観光ブログの評価表現抽出による地域情報獲得

2006MI176 竹市 紘一郎 2006MI178 武野 佑基
指導教員　河野 浩之

1　はじめに

現在, WEB ページ上では多くのマップサイトが存在し, ユーザは観光の手がかりにしているが, その情報には PR 目的のものが多い. その代わりにブログなどに目を通して情報を得るというユーザも少なくない. しかしブログは無数にあり, 必要な情報に行き着くまでに時間がかかるという問題点がある.

本研究では, この問題を解決するために以下のようなシステムを構築した. 一括収集した観光ブログに対して形態素解析ツールの ChaSen を使用し, ブログの分かち書き, 品詞情報の付加を行う. その解析結果をもとにブログが示している地名およびブログ内に書かれている評価を抽出する. 抽出した地名は CSV アドレスマッチングサービスを用いて作成した地名テーブルと照合し, 緯度経度に変換する. 評価表現の抽出は ChaSen が形容詞, 形容動詞とみなしたものを抽出する. 抽出した単語を辞書と照合することにより, スコアリングを行う. ユーザがキーワードを入力し, それについての記述があるブログを評価スコア付きで Google Map 上に表示していく.

2　ブログ解析技術に関する諸研究

ブログ解析の技術を取り入れてシステムとして実装している研究を表 1 にまとめる. このようなシステムに用いられている地名抽出, および評価表現抽出に関する要素技術を以下より紹介していく.

2.1　評価表現抽出の先行研究

藤村ら [4] は候補語が肯定的な批判で出現する確率と否定的な批判で出現する確率の差分をスコア化することによって評価語のスコアリングを行っている. 精度は 78.6% である. また, 鍛冶ら [5] は候補語が肯定極性の代表語と否定極性の代表語のどちらとより多く共起しているかで極性判定, スコアリングを行っている. この手法は 84% という高い精度を出している.

2.2　地名抽出の先行研究

倉島ら [1] は地名と格助詞「へ/に」と連体化の助詞「の」との共起回数をカウントすることにより, 抽出を行なっている. 安村ら [6] は同一地名で違う場所を示すという, 地名のかぶりを地理的抱合関係を考慮して解決している. また, 白石ら [7] は住所地名テーブルを作成し, ドキュメント中の地名とテーブルの地名を照合することで, 抽出を行なっている. さらにジオコーディングシステムを用いて住所を経度, 緯度に変換し, 地図上に表示をしている.

3　地域情報獲得システムの構築

3.1　地域情報獲得システムの構築案

まず, ブログの解析は予め収集済みのブログを, 解析プログラムに通し, データベースに格納していくというバッチ処理の形を取る. 出力先の地図は, Google Map を選択した. Google Map は指定した場所をマーカーで分かりやすくすること, 拡大縮小が手軽に出来ること, などの便利な機能も数多く組み込まれており, API も公開されているため, システムに組み込みやすいと判断した. 全文検索は表示するブログを select する際の条件指定により, 実現させる.

ここでは, 本研究の評価表現抽出における地域情報獲得システムの構築を図 1 に従って説明する.

下記の項目 (1), (2) はバッチ処理であり, トランザクション処理は (3) から始まることとなっている.

(1) 下準備の始めとして, 観光ブログの記事を Wget を用いて収集しておく. 収集した観光ブログのファイルは全部同一ディレクトリに保存し, Perl のプログラムから読み込みやすいようにしておく. 拡張子はファイルをダウンロードしてきた時のデフォルト設定である html となっている.

(2) Perl のプログラムを実行して, データベースに観光ブログの内容を格納していく. ただし, 同時に形態素解析, 地名抽出, 評価表現抽出, 評価語のスコアリング, 抽出された地名の緯度経度の照合, 全文検索を行うためのインデックス作成もこのプログラム内で行う. 格納していく内容は, ファイル名, マップに載せる地名, 評価語のスコア, マップに載せる地名の緯度経度, 全文検索を行なうための全文インデックスである.

(3) ここからが実際にユーザーが扱う動作で, ユーザーが検索したい用語をキーワード検索画面に入力する.

(4) キーワード検索画面から受け取ったキーワードを, 検索用 PHP プログラムに送る.

(5) 送られてきたキーワードを元に, PHP 上から SQL 文にのせてデータベースにアクセスをして, キーワードが記載されているデータを MySQL の全文検索機能を用いて検索し, 検索にヒットした情報をマップに載せることが出来るようにする.

(6) データベースに登録してある情報を元に, 地名, 評価語のスコア, ブログのリンク先 URL を記載したマップを作成する.

表 1　ブログ解析の技術を取り入れたシステム

タイトル	データ対象	抽出データ	抽出方法
Blog からの街の話題抽出法の提案 [1]	一般ブログ	地名, 対象	格助詞との組合せを抽出
地理的抱合関係を用いた自動ニュースマップの実装 [2]	地域ニュース	地名	地理的包含関係
評価表現に基づく飲食店評判マルチファセット検索システム [3]	飲食店サイトレビュー	評価 (スコア)	[4] の手法

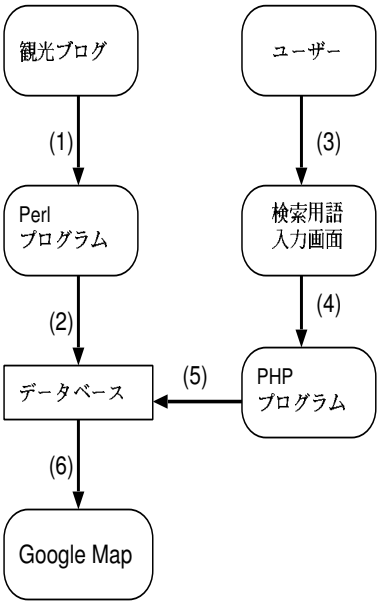

図 1　地域情報獲得システムの構築図

3.2　地名テーブル作成と緯度経度の獲得

抽出した地名に緯度経度の位置情報を付加するために, 地名テーブルを作成しておく. 本研究では, 緯度経度を CSV アドレスマッチングサービスを利用して取得する. このサービスは, 東京大学空間情報学研究センターが提供している. CSV 形式で保存されている住所データを, 緯度経度を付加したデータに変換を行なっている. 緯度経度の変換に利用されているデータは国土交通省の提供する「街区レベルで位置情報参照情報」である. しかし,「街区レベルで位置情報参照情報」は街区単位の位置情報を整備したもので, データ量が膨大である. これを整理して研究するのは困難なため, その処理を効率的に行うサービスが CSV アドレスマッチングサービスである. また, このサービスで提供される緯度経度は世界測地系であるため, Google Map の仕様にも適している.

本研究では CSV 形式の住所データを日本郵政のサイトから市町村レベルで取得し, CSV アドレスマッチングサービス を通して地名に緯度経度を付加したテーブルを作成する. 抽出した地名をこのテーブルに通すことで, 緯度経度に変換していく.

4　地域情報獲得システムの実装

4.1　観光ブログデータの収集

本研究で使用するデータの対象は,「にほんブログ村-旅行ブログ」から収集したものである. 収集するブログの記事は, 中部地方のブログ 100 件と限定した. これは本研究のシステムはプロトタイプ作成のためである. ブログの記事をダウンロードするにあたり, クローラの Wget を使用した.「wget -r ブログの URL」 と入力することでサイトにある記事を再帰的にダウンロードする.

4.2　地名抽出

地名抽出のプログラムは, 地名抽出プログラム枠内にあるプログラムとなっている. `chomp($_)` によって行末の改行コードを削除する. 読み込んだファイルは空白によって単語が区切られているため, `@blog = split(/\s+/, $_);` によって空白で分割して配列 `@blog` に一つずつ格納していく. `$blog[$i] =~ s/( | )+//g;` では全角, 半角の空白を削除する.

具体的な地名抽出の方法としては, 形態素解析で得た結果を用いて行なっていく. 形態素解析の結果における地名の品詞は「名詞-固有名詞-地域-一般」と分析されているので, 基本的にはこの文字列が出てきた時に地名の抽出を行っていく. ただし例外があり, 例えば「岐阜県岐阜市」のように県名と地名が重なった場合, 正確な地名の出現回数が分からなくなってしまので, `$hozon = $blog[ ];` を用いて 1 回前に出現した単語と形態素解析が分析した品詞を記憶し, 地名の後に「県」という単語が出現した場合は地名抽出を行わないことにした. 地名の出現回数に関しては, 地名が抽出された時 `$count++;` によってカウントしている.

地名抽出プログラム

```
while(<EXTR>){
 chomp ($_);
 @blog = split(/\s+/, $_);
  for($i=0; $i<@blog; $i++) {
      $blog[$i] =~ s/( | )+//g;}
  for($i=1; $i = 2; $i++){
   if(($hozon2 =~ /名詞-固有名詞-地域-一般
/) && ($blog[0] !~ /県/)){
      $count++;
      print OUT "$hozon1 ";
      $hozon1 = $blog[0];
      $hozon2 = $blog[3];
      last;}
   else{
    $hozon1 = $blog[0];
    $hozon2 = $blog[3];
    last;}
  }
}
close(EXTR);
```

4.3 評価語のスコアリング

評価語のスコアリングは東京大学の鍛冶ら [5] が構築した辞書を使用する. この辞書は鍛冶らの手法により,WEB 上に存在する大規模な評価文コーパスから構築した辞書であり, 評価語と極性値のペアが約 10000 組登録されているスコア辞書である. 一般表現が多く含まれているため, 汎用性が高く, 観光ブログの解析にも適していると言える.

評価語のスコアリングのプログラムは, 評価語のスコアリングプログラム枠内にあるプログラムとなっている. `while(<JISYO>)` の内部でスコア辞書の改行を削除し, 空白で分割して `@data` という配列に内容を格納する. 評価語の内容は `$ward` に格納されていて, 評価語の内容とスコア辞書の内容が一致した時に `$score += $data[0];` でスコアの合計を計算している. また, もし全てが評価語でなかった場合は「評価語ではありません.」と出力される.

4.4 地域情報獲得システムの各プログラム

地域情報獲得システム構築の際に, 5 つのファイルを作成した. 各ファイルの説明をし, 図 2 で地域情報獲得システムの実行例を示す.

● zen.pl

ブログの内容を読み込み形態素解析, 地名抽出, 評価表現抽出, 抽出された評価語のスコアリング, 抽出された地名の緯度経度の取得, データベースに各情報の格納を行なっている. データベースに格納する内容としては, 地名, 評価語のスコア, 緯度経度, また全文検索インデックスとしてブログの分かち書きを格納している.

● kennsaku.php

自分が検索したいキーワードを入力するための画面. キーワードを入力するとそのキーワードを BLOG.php に飛ばすようになっている.

● sql-info.php

データベースの接続の設定を行なう. ホストの名前, ユーザーネーム, パスワード, 接続するデータベースを入力してある.

● BLOG.php

kennsaku.php から受け取ったキーワードを元に, データベースに格納してあるブログ記事の内容に全文検索を行なうことと, XML データを作成し, map.html に XML データを飛ばすことを行なう. XML データの内容としては, データベースの属性内にある, ブログの識別番号, 抽出した地名, 評価語のスコア, 緯度, 経度である. また全文検索は検索用インデックスに対して, MySQL の much-against 文を用いて select した.

● map.html

BLOG.php から受け取った XML データを元にマップを作成する. 実行例を図 2 に示す.

評価語のスコアリングプログラム

```
while(<JISYO>){
 chomp ($_);
 @data = split(/\s+/, $_);
  for($k=0; $k<@data; $k++){
   $data[$k] =~ s/( | )+//g;}
 if($ward eq $data[1]){
  $score += $data[0];
  print OUT "$ward $data[0] \n";}
 elsif($data[1] eq "EOS"){
  print OUT "評価語ではありません。\n";}
 }
 close(JISYO);
}
```

図 2　地域情報獲得システムの実行例

5　地域情報獲得システムの考察評価

本研究では, 話題と評価表現に基づく地図検索システムを構築した. このシステムの性能評価を実施するために, 本システムと, 既存の 2 つのサイトを実際に利用した上で, アンケートに解答してもらった. アンケートは学生研究室の中の 15 人 を対象とする. アンケート内容は本研究と「全国観光マップ」, 「日本ブログ村」の中部地方のデータのみと比較してもらい, 次のような項目について 10 段階評価を行なってもらった.

(1) 観光ブログで訪れている場所の評価 (評判) を分かりやすく入手できた
(2) 観光ブログで訪れている場所の評価 (評判) を素早く知ることができた
(3) 自分が検索したい項目 (キーワード) を簡単に調べることができた
(4) システムに満足した (操作面, 情報収集面, 労力面, 検索時間などの総合評価)
(5) 本システムの不満な点, 改善点の記入

アンケートの結果を表にまとめると表 2 のようになった. 「観光マップ」,「ブログ村」におけるスコアの () 内の数字は本研究とのスコアの差となっている.

表 2　地域情報獲得システムのアンケート結果

質問項目	本研究	観光マップ	ブログ村
(1)	6.75	3.87(2.88)	5.12(1.63)
(2)	7.37	3.87(3.5)	5.87(1.5)
(3)	7.12	4.87(2.25)	7.25(-0.13)
(4)	8.12	4.87(3.25)	7.25(0.87)

アンケート項目の (5) より, 評判情報については, スコアが降順になっていて手軽に検索できるという高評価を得ることが出来た. 半面, 評価の情報が適切かどうか, 評判のスコアの小数点の設定, リンク先のタイトル情報, といった課題も見られた. しかし 3 章で挙げたような問題点は解決することが出来た.

またシステム全体については, 全体の満足度に関しては高評価が得ることが出来た. しかし, キーワードを簡単に検索できるという点では, それほど高い結果は得ることが出来なかった. これは, 本研究はプロトタイプのシステムなので, 観光ブログの情報が少ないためにこのような結果になった. 観光ブログの情報量を増やすことにより, 劣っていたアンケート結果は改善されるので, 本研究で用いたシステムは一般的な観光ブログ検索サイトとしておおよそ利用できるのではないかと考えられる.

6　まとめ

本研究では, WEB 上に存在する未整理のブログからの観光情報の読み取りが難しいという問題点を, ブログの形態素解析をし, 地名と評価を抽出し, GoogleMap 上に表示することによって解決を試みた.

また, 各項目 10 段階評価のアンケートにより本研究とマップサイトおよびブログサイトとの性能比較を行った. 評判情報の分かりやすさに関して 6.75, 素早さは 7.37 という結果が得られた. キーワード検索のヒットに関しては 7.12 となった. また, システム全体の満足度は 8.12 であった. 他のサイトと比べて低かった項目はキーワードのヒットに関して,「ブログ村」の方が 0.13 上回っていた.

評価表現のスコア化に関しては数値化されたことで見やすくなった半面, 基準の値が定まってないという問題があったため, ブログをスコアで昇順ソートすることによって解決した.

地名の抽出精度の関しては 83% という高い数値が出た. 観光ブログは市町村名を記述してるものが多いため, 市町村名に的を絞った抽出は観光ブログに関して実践的であると言える.

また今後の課題としては, 建物などの地名ではない場所を表す有力な情報を抽出し緯度経度に変換していくこと, 係り受け関係に着目した評価表現の抽出などが挙げられる.

参考文献

[1] 倉島健, 手塚太郎, 田中克己, “Blog からの街の話題抽出手法の提案,”DEWS2005, 2C-i10, 2005.

[2] 櫻井敦規, “地理的抱合関係を用いた自動ニュースマップの実装,” 南山大学, 数理情報学部, 情報通信学科 2007 年度, 卒業論文要旨集, pp.164-165, 2007.

[3] 佐藤誠也, 祖父江達師, 稲垣論, “評価表現に基づく飲食店評判マルチファセット検索システム,” 南山大学, 数理情報学部, 情報通信学科 2008 年度, 卒業論文要旨集, pp.172-175, 2008.

[4] 藤村滋, 豊田正史, 喜連川優, “電子掲示板からの評価表現及び評価情報の抽出,” 人工知能学会全国大会 (第 18 回), 3F1-03, 2004.

[5] 鍛治伸裕, 喜連川優, “自動構築した評価文コーパスからの評価表現辞書の構築,” 日本データベース学会 Letters Vol.6, No.1, pp.1-4, 2007.

[6] 安村祥子, 池崎正, 渡邉豊英, 牛尼剛聡, “blog マッピングを用いたイベント情報抽出,” DEWS2007, D8-3, 2007.

[7] 白石陽, 有川正俊, 相良毅, 浅見泰司, “空間ドキュメント管理システムの設計と実装,” DEWS2007, B7-10, 2007.