Wikipediaを用いた木構造カーネルによるコメント分類

武田 昌大[1]　竹村 直規[2]　小林 伸行[3]　椎名 広光[4]
[1,2] 岡山理科大学大学院 総合情報研究科 情報科学専攻
[3] 山陽学園大学 総合人間学部 生活心理学科
[4] 岡山理科大学 総合情報学部 情報科学科
i15im02tm@ous.jp[1], i15im03tn@ous.jp[2], koba_nob@sguc.ac.jp[3],
shiina@mis.ous.ac.jp[4]

1　はじめに

インターネットショッピングサイトにおける商品レビューやTwitterのような短いテキスト(コメント)を投稿するWebサービスの充実により，それらの投稿された文章のカテゴリや意味をナイーブベイズ(NB)やサポートベクタマシン(SVM)等の機械学習手法を用いて自動的に分類させることが文章分類の研究における一つのトレンドとなっている．自動的かつ正確に文章の分類を行う手法が確立できれば，新しいWebサービスやシステム開発の発展に繋がると考えられる．

文章分類の入力データに関する基本的な手法としては，bag-of-wordsによる単語の出現頻度や文法構造を素性とした特徴ベクトルを用いられることが多い．すなわち，用意した教師データのみから直接的に文章の特徴を見出そうとする手法が広く用いられる．しかし，Twitterのようなコメントデータを学習させたい場合，特徴要素となる単語数が少ないことや文法構造がコメントによって大きく変化する等の要因があるため，文章の内容を判別することが困難となりうる．また，分類したいコメントによっては，教師データを用意する作業が煩雑となる場合がある．そこで，本研究では，大規模オンライン百科事典であるWikipediaを用いて，教師データを自動的に生成し，尚且つ，コメントのような特徴要素の少ない文章でも高精度の文書分類が実現できる手法を提案する．

Wikipediaは，語彙が豊富な上，各記事がカテゴリ構造として整理されている等，知識リソースとして優れていることもあり，これまでにWikipediaを用いた文章分類に関する研究は盛んに行われている．例えば，「ナイーブベイズによる文章分類のためのWikipediaカテゴリグラフ解析」[1]では，Wikipediaのカテゴリ構造をグラフとみなし，確率的手法により，文章分類を行っている．本研究では，Wikipediaにおけるカテゴリ構造を用いることで，コメントデータの木構造化を行い，さらに木構造データ各々の木の類似度を計算することで，分類精度の向上を試みる．コメントデータをカテゴリの木構造と捉えることによって，潜在的意味を包含した特徴ベクトルが生成できると考えられる．Wikipediaのカテゴリ構造は最短経路により木構造化を行う，また，コメントデータの所属カテゴリはナイーブベイズを利用して決定する．コメントデータの所属する確率の高い複数のカテゴリから木構造データを生成し，SVMにおける木構造カーネルを用いて，文章分類及び分類精度の測定を行う．

2　木構造データについて

本研究では，上位下位の関係を持ち，データ間の類似度測定が容易な木構造をもつ教師データを生成する手法を提案する．そこでまず，Wikipediaのカテゴリ構造を木構造に変換することで，文章における意味情報の拡張について述べる．

2.1　Wikipediaのカテゴリ構造

Wikipediaでは，図1で示すように各記事を包含するカテゴリからなるネットワーク構造を形成している．各記事は一つ以上のカテゴリに属しており，関連度の高い記事は同一のカテゴリ，若しくはそのカテゴリと距離の近いカテゴリに割り当てられている．また，カテゴリにはそのカテゴリのトピックを持つ記事が複数所属している．言い換えると，カテゴリは関連する記事の意味集合であるといえる．これらの性質から，ある文章に対して所属する確率の高い複数のカテゴリを決定できれば，それらカテゴリ同士をノードで辿るこ

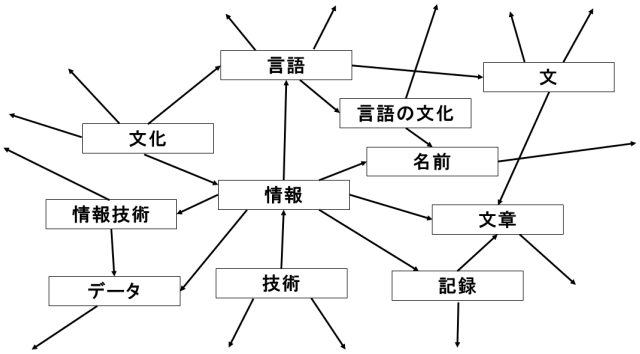


図 1: Wikipedia のカテゴリ構造

とで，文章の意味をカテゴリ構造で表現できると考えられる．しかし，Wikipedia のカテゴリ構造は，複雑な親子関係やループといった性質を持つネットワーク構造のため，単純にカテゴリ同士のノードを辿ると，その間にはまったく関係のないカテゴリが出現することが頻繁に起こりうる．また，WIkipedia のカテゴリはその数が非常に多いことから，ノードの組み合わせで表される文章の構造表現も膨大になり，うまく文章の特徴を捉えることができない懸念がある．

2.2 Wikipedia のカテゴリ構造を利用した木構造データ生成手法

Wikipedia のカテゴリ構造を図 2 のように，あるカテゴリの通るパスを一つに確定し，それを根 (ルート) となるカテゴリと繋げる木構造にすることで，カテゴリ構造の表現がシンプルになる．また，カテゴリの上位下位関係も文章の素性となるため，文章の意味情報が拡張され，分類精度が向上すると考えられる．例えば，ある文章が“大学”というカテゴリに分類された場合，そのカテゴリは“高等教育”という上位概念の親を持つため，同一概念を親とする“大学院”や“短期大学”等のカテゴリに分類された文章とは類似度が高いと判別されるようになる．カテゴリの木構造化に関しては,「Wikipedia を用いた多言語情報アクセスに関する研究:言語間リンクの分析と応用」[2] の最短経路による木構造への変換を参考にした．これは図 3 で示すように，あるカテゴリについて，根までの経路が最短となるような親カテゴリを選択することにより，そのカテゴリのパスを一つに定める手法である．最短経路の手法以外にも確率的にノードを辿る手法等が提案されているが，最短経路は適切なカテゴリパスになりやすい点と計算量の削減という点からこの手法を採用した．本研究では，木構造の根となるカテゴリには Wikipedia の主要カテゴリを利用し，2015 年 3 月のデータベース・ダンプを使用した．

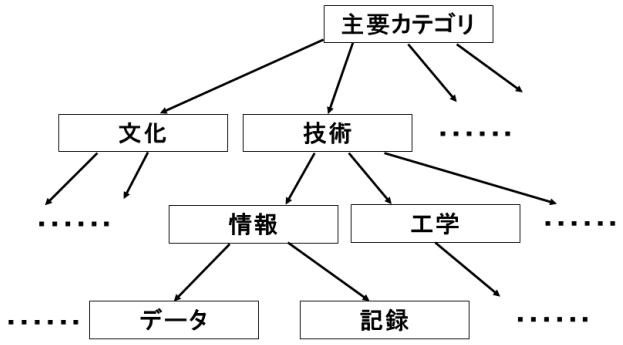


図 2: カテゴリの木構造化例

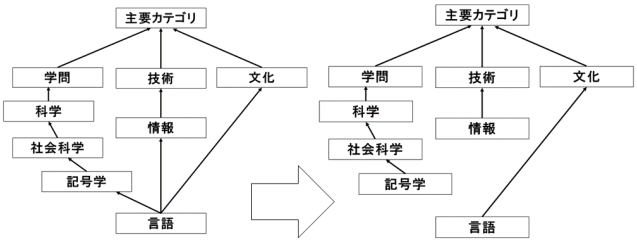


図 3: 最短経路によるパスの決定例

3 木構造データの生成手法

3.1 ナイーブベイズによる文章の所属カテゴリ決定手法

分類対象の文章に対するカテゴリ分類にはナイーブベイズを用いる．ナイーブベイズは高速かつ高精度に分類処理が行えるため，Wikipedia のようにサイズの大きなデータを扱う場合に適している．本研究で用いるナイーブベイズは以下のように定義する．

$$P(cat|doc) = \frac{P(cat)P(doc|cat)}{P(doc)} \propto P(cat)P(doc|cat)$$

$P(cat|doc)$ は事後確率と呼ばれ，入力文章 doc が得られたときの仮定がカテゴリ cat である確率を表す．入力文章は，事後確率がもっとも高いカテゴリへ分類される．

ナイーブベイズで学習させる Wikipedia の記事は MeCab[6] を用いて形態素解析を行い，固有名詞のみを素性とする．また，MeCab の初期辞書では多様な固有名詞を抽出できないため，mecab-ipadic-NEologd[7] を用いて辞書の更新を行った．

本研究では，カテゴリの木構造を構築するにあたり，複数のカテゴリへ分類させる．例えば，葉ノードの数が 3 つになるような木構造を構築する場合，カテゴリ

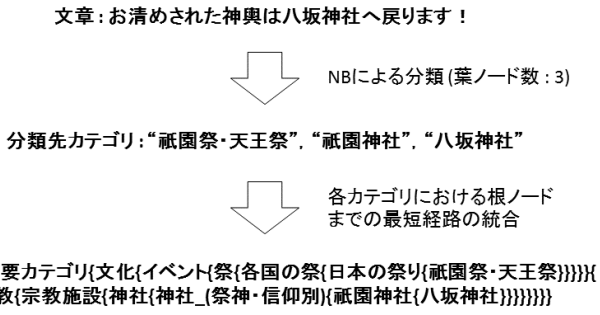


図 4: 木構造データの生成例

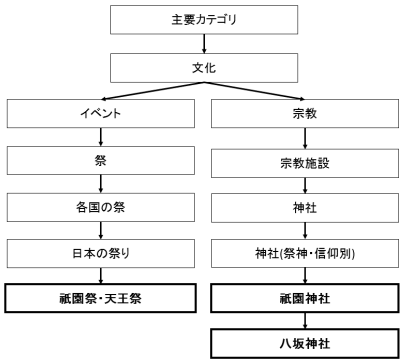


図 5: 木構造データの具体例

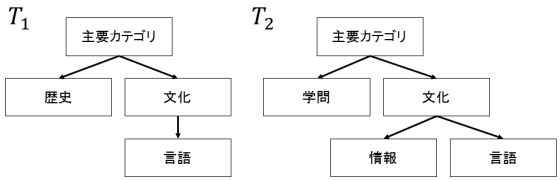
を事後確率の高いものから順に 3 つ選ぶ．文章のカテゴリが決定できれば，図 4 の手順に従いカテゴリ別の最短経路によるノード同士を統合させることで，木構造の構築を行う．生成された木構造の具体例を図 5 のようになる．

3.2 木構造カーネルを用いた SVM による分類手法

Wikipedia のカテゴリによる木構造を決定した文章に対し，木構造カーネル [3] を用いて内積 (類似度) を算出し，SVM に学習させる．木構造カーネル $K(T_1, T_2)$ は以下のように定義する．

$$K(T_1, T_2) = \sum_{p \in S} \gamma \ num(T_{1p}) \cdot num(T_{1p})$$

S は木 T_1，T_2 の部分パスの集合であり，p は部分パスに含まれるノードの数である．また，$num(T_{1p}) \cdot num(T_{2p})$ は，それぞれ木 T_1，T_2 に含まれる部分パス p の個数である．さらに γ は重みパラメータである．ここで，木 T_1，T_2 における共通する部分パス集合の例を図 6 に示す．例えば，{ 主要カテゴリ { 歴史 }{ 文化 { 言語 }}} と { 主要カテゴリ { 学問 }{ 文化 { 情報 }{ 言語 }}} という 2 つの木の場合，それぞれの木における共通部分パスの個数は 6 となる．以上の性



T_1とT_2に共通の部分パス集合

{主要カテゴリ}, {文化}, {言語},
{主要カテゴリ{文化}}, {文化{言語},
{主要カテゴリ{文化{言語}}}

図 6: 共通する部分パス集合の例

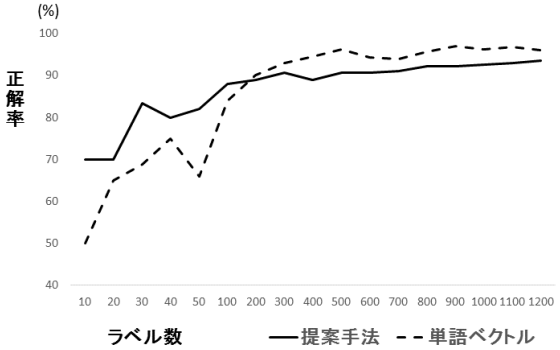


図 7: グラフによる精度比較

質から，この木構造カーネルは，特徴ベクトルとして木 T の全ての可能な部分パスの列挙を考え，それらを部分パスの長さに基づく重みで内積をとったものと考えることができる．

4 評価

文章分類は文章の自動分類それだけで完結することは少なく，そのほとんどが何かのシステムの応用先として期待される．本研究では，提案手法を観光情報に関する有益な情報を提供するシステム [4] への応用先として考慮し，Twitter を対象として，ある Tweet を「観光情報」と「それ以外」というカテゴリへ自動分類し，「観光情報」のみの Tweet を取得するという情報推薦のタスクを考える．

精度評価では，人手で正解ラベルを付けた 1200 件の Tweet に対し，10 分割交差検証を用いて行う．評価で用いる木構造データの葉ノード数は 3 に設定した．ベースラインの SVM の実装としては，UCI が開発した LIBSVM(v3.20)[8] を用いた．SVM モデルは C-SVM を用いた．比較として，TF-IDF による重み付けをした単語ベクトルの結果も提示する．

図 7 及び表 1 の結果より，単語ベクトルの精度が平均的に木構造データの数値を上回っているが，学習量

表 1: Tweet の分類正解率

ラベル数	提案手法	単語ベクトル
10	0.700	0.500
20	0.700	0.650
30	0.833	0.688
40	0.800	0.750
50	0.820	0.660
100	0.880	0.840
200	0.890	0.895
300	0.907	0.933
400	0.890	0.945
500	0.906	0.962
600	0.907	0.943
700	0.911	0.939
800	0.921	0.956
900	0.922	0.970
1000	0.925	0.962
1100	0.929	0.967
1200	0.935	0.960

が少ないときには木構造データの精度が良い数値を示していることがわかる．これは，文章がカテゴリの木構造で表されることで，単純に木構造の類似度で特徴空間に特徴ベクトルを射影することができるようになるため，少ない学習量でも効率よくカテゴライズできるからだと考えられる．一方で，単語ベクトルの場合は，未学習の単語に弱く，十分な学習量を与えないと精度が安定しないことがわかる．木構造データの平均的精度が単語ベクトルにわずかに及ばない要因としては，木構造データにおける上位概念の頻出カテゴリやナイーブベイズによって誤分類されたカテゴリ等が教師データ上のゴミとなり，精度の低下を誘発しているものを考えられる．

観光情報に分類された Tweet の具体例としては,「仁和寺の御室桜綺麗でしたよ＾＾」や「平野神社の桜は珍しい種類が多いですね。」等が挙げられる．上記のように地域や施設の紹介を含んだ Tweet が 9 割程度の精度で取得された．さらに提案手法の場合は，わずかなテキストからでも，自動的に高精度の分類を実現する教師データを生成できるため，たくさんの地域や施設，若しくは単語による教師データの用意が困難な情報の特徴を取得したい場合に有効である．このことから，さらに精度の改良を重ねていくことで，より良いシステムが提供できるようになると期待できる．

5 おわりに

本研究では，Wikipedia のカテゴリ構造を用いて，分類対象の文章を木構造に変換し，文章の意味情報を拡張することにより，Tweet のような短いテキストでも単語ベクトルにあまり劣らぬ精度で分類が行えることを示した．今後は木構造の変形や部分パスの取り方によって，分類精度の向上があるか研究していきたい．

参考文献

[1] 白川真澄, 中山浩太郎, 原隆浩, 西尾章治郎：ナイーブベイズによる文書分類のための Wikipedia カテゴリグラフ解析, 人工知能学会全国大会論文集 26, pp.1-4, 2012.

[2] 新井嘉章，福原知宏，増田英孝，中川裕志：Wikipedia を用いた多言語情報アクセスに関する研究：言語間リンクの分析と応用, 第 20 回セマンティックウェブとオントロジー研究会,SIG-SWO-A803-15, 2009.

[3] 木村大翼，久保山哲二，渋谷哲朗，鹿島久嗣:部分パスに基づいた木カーネル, 人工知能学会論文誌 26(3), pp.473-482, 2011.

[4] Masahiro Takeda, Hiromitsu Shiina, Fumio Kitagawa, Nobuyuki Kobayashi “Regional Information Video Searches Using Word Searches Generated by Twitter Posts,” Proceedings of IIAI 2015, pp.127-131, 2015.

[5] John Shawe-Taylor, Nello Cristianini, 大北剛, “カーネル法によるパターン解析,” pp.419-490, 2010.

[6] Taku Kudo, “MeCab: Yet Another Part-of-Speech and Morphological Analyzer” http://mecab.googlecode.com/svn/trunk /mecab/doc/index.html, 2015/08/20 アクセス.

[7] Toshinori Sato, https://github.com/neologd /mecab-ipadic-neologd, 2015/12/01 アクセス.

[8] Chih-Chung Chang, Chih-Jen Lin, “LIBSVM – A Library for Support Vector Machines,” http://www.csie.ntu.edu.tw/c̃jlin/libsvm/, 2015/05/01 アクセス.