Vol.2017-ICS-188 No.2
2017/7/20

複合的な文書分析技法を用いた 地元企業の口コミ分類補助システムの開発

安藤瞭[†1]　山本堅嗣宣[†2]　山本純子[†2]　大貫勉[†2]　床井真理子[†2]
原紳[†3]　渡邊信一[†3]

概要：地元メディアサイトの要望に従い，感情分析や類似文章検知などの複合的な文書分析技術を用いて同社の規約に違反するクチコミを自動分類し，運用を通じて逐次学習できるクチコミ分類の補助システムを開発した．

キーワード：自然言語処理，感情分析，誹謗中傷

Development of a Support System to Classify Reviews Using Multiple Language Processing Techniques

HARUKA ANDO[†1]　MITSUNOBU YAMAMOTO[†2]　JUNKO YAMAMOTO[†2]
TSUTOMU OHNUKI[†2]　MARIKO TOKOI[†2]
SHIN HARA[†3]　SHINICHI WATANABE[†3]

Abstract: We have developed a support system to classify reviews using multiple language processing techniques which can improve the precision of classify through incremental learning.

Keywords: NLP, Sentiment Analysis, Abuse

1. はじめに

近年，情報技術の発達により作成される電子情報は毎年40%近くの増加率となり，2020 年には 2015 年の 50 倍ものデータ量になることが予想されている[1]. Google での 1 日の処理データ量は 24 ペタバイトにのぼるなど[2]，日々人間が扱わなければならない情報量は爆発的に増大し続けており，これらの事情から様々な情報を自動的に分類する技術が求められている．

本研究は栃木県の地元密着系情報サイトである「栃ナビ！」を運営しているヤマゼンコミュニケイションズ株式会社の問題解決を行うための開発である．ポジティブな点にフォーカスすることによって地域振興に貢献するという同社が掲げる地域支援サイトの考え方に基づき，各種文書処理の技術を用いて自動的に投稿文の規約違反分類を処理するシステムの開発を行った．

投稿されるレビューの規約違反などのチェックは人力で行なっていたが，それには人件費がかかるという問題があった．今回はこれらの処理の一部を自動化し，支援することで人件費の削減を目的に開発を行った．本研究の要求をまとめると以下の通りである．

(1) 重複投稿などで規約に違反する文書の判別
(2) 使いまわしている文書の判別
(3) 誹謗中傷の目的で書かれた文書の判別

この 3 点の要求を満たすため、今回は感情分析や文書類似度の検知手法などを使用して第一弾となる Python ベースの API システムを開発し、運用のための試験を行った．

2. 現状と目標

現在，月平均 2,000 件の投稿記事の内，約 96.4%は有効データである. 問題になっている 3.6%の無効データ以外は可能な限り人間が見ずに掲示許可のラベリングを行うことが理想となる．

しかしながら，現実問題として 100%の精度で誹謗中傷などを検知することは困難である．このため，機械的な処理は支援にとどめ人間が常に関わるシステムとする．かつ逐次学習的な学習方針を取り，運用する中でより高い精度を得られるように意図した開発を行う．

また，当面の目標とする分類精度は誹謗中傷検知が難しいと考えられるため, これに合わせて基準設定を行なった．誹謗中傷に用いる感情分析の精度から，2016 年時点での感情分析研究の精度で最も高い 95%を目標にする[3].

さらに，分析のリクエスト 1 件に対して 200ms 以内に完了することを目標にする．この数値の根拠は，今後のリクエストの増大を考慮して，Mobage など大手 Web サービスのレスポンス速度を参考にした[4].

†1 宇都宮大学大学院 工学研究科
Graduate School of Engineering, Utsunomiya University.
†2 ヤマゼンコミュニケイションズ株式会社
Yamazen Communications Co., Ltd.
†3 宇都宮大学 工学部
School of Engineering, Utsunomiya University

3. システム構成

本件開発では与えられた自然言語処理タスクを実行するための学習器システムとそれを実運用するための API フレームワークによって構成されている．API のフレームワークでは実際に運用をするにあたり信頼性が高い Tornado フレームワークを用いて実装を行なった．

また，文書処理には Gensim，形態素解析には MeCab を用いて開発を行なった．

3.1 規約違反の文書分類システム

同じユーザーかつ同じ投稿という「重複投稿」は送信ボタンの連続クリックなどによる連投によって生じるものであり，掲載拒否されるもののうち，最も多いケースである．そこで，単純に文書間を比較し，同じものであれば無効のラベルをつけるものにした．これは原始的な手法で十分解決できることから判定文を用いた原始的な方法で構築した．

3.2 類似度判定システム

投稿に対してポイント制を定めて特典を出しているため，例えば「このお店の**カステラ**は絶品でした！また行きたくなるいいお店です」と「このお店の**ラーメン**は絶品でした！また行きたくなるいいお店です」のように，同じ文書で品物の名前だけ変更するケースも規約として禁止している．

そこで，Tata and Patel の TF-IDF cos 類似度評価法[5]によって検知を行うシステムを構築した．これは TF-IDF と cos 類似度判定を組み合わせたものである．

比較対象とするのは計算時間の関係から同じ記事内の投稿と同じユーザーの投稿のうち，最近の数件である．この件数は運用を通して決定する．

3.3 誹謗中傷判定システム

今回の課題となっている誹謗中傷判別において，大多数なものは特定の店舗や商品に対して批判を試みるものである．その次に皮肉などの要素を含んだものが存在する．そこで，今回我々は感情分析を用いて大多数の誹謗中傷記事をまずは排除することを試みた．

感情分析（Sentiment Analysis）とは，文章の感情を分析する手法であり，これらを使った誹謗中傷の検出は石坂ら[6]などが実際に応用して成果を出している．

今回の感情分析においては，BoW（Bag of Words）形式の特徴ベクトルを TF-IDF と LDA を用いて特徴抽出と次元圧縮を行い，教師付き学習について機械学習器によって学習を行った．

また，運用を通してさらに精度の向上と人間の分類精度に近づけることができるようにオンライン学習器である AROW(Adaptive Regularization of Weight Vectors)を導入した．

3.4 使用手法

(1) TF-IDF cos 類似度判定

TF-IDF cos 類似度判定は特徴抽出手法である TF-IDF と cos 類似度判定を組み合わせたものであり，Tata と Patel によって提唱された．

TF-IDF(Term Frequency – Inverse Document Frequency)は単語の出現頻度に基づいて単語の重み付けを行う特徴抽出手法であり，TF と IDF に分けて以下のように計算する．

$$\mathrm{TF}(t,d) = n_{t,d} / \sum_{s \in d} n_{s,d}$$

ここで，TF(t,d)は文書 d 内のある単語 t の TF 値であり，$n_{t,d}$ある単語 t の文書 d 内での出現回数，$\sum_{s \in d} n_{s,d}$は文書 d 内のすべての単語の出現回数の和である．

$$\mathrm{IDF}(t) = \log \frac{N}{\mathrm{df}(t)} + 1$$

ここで IDF(t)はある単語 t の IDF 値であり，N は全文書数，df(t)はある単語 t が出現する文書の数である．

これを用いて，文書 d における単語 t に対する TF-IDF 値は次のように計算できる．

$$\mathrm{TF \cdot IDF}(t,d) = \mathrm{TF}(t,d) \cdot \mathrm{IDF}(t)$$

さらに，ここで特徴抽出した特徴ベクトルについて LSI による次元圧縮を行い，以下の式で cos 類似度を計算する．ここで，得た値が 1 に近いほど文章の類似度が高く，-1 に近いほど，異なる文章であると言うように評価することができる．

$$\text{cosine similarity} = a \cdot b / |a||b|$$

本件では 0.8 以上の値を得た場合に，類似の文書として評価する．

(2) LDA(Latent Dirichelet Allocation)

LDA とは言語モデルの一つであり，次元圧縮手法の一つでもある[7]．潜在的ディレクレ配分法とも言われる．Hofmann によって考案された pLSA(Probabilistic Latent Semantic Analysis)[8]や LSA/LSI(Latent Semantic Indexing)を発展させた手法として知られている．

単語は独立に存在しているのではなく，潜在的なトピックを持ち，同じトピックをもつ単語は同じ文章に出現しやすいと言う特徴に着目し，文章中の単語のトピックを確率的に求める手法である．これを用いて単語をトピックごとにまとめることで次元圧縮をすることができる．

(3) Adaptive Regularization of Weight Vectors

Adaptive Regularization of Weight Vectors（AROW）[9]は K.Crammar が提案したオンライン線形分類器の一つである．Label noise による訓練例の急な変化にも頑健であり，過学習を起こしにくいアルゴリズムとされる．

4. 動作検証

試作では OS 環境を Mac OSX Sierra，サーバー環境を

Apache Apache/2.4.23 (Unix)、MySQL5.7.17、PHP 5.6.25 の環境を設置して動作検証を行った．

動作検証では現実の動作環境を仮定し，実験用のコンピュータに実行サーバーを立て，それに対して LAN を介して別のブラウザから POST リクエストを送信し，レスポンスを確認し，正常な動作を確認した．

運用環境での実装においては現在実装準備を行なっていることから当日その成果を公開する．

5. 評価指標

システム全体の評価指標として「全体精度(Accuracy)」「精度(Precision)」「再現率(Recall)」とその調和平均である「F1 値(F1-measure)」によって評価を行なった．その定義を以下に示す．

a：正しく掲載可能(Pos)と分類されたサンプル数
b：間違えて掲載可能(Pos)と分類されたサンプル数
c：間違えて掲載不可(Neg)と分類されたサンプル数
d：正しく掲載不可(Neg)と分類されたサンプル数

$$\text{Accuracy} = (a+d)/(a+b+c+d)$$
$$\text{Precision(Pos)} = a/(a+b) \quad \text{Recall(Pos)} = a/(a+c)$$
$$\text{Precision(Neg)} = d/(c+d) \quad \text{Recall(Neg)} = d/(b+d)$$
$$\text{F1-measure} = 2 * \text{Precision} * \text{Recall} / (\text{Precision} + \text{Recall})$$

6. システムの性能評価と考察

前項で定義した評価指標に基づいてシステム全体の性能評価を行なった．実験ではヤマゼンコミュニケイションズ株式会社から提供されたデータセットを用いた．性能評価においては全体の性能評価と比較的検出が難しい誹謗中傷の性能評価を個別に行なった．

6.1 全体の性能評価

全体の性能評価としては「栃ナビ！」にこれまで投稿されたものの内，135 件を「投稿許可」50 件と「重複投稿」25 件，「文章類似」25 件，「誹謗中傷」35 件のエラー対象に分類し，それぞれの精度を確認した．その結果を以下の表 1 に示す．

表 1．全体の性能評価

	Precision	Recall	F1-Score
許可	0.98	0.92	0.95
重複	1.00	1.00	1.00
文書類似	1.00	0.88	0.94
誹謗中傷	0.83	0.97	0.89
合計	0.95	0.94	0.94

全体の精度として 95%の精度で分類ができている．特に許可が 92%の再現率を確保しており非常に良い精度で分類もできていることがわかる他，誹謗中傷の再現率が 97%と非常によい分類精度を保っているが，精度自体は事前実験で行なっていた SVM の分類精度よりも低くなり, 83%であった．これは NG ワード検知との組み合わせで他の文書も検知してしまっていることが考えられる．しかしながら，これは AROW に対して再学習を行わせていくうちに改善されるのではないかと考えられる．

さらに，文書類似の再現率が極端に低いが，これはラベル設定の際に厳しく似た文書をつけたことが原因だとは思われる．また，実際には文書類似で掲載拒否される文書は少ないことから，これの再現率は将来的に向上させるべきとしても実用に耐えうると考えられる．

一方で，1 件の分析時間には 232ms かかっており，時間がかかっていることがわかる．この原因として最も大きかったのは，文書類似である．これは関連すると考えられる文書を全て特徴ベクトルに変換し、対象のレビューと関連文書を全て計算するのに時間がかかっているからであると考えられる．将来的には実用可能な水準になるように検知する対象の文書数を調整することで目標時間に近づけることも考慮に入れる．

6.2 誹謗中傷検知の性能評価

誹謗中傷検知の性能評価では「栃ナビ！」にこれまでに投稿された記事のうち，投稿許可(Pos)を 2,000 件，投稿不可(Neg)を 2,000 件の計 4,000 件を用いて評価を行なった．AROW を用いたシステムと SVM を用いたシステムの分類結果を以下に示す. SVM についてはグリッドサーチを用いてパラメーターの最適化を行なっている．

表 2．AROW を用いた誹謗中傷検知

	Precision	Recall	F1-Score
Pos	0.88	0.88	0.88
Neg	0.89	0.90	0.90
Total	0.89	0.89	0.89

表 3．SVM を用いた誹謗中傷検知

	Precision	Recall	F1-Score
Pos	0.93	0.93	0.93
Neg	0.92	0.92	0.92
Total	0.93	0.93	0.93

表 2 と表 3 を見てわかるように, AROW の誹謗中傷の検知精度は SVM と比較してあまり高くない. しかし, AROW について 5 回まで再学習を行なった結果，精度，再現率ともに 1%増大した．

これらの結果から，実運用が始まればある程度は精度を高めることができることが推測される．また，今後の改良においては AROW に対してパラメータ最適化手法などを実装し，学習効率を向上させることを検討している．

7. おわりに

本開発ではヤマゼンコミュニケイションズ株式会社の支援システムの開発に自然言語技術を導入し，文書分類支援システムを開発し，約 95%の精度で必要な分類が可能であるシステムを構築することができた．特に許可については 92%の再現率で保証することができており，かなり高い精度であることが考えられる．

特に誹謗中傷検知において感情分析手法を利用することで同社のデータセットにおいて最高 93%の検出精度が出せることが分かったことも一つの知見である．

今後は実際の運用を通じた性能評価と運用時に発生した問題解決を行うとともに，皮肉表現による誹謗中傷の抽出やスパム表現の検出などの実装なども行い，さらに高度な分類システムの開発を行いたい．

謝辞　本件開発にあたり，データセットの提供などにご協力いただいた，ヤマゼンコミュニケイションズ株式会社関係者各位と開発に協力いただきました株式会社 HitBit の碓氷様に感謝を申し上げます．

参考文献

[1] “World’s data volume to grow 40% per year & 50 times by 2020”. http://e27.co/worlds-data-volume-to-grow-40-per-year-50-times-by-2020-aureus-20150115-2/ , (参照：2017-02-03)
[2] 日立ソリューションズ ”ビッグデータとは”. http://www.hitachi-solutions.co.jp/belinda/sp/special/landing01/bigdata.html, (参照：2017-02-03)
[3] Vidisha M. Pradhan, Jay Vala, Prem Balani,. A Survey on Sentiment Analysis Algolithms for Opinion Mining. International Journal of Computer Applications, Volume 133 – No 9. January 2016
[4] “第 9 回　高速な Web API の実装とテスト―Mobage API を支えるノウハウ（1）” http://gihyo.jp/dev/serial/01/perl-hackers-hub/000901, (参照：2017-06-22)
[5] Sandeep Tata, Jignesh M.Patel.. Estimating the Selectivity of tf-idf based Cosine Similarity Predicates. ACM SIGMOD Record, Vol.36, 2, pp.7-12,2007
[6] 石坂 達也，山本 和英,. “Web 上の誹謗中傷を表す文の自動検出”. 言語処理学会 第 17 回年次大会 発表論文集(2011 年 3 月)
[7] David M. Blei, Andrew Y. Ng, Michael I. Jordan,. Latent Dirichlet Allocation. The journal of Machine Learning Research, Volume 3, 3/1/2003, Pages 993-1022
[8] Thomas Hofmann, Probabilistic Latent Semantic Indexing, Proceedings of the Twenty-Second Annual International SIGIR Conference on Research and Development in Information Retrieval (SIGIR-99), 1999
[9] Crammer, K., Kulesza, A. & Dredze, M.. Adaptive Regularization of Weight Vectors. Machine Learning. May 2013, Volume 91, issue 2, pp 155-187