依存構造を考慮した評価文書の分類

鍛治伸裕　　喜連川優

東京大学 生産技術研究所
〒 153-8505 東京都目黒区駒場 4-6-1
{kaji,kitsure}@tkl.iis.u-tokyo.ac.jp

評価文書の分類は近年になって注目を集めてきているタスクであり，これまでに様々な手法が提案されてきている．その中でも主流になっているのは，単語を素性にして分類器を学習するという方法である．だが，こうした手法には，係り受けを扱えないという問題がある．そこで我々は，文節間の係り受け関係を考慮した確率モデルを考案して，評価文書の分類精度を向上させることを試みた．実験の結果，提案モデルは，単語素性を用いた手法よりも高い分類精度を示すことが確認できた．

キーワード：評価文書の分類，依存構造

Dependency-based Sentiment Classification

Nobuhiro Kaji　　Masaru Kitsuregawa

Institute of Industrial Science, The University of Tokyo
4-6-1 Komaba, Meguro-ku, Tokyo 153-8505
{kaji,kitsure}@tkl.iis.u-tokyo.ac.jp

In the last few years, a lot of attention has been given to a new text classification task, sentiment classification. In this task, a popular approach is to learn a classifier that uses bag-of-words features. One flaw of such method is that it cannnot handle dependencies well. In this paper, we propose a dependency-based probabilistic model for sentiment classification, and show its effectiveness through our experiment.

Keywords: Sentiment Classification, Dependency Structure

1　はじめに

インターネットを見ると，いわゆる口コミのような情報をよく目にする．例えば，新製品の評価が掲示板に書き込まれていたり，映画の感想がブログに書かれていたり，といった具合である．このような，評価や感想が記述されたテキストのことを，ここでは評価文書と呼ぶ．

インターネット上の評価文書には様々な活用方法が考えられる．例えば，企業ならマーケティングに使えるだろうし，消費者であれば新商品の情報収集などに利用できるだろう．しかし，現在の技術では，インターネット上に散らばる評価文書を効率的に検索，閲覧することは難しい．

このような背景から，評価文書の検索，分類，加工など，評価文書に関連する処理技術が盛んに研究されている．その中の一つが，評価文書を肯定的な内容のものと否定的な内容のものに分類する処理である．これを評価文書の分類と呼ぶ．

評価文書の分類については，これまでに様々な手法が提案されてきている．その中でも主流になっているのは，単語を素性にして分類器 (ナイーブベイズや SVM など) を構築するという方法である [1, 5].

こうした手法が抱える問題の一つは，係り受けを扱えないことである．例えば次の文を考える．

(1)　印刷速度が今までの機種より早いです.
(2)　インクの減りがかなり早い.

(1) は肯定的，(2) は否定的な内容である．これらを正しく分類するには「印刷速度が早い」「減りが早い」といった係り受けの情報が必須であり，単語素性に基づく手法ではうまく分類できないだろう．

そこで我々は，文節間の係り受け関係を考慮した確率モデルを考案して，評価文書の分類精度を向上させることを試みた．このモデルでは，文は依存構造木として表現される．そして，文節の生起確率は，その親文節が観測されたもとでの条件付確率として定義される．

モデルの評価には，パソコン関連の掲示板から収集したデータを用いた．その結果，提案モデルは，単語素性を用いた手法よりも高い分類精度を示すことが確認できた．

本論文の構成は以下のとおりである．まず 2 節で関連研究を紹介する．次の 3 節では依存構造木について簡単な説明を行い，4 節で提案モデルの詳細について述べる．5 節では実験結果の報告を行い，誤り分析などの議論を行う．そして，最後に 6 節でまとめをする．

2 関連研究

これまで，単語素性に基づく分類手法を改良するために，様々な手法が提案されてきている．最もよく議論されるのが，単語 n-gram や系列パターンを素性として使う方法である [1, 8, 5, 10, 11]．これらの中には，係り受け関係を扱うことを目的として，n-gram や系列パターンを導入している研究もあるが，あくまでも近似的な扱いである．

Kudo らや Matsumoto らは，単語をノードとする依存構造木にテキストを変換して，その任意の部分木を素性に使う分類手法を提案している [2, 3]．しかし，このような手法では，機能語しか含まない部分木も素性として利用されてしまう．少なくとも日本語の場合，単語ではなく文節をノードとする依存構造木を考えたほうが自然である．

3 依存構造木

提案モデルは文を依存構造木で表現する．例として「印刷速度が今までの機種より早いです」という文を依存構造木に変換したものを図 1 示す．

この依存構造木は 4 つの文節 $b_1 \ldots b_4$ で構成されている．図中の括弧は文節，矢印は文節間の係り受け関係を表す．太字になっている単語は文節の主辞である．

以下では，依存構造木が分類システムの入力として与えられるという前提で議論を進める．文を依存構造木に変換するためには，文節間の係り受け関係と，文節の主辞を判定しなくてはならない．係り受け関係は構文解析システム KNP[1] を用いて判定し，主辞は文節内で最も後方に位置する自立語とした．

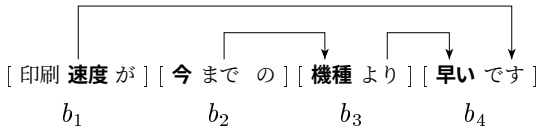

図 1: 依存構造木の例

4 依存構造に基づく確率モデル

本節では提案する確率モデルを説明する．評価文書の分類は，与えられた文書を肯定的と否定的の二クラスに分類するタスクである．これは，文書 d がクラス c に属する確率 $P(c|d)$ が与えられたとき，その確率を最大化するクラス c^* を求める問題としてモデル化することができる．

$$c^* = \arg\max_c P(c|d) \tag{1}$$

右辺はベイズ則を使って次のように変形できる．

$$\arg\max_c P(c|d) = \arg\max_c \frac{P(c)P(d|c)}{P(d)} \tag{2}$$

$$= \arg\max_c P(c)P(d|c) \tag{3}$$

$$\cong \arg\max_c P(d|c) \tag{4}$$

ただし $P(c)$ は一様分布と仮定している．

文書 d に含まれる文の数を $|d|$，先頭から i 番目の文の依存構造木を dep_i とすると，式 (4) は以下のように変形できる．ただし，依存構造木は互いに独立と仮定している．

$$c^* = \arg\max_c P(d|c) \tag{5}$$

$$= \arg\max_c P(dep_1, dep_2 \ldots dep_{|d|}|c) \tag{6}$$

$$\cong \arg\max_c \prod_{i=1\ldots|d|} P(dep_i|c) \tag{7}$$

式 (7) を見ると，結局モデルにとって重要なのは $P(dep|c)$ であることが分かる．以下，4.1 節では，依存構造木の生成確率 $P(dep)$ を定義し，それを元に $P(dep|c)$ を決定する．そして 4.2 節ではモデルのパラメータを推定する方法を述べる．

[1] http://www.kc.t.u-tokyo.ac.jp/nl-resource/knp.html

4.1 依存構造木の生成確率

まず，基本的な考え方を説明するために，図 1 に示した依存構造木 dep_{ex} が生成される確率 $P(dep_{ex})$ を考える．文節の生成確率はその親文節にのみ依存すると仮定すると，この依存構造木の生成確率は以下のようになる．

$$\begin{aligned} & P(dep_{ex}) \\ &= P(b_4)P(b_3|b_4)P(b_2|b_3,b_4)P(b_1|b_2,b_3,b_4) \\ &\cong P(b_4)P(b_3|b_4)P(b_2|b_3)P(b_1|b_4) \\ &= \prod_{i=1\ldots4} P(b_i|p_i) \end{aligned}$$

これは，いわゆる 2-gram を依存構造木に対して単純に拡張した形になっている．ここで p_i は文節 b_i の親文節を表す．b_4 は親文節を持たないが，文末にダミー文節を置いて考える．

同様の議論は，任意の依存構造木 dep についてもあてはまるので $P(dep)$ は

$$P(dep) = \prod_{i=1\ldots|dep|} P(b_i|p_i) \tag{8}$$

と定義できる．ただし $|dep|$ は dep の文節数である．

では次に，今までの「文節 b_i の生成確率はその親文節 p_i にのみ依存する」という仮定を拡張する．b_i の生成確率は p_i だけでなく，p_i の親文節 p_i^2，p_i^2 の親文節 $p_i^3 \ldots p_i^{n-2}$ の親文節 p_i^{n-1} にも依存していると仮定する (図 2 参照)．そうすると依存構造木 dep の生成確率は以下のように定義できる (cf.n-gram)．

$$P(dep) = \prod_{i=1\ldots|dep|} P(b_i|p_i, p_i^2 \ldots p_i^{n-1}) \tag{9}$$

式 (9) を元に $P(dep|c)$ を以下のように定めた．

$$P(dep|c) = \prod_{i=1\ldots|dep|} P(b_i|p_i, p_i^2 \ldots p_i^{n-1}, c)$$

これを式 (7) に代入したものが提案モデルとなる．実際の実験では n の値は 2 と 3 を試した．

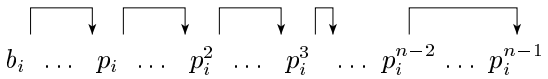

図 2: 文節間の係り受け関係

4.2 パラメータの推定

次は，モデルのパラメータを訓練データから推定する方法を述べる．$n = 2$ のときも $n = 3$ のときも全く同様なので，ここでは $n = 2$ の場合だけを考える．

$n = 2$ のとき，推定するべきパラメータは $P(b_i|p_i, c)$ だが，データスパースネスの問題があるため，訓練データから直接推定することは難しい．そこで以下のようにスムージングを行う．

$$\begin{aligned} P(b_i|p_i, c) = \lambda_1 P_e(b_i|p_i, c) + \lambda_2 P_e(b_i|c) \\ + (1 - \lambda_1 - \lambda_2)\frac{1}{Type(c)} \\ (0 < \lambda_1, \lambda_2 < 1) \end{aligned}$$

$P_e(\cdot)$ は訓練データからの推定値，$Type(c)$ はクラス c の訓練データに現われる文節の異なり数を表す．λ_1 と λ_2 はディベロップメントデータを用いて推定する．

残る問題は $P_e(\cdot)$ である．単純に考えるならば，$P_e(\cdot)$ は訓練データからの最尤推定値とすれば良いだろう．その場合は次のようになる．

$$P_e(b_i|c) = \frac{f(b_i, c)}{\sum_b f(b, c)} \tag{10}$$

$$P_e(b_i|p_i, c) = \frac{f(b_i, p_i, c)}{\sum_b f(b, p_i, c)} \tag{11}$$

$f(b, c)$ は文節 b が，クラス c の訓練データに出現する回数である．同様に $f(b, p, c)$ は，文節 b が親文節 p を伴って出現する回数である．$\sum_b$ の部分では，クラス c の訓練データに出現するあらゆる文節に対して和をとっている．

しかし，$P_e(\cdot)$ を最尤推定値とするのは問題がある．なぜなら，下のような文節を別々のものとして扱ってしまうからだ．

(3) a. 音質が [良かったですよ].
b. 音質が [良いです].
c. 音質が [良いですな].

では，主辞が同じ文節は全て同じものとして扱えば良いのだろうか．しかし，これも次のような例をうまく扱えない．

(4) a. 音質が [良くない].
b. 音質が [良いとは] 思いません.
c. 音質が [良いだけに] 残念です.

(3) の「良い」と (4) の「良い」では，性質が異なっていると考えられる．上のような表現をうまく扱うには，例えば言い換え技術を用いて，表現を正規化する方法などが考えられる．しかし，そのような手法は現状では困難であるので，以下で述べるような近似的な解決方法をとることにした．

まず，(4a) のような典型的な例に対しては特別な前処理を行う．具体的には，ある文節が否定または逆接を表す語 (「ない」「けど」など) を含む場合，その文節主辞にはタグを付与して，(3) のような場合とは明確に区別した．

そして，次に主辞を含む部分単語列に着目した．例えば (3a) の「良かったですよ」という文節を考える．この文節は「良い」「です」「よ」という 3 つの単語から成り，その主辞は「良い」である．したがって，主辞を含む部分単語列は s_1, s_2, s_3 となる (表 1)．ただし，単語はすべて原形で考えている．また別の例として，(4a) の「良くない」の場合も同じ表に示す．⟨ 否定 ⟩ というのは，否定を表す語 (この場合は「ない」) が文節に存在することを表すタグである．

表 1: 主辞を含む部分単語列の例

s_1	良い	です	よ
s_2	良い	です	
s_3	良い		
s_4	良い ⟨ 否定 ⟩	ない	
s_5	良い ⟨ 否定 ⟩		

我々は，この部分単語列を利用して $P_e(\cdot)$ を定めることを考えた．例えば「良かったですよ」という文節に対して $P_e(b|c)$ を次のように定義することにした．

$$P_e(b|c) = \frac{1}{3} \sum_{i=1,2,3} \frac{f(s_i, c)}{\sum_b f(b, c)}$$

$f(s_i, c)$ は，単語列 s_i を含む文節が，クラス c の訓練データに出現する回数である．

一般の場合 $P_e(b_i|c)$ と $P_e(b_i|p_i, c)$ は以下のようになる．

$$P_e(b_i|c) = \frac{1}{|S(b_i)|} \sum_{s \in S(b_i)} \frac{f(s, c)}{\sum_b f(b, c)}$$

$$P_e(b_i|p_i, c) = \frac{1}{Z} \sum_{s \in S(b_i), s' \in S(p_i)} \frac{f(s, s', c)}{\sum_b f(b, s', c)}$$

$$Z = |S(b_i)| \times |S(p_i)|$$

ただし $S(b)$ は文節 b の部分単語列の集合で，$|S(b)|$ はその要素数である．

5 実験と議論

提案モデルの有効性を検証するために，パソコン関連の掲示板から収集したデータを用いて実験を行った．

5.1 データ

実験に必要な訓練データと評価データは，インターネットサイトのパソコンに関する掲示板から集めた．収集に利用したサイトは「価格コム[2]」と「なんでもベスト店[3]」の二つである．

「価格コム」からは約 20,000 の評価文書を集めることができた．ここから無作為に抽出した約 18,000 文書を訓練データにし，残りを評価データ A とした．一方「なんでもベスト店」からは約 800 の評価文書が集った．これを全て評価データ B とした．表 2 に詳細な数字と，内訳 (肯定的か否定的か) を示す．括弧の中の数字は，一つの文書に含まれる平均文数である．

表 2: 訓練データと評価データの大きさ

	肯定的	否定的
訓練データ	11,245 (7.6)	6,867 (6.6)
評価データ A	1,258 (7.5)	780 (6.9)
評価データ B	473 (1.4)	349 (2.3)

5.2 実験結果

表 3 に，提案モデル ($n = 2, 3$) の分類精度を示す．λ の値は，訓練データの一部をディベロップメントデータに使って推定した．

比較のために，単語を素性とするナイーブベイズ (NB) と SVM の精度も併記する．SVM のカーネル関数は線形関数を使用した．ソフトマージンパラメータは，各評価データに対して最良の精度を出した値を採用した．また，素性には全ての単語を使うのではなく，自立語のみを利用した．否定や逆接の処理も，提案モデルと同様に行っている．

5.3 議論

提案モデルは，両方の評価データにおいて，他の二つの手法よりも精度が高い．この結果は，係り受

[2] http://www.kakaku.com
[3] http://www.nandemo-best10.com

表 3: 分類精度

			提案モデル	
	NB	SVM	n = 2	n = 3
評価データ A	82.1	82.4	83.7	83.3
評価データ B	81.2	76.3	83.9	85.3

け関係を考慮することの有効性を示唆している．

表 4: 分類に有効な係り受け

	提案モデル	NB
[コストパフォーマンスが] [高い]	2.190	2.331
[愛着が] [湧いてきます]	7.538	−0.148
[高い] [買い物だ]	−0.291	1.918
[メモリが] [少ない]	−0.812	1.357
[ノイズは] [気に] [ならない]	2.940	0.148
[言う] [こと] [なし]	1.698	−0.676
[買わない] [方が] [良い]	−2.301	−0.955
[ファンの] [音が] [うるさい]	−0.520	0.260

分類に有効であった係り受けを表 4 に示す．表中の 2 列目の数字は $log\frac{P(dep|c_+)}{P(dep|c_-)}$ の値を提案モデルで求めたものである．ここで dep は係り受け，c_+ と c_- は肯定的，否定的の二つのクラスを表わす．以下，この値のことをスコアと呼ぶ．スコアが正であれば肯定的，負であれば否定的といえる．提案モデルのパラメータ n は，表の上半分の係り受けには $n = 2$，下半分には $n = 3$ としている．また，一番右の列の数字は，単語素性に基づくナイーブベイズで求めたスコアである．すなわち $P(dep|c) = \prod_{w \in dep} P(w|c)$ とした値である (w は係り受けに含まれる自立語)．

この表からも，提案モデルが係り受けをうまく扱えていることが分かる．さらに，単語素性に基づくナイーブベイズでは，このような係り受けの扱いが十分でないことも確認できる．例えばナイーブベイズは「愛着が湧いてきます」に負のスコア (=否定的) を与えている．その原因を調べると「湧く」という語が次のような否定的な文脈で多く使われていることが分かった．

(5) a. 品質にも疑問が 湧いて 来ます．
b. 「いちいち手間取らせるな！」という感情が 湧いて しまいます．

「高い買い物だ」の場合も同様であった.「買い物」という語が，下に示すように，肯定的な使われ方をしていた．

(6) a. 値段の割にはいい 買い物 をした．
b. 十分満足できる 買い物 でした．

5.4 誤りの分析

表 5 に，提案モデルでうまく扱えなかった係り受けの例を示す．どちらとも否定的な表現だと考えられるが，提案モデルは正のスコアを与えている．以下では，この二つの誤りの原因を分析する．

表 5: 誤り例

[ディスプレイが] [見難い]	0.701
[強度が] [弱い]	9.736

まず「ディスプレイが見難い」を誤って肯定的だと判断してしまった原因を分析するにするため，訓練データを調べた．その結果「見難い」という語が，次のような形で肯定的な文書に多く出現していることが分かった．

(7) 画質は満足．色に関しては…(中略)…置き場所によっては 見難くなる．
(8) ×××の時は最大化でテレビを見るととても 見難かった のに対し，○○○はとても綺麗です．

(7) は全体的には肯定的な内容であるが，最後の部分で色に関して否定的なことが書かれていて，その中に「見難い」という語が使われている．(8) では，新しく購入した製品の感想に混じって，今まで使っていた製品について否定的な内容を述べている．しかし，全体としては，新しい製品に対する肯定的な内容となっている．

このように，全体としては肯定的/否定的な内容である文書の中に，否定的/肯定的な表現が紛れこむ問題は，映画のレビューを分類するさいにも報告されている．こうした現象への対応は今後の課題の一つである．

次の「強度が弱い」を間違った原因は,「強度が弱い」という係り受けが，否定的なほうの訓練データに一度も出現しなかったことであった．こうした問題には，言い換えや単語のクラスタリングなどが有効だろう．

5.5 今後の課題

上で議論したこと以外では，例えば次のようなことが今後の課題であると考えている．

提案モデルの問題点として，あらゆる文節を考慮して分類を行っているため，直感的には評価と関係のない表現まで分類に利用されていることがあげられる．そこで今後は，分類に有効な表現とそうでない表現を正しく認識して，有効なものだけを利用することが重要であろう．そして，そのためには，大規模な評価表現辞書を整備することが必要であると考えている．評価表現辞書を構築するには，人手で収集する手法，国語辞典やコーパスから学習するアプローチなどを検討している [6, 7, 9]．

もう一つの課題として，分類だけでなく検索にも提案モデルを適用することを考えている．提案モデルによって計算される $log\frac{P(d|c_+)}{P(d|c_-)}$ の値は，文書 d をランキングするときにも有効に使えると考えている．

6 おわりに

本論文では，評価文書の分類精度を向上させるために，文節間の係り受け関係を考慮した確率モデルを提案した．そして，そのモデルが，従来の単語素性に基づく手法よりも優れていることを実証的に示した．今後は，評価表現辞書の整備や言い換え表現の扱いを中心に研究を進めていく予定である．また将来的には，分類だけでなく検索というタスクにも取り組みたい．

参考文献

[1] Kushal Dave, Steve Lawrence, and David M.Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product revews. In *Proceedings of WWW*, pp. 519–528, 2003.

[2] Taku Kudo and Yuji Matsumoto. A boosting algorithm for classification of semi-structured text. In *Proceedings of EMNLP*, 2004.

[3] Shotaro Matsumoto, Hiroya Takamura, and Manabu Okumura. Sentiment classification using word sub-sequences and dependency subtrees. In *Proceedings of PAKDD*, 2005.

[4] Bo Pang and Lillian Lee. A sentimental education: Sentiment analyis using subjectivity summarization based on munimum cuts. In *Proceedings of ACL*, pp. 271–278, 2004.

[5] Bo Pang, Lillian Lee, and Shivakumar Vaihyanathan. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of EMNLP*, 2002.

[6] Ellen Riloff and Janyce Wiebe. Learning extraction patterns for subjective expressions. In *Proceedings of EMNLP*, 2003.

[7] Hiroya Takamura, Takashi Inui, and Manabu Okumura. Extracting semantic orientation of words using spin model. In *Proceedings of ACL*, pp. 133–140, 2005.

[8] Peter D. Turney. Thumbs up or thumbs down? senmantic orientation applied to unsupervised classification of reviews. In *Proceedings of ACL*, pp. 417–424, 2002.

[9] 小林のぞみ, 乾健太郎, 松本裕治, 立石健二, 福島俊一. 意見抽出のための評価表現の収集. 自然言語処理, Vol. 12, No. 3, pp. 203–222, 2005.

[10] 藤村滋, 豊田正史, 喜連川優. 文の構造を考慮した評判抽出手法. 電子情報通信学会第 16 回データ工学ワークショップ, 2005.

[11] 筏島郁子, 嶋田和孝, 遠藤勉. 系列パターンを利用した評価表現の分類. 言語処理学会第 11 回年次大会発表論文集, pp. 448–451, 2005.