Vol.2016-NL-227 No.15
2016/7/30

重み付き木構造カーネルと共起重みによるTwitterの自動分類手法

武田 昌大[1,a)]　椎名 広光[2,b)]　小林 伸行[3,c)]

概要：観光施設や商品のレビューやTwitterのような短いコメントを投稿するWebサービスの盛り上がりから，コーパスの作成等の手間を抑えつつ，自動的かつ正確に文章を識別する手法を求める要求が高まっている．また，大学の講義で利用される講義アンケートについても，自由回答記述は、ほとんど短文であり，短文の分類が講義評価に利用できると考えられる．これまでの多くは，文章分類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を利用している．それに対して本研究では，Tweetなどのショートコメントをカテゴリ分類を行う木構造データに変換し，更に木構造カーネルの上で木のノードの部分に重みを置く重み付き木構造カーネルによる分類を行った．また，Tweetの修飾語の包含関係から重みを計算し重み付き木構造カーネルに加えることで，分類精度の向上を試みた．

キーワード：サポートベクターマシン，木構造カーネル，Wikipedia，Twitter，ナイーブベイズ

Automatic classification of Twitter by and weighted tree kernel and co-occurrence weight

Masahiro Takeda[1,a)]　Hiromitsu Shiina[2,b)]　Nobuyuki Kobayashi[3,c)]

Abstract: Many web services posting short comments such as product reviews and Twitter have been provided. In addition, for the lecture questionnaire at universities, comment is mostly short in free answer column. The short classification is considered to be available for lectures evaluation.We consider that automatic and accurate text classification may lead to develop new web services and system. In the past, the frequency of appearance of words by bag-of-words have been often used for text classification as a basic technique. In contrast, we propose a technique to classify tweets using tree kernels created by the categories of Wikipedia in this study. In addition, we extened classification by weighted tree kernel. As evaluation experiment, we developed a retrieval system for tourism videos by applying the technique to tweets related to tourism.

Keywords: Support Vector Machine, Tree Kernel, Wikipedia, Twitter, Naive Bayes

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コメントやTwitterのような短いテキストを投稿する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やサポートベクターマシン [1] 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が行われている．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

1 岡山理科大学大学院総合情報研究科
Graduate school of Informatics, Okayama University of Science, Ridaicho 1-1 Kitaku Okayama, Tokyo 700–0005, Japan
2 岡山理科大学総合情報学部
3 山陽学園大学総合人間部
a) i15im02tm@ous.jp
b) shiina@mis.ous.ac.jp
c) koba_nob@sguc.ac.jp

テム開発の発展に繋がると考えられる．

文章分類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，本研究は，Wikipedia を用いて，明示的には教師データを作成しないで，か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．また，本研究では，観光動画検索するシステムへの適用として，Tweet の分類に「観光情報」と「それ以外」というカテゴリに自動分類し，「観光情報」のみの Tweet を取得し，二次的な関連検索語を抽出に利用した．

本研究で利用している 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多く行われている．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」[2] では，Wikipedia のカテゴリ構造をグラフとみなし，統計的手法により，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，コメントデータを木構造データに変換し，更に木構造データ間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木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る．複数の Wikipedia のカテゴリ構造の最短経路を合成によって木への変換を行う．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 の木構造カーネル [3] を用いて，文章分類及び分類精度の測定を行う．

更に木構造カーネルの上で木の葉の部分に重みを置く重み付き木構造カーネルによる分類を行った．また，Tweet の修飾語の包含関係から重みを計算し重み付き木構造カーネルに加えることで，分類精度の向上を試みた．

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．

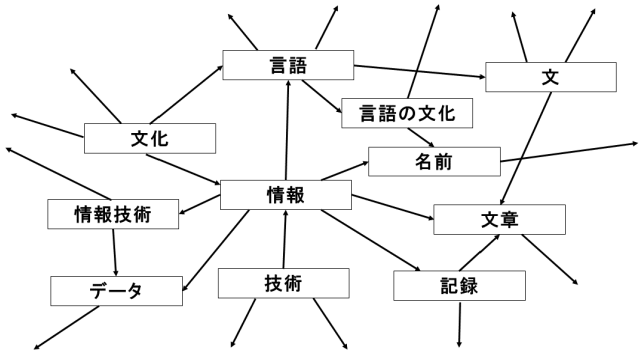

図 1 Wikipedia のカテゴリ構造
Fig. 1 Category structure of Wikipedia

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」[4] 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

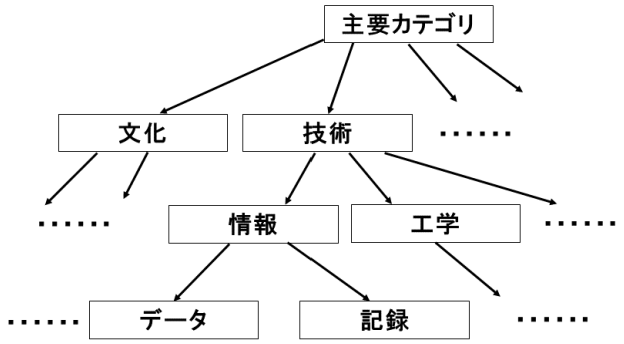

図 2 カテゴリの木構造化例
Fig. 2 Tree structure converts a category

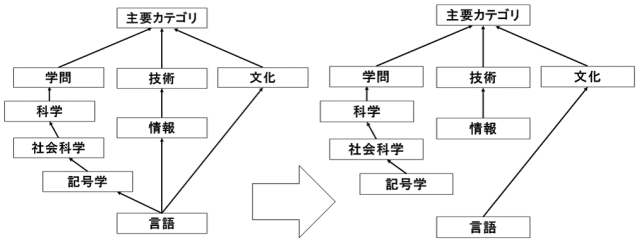

図 3 最短経路によるパスの決定例
Fig. 3 Make of the shortest pass

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．

3. 木構造データの生成手法

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．

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)} \propto P(cat)P(doc|cat) \qquad (1)$$

$P(c|d)$ は事後確率と呼ばれ，入力文章 d が得られたときの仮定がカテゴリ c 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[5] を用いて形態素解析を行い，固有名詞のみを素性とする．

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

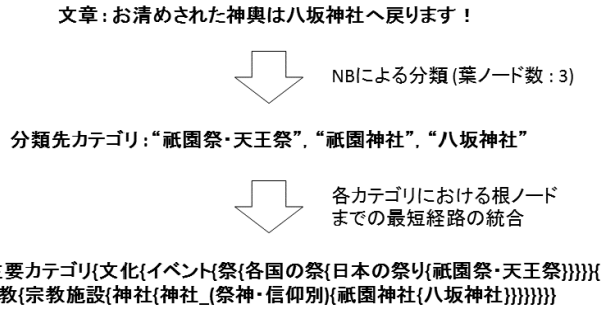

図 4 木構造データの生成例
Fig. 4 Output of tree structure data

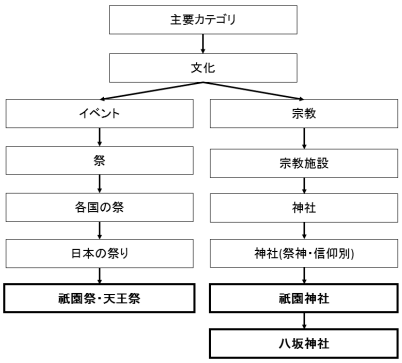

図 5 木構造データの例
Fig. 5 Example of tree structure data

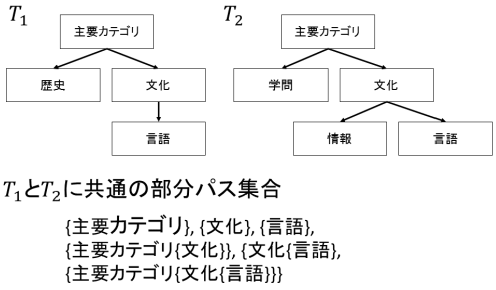

図 6 共通する部分パス集合の例
Fig. 6 Set of the subtree

になるような木構造を構築する場合，カテゴリ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．

4. 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 [6] 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．

$$K(T_1, T_2) = \sum_{p \in S} \gamma \cdot num(T_{1p}) \cdot num(T_{1p}) \quad (2)$$

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．

5. 重み付き木とカーネル

5.1 重み付き木

通常の木間の類似度を測るのでは，その部分的な一致性が重要となる．しかし，部分木の一致でも，木の葉に近い部分が一致するほうが重要度が高いと考えられる．例えば，歴史や文化等の抽象的であるような上位カテゴリに属するノードと比較して，祇園祭や八坂神社等の具体的な意味合いを持つようなカテゴリに属するノードがより特徴的な要素であると考えられる．そこで，カテゴリ木の各ノードにその重要度を反映した重み付けを行うことで，類似度測定による精度の向上を図る．

重み付き木については，三上ら [7] による文分類の方法が提案されており，木編集距離に TF–IDF をによる重み付け処理を行う手法を提案している．

本研究では，Wikipedia のカテゴリ構造におけるカテゴリ間ノードの出現頻度による TF–IDF を以下の式で定義する．

$$TFIDF(v) = tf(v) \times idf(v) \quad (3)$$

$$tf(v) = \frac{n_v}{\sum_{\forall k \in d \forall d \in D} n_k} \quad (4)$$

$$idf(v) = log\left\{\frac{|D|}{|\{D : v \in d\}|}\right\} \quad (5)$$

ここで，n_v は Tweet の文章集合から生成されたカテゴリ木におけるノード v の出現する頻度，D は Tweet の文章集合，d は Tweet の文章集合 D に含まれる単体の Tweet の文章である．$tf(v)$ は，各カテゴリノード n_v の出現数を Tweet の文章集合に出現するカテゴリノードの個数の総和で割ったものであり，出現頻度の高いノードほど TF 値は大きくなる．一方で，$idf(v)$ は，多くの Tweet の文章に出現するノードほど IDF 値下げ，出現頻度の低い特徴的なノードほど IDF 値を上げるノードフィルタとしての役割を持つ．

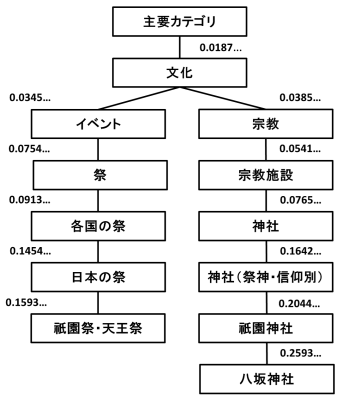

図 7 重み付き木の例
Fig. 7 Example of weighted tree

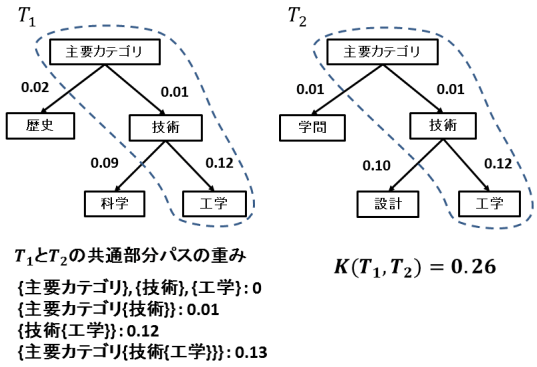

図 8 重み付き木の類似度
Fig. 8 Similarity of weighted trees

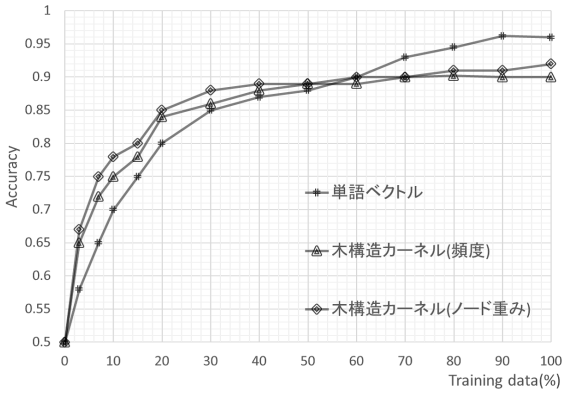

図 9 重み付き木の類似度の比較
Fig. 9 Conpare of similarity of weighting trees

図 7 に重み付け処理を行ったカテゴリ木の例を示す．図より，抽象的な意味のカテゴリのノードは TF–IDF 値が小さく，より具体的な意味を持つカテゴリのノードの TF–IDF 値は大きくなっていることがわかる．例えば，「主要カテゴリ」と「文化」の間におけるノードのスコアは 0.0187 に対して，「日本の祭」と「祇園祭・天王祭」の間におけるスコアは 0.1593 であり，各ノードの重要度を反映した結果となっていると考えられる．

5.2 重み付き木構造カーネルによる類似度の計算の例

重み付き木構造カーネルによる類似度の計算過程を図 8 を用いて説明する．本研究では，T_1 及び T_2 の共通部分パスに対し，それぞれのパスに付随するノードの重みの総和を木間の類似度として計算を行う．ここで，{ 主要カテゴリ } や { 技術 } 等のカテゴリ間のノードを持たない部分パスの重みは 0 とし，{ 主要カテゴリ { 技術 }} のようにカテゴリ間のノードを持つ部分パスは，内包するノードの重みの和を取ったものとする．さらに，共通部分パスの重みの総和を木間の類似度として算出を行う．

5.3 重み付き木の評価

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 [8] 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．

評価手法としては，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．また、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)[9] 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 9)．

6. 共起重みによる手法

6.1 修飾語の有無に関しての重みづけ

前章の手法では，少し異なる Tweet でも同じ類似度として評価されてしまうことがある．例えば，修飾語を含まれていても重要単語から木を生成するために，重要単語とみなされない部分については，欠落してしまう．そこで，修飾語をについては，Tweet から生成される木とは別に修飾語だけを評価し，その評価を木間の類似度に掛け合わせることを考える．修飾語評価に関しては，2 つの木のもととなる Tweet で含まれている修飾語 (形容詞，副詞) 同士の包含関係とその重みから修飾語による評価とする．

例えば，図 10 の 2 つの木に対して，形容詞，副詞の共起 [10],[11] している語 a, b, c と a, c, d とする．この場合，$\frac{\{a,b,c\} cap \{a,c,d\}}{\{a,b,c\} \cup \{a,c,d\}} = \frac{2}{4}$ を共起係数として掛け合わせることにする．

6.2 共起重みの評価実験

前章の重み付き木の評価実験と同じタスクで，共起重みの有無による評価実験を行った．図 11 に共起重みの精度評価グラフを示す．

また，Tweet から生成される木の構造が類似している場合での共起重みの評価実験を結果を図 12 に示す．

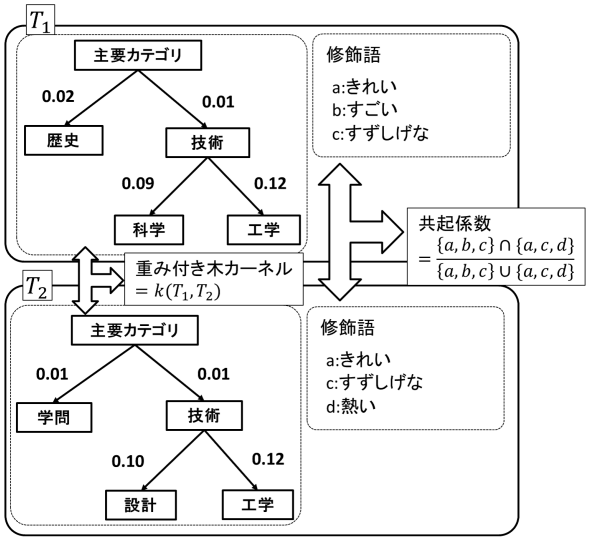

図 10 修飾語の包含関係を用いた類似度

Fig. 10 Similarity of tweets using inclusion property of modifier

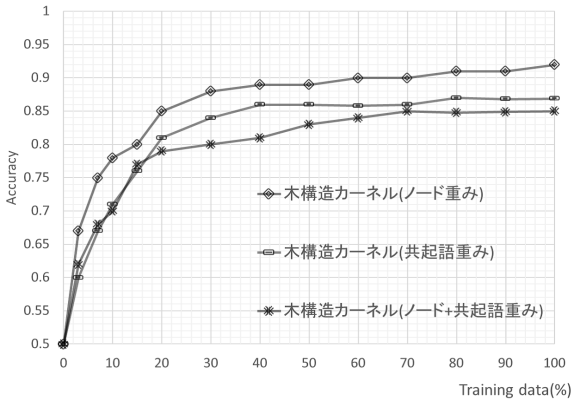

図 11 共起重みの精度比較

Fig. 11 Compare of accuracy with cooccurrence weight

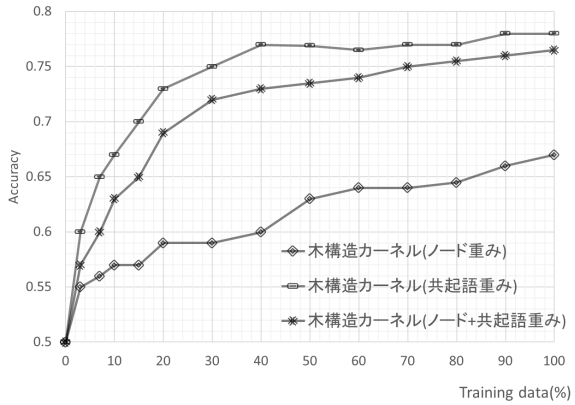

図 12 類似木での共起重みの精度比較

Fig. 12 Compare of accuracy with cooccurrence weight in case of similar tree

7. 評価実験からの考察

まず，図 9 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ノイズとなり，識別精度の低下を誘発しているものを考えられる．

木構造カーネルにおける共通部分パスの頻度による計算とカテゴリ木のノードに重み付けを行った計算での識別精度の比較では，ノードに重み付けを行ったほうの識別精度が向上していることがわかる．これは，カテゴリ木に対してノードの重み付けをすることで，Tweet に対するカテゴリ木の特徴がより反映され，識別精度が向上したものと考えられる．また，学習量が少ないときから精度が伸びているため，小規模の学習データに向いているモデルであると考えられる．

次に，図 11 の結果より，共起重みを用いた手法では全体的な識別精度は下がる結果となった．精度の下がる要因としては，共起重みを用いたことで，カテゴリ木全体において適切な類似度が算出することができなかったからだと考えられる．その一方で，図 12 の類似木データを用いた結果では，共起重みを用いた手法の識別精度が高いことがわかる．これから，木構造が類似しているデータにおいては，共起重みを用いることで，うまく木構造の差別化ができ，精度の向上に繋げることができると考えられる．

観光情報に分類された Tweet の具体例としては，「京都の本能寺、実は現在の場所じゃなかったって知ってました？」や「昨日は大阪の友達と久々に五行のラーメン食べてきました。ここの焦がしラーメンは最高♪」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，多くの地域や施設，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．

8. 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．また，カテゴリ木にノードの重要度に応じた重み付けを行うことで，識別精度の向上を行った．さらに，木構造が類似しているデータにおいては，共起重みを用いることが有効であることを示した．今後は，カテゴリ木のノードや共起語による重み付けの手法を改善していくことで，識別精度の向上を図っていきたい．

参考文献

[1] 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).
[2] 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).
[3] Alessandro Moschitti, “Making Tree Kernels practical for Natural Language Learning”, 11th Conference of the European Chapter of the Association for Computational Linguistics, pages 113–120(2006).
[4] 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).
[5] 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.
[6] 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).
[7] 三上崇志, 平野敬, 川又武典, “木の編集距離を用いた文の類似度計算方式”, 情報処理学会研究会報告, Vol. 2010-NL-196,No.3, 1-6(2010).
[8] 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).
[9] Chih-Chung Chang, Chih-Jen Lin, “LIBSVM – A Library for Support Vector Machines,” http://www.csie.ntu.edu.tw/c̆jlin/libsvm/, 2015/05/01 アクセス.
[10] 湯浅夏樹，上田徹，外川文雄，“大量文書データ中の単語間共起を利用した文書分類”，情報処理学会論文誌，Vol.33, No.9(1995).
[11] 藤井洋一，鈴木克志，今村誠，高山泰博，“共起情報を利用した文書の自動分類”，情報処理学会研究報告，自然言語処理研究会報告，Vol.97，No.29(1997).