分野の階層構造を利用したコーパスの誤り修正

濱野 秀俊 † 福本 文代 ‡

山梨大学工学部

g04mk020@ccn.yamanashi.ac.jp†, fukumoto@yamanashi.ac.jp‡

1 はじめに

近年，様々なタグが付与された大量のコーパスが利用可能になったことを背景に，教師付き学習を用いた研究が多く行われている．一般に教師付き学習の精度はタグ付きコーパスの質に依存する．しかし，タグ付きコーパスは人手により作成されるため，コーパスの量が膨大になると誤りの混在は避けられない．従って精度良く学習するためには，分類対象となるコーパス中のタグの誤りを自動的に検出し，修正する技術が必要となる．

タグの誤り検出や修正に関する研究では，機械学習の結果得られる事例の重みを利用する方法と，誤りである可能性を確率値として求める方法がある．重みを利用する手法として，Abney らはコーパス中の品詞タグの誤りと構文情報タグとして前置詞句の係り先に関する誤りに注目し，ブースティングを用いて学習した結果得られる事例の重みを利用することで誤りが検出できることを示している [1]．中川らは，品詞タグの誤り検出についてコーパスの一貫性に注目し，SVMs を用いることで一貫性を乱す例外的な記事を抽出することにより誤り検出を行う手法を提案している [2]．Penn Treebank WSJ を用いた実験では，適合率 90% 以上という高い精度が得られている．確率値を用いる手法には村田らによる研究がある．村田らは品詞タグと構文情報タグとしての単語の係り先の誤りに注目し，タグの誤っている確率を決定リストや用例ベース手法を用いて求め，誤り検出と修正を行う手法を提案した [3]．京大コーパスを用いた実験では，決定リストを用いた場合の方が用例ベース手法より優れ，構文情報タグでは適合率が 20% であるものの，品詞タグでは最高 78% の適合率で誤りが修正できることが報告されている．しかし，これらの研究で扱った品詞タグや係り先タグは複数の解を持つことがないため，文書分類のように文書に複数の分野名が付与されている multi-label の問題に対しても有効であるかは検証の余地がある．

本稿では人手により複数の分野名が付与された新聞記事を対象とした分野名の誤り修正手法を提案する．機械学習として kNN(k-Nearest Neighbors) [4] を用いる．我々は誤り修正の手がかりとして 2 点に注目した．1 点目は，誤りか否かを判定する記事から見た類似度の上位の記事が属する分野情報である．2 点目は，誤りか否かを判定する記事が他の記事から見て類似度の上位になる場合のその記事の分野情報である．また分野名タグの階層構造を利用し，上位分野の誤り修正結果を下位分野に適用することで高精度な修正を行う．

2 誤り修正手法

誤り修正は機械学習 kNN を用いて行う．先ず kNN について述べた後，kNN を用いた誤り修正手法と階層構造の利用について説明する．

2.1 kNN

kNN は類似度に基づく分類アルゴリズムである．分類を行うテスト記事を $\mathbf{x} = (x_1, \cdots, x_t)$ (x_i は記事の出現単語) とし，分野名タグの付与された訓練記事の集合を Y とする．$\mathbf{x}$ が既知のどの分野に分類されるかは，$\mathbf{y}_i = (y_{i1}, \cdots, y_{it}) \in Y$ との類似度に基づき判断される．本手法では類似度尺度として式 (1) で示される余弦尺度を用いた．

$$sim(\mathbf{x}, \mathbf{y}_i) = \frac{\sum_{j=1}^{t} x_j \cdot y_{ij}}{\sqrt{\sum_{j=1}^{t} x_j^2 \times \sum_{j=1}^{t} y_{ij}^2}} \quad (1)$$

kNN では訓練記事集合 Y の中から $\mathbf{x}$ との類似度が高い上位 k 個の訓練記事を用い，$\mathbf{x}$ がどの分野に属するかの判定を行う．$\mathbf{x}$ が分野 c_n に属するか否かは以下の式 (2) で求める．

$$d(\mathbf{x}, c_n) = \sum_{\mathbf{y}_i \in kNN} sim(\mathbf{x}, \mathbf{y}_i) d(\mathbf{y}_i, c_n) - b_n \quad (2)$$

式 (2) において $d(\mathbf{y}_i, c_n) \in \{0, 1\}$ は訓練記事 $\mathbf{y}_i$ が分野 c_n に属するか (1) 否か (0) を示す．b_n は各分野に設定された閾値を示す．本手法では式 (2) により求められた $d(\mathbf{x}, c_n)$ の値が 0 より大きい分野全てをテスト記事 $\mathbf{x}$ の分野とする．このようにして分野ごとに式 (2) で求め

た値が閾値を越えたかを求め，複数で越えた場合に $\mathbf{x}$ は複数の分野が付与される．

2.2 kNN を用いた誤り修正

我々は修正の手がかりとして 2 点に注目し分野タグの誤り修正を行なった．1 点目は誤りか否かを判定する記事から見た類似度の上位の記事が属する分野情報である．これは誤りか否かを判定したい記事をテスト記事として kNN により分類することで求める．kNN により付与された分野 (分野の集合) を誤り修正の候補とする．2 点目は誤りか否かを判定する記事が他の記事から見て類似度の上位に属する場合におけるその記事の分野情報である．これは誤りか否か判定する記事を kNN の訓練記事とし，他の記事をテスト記事として kNN で分類を行う．その際に類似度の上位 k 個として選択された場合，テスト記事との類似度とテスト記事に予め付与された分野の情報から各分野の重みを求める．この重みを用いて修正候補に属するか否かを判断する．

誤り修正の処理の流れを以下に示す．誤り修正を行う記事の集合を S とし，S の要素を $s_i = (\mathbf{t}_i, \mathbf{c}_i, \mathbf{kc}_i, \mathbf{lc}_i, \mathbf{wc}_i)$ とする．$\mathbf{t}_i$ は記事を示すベクトルであり，$\mathbf{c}_i = (c_{i1}, \cdots, c_{im})$，$\mathbf{kc}_i = (kc_{i1}, \cdots, kc_{im})$，及び $\mathbf{lc}_i = (lc_{i1}, \cdots, lc_{im})$ は記事が各分野に属するか (1) 否か (0) を m 次元 (m は分野の総異なり数) のベクトルで表し，$\mathbf{c}_i$ は予め付与された分野，$\mathbf{kc}_i$ は kNN のテスト記事としたときに付与される分野，そして $\mathbf{lc}_i$ は最終的な誤り修正先を示す．$\mathbf{wc}_i = (wc_{i1}, \cdots, wc_{im})$ は m 次元のベクトルとし，ベクトルの各次元の値は各分野の重みの値とする．

1. S の中から 1 つをテスト記事 s_i とし, 残り $S - s_i$ を訓練記事集合として kNN を用いて s_i に分野 $\mathbf{kc}_i$ を付与する．同時に，訓練記事 s_j が s_i との上位 k 個の要素に含まれる ($s_j \in kNN$) 場合，式 (3) を用いて s_j の各分野の重みの値を計算する．

$$wc_{jk} = wc_{jk} + sim(\mathbf{t}_i, \mathbf{t}_j)c_{ik} \ (1 \leq k \leq m) \quad (3)$$

2. 手順 1 を S の中の全ての記事に対して適用する．
3. 以下の条件により s_i の誤り修正先 $\mathbf{lc}_i$ を求める．k は各分野 ($1 \leq k \leq m$) を示す．$threshold_k$ は各分野の重みの閾値であり，実験的に求めた．
   (a) **if**$(c_{ik} == kc_{ik})$ **then** $lc_{ik} = c_{ik}$
   (b) **else if**$((c_{ik}\ or\ kc_{ik}) == 1\ and\ (wc_{ik} > threshold_k))$
   then $lc_{ik} = 1$
   (c) **else** $lc_{ik} = 0$

各分野について予め付与された分野と kNN の付与した分野の一致を求め，一致しない分野は分野の重みが閾値を越えているならその分野に属するとする．

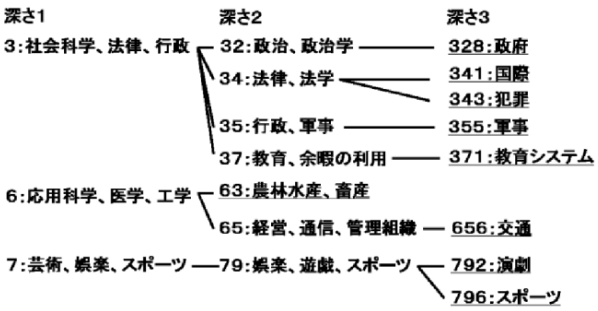

図 1　使用分野とその階層構造

2.3 階層構造の利用

本手法では multi-label に対応した誤り修正を高精度で行うために，分野名の階層構造を利用した．分野名タグが階層構造を持つ場合，上位ほど分類が容易であり下位になるほど分類が難しくなる傾向がある．そこで Boolean function [5] を用い，上位階層における誤り修正の結果を反映して修正候補となる分野を絞りこみ，その分野に属する記事のみを使用して誤り修正を行った．

3 実験

3.1 実験データ

実験では RWCP コーパス [6] を使用した．これは 1994 年度の毎日新聞 30,207 記事からなるコーパスであり，複数の国際十進分類法のタグが付与されている．実験では，この国際十進分類法のタグの中から図 1 に示される最大で 3 の深さを持つ 18 分野を対象とした．また 30,207 記事の中から 10,766 記事を用いた．これらの記事は図 1 の分野の中で最下位に位置する分野となる下線で示される 9 つの分野とその上位の分野に属する記事である．1 つの記事には平均 3.07 個のタグが付与されている．各分野に属する記事数を表 1 に示す．

記事の素性として本文中の出現単語のうち一般名詞，固有名詞，サ変名詞の中から χ^2 素性選択を行い有効な

表 1　各分野の記事数

分野名	記事数	分野名	記事数
社会科学	5,750	政府	1,517
応用科学	1,228	国際	1,609
娯楽	3,788	犯罪	2,212
政治	1,517	軍事	569
法律	3,789	教育システム	436
行政	569	農林水産	447
教育	436	交通	786
経営	786	演劇	459
遊戯	3,788	スポーツ	3,331
		計	33,017

表 2　分野名の誤りの修正精度

	階層利用	階層なし
修正記事数	816	678
付与タグ数	3584	2719
適合率 (%)	88.8	89.9
再現率 (%)	88.7	83.6
F 値 (%)	**88.8**	86.6

単語を選択した．階層構造を用いない場合は全記事から χ^2 の値を求め，図 1 において下線の付いた各分野から上位 500 語を抽出し素性とした．階層構造を用いる場合は，誤り修正を行うグループ別に χ^2 の値を求めそれぞれに適した語数を選択し素性とした．記事中の単語は奈良先端大学で公開されている形態素解析システム茶筌 [7] を用いて抽出した．

3.2　分野名誤りの修正実験

multi-label を対象とした誤り修正手法の評価尺度として, 誤り修正の行われた各記事に対する適合率 (Prec), 再現率 (Rec)，及び F 値を定義した．

$$Prec = \frac{\text{正しく修正された分野数}}{\text{修正後の分野数}}$$

$$Rec = \frac{\text{正しく修正された分野数}}{\text{記事に付与されるべき分野数}}$$

$$F\text{ 値} = \frac{2 \cdot Prec \cdot Rec}{Prec + Rec}$$

10,766 記事に対して誤り修正を行い，適合率，再現率，及び F 値を用いて修正された記事に対する評価を求めた．また階層構造を用いずに修正を行った結果と比較した．表 2 に結果を示す．また修正例を表 3 に示す．

表 2 より本手法の F 値は 88.8% であった．誤り修正に失敗した場合として上位階層における誤り修正が失敗した場合がある．その原因として 2 点考えられる．1 点目はデータ中に含まれる他の誤りの影響を受けた場合である．特に，ほぼ同一の内容を持つ複数の記事に対して全て同じように誤った分野を付与されている場合があり，この場合にはそれらの誤りを検出できないだけでなく，他の記事を同様の誤りに修正してしまうことがあった．表 3 の修正失敗例がこの例であり，「米軍のヘリ誤射で撃墜」に関する複数の記事に誤った分野 (応用科学, 経営，交通) が予め付与されていたため正しい修正が行われなかった．2 点目は記事の特徴を上手く抽出できていない記事があることが挙げられる．本手法では単語を用いて記事を表現しているため単語が重複した記事間は高い類似度を持つ．他の記事との類似度が低い記事において，読者投稿記事の投稿要項の様なテンプレート中の単語や，国名や地名などの単語が重複する記事により高い類似度を持つ記事が構成される場合が存在した．このような場合では正しく修正することができなかった．

階層構造を利用した場合と利用しない場合の修正結果を比較すると，階層構造を用いることで修正記事数がおよそ 140 記事ほど増え，精度では F 値で 2.2% 程度の向上が得られた．特に再現率で良い結果が得られている．これは，上位階層の誤り修正結果を信頼することで，関係の無い分野の記事の影響を受けずに誤り修正を行うことができるためと考えられる．一方，適合率のみを見ると階層構造を用いない場合のほうが若干精度が良い．これは階層を用いた場合には，政治，法律及び行政など

表 3　誤り修正例

該当記事のタイトル	修正前	修正後
修正成功例		
賭けゴルフ容疑で三人を逮捕	娯楽 遊戯 スポーツ	社会科学 法律 犯罪
北朝鮮に柔軟な対応を要請 日韓首脳会談	社会科学 政治 政府	社会科学 政治 法律 政府 国際
修正失敗例		
「ヘリ誤射」真相究明へ	社会科学 法律 行政 国際 軍事	応用科学 経営 交通

表 4　人工的な誤りの修正精度

総誤り数	修正記事数	検出率	F 値
538(5%)	386	71.7	94.9
1076(10%)	775	72.0	93.8
2152(20%)	1468	68.2	89.3
3228(30%)	2073	64.2	84.3

重要単語の重複が多い下位分野間で分野が区別できず，kNN により正しく分類できない場合があり誤り修正に失敗する記事が存在したのに対し，階層構造を用いない場合には修正の容易な上位階層の分野のみが誤り修正で付与され，修正の難しい下位階層の分野には属さないとする修正結果があるためと考えられる．

今後はこれらの問題に対処するため，素性の種類を見直す必要があると考えられる．本手法では，本文中の一部の名詞を素性とし χ^2 素性選択を用いた．今後はタイトルの情報を加えたり，動詞や未知語を素性に加えることで素性選択を行う必要がある．

3.3　人工的な誤りデータの修正実験

一般にコーパス中の誤りの個数は既知ではないため，どの程度誤りが修正できているのかは不明である．そこで人工的に誤りを付与した記事集合に対して誤り修正を行うことで，コーパスに含まれる誤りをどれだけ修正できているかを求めた．この実験では予め付与された分野が正しいと仮定し，その分野を他の分野へ無作為に置き換えることで人工的な誤りを作成した．実験には誤り修正の実験と同様のデータを用い，階層構造を利用して誤り修正を行った．またコーパス中に含まれる誤りの量の影響を考慮し，誤りの量が 5%, 10%, 20%, 及び 30% の場合について結果を求めた．結果を表 4 に示す．

表 4 における検出率とは作成した誤りのうち修正が行われた記事の割合である．表 4 より，誤りが 10% 以下のときには 70% 程度の誤りに対して修正ができ，その際の誤りデータに対する F 値は 90% を越えていることがわかる．また含まれる誤りが少ないほど多くの誤りが修正でき，その F 値も高い．これは誤りの数が少ないため，誤りの記事を判別する際に他の誤りの影響を受けにくいためであると考えられる．

4　まとめ

本手法では，multi-label に対応した分野名タグの誤り修正手法を提案した．国際十進分類法のタグに対して実験を行った結果，88.8% の精度が得られた．また階層構造を利用しない場合と比較した結果，F 値で 2.2% の向上が見られたことから階層構造の有効性が確認できた．今後の課題としてはさらに修正の精度を向上させるため，素性選択で用いた素性の種類を見直す必要がある．また，大規模コーパスを用いて定量的な評価を行うと同時に適用する際の高速化の問題にも取り組む必要がある．

参考文献

[1] Steven Abney, Robert E. Schapire and Yoram Singer: Boosting Applied to Tagging and PP Attachment, *Proc. of the joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pp. 38-45, 1999.

[2] Tetsuji Nakagawa and Yuji Matumoto: Detecting Errors in Corpora Using Support Vector Machines, *Proceedings of the 19th International Conference on Computational Liguistics*, pp. 709-715, 2002.

[3] 村田 真樹, 内山 将夫, 内元 清貴, 馬 青, 井佐原 均, “決定リスト, 用例ベース手法を用いたコーパス誤り検出・誤り訂正”, 情報処理学会研究報告 2000-NL, 136, pp. 49-56, 2000.

[4] Yiming Yang and Xin Liu: A re-examination of text categorization methods, *Proceeding of the 22nd International Conference on Research and Development in Infomation Retrieval (SIGIR'99)*, pp. 42-49, 1999.

[5] Susan Dumais, Hao Chen: Hierarchical Classification of Web Content , *Proeedings of the 23rd Annual International ACM SIGIR Confreence on Research and Development in information Retrieval (SIGIR2000)*, pp. 256-263, 2000.

[6] RWC: *RWC Text Database (Japanese)*, Real World Computing 1995.

[7] 松本 裕治, 北内 啓, 山下 達雄, 平野 善隆, 松田 寛, 高岡 一馬, 浅原 正幸, “形態素解析システム『茶筌』version2.3.3 使用説明書”, 奈良先端科学技術大学院大学 情報処理科学研究科 自然言語処理学講座, 2003.