一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
LOIS2015-71(2016-03)

単語の分散表現を使用した観光地推薦システムの構築

開地 亮太† 檜垣 泰彦‡

†‡ 千葉大学大学院工学研究科〒263-8522 千葉県千葉市稲毛区弥生町 1-33
E-mail: †afpa3246@chiba-u.jp, ‡higaki.yashiko@faculty.chiba-u.jp

あらまし 観光情報を入手する手段として Web 検索が用いられている．Web 検索ではすでに知っている観光地名や単語でしか検索を行えない問題点がある．本研究では自分のお気に入りの観光地や好きな観光地の特徴，行きたい観光地のイメージなど，既知の情報から観光地を自動的に推薦するシステムを提案する．利用者の入力をベクトル化し，事前に作成した観光地の特徴ベクトルとのコサイン類似度を評価することで，利用者の入力に沿った観光地推薦を実現する．観光地の特徴ベクトルは Web 上にある膨大な観光情報をテキストデータとして取得し，ニューラルネットワークによる学習を行うことで獲得した．作成した観光地推薦システムを iOS アプリケーションとして実装した．実装したアプリケーションの動作を検証し，その有効性を確認した．

キーワード word2vec，観光地推薦システム，大規模文書集合，単語ベクトル

Recommendation system of tourist spots using distributed representations of words

Ryota KAICHI† Yasuhiko HIGAKI‡

†‡ Graduate School of Engineering, Chiba University 1-33 Yayoi-cho, Inage-ku, Chiba-shi, 263-8522 Japan
E-mail: †afpa3246@chiba-u.jp, ‡higaki.yasuhiko@faculty.chiba-u.jp

Abstract Web search has been used as a means to obtain tourist information. The Web search is a problem that Web search can't get only tourist place names or a word you already know. In this paper, we propose a system to automatically recommend tourist spots from known information such as favorite tourist spots and image of the tourist spots that you want to go. This system recommends tourist spots in consideration of the user's input to evaluate the cosine similarity between a vector from the user's input and feature vectors of tourist spots. Feature vectors of tourist spots were obtained by learning large amounts of information in a neural network. The authors have implemented the recommendation system of tourist spots as iOS application, and evaluated the effectiveness of it.

Keywords word2vec, Recommendation system of tourist spots, Large amounts of information, Word vector

1. はじめに

近年、格安旅行プランや LCC などの格安航空の参入などにより旅行にかかる費用低下し，旅行がより身近なものになっている．また，インターネットの爆発的な普及により，Web 上に非常に多くの観光情報が提供されるようになり，こうした観光情報を Web 検索することによって旅行計画を立てることが多くなった．Web検索を用いた旅行計画において具体的な旅行目的が決まっていたり，自分が知っている観光地を調べたりする場合は，検索クエリが決まっていることが多いため有効であることが多い．しかし、Web 検索では，自分の知らない観光地を検索クエリとすることが出来ないため，たとえ興味がある観光地であっても検索・発見することは困難である．また“京都 観光地 おすすめ”といった検索クエリで探したとしてもアクセス数が多い順にずらっと Web ページが表示されるだけで，提示された膨大な情報から自分の嗜好に合った観光地を探すことは困難である．さらに検索クエリによっては観光に関係のない検索結果が提示されることがある．観光地推薦において Web 検索では得られない新規性(Novelty)や発見性(Serendipity)が獲得できることが求められている．

そこで本稿では自分のお気に入りの観光地や好きな観光地の特徴，行きたい観光地のイメージなど，既知の嗜好情報から観光地を自動的に推薦するシステムを提案する．目的地を決定する段階での使用を想定し，このシステムにより Web 検索で観光地を発見する上での問題点を解決し，利用者の嗜好に沿った観光地の発見を支援する．

2. 関連研究

インターネット上で観光地を探す場合，キーワード検索を利用する．キーワード検索では利用者がすでに

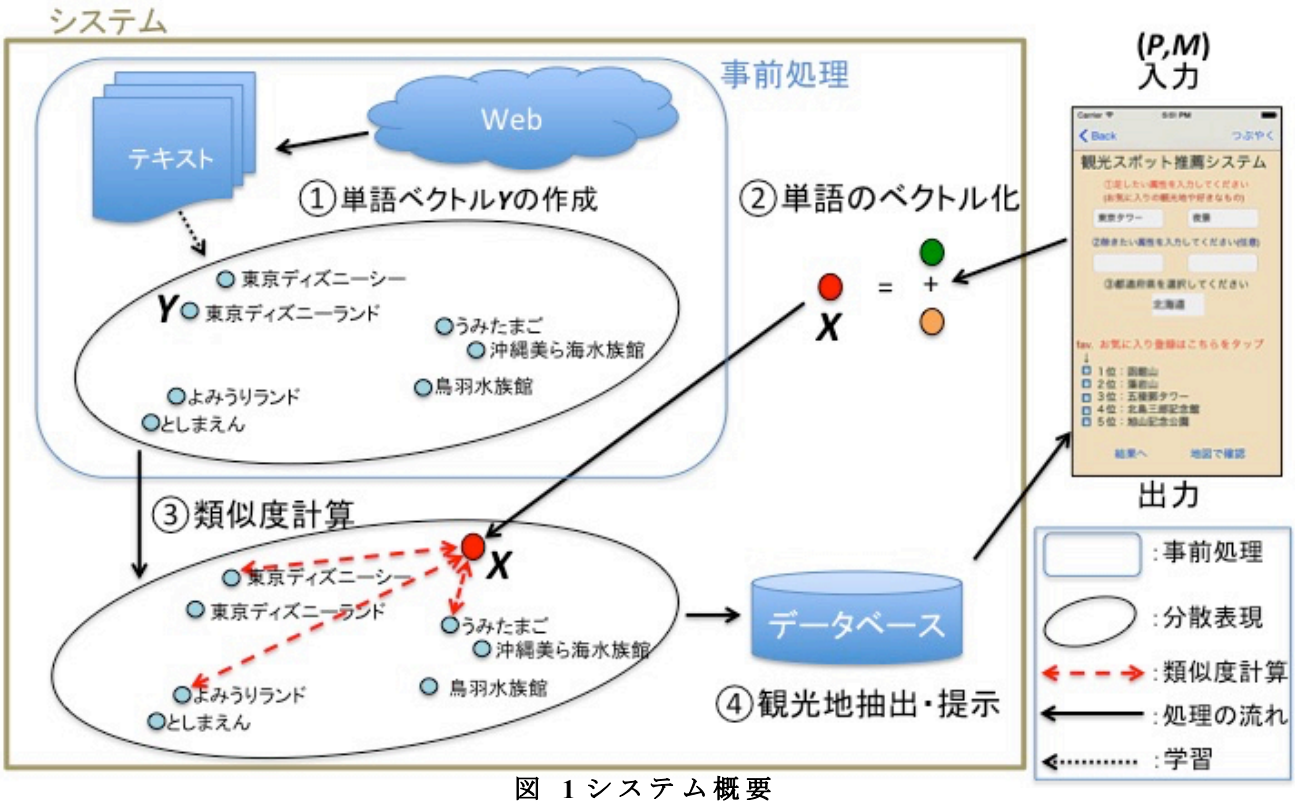

図 1 システム概要

知っている観光地名や単語のみでしか検索クエリにすることができない．そのため新たな観光地を探したい場合検索を行うことができない．また検索クエリを県名や市町村名で検索を行った場合，膨大な検索結果が出力されてしまう．そこから利用者の嗜好にあった情報を見つけることは非常に困難である．こういった問題点から観光地推薦に関連する研究が盛んに行われている．

利用者の既知の観光地から推薦を行うシステムとして上原らの研究[1] がある。上原らは Web から観光情報を抽出することで各観光地に対する特徴ベクトルを生成し類似度計算を行うことで，利用者のお気に入り観光地に類似した観光地を推薦するシステムを提案している．このシステムは利用者のお気に入りの観光地一つを入力するように設計されている．そのため複数の観光地や「綺麗な」や「緑豊かな」などの抽象的表現の利用など入力の多様性については考慮されていない．感情や情景など入力に自由度を持たせることは，より利用者の嗜好にあった観光地を推薦する点において必要である．

目的地と目的を選択肢から選んで観光地や観光キーワードを推薦するシステムとして松並らの研究[2]や杉浦ら研究[3]がある．これらのシステムでは探したい観光地の特徴などを選択肢から選んで，推薦が行われる．しかしあらかじめ設定した選択肢から選ぶ方式では，利用者が望む観光地の特徴が選択肢に存在しないという問題点や選択肢の組み合わせには限界あるという問題点がある．

このような問題背景から，我々は利用者のお気に入りの観光地や除きたい観光地，好きな特徴や情景などの入力から観光地推薦が行えるシステムを試作した[4]．試作したシステムでは入力に観光地が含まれることが前提であった．本研究では，入力に観光地が含まなくとも観光地が推薦される，自由な入力が行えるシステムを目指す．提案手法の独自性は，単語の分散表現を用いて観光地推薦を行う点，観光地や行きたいイメージなどの入力を検索システムを使用するように自由に設定できる点である．

3. 提案手法

本システムの概要図を図 1 に示す．本システムの構成は，次の 4 項目に分けられる．

① 単語ベクトルの作成
② 入力単語のベクトル化及び計算
③ 単語ベクトルの類似度計算
④ 観光地抽出・提示

3.1. 単語ベクトルの作成

本節では本研究の基礎となる分散表現について述べる．

3.1.1 単語の分散表現

本システムでは入力された単語の特性を足したり引いたりすることで，利用者の嗜好を判断する．そこで単語の特性を計算できる形にする必要がある．自然言語処理において単語が持つ意味情報をベクトルによっ

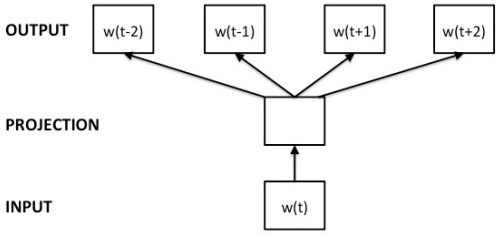

図 2 Skip-gram モデル

て表現する単語ベクトル空間モデル[5]が広く用いられてきた．このモデルは似た文脈に出現する単語は似た意味を持つという仮説である Distributional Semantic Models[6]に基づいている．近年では，ニューラルネットワークによる学習モデルである Skip-gram モデル[7]が Mikolov らによって新たに提案され，従来の手法に比べ飛躍的に精度が向上した．また Skip-gram モデルを用いることにより'Paris'-'France'+'Italy'="Rome"のような単語の特性を捉えた意味計算が可能となった．本システムでは，単語ベクトルを大規模文書集合(コーパス)から学習・作成するツールとして Mikolov らの手法[7]を実装した word2vec[1]を使用する．

3.1.2 word2vec

word2vec とはニューラルネットワークを用いた単語の分散表現(単語ベクトル)を作成する手法である．Skip-gram モデルの概略図を図 2 に示す．入力コーパスの 1 文$(w_1 \dots w_r)$が与えられると Skip-gram モデルは次式を最大化するような単語ベクトルを求める．

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c, j\neq 0} \log p(w_{t+j}|w_t) \qquad (1)$$

ここで c は中心単語w_tに対してどの距離までの単語を文脈とするか設定する定数である．確率$p(w_{t+j}|w_t)$は log-bilinear モデル[8]と呼ばれ以下のように定式化される．

$$p(w_{t+j}|w_t) = \frac{\exp(v_{w_t} \cdot u_{w_{t+j}})}{\sum_{w^T} \exp(v_{w_t} \cdot u_{w^T})} \qquad (2)$$

確率$p(w_{t+j}|w_t)$は単語w_tから周辺の単語w_{t+j}を予測するものである．周辺の単語の分布が似た単語ベクトル同士は似た値をとるように計算を行う．すなわち単語ベクトルの値が近いほど単語の意味も近くなると言える．word2vec によって単語を分散表現に変換することにより単語を意味空間上の 1 点に対応づけることができる．学習した単語ベクトルを主成分分析で可視化したものを図 3 図 4 に示す．コーパスは Yahoo!知恵袋[2]の観光カテゴリから収集した質問回答を用いた．

図 3 図 4 より観光地においても意味的に関連の強い観光地はベクトルが近くなることが見て取れる．学習させるコーパスを観光に関係あるものにし，観光地の

[1] https://code.google.com/p/word2vec
[2] http://chiebukuro.yahoo.co.jp

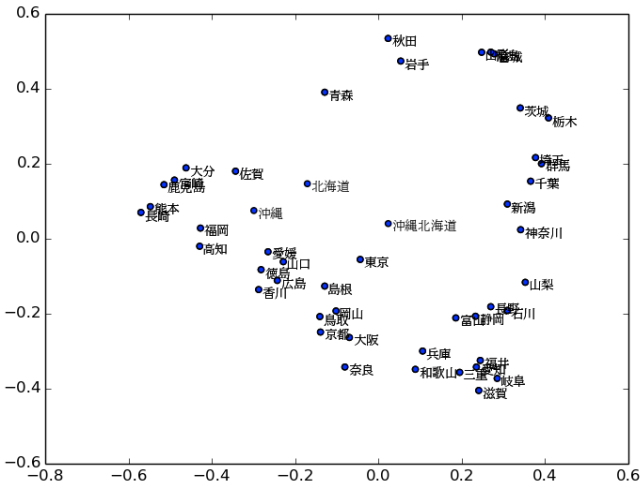

図 3 単語ベクトル可視化(都道府県)

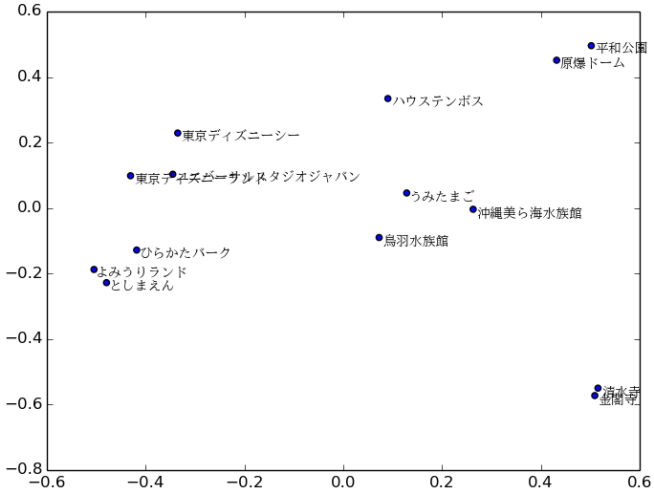

図 4 単語ベクトル可視化(観光地)

単語ベクトルを作成することで観光地同士の意味な計算ができると考えられる．本システムで用いるコーパスについては 4.2.節で述べる．

3.2. 入力単語のベクトル化及び計算

本システムでは利用者の入力を追加したい特性（P_n）と除きたい特性（M_n）の 2 つを想定している．3.1.節の手法で作成した単語ベクトルを利用者の入力に適応するとそれぞれの単語ベクトル$\boldsymbol{P_n}, \boldsymbol{M_n}$は，

$$\boldsymbol{P}_n = \{p_{n_1}, p_{n_2}, \dots, p_{n_{600}}\} \qquad (3)$$

$$\boldsymbol{M}_n = \{m_{n_1}, m_{n_2}, \dots, m_{n_{600}}\} \qquad (4)$$

と表せる．よって入力ベクトル $\boldsymbol{X}$ は，

$$\boldsymbol{X} = \boldsymbol{P}_1 + \boldsymbol{P}_2 \dots + \boldsymbol{P}_n - \boldsymbol{M}_1 - \boldsymbol{M}_2 \dots - \boldsymbol{M}_n \qquad (5)$$

と表現できる．

3.3. 単語ベクトルの類似度計算

本システムでは類似度計算処理としてベクトル空間モデルのコサイン類似度の計算を行う．3.2.節で事前に作成した単語 m 個の単語ベクトル$\boldsymbol{Y}_m$は，

$$\boldsymbol{Y}_m = \{y_{m_1}, y_{m_2}, \dots, y_{m_{600}}\} \qquad (6)$$

と表せる．また，入力ベクトル $\boldsymbol{X}$ と事前に作成した単語ベクトル$\boldsymbol{Y}_m$のコサイン類似度$\cos(\boldsymbol{X}, \boldsymbol{Y}_m)$は，

表 1 Google カテゴリ

カテゴリ				
・amusement_park	・establishment	・aquarium	・spa	・stadium
・place_of_worship	・museum	・park	・zoo	

$$\cos(\boldsymbol{X}, \boldsymbol{Y}_m) = \frac{\boldsymbol{X} \cdot \boldsymbol{Y}_m}{|\boldsymbol{X}||\boldsymbol{Y}_m|} \quad (7)$$

で計算される．この計算により入力された単語と学習済みの全単語に対して類似度の順位付けを行う．

3.4. 観光地の抽出・提示

単語ベクトルの特性上，利用者の入力が「楽しい」の場合，類似度上位にくる単語は「楽しみ」や「楽しさ」となる．この例のように観光地以外のものが推薦されることを避けるため，類似度で順位付けされたものから観光地のみを抽出し提示を行う．観光地抽出には観光地データベースを用いる．本システムで用いる観光地データベースについては 4.1.節で述べる．

4. システム設定

本稿ではシステムに用いる観光地データベース，コーパス，word2vec の学習設定について述べる．

4.1. 観光地データベース作成

本システムで用いる観光地データベースについて述べる．観光地データベースの精度を上げるため複数の手段で作成した観光地データベースについて評価を行い精度が一番良いものをシステムで使用する．

4.1.1. 観光地候補の収集

観光地候補の収集を以下の 2 つの手段で行う．

(a)日本語版 Wikipedia[3]の見出し語 100 万件の中で，緯度経度情報を持っているものを収集する．

(b)googleplace API[4]のプレイス検索で任意の緯度経度を入力とし，表 1 に示す Google によって定義された 9 種のカテゴリを持つ施設を取得する．

4.1.2. 観光地の選定

観光地かどうかの判断は人のそれぞれの主観によって変化する．そのため観光地の判定は客観的な判断で定める必要がある．客観的な判断で観光地か否かの判定をするため Yahoo!知恵袋のカテゴリ構造を利用する．カテゴリ分けがされていることから，あるキーワードを含む質問が他のカテゴリに比べ観光カテゴリに多く存在していたとすると，このキーワードは観光に関係の深いキーワードであると言える．この特性を用いることで観光地の判定，取得を行う．あるキーワードを含む質問本文や質問に対するベストアンサー，キーワードにマッチした質問の総数取得には質問検索 API[5]を使用した．

[3] http://ja.wikipedia.org/wiki
[4] https://developers.google.com/places/
[5] http://developer.yahoo.co.jp/webapi/chiebukuro/chiebukuro/v1/questionsearch.html

表 2 観光地候補と関連度

観光地候補(k)	ヒット件数(Jk)	ヒット件数(Ak)	関連度(Rk)
東京スカイツリー	3,693	6,703	29.779
千葉大学	9,306	43	0.249
新江ノ島水族館	369	484	41.208
清水寺	14,807	17,957	44.569
靖国神社	732	12,375	3.197
九州大学病院	2	112	0.965

本研究では 4.1 節で述べた 2 つの手段で収集した観光地候補 k をキーワードとして検索を行い，「地域，旅行，お出かけ」>「国内」カテゴリ(観光カテゴリ)でのヒット数J_k，カテゴリ全体でのヒット数A_kを取得する．Yahoo!知恵袋における観光カテゴリでの総質問数を S，全カテゴリでの総質問数を L とすると，あるキーワードを含む質問が存在する割合I_{S_k}，I_{L_k}は

$$I_{S_k} = \frac{J_k}{S} \quad (8)$$

$$I_{L_k} = \frac{A_k}{L} \quad (9)$$

と表せる．2015 年 11 月時点での観光カテゴリでの総質問数は 1,790,753 件，全カテゴリでの総質問数は 96,790,696 件であった．この 2 つの割合をもとに，観光地候補 k がどの程度観光に関係しているかを表す度合い(観光関連度)R_kを

$$R_k = \frac{I_{S_k}}{I_{L_k}} \quad (10)$$

と定義する．観光地候補のJ_k，A_k，R_kの値の例を表 2 に示す．観光光関連度は 0~54.05 の間で変化し，観光カテゴリでの質問割合が多いほど値は大きくなる．

表 2 より観光地として考えられる「東京スカイツリー」や「新江ノ島水族館」は観光関連度の値が大きくなっており，逆に観光に関係ないと考えられる「千葉大学」や「九州大学病院」は観光関連度の値は小さくなっている．観光関連度R_kが設定した閾値を上回る観光地候補を観光地としてデータベースに登録を行う．

4.1.3. 観光地データベース評価

閾値を①1，②13.5，③27 と変化させ観光地データベースを作成した．作成した観光地データベースの観光地登録数を表 3 に示す．

それぞれの観光地データベースについて，情報工学分野を専攻する学生 9 名に作成した観光地データベースに登録されている観光地をランダムに 30 件ずつ提示し，「観光地である」「知らない(分からない)」「観光地ではない」の 3 つの選択肢から選択してもらった．「知らない」観光地を減らすため回答者には自分の馴染みの深い県名を選択してもらい，選択した県に存在する観光地候補のみを提示した．また学生 36 名に思いついた観光地をあげてもらい，95 件の回答を得た．

表 3 観光地登録件数

データベース	閾値	観光地登録数(件)
Wikipedia①	1	9,182
Wikipedia②	13.5	4,906
Wikipedia①	27	2,997
Google①	1	13,510
Google②	13.5	8,384
Google③	27	6,050

表 4 アンケート結果

データベース	観光地	知らない	観光地ではない	適合率	観光地該当数	再現率	F値
Wikipedia①	45	43	91	0.331	39	0.411	0.366
Wikipedia②	62	69	49	0.559	32	0.337	0.42
Wikipedia③	65	82	26	0.714	23	0.242	0.362
Google①	40	75	65	0.381	43	0.453	0.414
Google②	66	80	34	0.660	34	0.358	0.464
Google③	62	89	29	0.681	28	0.295	0.411

観光地データベースの評価は再現率(recall)，適合率(precision)および F 値(F-measure)を利用する．適合率は提示した観光地の中にアンケート回答者が観光地だと判断する観光地の割合を表す．

再現率は作成した観光地データベースの中にアンケート回答者が思いついた 95 件の観光地がどれだけ含まれているかを表す．F 値は適合率と再現率の調和平均によって求められる．適合率，再現率，F 値は以下の式で表される．

$$適合率 = \frac{観光地}{観光地 + 観光地ではない} \tag{11}$$

$$再現率 = \frac{回答の内登録済みの観光地数}{回答で得た観光地数} \tag{12}$$

$$F値 = \frac{2 \times 適合率 \times 再現率}{適合率 + 再現率} \tag{13}$$

アンケート結果を表 4 に示す．

閾値を高く設定するほど観光地以外のものが含まれる確率は減少するが観光地の漏れが生じる確率は上昇する．適合率と再現率の調和平均である F 値が一番高い Google②データベースを本システムでは使用する．観光地データベースには観光地名，緯度，経度，県名が登録してある．

4.2. コーパス作成

word2vec はコーパスの量が大きいほど良い精度になる報告がされている[7]．つまり本論文においては観光地に関係のある文書を多く取得することが望ましい．本論文では Yahoo!知恵袋の「地域，旅行，お出かけ」>「国内」カテゴリを対象に観光地データベースに登録された観光地を検索クエリとし，ヒットした質問・

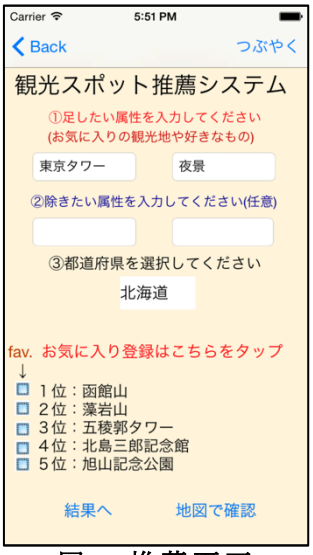

図 5 推薦画面

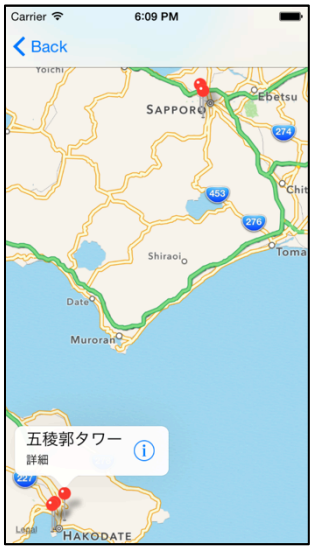

図 6 地図画面

ベストアンサーを取得する．取得した文書は形態素解析エンジン Mecab[6]を使用し文書を品詞ごとに分解する．「東京，タワー」のように観光地が品詞分解されることを防ぐため，Mecab の辞書に観光地データベースに登録した観光地を追加した．

4.3. 学習設定

4.2 節で作成したコーパスをもとに単語の分散表現を作成する．学習設定は表 5 のように設定した．学習時間は約 3 時間で学習後データは約 273MB であった．

5. アプリ実装・動作検証

本観光地推薦システムは iOS アプリケーションとして開発を行った．推薦システムでは前述のようにユーザーの入力と観光地間の類似度を計算することで観光地の推薦を行う．

5.1. 観光地推薦

本システムの入力画面を図 5 に示す．ユーザーは足したい属性と除きたい属性をそれぞれ 2 件まで入力し，結果へボタンを押すことで入力単語のベクトル化・計算とコサイン類似度が行われ，類似度が高い順に 5 件観光地が出力される．その際都道府県を指定することで指定した都道府県に存在する観光地のみが出力される．目的地決定する段階ではユーザーはより自分に合った観光地を発見するため，入力ク

表 5 学習設定

train:学習対象のテキストデータ	534MB
size:次元数	600 次元
windows:ウィンドサイズ	5 単語
hs:階層的ソフトマックス	1(使う)
sample:単語を無視する頻度の閾値	10^{-3}

[6] http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html

表 6 応答時間

	応答時間(s)
メモリ上保持	0.245
メモリ上保持しない	5.959

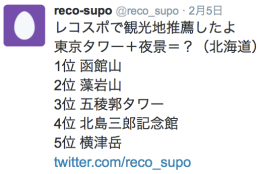

図 7 twitter 投稿

エリを変更しながら複数回利用することが想定される．推薦された観光地をあとでまとめて確認できる機能が必要となると考えられる．そこでお気に入り登録機能を実装した．推薦された観光地の横にあるチェックボックスをタップすることで観光地が登録される．お気に入りに登録した観光地は「お気に入り」メニューからいつでも確認することができる．

5.2. 観光地の地図上マッピング

目的地を決める段階において提示された観光地の場所がどこにあるか確認することは重要である．そこで本アプリケーションでは提示した観光地の場所を地図上で確認できる機能を追加した．地図画面を図 6 に示す．また地図上のピンから出る吹き出しの詳細ボタンを押すことで，その観光地についての情報を直接調べることができる．

5.3. ユーザビリティ

観光地推薦システムでは単に観光地名を提示するだけでなく，使いやすさや利用性などユーザビリティを考えた設計にすることも大事な要素である．ユーザビリティを考えた設計において，入力から推薦までの応答時間短縮も大事な要素である．本システムではコーパスを学習して得た学習後データや観光地データベースの情報を利用して推薦を行う．入力を行うたびに学習後データの読み込みを行うと出力まで時間がかかる．そこで入力前に学習後データや観光地データベースの情報をあらかじめメモリ上に読み込ませておくことで応答時間を短縮した．応答時間の変化を表 6 に示す．Nielsen Norman Group[9]によると人間の操作に対するシステムの反応速度で利用者の思考を止めない限度は 1 秒であるという報告がされている．あらかじめメモリ上にデータを保持させることで 5 秒以上応答時間を短縮し，利用者の思考を止めることのない応答時間を実現した．

5.4. twitter 連携

観光地推薦において単に観光地を提示して終わるのではなく，その後の旅行につながるようなシステム設計も大事な要素である．宿泊を伴う旅行の 84.1%は複数人で行われているという報告[10]がある．観光地推薦結果を自分だけで所有するだけでなく，発信・共有することで観光意欲を促すことができると考えられる．そこで推薦結果を Twitter に投稿できる機能を追加した．ログインにかかる手間を減らすため xcode の標準フレームワークである Twitter.framework を用いた．推薦結果が表示されたあと図 5 の右上にある「つぶやく」ボタンを押すことで図 7 に示すような投稿が自動的に行われる．

6. まとめ

本稿では自分のお気に入りの観光地や好きな観光地の特徴，行きたい観光地のイメージなど，既知の情報から観光地を自動的に推薦するシステムを提案した．本システムではユーザーの自由な入力を可能にするため，単語の分散表現を word2vec で学習・作成した．利用者の入力をベクトル化し，事前に作成した観光地の特徴ベクトルとの類似度計算による評価を行うことで，利用者の入力に沿った観光地推薦を実現した．作成した観光地推薦システムを iOS アプリケーションとして開発・実装を行った．

コーパスから学習を行う際，1 単語を 1 ベクトルで表現する．しかし「赤レンガ倉庫」など同名だが異なる観光地の場合，それぞれを異なる単語ベクトルで表す必要があるが，コーパス上では横浜にある「赤レンガ倉庫」も函館にある「赤レンガ倉庫」も同じ単語として扱われてしまう問題がある．

またアンケートによる推薦システムの精度評価，コーパスや学習設定の最適化はまだ行っていない．これらの問題を解決することが今後の課題である

文 献

[1] 上原 尚，嶋田和孝，遠藤 勉:Web 上に混在する観光情報を活用した観光地推薦システム:電子情報通信学会，言語理解とコミュニケーション研究会 (NLC)，第 4 回集合知シンポジウム，NLC2012-35，pp. 13-18，2012

[2] 松並 政治，清水 明宏:観光キーワード自動推薦システムの研究:平成 20 年度フロンティアプロジェクト

[3] 杉浦 孔明，石橋 直人，芳賀麻誉美，堀智織：階層型評価構造に基づく観光スポット推薦システムの構築と長期実証実験：観光情報学会第 8 回研究発表会，2013-11-30

[4] 開地亮太，檜垣泰彦：LOIS2015-14 潜在的興味に基づく観光地推薦システムの試作，電子情報通信学会技術研究報告,Vol.115,No.138,pp.29-34 (2015-07-13,14 はこだて未来大学)

[5] Peter Turney，Patrick Pantel：From frequency to meaning: Vector space models of semantics：Journal of Artificial Intelligence Research 37， 141–188，2010

[6] Stefan Evert，Marco Baroni，Alessandro Lenci：Distributional Semantic Models：NAACL HLT 2010 Tutorials，Los Angeles，1 June 2010

[7] Tomas Mikolov，Kai Chen，Greg Corrado，and Jeffrey Dean：Efficient Estimation of Word Representations in Vector Space：In Proceedings of Workshop at ICLR，2013.

[8] Andriy Mnih，Yee Whye The：A fast and simple algorithm for training neural probabilistic language models：Proceedings of the 29th International Conference on Ma- chine Learning，pp. 1751–1758 ，2012.

[9] Jakob Nielsen：Website Response Times：June 21,2010

[10] じゃらんリサーチセンター：Press Release：2015-7-28