゴリ間の意味的なつながりを考慮した拡張が必要となる．

なお，これらの見解はあくまでの著者らの主観によるものであり，客観的に提案手法の有効性を示すものではない．そこで，5 章の評価実験では，提案手法の応用としてテキスト分類（スニペット分類）を想定し，複数のデータセットを用いた評価を行う．次章では提案手法を用いたテキスト分類手法について説明する．

[*5] 複数の親を許容する木構造のこと．

4. テキスト分類への応用

前章で説明した語句の確率的分類の応用として，自然言語で記述されたテキストの分類を行う．提案手法では語句を確率的に分類していることから，その結果をナイーブベイズの教師データとして用いることが可能である．すなわち，Wikipedia のカテゴリ構造をより直接的な形でテキスト分類に利用できる．本研究では，確率的な語句の分類結果を教師データとし，ナイーブベイズを拡張した手法 [16] に教師データをあてはめることで，テキスト分類を行う．なお，拡張ナイーブベイズは入力系列が確率的に予測可能な場合に適用できる手法であり，自然文の入力に対して有効である [15]．通常のナイーブベイズを用いた場合，与えられた入力語句 $t_1, \cdots, t_N$ がすべてキーフレーズである（すなわちキーフレーズ集合 $T = \{t_1, \cdots, t_N\}$）とし，基底カテゴリ b への所属確率 $P(b|T)$ を，個々の確率 $P(b|t_1), \cdots, P(b|t_N)$ から算出する．一方，拡張ナイーブベイズでは，与えられた入力語句をそのまま用いるのではなく，キーフレーズ集合 T に含まれるか否かを確率的に定義してからナイーブベイズを適用する．これにより，特徴的な語句ほど基底カテゴリの推測に影響を与えやすくなる．具体的には，入力テキスト（入力語句 $t_1, \cdots, t_N$）が与えられたとき，そこからキーフレーズ集合 T を確率的に予測し，基底カテゴリ b への所属確率 $P(b|T)$ を，以下の式により算出する．

$$P(b|T) \propto \frac{\prod_{k=1}^{K}\Big(P(t_k \in T)P(b|t_k) + \big(1 - P(t_k \in T)\big)P(b)\Big)}{P(b)^{K-1}} \tag{9}$$

K は入力テキストに含まれるキーフレーズ候補の数，$P(t_k \in T)$ は語句 t_k がキーフレーズ集合 T に含まれる確率，$P(b|t_k)$ は語句 t_k が与えられたときにそれが基底カテゴリ b に属する確率，$P(b)$ は基底カテゴリ b の事前確率である．また，E をエンティティ集合とすると，$P(b|t_k) = \sum_{e_i \in E} P(b|e_i)P(e_i|t_k)$ である．ここでは，$P(e_i|t_k)$ および $P(t_k \in T)$ については拡張ナイーブベイズを用いた関連語取得に関する研究 [15] で使用しているものを利用する．すなわち，Wikipedia の情報を用いて，それぞれ以下の式により算出する．

$$P(t_k \in T) \approx \frac{CountDocuments(t_k \in Key)}{CountDocuments(t_k)} \tag{10}$$

$$P(e|t_k) \approx \frac{CountAnchortexts(t_k, e)}{\sum_{e_i \in E} CountAnchortexts(t_k, e_i)} \tag{11}$$

$CountDocuments(t_k)$ は語句 t_k が出現する記事数，$CountDocuments(t_k \in Key)$ は語句 t_k がアンカーテキストとして出現する記事数，$CountAnchortexts(t_k, e)$ は語句 t がアンカーテキストとしてエンティティ e の記事にリンクされている回数である．なお，式 (10) は Mihalcea らの研究 [5] の Keyphraseness，式 (11) は Milne らの研究 [6] の Commonness である．ここで E は Wikipedia で定義されているエンティティ（記事）集合である．

$P(b|e_i)$ は提案手法のカテゴリグラフカーネルを用いて算出できる（3 章）．また，$P(b)$ は基底カテゴリ b の一般度を表すものであることから，どの程度所属されやすいか，を算出することでおよその値が得られる．具体的には，以下の式により算出する．

$$P(b) \propto \sum_{e_i \in E} P(b|e_i) \tag{12}$$

なお，$\sum_{b \in B} P(b) = 1$ となるよう正規化する．これらの情報と式 (9) を用いることで，指定した基底カテゴリに対するテキストの分類が可能となる．

5. 評価

5.1 評価環境

提案手法の有効性を客観的に評価するため，テキスト分類において評価を行った．データセットとして，Phan らの研究 [10] で用いられている Web 検索結果のスニペット（Web データセット），および PhysOrg.com [*6] から取得した科学に関する記事のタイトルとスニペット（Sci. データセット）を利用した．データセットの各カテゴリの名前を基に，Wikipedia から基底カテゴリを選択し，べき乗法による反復回数を 50 回としてカテゴリグラフカーネルを構築し，4 章で説明した手法を用いて正しくスニペットを分類できるかどうかを検証した．Wikipedia のデータは，2009 年 3 月 6 日の英語版のダンプを使用した．なお，カテゴリ（ノード）数は 455,854，カテゴリ間のリンク（エッジ）数は 914,738 であった．

各データセットの統計データを**表 2**，**表 3** に示す．Web データセットは，各カテゴリに対して排他的になるよう選択された検索クエリによってそれぞれ 20 件または 30 件の検索結果のスニペットを取得したものである．基底カテゴリは Wikipedia の中から該当する 13 カテゴリ “Business,” “Economics,” “Computing,” “Culture,” “Arts,” “Entertainment,” “Education,” “Science,” “Engineering,” “Health,” “Politics,” “Society,” “Sports” を選択 [*7] した．Web データセットでは，トレーニングセットとテストセットが分けられているため，テストセットに対して評価を行った．Sci. データセットは，PhysOrg.com から各カテゴリの記事を 300 件ずつ取得し，タイトルとスニペットを取得したものである．基底カテゴリは Wikipedia の中から該当する 10 カテゴリ “Nanotechnology,” “Physics,”

[*6] http://www.physorg.com/

[*7] Wikipedia のカテゴリにおいて “Business” は主に「企業」という意味で用いられているため，「経済」の意味を包含する目的で “Economics” も “Business” の基底カテゴリとして選択した．

表 2　Web データセット
Table 2　Web dataset.

基底カテゴリ	トレーニングセット		テストセット	
	検索クエリ数	スニペット数	検索クエリ数	スニペット数
Business (Bus.)	60	1,200	10	300
Computers (Comp.)	60	1,200	10	300
Culture-Arts-Entertainment (Cult.)	94	1,880	11	330
Education-Science (Sci.)	118	2,360	10	300
Engineering (Eng.)	11	220	5	150
Health (Heal.)	44	880	10	300
Politics-Society (Pol.)	60	1,200	10	300
Sports (Spo.)	56	1,120	10	300
合計		10,060		2,280

表 3　Sci. データセット
Table 3　Sci. dataset.

基底カテゴリ	スニペット数
Nanotechnology (Nano.)	300
Physics (Phys.)	300
Space-Earth (Spa.)	300
Electronics (Elec.)	300
Technology (Tech.)	300
Chemistry (Chem.)	300
Biology (Bio.)	300
Medicine-Health (Med.)	300
合計	2,400

“Space,” “Earth,” “Electronics,” “Technology,” “Chemistry,” “Biology,” “Medicine,” “Health” を選択した．

評価対象は，提案手法（Wikipedia を用いた確率的な語句分類とナイーブベイズ），語句の分類を親カテゴリをたどる際のホップ数に応じて決定する手法，WordNet を用いた手法，教師ありナイーブベイズ（NB）手法とした．ホップ数ベースの手法では，N ホップ（$N = 1, \cdots, 6$）までの祖先カテゴリに所属すると見なし，語句の重要度を表す Keyphraseness [5] の重み付き和としてテキストの分類を行った．また，WordNet を用いた手法では，祖先カテゴリにすべて所属すると見なし，最も出現回数の多いカテゴリに分類した．これは，WordNet では DAG 構造により正確に上位下位関係が定義されており，単純に親カテゴリをたどる手法がうまく機能するためである．教師ありナイーブベイズでは，Web データセットにおいてはトレーニングセットを教師データとして利用し，使用するスニペット数を変化させた．また，Sci. データセットにおいては 5 分割交差検定を行った．これらの手法では，テキスト入力に対して基底カテゴリの順位付きリストを出力として返すため，評価指標として最上位の適合率に加え，正解のカテゴリの順位の逆数の平均（MRR）を用いた．MRR は，順位付けのタスクの評価指標としてよく用いられ，正解のカテゴリが上位であればあるほど高いスコアが与えられる．

5.2　評価結果

評価結果を**表 4**，**表 5**，**表 6**，**表 7** に示す．表ではそれぞれの基底カテゴリごとの評価指標と全体の評価指標を計算している．表 4，5 の結果からみると，親カテゴリをたどる際のホップ数で所属を決定する方法と比較して，所属を確率として表す提案手法のほうが全体的に安定して高い精度で分類できている．ホップ数ベースの手法では，ある 1 つのホップ数ではすべての基底カテゴリに対して高い精度を達成するのが難しいことが分かる．また，ホップ数が大きくなると，ほとんどの入力に対して少数の支配的なカテゴリ（“Culture” や “Society”）のスコアが高くなることが問題となっている．一方，提案手法では，確率的に語句を分類することにより，ナイーブベイズといった確率的な手法との組合せが可能になったことが精度向上や精度安定につながっていると考えられる．同様の傾向は Sci. データセット（表 6，7）においてもみられる．また，Sci. データセットでは，提案手法の場合 Technology に対して適合率が落ちているが，最上位以外のカテゴリについても考慮した指標である MRR では，ある程度良いスコアとなっている．これは，“Technology” と “Electronics” のスニペットが類似していることに加えて，2 つのカテゴリが Wikipedia のカテゴリネットワークにおいて近くに存在しており，“Technology” に属するべきテキストの多くが，“Electronics” に対してより強く属していると見なされたためである．このことから，分類したいカテゴリの意味と Wikipedia におけるカテゴリの意味のずれにより，類似した意味のカテゴリ間では分類が困難になることが問題としてあげられる．そのため，そのような類似したカテゴリをそれぞれ基底カテゴリとして選択したい場合，その違いを認識できるよう慎重に基底カテゴリを選択する必要がある．

WordNet を用いた手法についてみると，WordNet はスニペットの分類に対してあまり効果的でないことが分かる．これは，WordNet では固有名詞，専門用語，新語をあまり定義していないことや，親カテゴリが基本的に上位下位関係を表すものであることに由来する．実際，多くの

表 4　適合率（Web データセット）
Table 4　Precision (Web dataset).

基底カテゴリ	Bus.	Comp.	Cult.	Sci.	Eng.	Heal.	Pol.	Spo.	All
教師あり NB									
全部 (10,060)	0.787	0.837	0.879	0.853	0.773	0.830	0.700	0.883	0.821
1/2 (5,030)	0.727	0.760	0.836	0.760	0.640	0.780	0.660	0.857	0.761
1/5 (2,012)	0.720	0.777	0.785	0.760	0.700	0.780	0.563	0.817	0.741
1/10 (1,006)	0.623	0.653	0.752	0.743	0.620	0.697	0.520	0.793	0.680
1/20 (503)	0.600	0.657	0.621	0.740	0.593	0.627	0.330	0.730	0.614
1/50 (201)	0.537	0.427	0.482	0.583	0.027	0.563	0.327	0.623	0.474
1/100 (100)	0.527	0.370	0.376	0.663	0.033	0.580	0.160	0.447	0.418
WordNet	0.417	0.240	0.200	0.217	0.033	0.100	0.027	0.553	0.236
Wikipedia									
1 ホップ	0.363	0.273	0.358	0.183	0.080	0.193	0.283	0.177	0.263
2 ホップ	0.627	0.457	0.545	0.350	0.047	0.197	0.507	0.580	0.412
3 ホップ	0.703	0.723	0.685	0.520	0.120	0.500	0.610	0.667	0.594
4 ホップ	0.563	0.727	0.791	0.437	0.047	0.267	0.797	0.613	0.522
5 ホップ	0.303	0.610	0.879	0.410	0.013	0.097	0.873	0.337	0.436
6 ホップ	0.140	0.427	0.842	0.397	0.000	0.013	0.950	0.173	0.349
提案手法	0.737	0.837	0.658	0.630	0.547	0.713	0.513	0.797	0.687

表 5　MRR（Web データセット）
Table 5　MRR (Web dataset).

基底カテゴリ	Bus.	Comp.	Cult.	Sci.	Eng.	Heal.	Pol.	Spo.	All
教師あり NB									
全部 (10,060)	0.867	0.909	0.926	0.915	0.861	0.895	0.825	0.925	0.893
1/2 (5,030)	0.827	0.858	0.895	0.849	0.788	0.863	0.792	0.906	0.852
1/5 (2,012)	0.825	0.868	0.858	0.853	0.815	0.849	0.718	0.874	0.834
1/10 (1,006)	0.743	0.787	0.839	0.837	0.747	0.790	0.675	0.857	0.788
1/20 (503)	0.726	0.786	0.746	0.841	0.727	0.739	0.519	0.810	0.738
1/50 (201)	0.673	0.590	0.657	0.734	0.295	0.694	0.537	0.745	0.637
1/100 (100)	0.662	0.551	0.578	0.791	0.280	0.699	0.365	0.619	0.587
WordNet	0.452	0.263	0.221	0.268	0.037	0.108	0.047	0.608	0.264
Wikipedia									
1 ホップ	0.418	0.300	0.391	0.201	0.090	0.207	0.301	0.180	0.274
2 ホップ	0.715	0.541	0.577	0.438	0.081	0.288	0.585	0.629	0.509
3 ホップ	0.812	0.817	0.777	0.660	0.257	0.637	0.760	0.729	0.710
4 ホップ	0.729	0.813	0.889	0.644	0.237	0.492	0.888	0.743	0.711
5 ホップ	0.573	0.731	0.936	0.638	0.221	0.398	0.935	0.572	0.656
6 ホップ	0.473	0.586	0.917	0.634	0.211	0.320	0.974	0.429	0.596
提案手法	0.827	0.889	0.785	0.782	0.698	0.779	0.722	0.862	0.799

スニペットに対して，トピックの分類に利用できる語句が WordNet にまったく存在していなかった．WordNet は，語句間の上位下位関係により，推論を用いた様々なアプリケーションに適用できるが，実データ（特にテキストが短い場合）に対してトピックによる分類を行うには情報量が少ないと考えられる．

提案手法と教師ありのナイーブベイズによるテキスト分類手法を比較すると，Web データセット（表 4，5）において，教師データを 1,000 件程度用いた場合と同等の適合率および MRR となっている．提案手法では教師データを用いていないことから，Wikipedia がテキスト分類に対する正解データとして有効であるといえる．この結果から，教師データが十分に用意できない場合，あるいはそこまで高い精度が要求されない場合においては，提案手法を用いたテキスト分類が効果的であることが分かる．たとえば，Web 検索のスニペットをいくつかのカテゴリに分類することで，検索結果を見やすく表示するようなアプリケーションが考えられる．

6. おわりに

本研究では，Wikipedia のカテゴリ構造をグラフと見なして解析し，確率的に語句を分類する手法を提案した．具

表 6　適合率（Sci. データセット）
Table 6　Precision (Sci. dataset).

基底カテゴリ	Nano.	Phys.	Spa.	Elec.	Tech.	Chem.	Bio.	Med.	All
教師あり NB	0.687	0.570	0.763	0.820	0.607	0.470	0.647	0.717	0.660
WordNet	0.033	0.010	0.433	0.000	0.100	0.080	0.043	0.243	0.118
Wikipedia									
1 ホップ	0.320	0.323	0.243	0.043	0.173	0.407	0.420	0.380	0.289
2 ホップ	0.423	0.560	0.380	0.313	0.210	0.410	0.550	0.393	0.405
3 ホップ	0.190	0.530	0.630	0.630	0.233	0.363	0.643	0.687	0.488
4 ホップ	0.003	0.610	0.710	0.563	0.387	0.340	0.657	0.630	0.488
5 ホップ	0.000	0.580	0.753	0.400	0.557	0.283	0.543	0.633	0.469
6 ホップ	0.000	0.547	0.740	0.027	0.697	0.167	0.423	0.583	0.398
提案手法	0.537	0.473	0.713	0.623	0.157	0.400	0.480	0.707	0.511

表 7　MRR（Sci. データセット）
Table 7　MRR (Sci. dataset).

基底カテゴリ	Nano.	Phys.	Spa.	Elec.	Tech.	Chem.	Bio.	Med.	All
教師あり NB	0.822	0.729	0.851	0.886	0.740	0.681	0.778	0.824	0.789
WordNet	0.047	0.010	0.437	0.000	0.120	0.085	0.057	0.250	0.126
Wikipedia									
1 ホップ	0.364	0.374	0.266	0.049	0.186	0.453	0.453	0.424	0.325
2 ホップ	0.532	0.660	0.462	0.372	0.266	0.561	0.635	0.526	0.502
3 ホップ	0.387	0.683	0.740	0.746	0.423	0.571	0.770	0.798	0.640
4 ホップ	0.170	0.755	0.817	0.724	0.620	0.549	0.801	0.768	0.650
5 ホップ	0.140	0.739	0.846	0.628	0.753	0.507	0.733	0.779	0.641
6 ホップ	0.126	0.724	0.840	0.441	0.828	0.413	0.659	0.749	0.597
提案手法	0.639	0.671	0.820	0.726	0.521	0.565	0.672	0.800	0.677

体的には，親カテゴリへの遷移確率行列を作成し，分類したいカテゴリ（基底カテゴリ）を意図的にシンクとして自身に遷移するよう行列を修正した後，べき乗法によりカテゴリグラフカーネルを構築する．カテゴリグラフカーネルを用いることで，ある Wikipedia の記事（エンティティ）に対して，親カテゴリのベクトルから基底カテゴリへの所属確率を表すベクトルに変換できる．また，エンティティの確率的な分類の応用として，ナイーブベイズを基にしたテキスト分類手法を提案した．評価実験により，提案手法である確率的な語句分類の有効性を確認した．

今後の予定として，分類したいカテゴリと Wikipedia のカテゴリの意味の相違を考慮し，ユーザが正しく基底カテゴリを選択できるような仕組みを検討する．たとえば，ごく少数の正解データを与えることにより，大きく精度を向上させることができる可能性がある．あるいは，Wikipedia のカテゴリ構造を可視化することにより，ユーザが基底カテゴリを直感的に正しく選択できるようなインタフェースを導入することも重要であると考えられる．また，よりノイズの少ない語句分類のため，Wikipedia の記事間リンクを用いて関連記事どうしで分類結果の誤りを発見するような方法も考えられる．

謝辞　本研究の一部は，科学研究費補助金基盤研究 B (21300032)，および日本学術振興会特別研究員奨励費（24-807）の助成によるものである．ここに記して謝意を表す．

参考文献

[1] Domingos, P. and Pazzani, M.: On the Optimality of the Simple Bayesian Classifier under Zero-One Loss, *Machine Learning*, Vol.29, No.2-3, pp.103–130 (1997).
[2] Fellbaum, C.: *WordNet: An Electronic Lexical Database*, The MIT Press (1998).
[3] Kandola, J.S., Shawe-Taylor, J. and Cristianini, N.: Learning Semantic Similarity, *Proc. Advances in Neural Information Processing Systems* (*NIPS*), pp.657–664 (2002).
[4] Langville, A.N. and Meyer, C.D.: *Google's PageRank and Beyond: The Science of Search Engine Rankings*, Princeton University Press (2006).
[5] Mihalcea, R. and Csomai, A.: Wikify! Linking Documents to Encyclopedic Knowledge, *Proc. ACM Conference on Information and Knowledge Management* (*CIKM*), pp.233–241 (2007).
[6] Milne, D. and Witten, I.H.: Learning to Link with Wikipedia, *Proc. ACM Conference on Information and Knowledge Management* (*CIKM*), pp.509–518 (2008).
[7] 中山浩太郎，原　隆浩，西尾章治郎：人工知能研究の新しいフロンティア：Wikipedia，人工知能学会誌，Vol.22, No.5, pp.693–701 (2007).
[8] Nastase, V. and Strube, M.: Decoding Wikipedia Categories for Knowledge Acquisition, *Proc. National Conference on Artificial Intelligence* (*AAAI*), pp.1219–1224 (2008).
[9] Nigam, K., Mccallum, A.K., Thrun, S. and Mitchell, T.:

Text Classification from Labeled and Unlabeled Documents using EM, *Machine Learning*, Vol.39, No.2-3, pp.103–134 (2000).
[10] Phan, X.-H., Nguyen, L.-M. and Horiguchi, S.: Learning to Classify Short and Sparse Text & Web with Hidden Topics from Large-scale Data Collections, *Proc. International World Wide Web Conference* (*WWW*), pp.91–100 (2008).
[11] Ponzetto, S.P. and Strube, M.: Deriving a Large Scale Taxonomy from Wikipedia, *Proc. National Conference on Artificial Intelligence* (*AAAI*), pp.1440–1445 (2007).
[12] Rennie, J.D.M., Shih, L., Teevan, J. and Karger, D.R.: Tackling the Poor Assumptions of Naive Bayes Text Classifiers, *Proc. International Conference on Machine Learning* (*ICML*), pp.616–623 (2003).
[13] 佐藤次男，中村理一郎：よくわかる数値計算アルゴリズムと誤差解析の実際，日刊工業新聞社 (2001).
[14] Schönhofen, P.: Identifying Document Topics Using the Wikipedia Category Network, *Proc. IEEE/WIC/ACM International Conference on Web Intelligence* (*WI*), pp.456–462 (2006).
[15] 白川真澄，中山浩太郎，原　隆浩，西尾章治郎：Wikipediaとナイーブベイズを用いた自然文に対する関連語句取得手法，第4回データ工学と情報マネジメントに関するフォーラム（DEIM2012），D7-2 (2012).
[16] Shirakawa, M., Wang, H., Song, Y., Wang, Z., Nakayama, K., Hara, T. and Nishio, S.: Entity Disambiguation based on a Probabilistic Taxonomy, Technical Report MSR-TR-2011-125, Microsoft Research (2011).
[17] Strube, M. and Ponzetto, S.P.: WikiRelate! Computing Semantic Relatedness using Wikipedia, *Proc. National Conference on Artificial Intelligence* (*AAAI*), pp.1419–1424 (2006).
[18] Su, J., Shirab, J.S. and Matwin, S.: Large Scale Text Classification using Semi-supervised Multinomial Naive Bayes, *Proc. International Conference on Machine Learning* (*ICML*), pp.97–104 (2011).
[19] Suchanek, F.M., Kasneci, G. and Weikum, G.: YAGO: A Core of Semantic Knowledge, *Proc. International World Wide Web Conference* (*WWW*), pp.697–706 (2007).
[20] 隅田飛鳥，吉永直樹，鳥澤健太郎：Wikipediaの記事構造からの上位下位関係抽出，自然言語処理，Vol.16, No.3, pp.3–24 (2009).
[21] Syed, Z.S., Finin, T. and Joshi, A.: Wikipedia as an Ontology for Describing Documents, *Proc. International Conference on Weblogs and Social Media* (*ICWSM*), pp.136–144 (2008).
[22] Vapnik, V.N.: *The Nature of Statistical Learning Theory*, Springer-Verlag (1995).
[23] Yeh, E., Ramage, D., Manning, C.D., Agirre, E. and Soroa, A.: WikiWalk: Random Walks on Wikipedia for Semantic Relatedness, *Proc. Workshop on Graph-based Methods for Natural Language Processing* (*TextGraphs-4*), pp.41–49 (2009).
[24] Zesch, T. and Gurevych, I.: Analysis of the Wikipedia Category Graph for NLP Applications, *Proc. Workshop on Graph-based Methods for Natural Language Processing* (*TextGraphs-2*), pp.1–8 (2007).

白川 真澄　（学生会員）

大阪大学大学院情報科学研究科博士後期課程在学中．2010年同大学院情報科学研究科博士前期課程修了．2011年本学会山下記念研究賞受賞．Webマイニング，データベースに関する研究に従事．日本データベース学会，言語処理学会各学生会員．

中山 浩太郎　（正会員）

東京大学知の構造化センター特任講師．2007年大阪大学大学院情報科学研究科博士後期課程修了．同年4月から同大学院情報科学研究科特任研究員．人工知能，Webマイニングに関する研究に従事．IEEE，ACM，電子情報通信学会，人工知能学会，日本データベース学会各会員．

原 隆浩　（正会員）

1995年大阪大学工学部情報システム工学科卒業．1997年同大学大学院工学研究科博士前期課程修了．同年同大学院工学研究科博士後期課程中退後，同大学院工学研究科情報システム工学専攻助手，2002年同大学院情報科学研究科マルチメディア工学専攻助手，2004年より同大学院情報科学研究科マルチメディア工学専攻准教授となり，現在に至る．工学博士．1996年本学会山下記念研究賞受賞．2000年電気通信普及財団テレコムシステム技術賞受賞．2003年本学会研究開発奨励賞受賞．2008年，2009年本学会論文賞受賞．モバイルコンピューティング，ネットワーク環境におけるデータ管理技術に関する研究に従事．IEEE，ACM，電子情報通信学会，日本データベース学会各会員．

西尾 章治郎　（フェロー）

昭和 50 年京都大学工学部数理工学科卒業．昭和 55 年同大学大学院工学研究科博士後期課程修了．工学博士．京都大学工学部助手，大阪大学基礎工学部および情報処理教育センター助教授を経て，平成 4 年大阪大学工学部教授，平成 14 年同大学大学院情報科学研究科教授となり，現在に至る．その間，大阪大学サイバーメディアセンター長，大学院情報科学研究科長，理事・副学長を歴任．データベースシステムにおけるデータおよび知識管理に関する研究に従事し，紫綬褒章，立石賞功績賞等を授与される．日本学術会議会員．本会では理事を歴任し，論文賞，功績賞を受賞．IEEE，電子情報通信学会フェロー．

（担当編集委員　村田 真樹）