Web上における製品レビューの機械学習による皮肉文判別

M2011MM069 鈴木佑哉
指導教員：石崎文雄

1 はじめに

近年，Web サービスの充実化により誰もが容易に情報を発信できるようになった, それに伴い, Web 上には商品に対する感想の書き込みも増加している．これらは, 消費者が購入する製品を選ぶときや, 企業が次期製品開発の計画を練るときなど, 様々な場面で貴重な情報源となる．従って, ブログや Web 掲示板などから消費者の製品に対する意見を収集し, よりきめ細かく消費者の意見を分析する必要がある. しかしながら, ユーザの製品に対する意見は, Web 掲示板等の人手では読み切れない莫大な量のテキスト中に存在するために, 人の手のみで分析することは困難となる．そこで活用できる技術の一つとして文章をポジティブ・ネガティブに分類するものが挙げられる．この技術で重要なのは分類精度である. 分類精度の向上の妨げとなる要因として皮肉文があげられる．皮肉とは自分が書いたりしたことの逆の意味または誰かをあざ笑ったり怒ったりすることを意図することである．そのためポジティブ・ネガティブの分類方法は単語の極性がよく利用されるため皮肉文は誤った分類をしてしまう．

そこで本研究ではポジティブ･ネガティブの分類精度向上のために, ツイッター上の製品に関するつぶやきを英語の製品レビューにおける皮肉文判別で使用されていた SASI (Semi-supervised Algorithm for Sarcasm Identification) アルゴリズムを用いて各ツイートへ重み付けを行い, k 分割交差検定と SVM を用いて皮肉文と通常文への分類を行う．分類に使用する製品レビューは楽天の電子書籍端末である kobo Touch のレビューを用いる．ツイートは 1〜3 の 3 段階に分け分類する．1 に近いほど皮肉文とし, 3 に近いほど通常分とする．重み付けをする前に障害となる URL やハッシュタグ等を除き茶筅による形態素解析を行う. SASI アルゴリズムはツイートから文字列のパターンを抽出し, そのパターンを元に重み付けを行う．その工程は VBA (Visual Basic for Applications) でマクロを作成しエクセルで行う．重み付け後, 分類を SVM を用いて分類するが, 学習に用いたデータのラベルの偏りによって精度が変動するので K 分割交差検定をもちいて変動を減少させる．SASI アルゴリズムの性能評価には精度, 再現率, F 値を使用する．

2 皮肉・風刺の判別に関連する研究

2.1 日本語における皮肉文判別の研究

[1] は, アイロニー表現 (皮肉) の解釈機構の認知モデルおよびその計算モデルを提案している．この論文は, アイロニーは話し手の期待, 期待と現実の不一致, 否定的態度を暗黙的に提示するという考えに基づき図 1 の認知モデルを提案している．また,「話し手は〜を信じている」というようなある命題に対する心的態度を表す高次命題を扱えるように拡張した関連性に基づく解釈モデルを提案している．この計算モデルでは, 発言内容やその高次命題から得ることのできる文脈によって暗黙的啓示の成立を判断し, 成立すればアイロニーのための環境（話し手の期待など）の構成要素を含むという条件を付け処理を行っている．

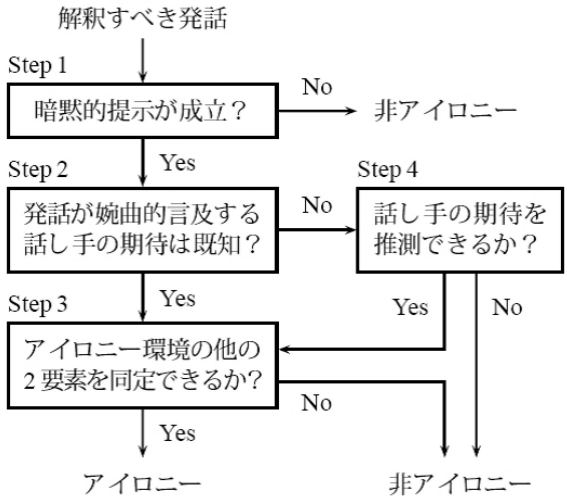


図 1 アイロニーの解釈仮定の認知モデル

2.2 英語における皮肉・風刺判別の研究

[2] は, 通販サイトのアマゾンにおける製品レビューを皮肉文かどうか 1 から 5 の 5 段階で判別をする．特徴付けと機械学習による分類方法を用いている．特徴付けには SASI というアルゴリズムを利用し, 分類方法は k 近傍法を改良したもので判別を行っている．SASI については 3 章で紹介をする．判別に用いた製品レビューは 66000 件で音楽プレーヤー, 本, デジタルカメラ, 携帯電話など 120 種類の製品から取得している．結果は, 精度 91.2%, 再現率 75.6%, F 値 82.7%.

[3] は, 新聞記事をデータセットとし, BNS (Bi-normal separation feature scaling) という重み付けアルゴリズムを利用し SVM で判別を行っている．(1) は BNS の計算式である．F^{-1} は標準正規累積分布の逆関数, tpr は文中のポジティブな単語の割合, fpr は文中のネガティブな単語の割合である．

$$|F^{-1}(tpr) - F^{-1}(fpr)| \quad (1)$$

判別に用いたデータは 4000 件の普通の記事と 233 件の風刺の記事で構成されている．

[1] の方法は精度を上げるためにはシステムの規模を大きくしなければならず, その点が問題となる．そこで, 海外の皮肉文判別で用いられている特徴を付ける手法を使用する．今回は, 製品レビューにおける皮肉文の判別を目

表 1　皮肉の分類方法の比較

論文名	対象言語	内容
[1]	日本語	皮肉表現の解釈機構の認知モデルおよびその計算モデルを提案
[2]	英語	アマゾンの製品レビューを皮肉と普通の文章に SASI で重み付けし, k 近傍法で分類
[3]	英語	新聞の記事を風刺と普通のものに BNS で重み付けし, SVM で分類

的としているため同じ製品レビューを対象とした [3] で使用されている SASI を重み付けに使用する.

3　皮肉文分類までの流れとデータセット

3.1　データフロー

本研究では実験において後の 3.2 節で説明するツイッターにおける投稿 (ツイート) を使用する. ここでは本研究の皮肉文分類の手順を (1)〜(5) で図 2 に示す.

(1) The Archivist Desktop というツールを使用しツイートを回収

(2) 重み付けを行うために後の 4 章で説明する前処理を行う

(3) 各ツイートに重み付けを行う

(4) SVM によって分類

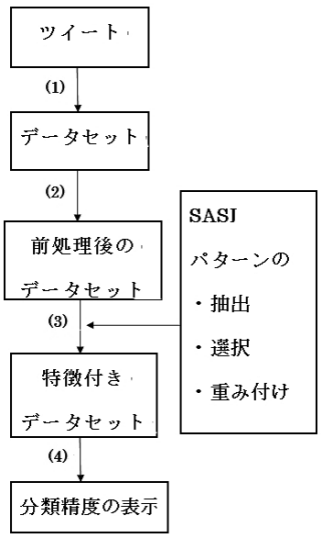
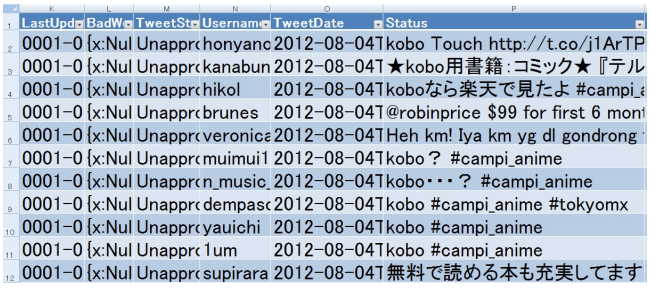


図 2　分類までのデータフロー

3.2　データセット

本研究で使用する実験データはツイッターのつぶやきを使用する. データの収集には The Archivist Desktop[1] というツールを使用した. The Archivist は, キーワードを入力するとそのキーワードを含むつぶやきを最新のものから取得することができる. 今回は, kobo に関するツイートを 8 月 4 日の 1556 件を取得した. データの中身としては, 検索キーワード, ユーザー名, ツイートした時間, ツイートの内容など 25 項目が記録されている. 今回使用する項目は, ツイートの内容のみである.

[1]http://visitmix.com/work/archivist-desktop/



LastUpde	BadW	TweetSta	Username	TweetDate	Status
0001-0	{x:Nul	Unapprc	honyanc	2012-08-04T	kobo Touch http://t.co/j1ArTP
0001-0	{x:Nul	Unapprc	kanabun	2012-08-04T	★kobo用書籍：コミック★『テル
0001-0	{x:Nul	Unapprc	hikol	2012-08-04T	koboなら楽天で見たよ #campi_a
0001-0	{x:Nul	Unapprc	brunes	2012-08-04T	@robinprice $99 for first 6 mont
0001-0	{x:Nul	Unapprc	veronica	2012-08-04T	Heh km! Iya km yg dl gondrong
0001-0	{x:Nul	Unapprc	muimui1	2012-08-04T	kobo？ #campi_anime
0001-0	{x:Nul	Unapprc	n_music_	2012-08-04T	kobo・・・？ #campi_anime
0001-0	{x:Nul	Unapprc	dempasc	2012-08-04T	kobo #campi_anime #tokyomx
0001-0	{x:Nul	Unapprc	yauichi	2012-08-04T	kobo #campi_anime
0001-0	{x:Nul	Unapprc	1um	2012-08-04T	kobo #campi_anime
0001-0	{x:Nul	Unapprc	supirara	2012-08-04T	無料で読める本も充実してます

図 3　データセット

4　重み付けと分類手法

4.1　前処理

各ツイートに重み付けをする SASI を適用するために前処理を行う. まず, ツイートごとに 1〜3 のラベルを付けておく. これはツイートの皮肉具合を数字化したもので, 1 に近いほど皮肉が文中に存在し, 3 に近いほど皮肉が文中に存在しない. ラベル付けは人間が行う. 次に, パターンを生成するために以下の様に置き換える.

表 2　単語の置き換え

置き換え前	置き換え後
製品	[製品]
会社	[会社]
タイトル	[タイトル]
著者	[著者]

表 2 の置き換え例を示す. kobo という単語の場合, 製品に当てはまるので [製品] と置き換える. また, すべての URL と特殊記号を削除する.

4.2　重み付けアルゴリズム SASI

4.2.1　パターンの抽出

ここでは, HFW (high frequency word) と CW (content word) に単語を分類する. これはデータセット中の出現率を元に定義する. HFW の条件は, 全単語中 0.1%以上の出現率であること. CW の条件は, 全単語中 0.01%以下の出現率であること. この出現率を調べるために形態素解析を行う. 今回は茶筌を使用した. その結果を Excel に格納し, 各単語の出現率を調べた. 次に, HFW と CW を利用してパターンの抽出を行う. 1 つのパターンは HFW が 2〜6 個, CW が 1〜6 個で構成される. また, パターンは始まりと終わりが HFW でなければならない. 例として, ”英語を公用語にした楽天が日本語プログラムの不手

際。”という文からいくつかのパターンを生成する. 生成されるパターンは, ”を公用 CW に”, ”を公用 CW にした”, ”[会社] が日本 CWCW の”といったものがある.

4.2.2 パターンの選択

パターン抽出で様々なパターンの抽出を行った. しかしその多くは一般的なパターン, もしくは限定的なパターンであることが多い. 取得したパターンの中から有用なものを選択するために以下の 2 つの条件に当てはまるものは削除した.

1. ラベル 1 と 3 のツイートから抽出したパターン
2. 文中に商品名が出てくる場合

2 の例には“ CW の本を捜して ”(CW にはカメラが入る) といったものが挙げられる.

4.3 パターンによる重み付け

パターンを選択したら, 重み付けのためにパターンを使用する. 各文の計算のために次のような計算を行う.

$$
\begin{cases}
1 & \text{完全一致　すべてのパターンの単語が文の中に現れ, 追加の単語もない場合} \\
\alpha & \text{部分一致　ほぼパターンが一致しているが, パターンを構成する単語が足りない場合 ただし, パターンの始めと終わりは HFW でなければならない} \\
\gamma * n/N & \text{部分一致　ほぼパターンが一致しているが, パターンを構成する単語が多い場合. ただし, パターンの始めと終わりは HFW でなければならない} \\
0 & \text{完全不一致　まったくパターンと一致しなし場合}
\end{cases}
$$

αとγはパターンが不完全一致の場合のスコアを縮小させるために使用するパラメータである.それぞれの範囲は $0 \leq \alpha \leq 1, 0 \leq \gamma \leq 1$ とする.今回は $Dmitry$ ら [2] と同じ$\alpha = \gamma = 0.1$ で計算を行った.それぞれの状況の例を紹介する.“ 英語を公用語にした楽天が日本語プログラムの不手際。”という文の場合,“ [会社] が日本 $CWCW$ の ”は完全一致で 1.“ [会社] が日本 CW の ”は CW が足りないため 0.1.“ [会社]CW が日本 $CWCW$ の ”は CW が多いため $0.1 * 6/7 = 0.08$.

4.4 句読点における重み付け

パターンを用いた機能に加え, 句読点を用いた重み付けも使用する. 以下が重み付けに使用するものである.

1. 文中の単語の数
2. 文中の!の数
3. 文中の?の数
4. 引用符の数 (「」など)
5. すべて大文字の単語の数

これらをデータセットの最大値で割ったものを重みとして使用する.

4.5 分類・評価

この章では重み付けしたツイートの分類方法と分類結果に対する評価方法について述べる. 重みを付けたツイートを分類するために k 分割交差検定と SVM を使用し, 評価方法には F 値を使用する.

4.5.1 K 分割交差検定

学習用データと評価用データに偏りがあった場合, 分類精度に影響が出ることがある. この影響を小さくするため k 分割交差検定を使用する. 方法は, データセットを k 個に分割する. そして分割したうちの 1 つを評価用, 残りを学習用のデータとして使用し, K 回繰り返す. こうして出された K 個の精度の平均が最終的な精度となる. 例えば, K=3 の時は以下の表 3 のように実行される.

表 3　3 分割交差検定

	A	B	C
1 回目	評価用	学習用	学習用
2 回目	学習用	評価用	学習用
3 回目	学習用	学習用	評価用

この 3 回分の精度の平均を分類方法の精度とみなす. 今回は K=5 で使用する.

4.5.2 Support Vector Machine

SVM (Support Vector Machine) は線形識別器の一つであり, サンプル (2) の識別関数を (3) に示す.

$$x = (x_1, \cdots, x_h)^T \tag{2}$$

$$f(x) = \sum_{j=1}^{h} w_j x_j + b \tag{3}$$

w_jは重みと呼ばれるパラメータで,b はバイアス項と呼ばれるパラメータである.この識別器の $f(x) = 0$ は超平面となる.SVM では訓練サンプルを完全に識別する超平面の中で最適解を探さなければならない.最適解はは超平面と訓練サンプルとの最小距離 (マージン) を評価関数として用いて, これを最大にするような超平面を選ぶ..線形識別面でクラス分けを行う際に, どの位置が最適な解なのかがわからない.そのため, マージンと呼ばれる線形識別面からサンプル点への最近傍までの距離を最適な選択はマージンが最大の時である.最適な選択を行うためのには式 (3) の条件の元で式 (4) を満たせば良い.これを満たすことが出来ればマージンが最大となる.

$$2/|w| = 1/|w| + 1/|w| \tag{4}$$

4.6 評価方法

実行結果の評価方法には先行研究で用いられていた精度, 再現率, F 値の 3 つの指標を用いる. 精度は正と予測

したデータのうち実際に正であるものの割合, 再現率は実際に正であるもののうち正であると予測されたものの割合, F 値は精度と再現率の調和平均である. それぞれの計算方法を以下に示す. 表 4 は精度等の式で使用されている TP などを示している.

表 4 分類結果の例

		データセット	
		正	負
予測結果	正	TP	FP
	負	FN	TN

$$精度 = \frac{TP}{TP + FP} \tag{5}$$

$$再現率 = \frac{TP}{TP + FN} \tag{6}$$

$$F\ 値 = \frac{2・再現率・精度}{再現率 + 精度} \tag{7}$$

5 評価

本章ではツイートへの前処理と重み付けについて述べる.

5.1 前処理

今回は 1556 件収集した中から広告とラベル 3 からランダムで 200 件抜いた 360 件を使用した. 表 5 は 360 件のラベルの内訳である.

表 5 ラベルの内訳

ラベル	件数
1	74
2	117
3	169

5.2 特徴付け

ツイートに特徴付けをした結果の一部を図 4 に示す. 特徴として使えるパターンと句読点は合わせて 3978 個だった.

表 6 実行結果

特徴付けの手法	精度	再現率	F 値
tf-idf	0.524	0.503	0.510
BNS	0.156	0.320	0.210
SASI	0.562	0.525	0.543

表 6 のように SASI は精度において 3.8%と 40.6%, 再現率においては 2.2%と 20.5%, F 値においては 3.3%と 33.3%tf-idf と BNS よりも高い数値を示した. BNS が他の手法に比べ精度等が低い値になった原因は特徴として使用できる単語もしくはパターンの少なさに原因があると考えられる. tf-idf は 1748 個, SASI は 3978 個の特徴があるが BNS は 27 個であった. 特徴として使用できる単語もしくはパターンが少ないほど精度など 3 つの値がが低いため, 精度等を上げるには追加のデータ収集によりパターンを追加する必要があると考えられる.

	KG	KH	KI	KJ	KK	KL
1	のおかげでCWD		でCWDCV	でCWDCV	でCWDCV	XCWD1
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0.1	0.1	0.166667	0.1125	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	1
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0.2	0.2	0	0	0.1
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0

図 4 重み付けの結果

6 まとめ

本研究では, SASI は精度において 3.8%と 40.6%, 再現率においては 2.2%と 20.5%, F 値においては 3.3%と 33.3%tf-idf と BNS よりも高い数値を示した. その結果 SASI の有効性を示すことができた. しかし, 精度・再現率・F 値が 56.2%, 52.5%, 54.3%とまだまだ低い値であり, 皮肉文または皮肉の可能性がある文の分類精度が低いという問題がある. 今後はこれらの問題に対し, 皮肉文または皮肉の可能性がある文には引用符が用いられる傾向があったため, 句読点による特徴付けのパラメータ変更, そして本研究で使用したデータが 1 つの商品のツイート 360 件と種類, 量ともに参考文献に比べ少ないため更にデータを収集する必要がある.

参考文献

[1] 内海彰,“ アイロニー解釈の認知・計算モデル, ”情報処理学会論文誌, Vol.41, No.9, pp.2498-2509, 2000.

[2] Dmitry Davidov, Oren Tsur, Ari Rappoport,“ Semi-supervised recognition of sarcastic sentences in Twitter and Amazon, ”Proceedings of the Fourteenth Conference on Computational Natural Language Learning, pp.107-116, 2010.

[3] Clint Burfoot, Clint Burfoot,“ Automatic Satire Detection: Are You Having a Laugh?, ”Proceedings of the ACL-IJCNLP, pp.161-164, 2009.

[4] Rada Mihalcea, Stephen Pulman,“ Characterizing Humour: An Exploration of Features in Humorous Texts, ”Computational Linguistics and Intelligent Text Processing, pp.337-347, 2007.