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
LOIS2016-85(2017-03)

観光地推薦システムへの単語分散表現の適用

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.yasuhiko@faculty.chiba-u.jp

あらまし 本研究の目的は，単語の分散表現を観光地推薦システムに適用する上での最適条件の検討と単語分散表現を適用した観光地推薦システムの有効性を検証することである．本システムは単語ベクトルの作成，データの入力・ベクトル化，類似度計算，観光地抽出・提示の 4 つのプロセスから構成される．本システムにおいて，観光地データベースとコーパスを選定することでさらに精度が上昇すると考え，事前実験による選定を行った．既存の手法との比較実験の結果，成功観光地数が既存手法の 18 ヶ所中 6 ヶ所に比べ，提案手法は 18 ヶ所中 11 ヶ所となり精度の上昇が見られた．また，カイ二乗検定を行ったところ，優位水準 0.1 で提案手法が優位であり，提案手法の有効性を確認した．

キーワード 観光地推薦システム，単語ベクトル，word2vec，自然言語処理

Application of distributed representations of words to tourist spots recommendation system

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 The purpose of this study is to examine the optimum condition for applying distributed representations of words to recommendation system of tourist spots and verify the effectiveness of this system applying word distribution expression. The system consists of four processes: word vector creation, data input and vectorization, similarity calculation, tourist spots extraction and presentation. In order to raise accuracy in this system, we conducted preliminary experiments on tourist spots databases and corpus. As a result of the evaluation experiment, the proposed method has better results than the existing method and the effectiveness was confirmed.

Keywords Tourist Spots Recommendation System, Word Vector, Word2vec, Natural Language Processing

1. 序論

1.1. 研究背景

Web 検索で得られる観光情報は膨大であり，提示された情報から自分の嗜好に合った観光地を見つけ出すことは困難である．こうした背景から観光地推薦に関する研究が行なわれてきた．従来の観光地推薦システム[1]では，単語の出現頻度と逆文書頻度の２つの指標から計算される TF-IDF から観光地の特徴化を行う手法が広く用いられている．しかし，それらの手法は，単語自体の類似性を考慮したものではない．

1.2. 研究目的

そこで本論文では単語の類似性に考慮した観光地推薦システム構築のため，単語分散表現に注目した．近年，ニューラルネットワークによる学習が注目を浴び，単語が持つ意味を保持したままベクトルとして表現できるようになった．これらの手法は自然言語処理の分野で注目されており，情報検索や機械翻訳，レコメンドに応用されている．特にレコメンドにおいては，音楽推薦システム[2]や購入商品の推薦[3]，代替食材の発見手法の提案[4]など様々な応用例があるが，観光地推薦システムにおいてまだ応用されていない．

そこで本研究では，単語の分散表現を観光地推薦システムに適用する上での最適条件の検討と単語分散表現を適用した観光地推薦システムの有効性を検証することを目的とする．

2. 単語分散表現

2.1. 単語の分散表現

自然言語処理において単語が持つ意味情報をベク

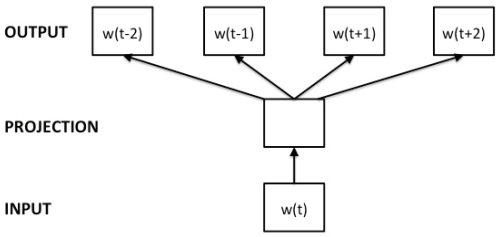

図 1 skip-gram モデル

トルによっ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，似た文脈に存在する単語は似た意味を持つという仮説である Distributional Semantic Models[6]に基づいてる．この考え方に基づき単語の分散表現を大規模文書集合(コーパス)から自動的に学習を行う，単語ベクトル空間モデルが広く用いられてきた．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]が新たに提案され，従来手法に比べ飛躍的に精度が向上した．

2.2. word2vec[1]

word2vec[4]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)を作成するモジュールである．skip-gram モデルの概略図を図 1 に示す．入力コーパスの 1 文($w_1 \dots w_r$から構成される)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．

$$p(w_{t+j}|w_t) = \frac{\exp\left(v'_{w_{t+j}} \cdot {}^{T}v_{w_t}\right)}{\sum_{w=1}^{W}\exp\left(v'_{w} \cdot {}^{T}v_{w_t}\right)} \qquad (2)$$

ここでv_w, v'_wは「入力」「出力」のベクトル表現であり，w, W は文章中の語彙数である．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．したがって周辺の単語の分布が似ていれば単語ベクトルは似た値をとる．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で意味空間上の 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によって可視化したものを図 2 に示す．コーパスは日本語 Wikipedia[2] から収集した質問回答を用いた．図 2 に示すように高い精度をもつ word2vec は自然言語処理の分野で注目されており，情報検索や機械翻訳，感情分析，レコメンドに応用されている．特にレコメンドにおいては，音楽推薦システム[2]や代替食材の発見手法の提案[4]など様々な応用例があるが．観光地推薦システムにおいてまだ応用されていない．

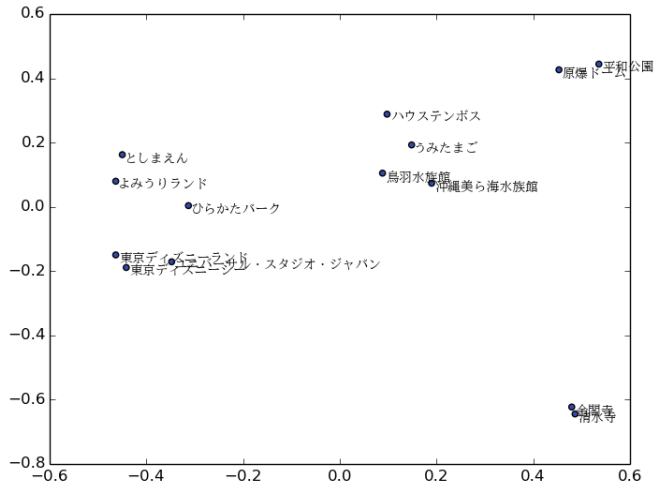

図 2 単語ベクトルの可視化

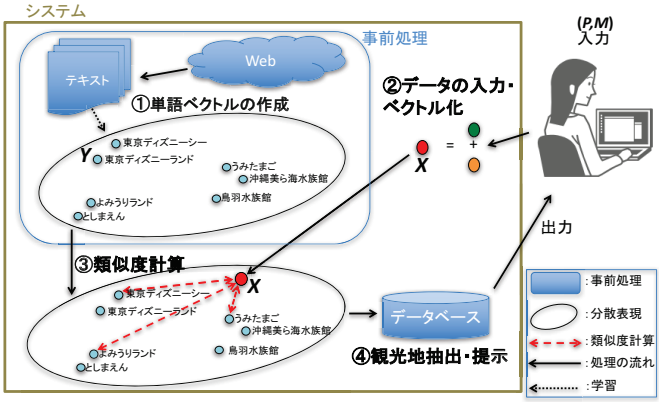

図 3 システム概要

3. 提案手法

本システムで用いる単語ベクトルは 2.2 節で述べた word2vec を使用して作成する．本システムの概要図を図 3 に表す．本システムは大きく分けて，単語ベクトルの作成，データの入力・ベクトル化，類似度計算，観光地抽出・提示の 4 つのプロセスから構成される．

3.1. 単語ベクトルの作成

本システムではまず事前処理として Web から取集したテキストデータを word2vec で学習させることにより単語ベクトルを獲得する．収集したテキストから作成される単語 g の単語ベクトル$\boldsymbol{Y}_g$は v 次元で表すと，

$$\boldsymbol{Y}_g = \{y_{g_1}, y_{g_2}, \dots, y_{g_v}\} \qquad (3)$$

と表せる．作成された単語ベクトルは学習後データセットとして保存し，データの入力・ベクトル化，類似度計算プロセスで使用する．

[1] https://code.google.com/p/word2vec

[2] https://ja.wikipedia.org/wiki/

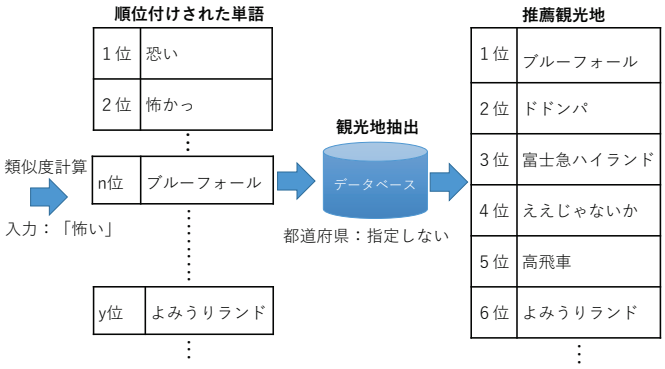

図 4 観光地の抽出・提示

3.2. データの入力・ベクトル化

本システムでは利用者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を入力とし，それを利用者の嗜好情報であると考える．本システムでは利用者の入力を，追加したい特性 n 個と除きたい特性 n 個の２つを想定している．事前処理で作成した単語ベクトルを利用者の入力に適用するとそれぞれの単語ベクトル$\boldsymbol{P_n}, \boldsymbol{M_n}$は，

$$\boldsymbol{P_n} = \{\boldsymbol{p_{n_1}}, \boldsymbol{p_{n_2}}, \dots, \boldsymbol{p_{n_v}}\} \quad (4)$$

$$\boldsymbol{M_n} = \{\boldsymbol{m_{n_1}}, \boldsymbol{m_{n_2}}, \dots, \boldsymbol{m_{n_v}}\} \quad (5)$$

と表せる．よって入力ベクトル$\boldsymbol{X}$は，

$$\boldsymbol{X} = \boldsymbol{P_1} + \boldsymbol{P_2} \dots + \boldsymbol{P_n} - \boldsymbol{M_1} - \boldsymbol{M_2} \dots - \boldsymbol{M_n} \quad (6)$$

と表現できる．

3.3. 類似度計算

本システムでは利用者の入力ベクトルと事前処理で作成された単語ベクトルとの類似度を計算することで推薦を行う．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を用いる．入力ベクトル$\boldsymbol{X}$と事前に作成した単語ベクトル$\boldsymbol{Y_g}$のコサイン類似度 $\cos(\boldsymbol{X}, \boldsymbol{Y_g})$は，

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

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

観光地抽出・提示の模式図を図 4 に示す．単語の分散表現を導入したことで，利用者の入力を観光地に限定することなく自由な入力のシステムを実現できる．しかし図 4 のように，単語ベクトルの特性上，利用者の入力が「怖い」の場合，類似度上位にくる単語は「恐い」「怖かっ」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必要がある．この問題を解決するため本システムでは観光地抽出を導入する．事前に観光地データベースを作成し，類似度で順位付けされた単語から観光地のみを抽出し提示を行う．観光地抽出・提示を導入することで，観光地以外の単語を利用者に推薦することなく観光地推薦を行える．

表 1 Google カテゴリ

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

4. 事前実験

事前実験により，単語の分散表現を観光地推薦システムに適用する上での最適条件の検討を行う．

4.1. 観光地データベース事前実験

本研究では，集合知である Web から観光地候補を収集し，分類することで観光地データベースを作成する．そこで，観光地候補の分類手法を使用した観光地データベースを評価することで，分類手法の最適条件の検討を行う．

4.1.1. 観光地データベース作成

観光地候補収集手段は以下である．

a. 日本語版 Wikipedia の見出し語を収集

b. Google Places API[3] 検索にて任意の緯度経度を入力し，表 1 に示すカテゴリを持つ施設を収集

収集した観光地候補について，英数字や記号だけで構成されている語は不要語とみなし，それらを削除した．Wikipedia から収集した観光地候補は 914,843 語，Google Places API から収集した観光地候補は 170,853 語である．以上の２つの手段で得られた観光地候補を分類することで観光地データベースを作成する．観光地の選定として以下の手段が考えられる．

(1) 緯度経度情報を持っているか

(2) 観光関連語であるか

(3) 観光関連度が閾値以上であるか

観光地候補の分類手法を使用した観光地データベースを評価することで，分類手法の最適条件を検討する．

4.1.2. 評価方法

作成したデータベースと正解データを比べることで観光地データベースの評価を行う．正解データはるるぶ.com[4] に登録してある観光地とした．るるぶ.com に登録されている観光地には居酒屋やレストランなど普通の飲食店も含まれているため，飲食店を持つカテゴリは排除して収集した．得られた正解データであるるるぶ.com データベースの登録観光地数は 17119 件となった．観光地候補の分類手法を使用した観光地データベースの評価は，るるぶ.com データベースと，適合率，再現率および F 値を利用する．適合率，再現率，

[3] https://developers.google.com/places/

[4] http://www.rurubu.com/domestic/

表 2 事前実験(1)

データベース	登録観光地	存在数	F 値
観光地候補の(Wiki)	914,843	4,984	0.0107
緯度経度持ちの Wiki	60,512	3,138	0.0808

表 3 事前実験(2)

データベース	登録観光地	存在数	F 値
観光地候補(Wikipedia)	914,843	4,984	0.0107
観光関連語(Wikipedia)	119,482	3,954	0.0579
観光地候補(Google)	170,853	4,983	0.0530
観光関連語(Google)	23,465	3,272	0.1612

F 値は以下の式で表される.

$$適合率 = \frac{正解データに存在した観光地数}{登録観光地数} \quad (8)$$

$$再現率 = \frac{正解データに存在した観光地数}{るるぶデータベースの登録観光地数} \quad (9)$$

$$F値 = \frac{2 \cdot 適合率 \cdot 再現率}{適合率 + 再現率} \quad (10)$$

4.1.3. 実験結果

(1) 緯度経度を持っているか

実験結果を表 2 に示す. 表 2 より, 緯度経度情報を持つ観光地候補を抽出することで F 値が上昇していることが分かる. よって観光地候補の分類に緯度経度情報を用いることは有効な手段である.

(2) 観光関連語であるか

実験結果を表 3 に示す. 観光関連語は各観光地候補に対し, Yahoo!知恵袋[5]の「地域, 旅行, お出かけ」>「国内」カテゴリ (以下観光カテゴリ) 内で検索を行い, ヒットしたものと定義される. 表 3 より, Wikipedia と Google Places どちらにおいても観光関連語である観光地候補を抽出することで F 値が上昇していることが分かる. よって観光地候補の分類に観光関連語を用いることは有効な手段である.

(3) 観光関連度が閾値以上であるか

観光関連度は観光地候補がどれだけ観光に関係が深いかを表す数値であり, 次の計算から定義される.

観光地候補kを検索クエリとして検索を行い, Yahoo!知恵袋の観光カテゴリでのヒット数J_k, 全カテゴリでのヒット数A_kをそれぞれ取得する. 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,

$$I_{S_k} = \frac{J_k}{S} \quad (11)$$

$$I_{L_k} = \frac{A_k}{L} \quad (12)$$

[5] http://chiebukuro.yahoo.co.jp

表 4 観光地候補と観光関連度

観光地候補k	ヒット件数J_k	ヒット件数A_k	観光関連度R_k
東京スカイツリー	3,693	6,703	29.8
千葉大学	43	9,306	0.249
新江ノ島水族館	369	484	41.2
清水寺	14,807	17,957	44.6
靖国神社	732	12,375	3.20
九州大学病院	2	112	0.965

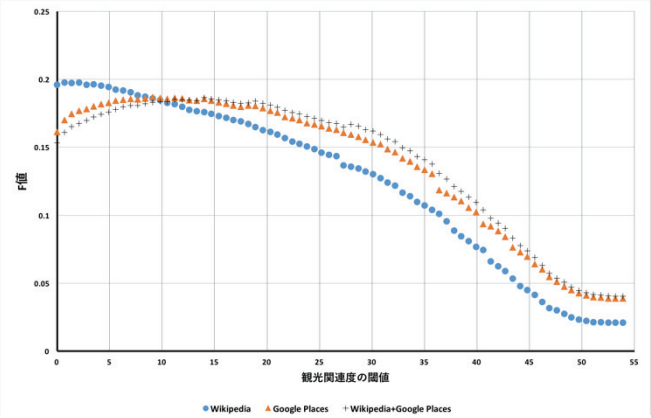

図 5 事前実験(3)

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である観光関連度R_kは

$$R_k = \frac{I_{S_k}}{I_{L_k}} \quad (13)$$

と定義できる. 観光地候補のJ_k, A_k, R_kの値の例を表 4 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 4 より, 観光地と考えられる「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高くなっており, 逆に観光に関係が浅いと考えられる「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.

実験結果を図 5 に示す. 実験には, (1), (2)の分類手法によって分類された単語に対し, 観光関連度の閾値を 0.1 ずつ変化させて作成した観光地データベースを使用した.

図 5 より, 全てのデータベースにおいて最適な観光関連度を設定することで F 値が上昇していることが分かる. よって観光地候補の分類に観光関連度を用いることは有効な手段である.

事前実験により一番精度が高い観光地データベースは Wikipedia から収集した観光地候補を(1)緯度経度を持つ, (2)観光関連語である, (3)観光関連度が 0.9 以

表 5 評価するコーパス詳細

コーパス	コーパス容量	ユニークワード数	学習時間	学習後容量
(1)	6.6GB	161 万単語	34 時間	3.6GB
(2)	4.5GB	42 万単語	9 時間	959MB
(3)	15GB	110 万単語	74 時間	2.5GB
(4)	11GB	178 万単語	338 時間	4.9GB

表 6 実験結果

コーパス	成功数(100 件中)	平均順位
(1)	71	9.70
(2)	75	8.56
(3)	72	9.30
(4)	66	10.56

上である，で分類した時であり，これを本システムの観光地データベースとして採用した．

4.2. コーパス事前実験

word2vec はコーパスの量が大きいほど良い精度になるという報告[4]がされているが，観光地推薦システムにおける最適なコーパスは不明である．そこで複数のコーパスを用意し，それぞれを使用した観光地推薦システムの推薦結果を評価することで最適なコーパスの検討を行う．

4.2.1. コーパス作成

大規模なコーパスが必要であることから評価対象となるテキスト対象を以下の 4 つ選択した．

(1) Wikipedia の全文
(2) Yahoo!知恵袋の観光カテゴリでの質問・回答
(3) Yahoo!知恵袋の全カテゴリでの質問・回答
(4) (1)と(2)を足し合わせたテキストデータ

選択したテキストデータを Mecab[6]で形態素解析することでコーパスを作成する．作成したコーパスを word2vec で学習することにより単語ベクトルを獲得する．word2vec の学習設定は size=600, window size=5, 階層的ソフトマックスは使わない(hs=0)とした．それぞれのコーパスの容量，学習時間、ユニークワード数を表 5 に示す．

表 5 より，コーパス容量が多いほど，ユニークワード数が多いほど学習時間が指数関数的に増加していることが分かる．作成した単語ベクトルを用いた観光地推薦システムの推薦結果を評価することで観光地推薦システムに最適なコーパスの検討を行う．

4.2.2. 評価方法

本実験では，入力に沿った推薦結果が提示されているかで評価を行う．観光地抽出にるるぶ.com データベースを使用し，入力観光地と推薦観光地のカテゴリ情報を比べることで判定を行う．都道府県を指定して観光地を入力し，入力観光地のカテゴリと同じカテゴリを持つ観光地が,何位に提示されているかを取得する．推薦結果の上位 5 位以内に存在すれば期待どおりに観光地が推薦できたとし.成功観光地と呼ぶ.るるぶ.com の観光ランキング[7]から選択された観光地 10 件について 10 都道府県で実験を行い，総成功観光地数と平均順位から評価を行う．実験に使用する都道府県は観光庁が調査した観光入込客統計[9]を基に入込人数が多い順に 10 県選択した．

表 7 実験に用いる観光地名

観光地のカテゴリ	観光地名
山(A)	阿蘇山,富士山,八ヶ岳
温泉(B)	別府,指宿,黒川
ドーム球場(C)	福岡ドーム,東京ドーム,札幌ドーム
美術館(D)	熊本市現代美術館,国立国際美術館,国立西洋美術館
城(E)	熊本城,名古屋城,大阪城
遊園地(F)	ディズニーランド,USJ,スペースワールド

4.2.3. 実験結果

実験結果を表 6 に示す．

表 6 の結果より，全ての指標で(2)のコーパスを用いた推薦結果が優れているということが分かった．そこで本システムでは(2)のコーパスを採用する．

5. 評価実験

提案手法と既存手法の推薦精度を比べることで単語分散表現を適用した観光地推薦システムの有効性を検証する．

5.1. 評価対象

本実験で用いる観光地名を表 7 に示す．表 7 に示した観光地は，山，温泉，ドーム球場，美術館，城，遊園地という 6 種類のカテゴリに属するものを 3 ヶ所ずつ選んである．実験に使用する学習後データセットは 4.2 節で行った事前実験により選択したコーパスから作成したものを用いた。

18 ヶ所の観光地の内 1 ヶ所を利用者の好きな観光地とみなし，その他の 17 ヶ所の観光地を利用者に提案する観光地とみなして，これらの観光地間の類似度を求める．

5.2. 評価方法

既存手法である伊達らの研究[1]に基づき，利用者は好きな観光地と同じカテゴリに観光地を好むという想定で評価する．利用者の好きな観光地と同じカテゴリに属する 2 ヶ所の観光地が，コサイン類似度の大きさの上位 3 位以内に入っているものを期待どおりに観光

[6] http://taku910.github.io/mecab/

[7] http://www.rurubu.com/ranking/Dom Sight.aspx

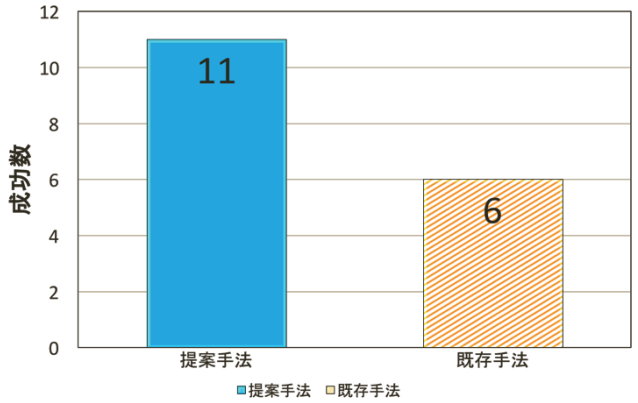

図 6 評価実験

地が推薦できた観光地とし，成功観光地と呼ぶ．既存手法と提案手法の成功観光地数を比べることで提案手法の有効性を検証する．

5.3. 実験結果

実験結果を図 6 に示す．提案手法は 18 ヶ所中 11 ヶ所の観光地で成功し，既存手法の 18 ヶ所中 6 ヶ所に比べて良い結果となった．この結果により提案手法の方が観光地の特性をより捉えた推薦ができていることが分かる．また，カイ二乗検定を行ったところ，優位水準 0.1 で提案手法が優位であることが認められた．

6. 考察

実際のシステム使用例から提案手法の有効性を考察する．システム利用例を表 8 に示す．

一般的な観光地推薦システムでは観光地のみの特徴化を行うことが多い．そのため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．表 8 のように提案手法では，全ての単語に対して特徴化を行っているため，自由な入力で推薦システムを使うことができる．また，足し引きロジックを使用できることも大きな利点である．表 8 のように釣り+海−川を計算することで川に関係の深いものは推薦させず「磯ノ浦」「名護湾」「みんな島」など海に関係の深いものが新たに推薦されている．この結果が妥当かどうかは人によって異なるため，議論の余地があるが，提案手法が観光地の特徴の足し引きロジックができていることの実例であることは確かである．このような足し引きロジックを用いた推薦は他の観光地推薦システムでは見られない．このように，足し引きロジック使って推薦できる提案手法の有効性を見ることができる．

7. まとめ

本研究では，単語の分散表現を観光地推薦システムに適用する上での最適条件の検討と単語分散表現を適用した観光地推薦システムの有効性を検証することを目的とした．本システムにおいて，観光地データベースとコーパスとを選定することでさらに精度が上昇すると考え，事前実験を行った．事前実験により，日本語版 Wikipedia から収集した見出し語を(1)緯度経度を持つ，(2)観光関連語である，(3)観光関連度が 0.9 以上である，の条件で分類して作成した観光地データベースが一番良い結果となったため，本手法に採用した．コーパスは，Yahoo!知恵袋の観光カテゴリから収集したコーパスを用いた場合が一番良い結果となったため，本手法に採用した．評価実験により，既存手法に比べ精度の上昇が見られ，提案手法の有効性が認められた．

今後の課題として，表記ゆれに対応した観光地データベースの作成を行うこと，同名観光地の区別化を行うことが考えられる．

表 8 システム使用例

釣り			釣り ＋ 海 − 川		
順位	観光地	類似度	順位	観光地	類似度
1	鹿島川	0.289	1	果無山脈	0.286
2	片男波海水浴場	0.280	2	片男波海水浴場	0.279
3	門池	0.262	3	磯ノ浦	0.273
4	用宗漁港	0.260	4	名護湾	0.251
5	滝畑ダム	0.252	5	水納島	0.231

文 献

[1] 伊達賢志，北須賀輝明，糸川剛，有次正義:旅先での観光地選び支援のためのブログを用いた観光地の印象抽出手法，マルチメディア,分散協調とモバイルシンポジウム 2011 論文集 2011,1566-1579,2011-06-30.

[2] 和田計也，福田一郎:音楽聴き放題サービス AWA におけるレコメンド手法の検討(artist2vec の試み)，人工知能学会 合同研究会 第 9 回 データ指向構成マイニングとシミュレーション研究会(SIG-DOCMAS2015), 2015.

[3] Mihajlo G, Vladan R, Nemanja D, and Narayan B: E-commerce in Your Inbox: Product Recommendations at Scale, KDD'15, (2015). (著書，編書の場合) 著者名，書名，編者名，発行所，発行都市名，発行年.

[4] 野沢健人，中岡義貴，山本修平，佐藤哲司:word2vec を用いた代替食材の発見手法の提案，電子情報通信学会技術研究報告. DE, データ工学 114(204), 41-46, 2014-09-03.

[5] Peter Turney, Patrick Pantel:From frequency to meaning:Vector space models of semantics, Journal of Artificial Intelligence Research 37, 141-288, 2010.

[6] Harris:Distributional structure, Word.10(23), 146-162.1054.

[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] Mnih, A. and Teh, Y. W.: 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] 観光庁 統計情報:観光入込客統計,平成 26 年.