Vol.2020-GN-109 No.14
Vol.2020-CDS-27 No.14
Vol.2020-DCC-24 No.14
2020/1/23

自由記述文にもとづく観光スポットレコメンドシステムの提案

長尾将宏[1]　草野大智[2]　園田泰子[1]　本橋洋介[1]

概要：近年，来日外国人観光客の増加にともない，観光客ごとにスポットを推薦するニーズが高まっている．従来の観光スポットレコメンドシステムの課題として，レコメンドが有名スポットに偏ってしまうこと，観光客の多様な要望を反映することが難しいことが挙げられる．本稿では，自由記述文を入力とすることで多様な要望を反映した観光スポットのレコメンドを行うシステムを提案する．また，実際に自由記述文にたいして適切に観光スポットをレコメンドできるか評価実験を行う．

キーワード：推薦システム，自由記述文，自然言語処理，観光スポット

Proposal of sightseeing spots recommendation system based on free description sentences

MASAHIRO NAGAO[†1]　DAICHI KUSANO DAICHI[†2]
YASUKO SONODA[†1]　YOSUKE MOTOHASHI[†1]

1. はじめに

近年，来日外国人観光客の増加にともない，観光客ごとにスポットを推薦するニーズが高まっている．一般的に観光客は旅行ガイドや，おすすめサイトなどの情報をもとに観光するスポットを決定する．しかしながら，これらに掲載されているのは人気の観光スポットであることが多く，人気の観光スポット以外に自身の要望にあった観光スポットを見つけることは難しい．

本稿では，これらの多様な観光客の要望を自由記述文として入力してもらうことで，最適な観光スポットのレコメンドを行うシステムを提案する．

2. 関連研究

本章では，観光スポットをレコメンドする関連研究/ツールについて述べる．

1 つ目は，観光客の属性情報などをもとにクラスタリングを行い，クラスタごとに観光スポットをレコメンドする方法である．Laura ら[1]は，観光客の属性や興味関心情報をもとに，観光客を 8 クラスタに分け，クラスタごとの観光ルートをレコメンドする手法を提案している．Konomy[2]では，旅行での価値観に関する 7 つの質問に答えることで，観光スポットのレコメンドを行う．これらの手法では，観光客の属性などに応じて様々な観光スポットをレコメンドすることができるが，質問などで価値観や属性情報を入手する必要がある．

2 つ目は，入力した観光スポット名から類似する観光スポットをレコメンドする方法である．開地ら[3]は，単語分散表現に基づき観光スポットをレコメンドすることが有効であると示している．鎌倉 NAVITIME Travel[4]では，ユーザーが入力した観光スポットに類似する観光スポットをレコメンドする．これらの手法では，観光客が観光スポットを入力するだけで簡便に類似スポットを知ることができるが，観光スポット名にたいしてレコメンドされる内容が同一となるため，検索結果の多様性に課題がある．

本研究では，観光客の属性情報等を必要としない，自由記述文に応じた多様性のある観光スポットのレコメンドを行うシステムを提案する．

3. 観光スポットレコメンドシステム

本章では，観光スポットレコメンドシステム（以下，本システム）の構成を説明する．本システムは，図 1 のとおり 2 つの要素から構成される．

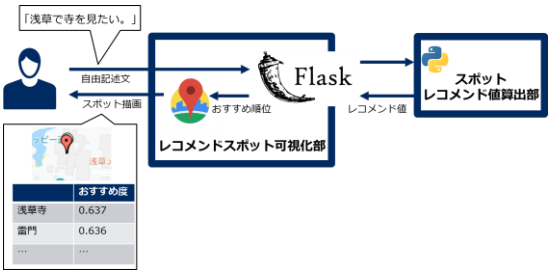

図 1　システム構成
Figure 1　System Architecture

1 NEC
2 滋賀大学

Vol.2020-GN-109 No.14
Vol.2020-CDS-27 No.14
Vol.2020-DCC-24 No.14
2020/1/23

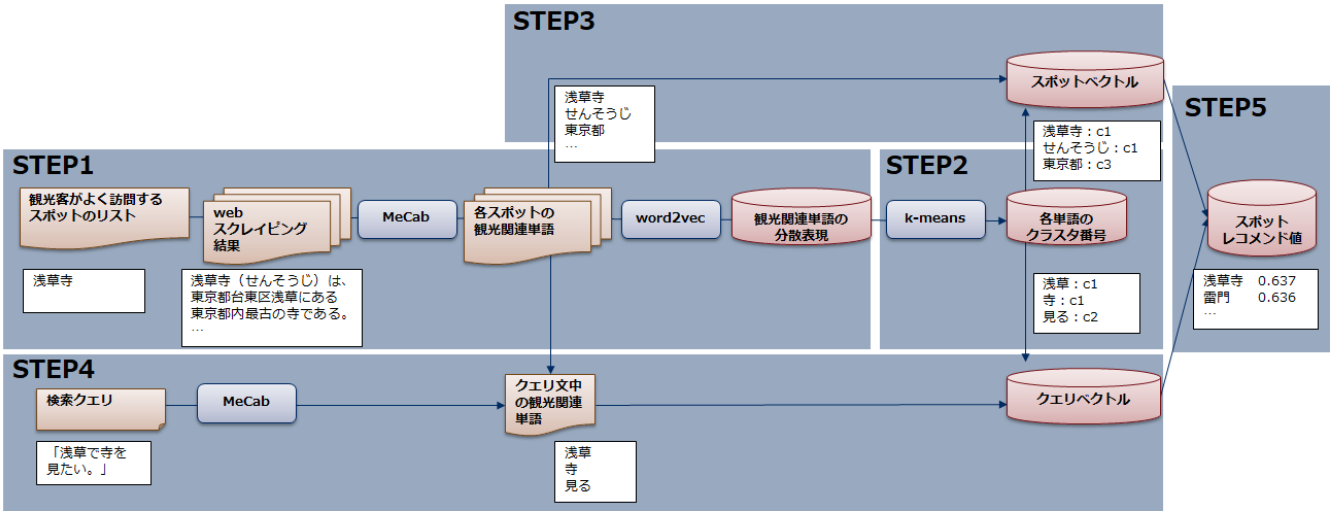

図 2 スポットレコメンド値出力手順

Figure 2 Calculation procedure of scores for spot-recommendation

① スポットレコメンド値算出部
観光客の入力した自由記述文から得たクエリベクトルと，観光スポットごとのスポットベクトルの類似度をスポットレコメンド値として算出する．

② レコメンドスポット可視化部
①で算出したスポットレコメンド値をもとに，値が高いスポットを地図上に可視化する．

次章以降，各部の構成を説明する．

4. スポットレコメンド値出力部

スポットレコメンド値出力部は，スポットの推薦度合として，スポットレコメンド値を出力する．スポットレコメンド値は，スポットベクトルとクエリベクトルの類似度として算出される．

スポットベクトルは，観光関連単語の分散表現および単語のクラスタリング結果から算出する．

クエリベクトルは，ユーザーの自由記述文における単語に対してスポットベクトルと同様の処理を行い算出する．以下，スポットレコメンド値を計算する手順を記す．図 2 はその手順を図示したものである．

STEP1. 観光関連単語の分散表現

後述する単語クラスタリングのために，まず観光関連単語の分散表現を作成する．観光関連単語の分散表現は以下の手順により作成する．

1. 観光客がよく訪問する 2000 個の観光スポット名を抽出する．今回は東京近郊の観光スポット名に限定して抽出した．
2. 観光スポット名それぞれについて，それを説明する文章を web スクレイピングにより取得する．これをスポットiに対する文章S_iとする．
3. MeCab[5]により形態素解析をし，各単語にたいして word2vec[6]によるベクトル化を行う．word2vec の設定は表 1 のとおりである．

表 1 word2vec 設定

Table 1 Configuration for word2vec

対象品詞	名詞，形容詞，形容動詞，動詞
単語の最小頻度	5
埋め込み次元数	200
周辺単語数	10

STEP2. 単語クラスタリング

前節で獲得した観光関連単語の分散表現に基づき，各単語を k-means によりクラスタリングする．k-means の設定は表 2 のとおりである．

表 2 k-means 設定

Table 2 Configuration for k-means

単語数	9000
クラスタ数（C）	95
距離	ユークリッド距離

STEP3. スポットベクトル算出

文章S_iを表現するベクトルをスポットベクトル$\boldsymbol{s_i}$とし，次式で定義する．

$$\boldsymbol{s_i} = \begin{bmatrix} n_1 \\ \vdots \\ n_C \end{bmatrix}$$

n_j ：*文章S_iに含まれる単語のうちクラスタjに含まれる単語数*

例えば，浅草寺の検索結果にクラスタ 1 の単語が 5 つ，クラスタ 2 の単語が 3 つだけが含まれる場合，浅草寺のスポットベクトルは次式で表される．

Vol.2020-GN-109 No.14
Vol.2020-CDS-27 No.14
Vol.2020-DCC-24 No.14
2020/1/23

$$s_{浅草寺} = \begin{bmatrix} 5 \\ 3 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

STEP4.　クエリベクトル算出

観光客が入力する自由記述文（クエリ）Qを MeCab により形態素解析を行い，観光関連単語（STEP1）を抽出，各観光関連単語のクラスタ番号（STEP2）を得る．クエリQを表現するベクトルをクエリベクトル$\boldsymbol{q}$とし，次式で定義する．

$$\boldsymbol{q} = \begin{bmatrix} n_1 \\ \vdots \\ n_C \end{bmatrix}$$

n_j：*検索クエリQに含まれる単語のうちクラスタjに含まれる単語*

STEP5.　スポットレコメンド値算出

スポットベクトル$\boldsymbol{s_i}$とクエリベクトル$\boldsymbol{q}$の類似度c_iを次式のコサイン類似度で算出する．これをスポットレコメンド値とする．

$$c_i = \frac{\boldsymbol{s_i} \cdot \boldsymbol{q}}{|\boldsymbol{s_i}||\boldsymbol{q}|}$$

以上のステップで作成したスポットレコメンド値を，すべての観光スポットに対して算出し，出力する．

5.　レコメンドスポット可視化部

前章の処理により得られたスポットレコメンド値をもとに上位のスポットをレコメンドスポットとして可視化する．可視化には地図アプリサービス（Google Maps API[7]）を用いる．

6.　動作検証と考察

作成したシステムの動作を検証するために，表 3 のクエリに対して実験を行った．

表 3　クエリ一覧

Table 3　List of queries

クエリ番号	クエリ
Q_1	浅草で寺を見たい。
Q_2	原宿でタピオカを飲みたい。
Q_3	鎌倉で大仏を見てから、横浜、東京で観光したい。
Q_4	大人っぽい落ち着いた雰囲気のところ。

なお，レコメンド可視化結果図における赤ピンはスポットレコメンド値上位 10 位のスポットを表しており，灰色ピンは上位 11～40 位のスポットを表している．

6.1　動作検証結果

6.1.1　クエリQ_1へのレコメンド結果

クエリQ_1（浅草で寺を見たい。）へのレコメンド結果は表 4，図 3 のとおりである．「浅草寺」「雷門」「伝法院通り」などの浅草周辺の寺に関するスポットを適切にレコメンドすることができた．

この結果から”浅草”，”寺”の単語から「雷門」や「伝法院通り」が近くなるように学習ができていることがわかる．

表 4　Q_1へのレコメンド結果（上位 5 スポット）

Table 4　Result of recommendation for Q1

スポット名	スポットレコメンド値
伝法院通り	0.683
鏡商店（伝報院通り）	0.663
二天門通り	0.660
浅草寺	0.637
雷門	0.636

図 3　Q_1へのレコメンド可視化結果

Figure 3　Visualization of results of recommendation for Q1

6.1.2　クエリQ_2へのレコメンド結果

クエリQ_2（原宿でタピオカを飲みたい。）へのレコメンド結果は表 5，図 4 のとおりである．タピオカのキーワードから実際にタピオカを販売している「カワイイ モンスターカフェ」「カフェクレープ（ラフォーレ原宿店）」（以下，タピオカ店）をレコメンドすることができた．

“原宿で飲みたい．”のクエリには居酒屋関連のお店が多く出たことから，”タピオカ”とタピオカ店を関連付けたことがわかるが，タピオカ店の web スクレイピング結果に”タピオカ”の単語は含まれていなかった．関連付けることができたのは k-means による単語クラスタリングが適切だったためといえる．実際に，”タピオカ”と同じ単語クラスタには，”シュークリーム”, ”ストロベリー”, ”アイスソフト”, ”

Vol.2020-GN-109 No.14
Vol.2020-CDS-27 No.14
Vol.2020-DCC-24 No.14
2020/1/23

チョコレート”などタピオカ店のスクレイピング結果に多く含まれる単語が属していた．

表 5　Q_2へのレコメンド結果（上位 5 スポット）

Table 5　Result of recommendation for Q2

スポット名	スポットレコメンド値
カワイイ モンスター カフェ	0.518
えん（渋谷店）	0.484
カフェクレープ（ラフォーレ原宿店）	0.430
自然派グリルバル 原宿 Hutte	0.422
ナポリス ピッツァ&カフェ（渋谷センター街）	0.417

図 4　Q_2へのレコメンド可視化結果

Figure 4　Visualization of results of recommendation for Q2

6.1.3　クエリQ_3へのレコメンド結果

クエリQ_3（鎌倉で大仏を見てから、横浜、東京で観光したい。）へのレコメンド結果は表 6，図 5 のとおりである．「鎌倉大仏」「横浜赤レンガ倉庫」「東京国立近代美術館」と特定の地域に限らず，東京・横浜・鎌倉のそれぞれの観光名所をレコメンドすることができた．

しかしながら，本システムでは係り受け解析を行っていないため”横浜で大仏を見てから，鎌倉，東京で観光したい．”としても同様の結果が得られてしまうという問題があることが確認された．

表 6　Q_3へのレコメンド結果（一部）

Table 6　Result of recommendation for Q3

順位	スポット名	スポットレコメンド値
3	鎌倉大仏（高徳院）	0.620
…	…	…
7	横浜赤レンガ倉庫	0.563
…	…	…
30	東京国立近代美術館	0.424

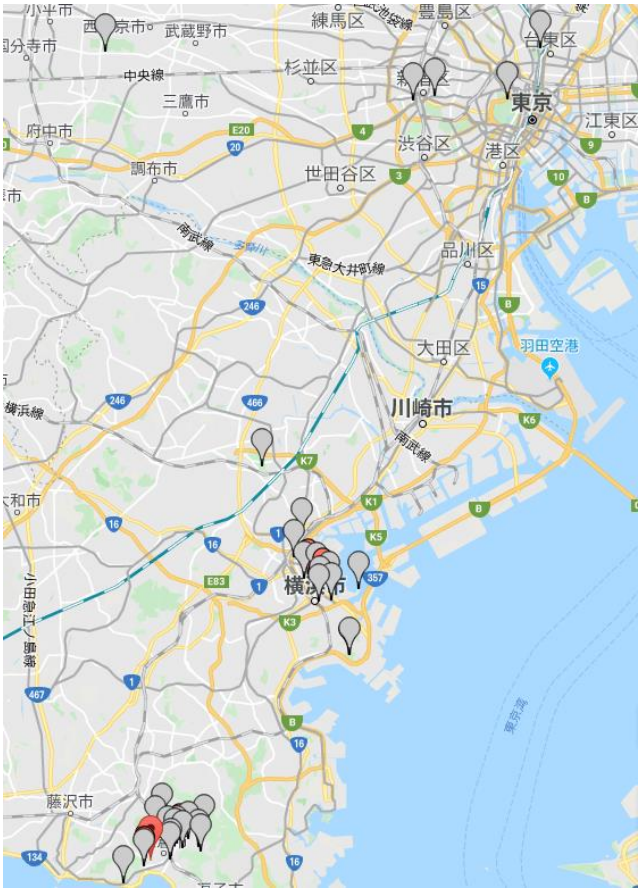

図 5　Q_3へのレコメンド可視化結果

Figure 5　Visualization of results of recommendation for Q3

6.1.4　クエリQ_4へのレコメンド結果

クエリQ_4（大人っぽい落ち着いた雰囲気のところ。）へのレコメンド結果は表 7，図 6 のとおりである．”大人っぽい”・”落ち着いた雰囲気”という曖昧な表現の単語にも適切にレコメンドができていることがわかる．実際に，「スプリングバレーブルワリー東京」は代官山にある 6 種類のクラフトビールが楽しめる成人向けのお店であり，「伝統工芸青山スクエア」は焼き物や織物等を扱う専門店である．その他の上位 5 スポットに関しても”大人っぽい”・”落ち着いた雰囲気”に当てはまるお店といえる．

表 7　Q_4へのレコメンド結果（上位 5 スポット）

Table 7　Result of recommendation for Q4

スポット名	スポットレコメンド値
スプリングバレーブルワリー東京	0.564
伝統工芸 青山スクエア	0.550
金魚坂	0.523
和雑貨 渋谷丸荒渡辺	0.516
まいか 米香（アトレ上野店）	0.503

Vol.2020-GN-109 No.14
Vol.2020-CDS-27 No.14
Vol.2020-DCC-24 No.14
2020/1/23

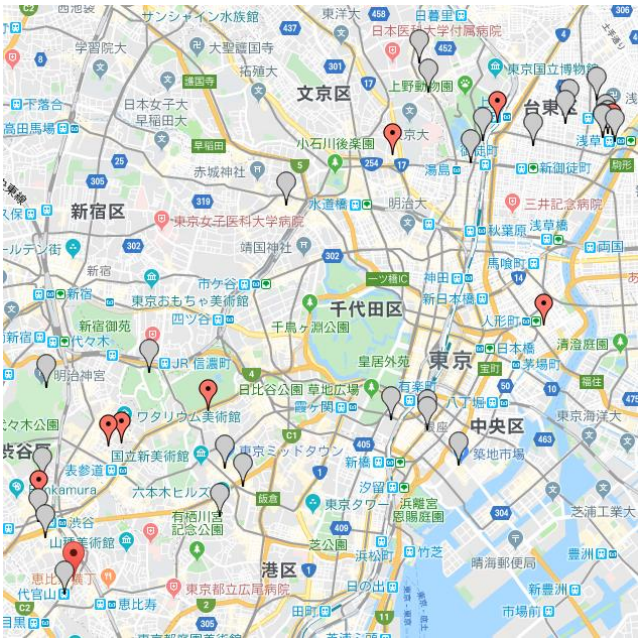

図 6　Q_4へのレコメンド可視化結果
Figure 6　Visualization of results of recommendation for Q4

6.2　考察

以上の結果から，本システムは今回実験した 4 件の検索クエリにたいして，それぞれ適切なレコメンドができていると言える．また，各クエリの結果から次の 4 点が言える．

- クエリQ_1の結果から，“浅草”＋”寺”→「雷門」などの単語を組み合わせたレコメンドが可能なこと
- クエリQ_2の結果から，タピオカ店のスクレイピング結果に”タピオカ”の単語が存在しない場合でも，類似単語をタピオカ店に紐付け，レコメンドが可能なこと
- クエリQ_3の結果から，複数都市の観光スポットにたいするレコメンドが可能なこと
- クエリQ_4の結果から，“大人っぽい”・”落ち着いた雰囲気”といった曖昧な表現の単語にたいしてレコメンドが可能なこと

7.　まとめと今後の課題

本稿では，観光スポットの特徴を単語の分散表現とクラスタリングにより学習し，自由記述文に対して適切な観光スポットのレコメンドを行うシステムを開発した．また動作検証実験を行い，今回実験した 4 つのクエリに対して有効性を確認することができた．

今後の課題として，3 点挙げられる．

1 点目は，web スクレイピング結果に存在しない単語を考慮できていない点である．クエリ中の単語が web スクレイピング結果に含まれない場合は，その単語を無視する実装になっている．これによりレコメンド精度が下がっている可能性がある．解決策として，次の手順が考えられる．

1. web スクレイピング結果に含まれない単語（w）について Wikipedia などの学習済み word2vec モデル（$Model_{wiki}$）でのベクトル（v_w）を取得する
2. $Model_{wiki}$においてv_wと距離が近く，web スクレイピング結果に存在する単語（w^*）を取得する
3. w^*のベクトル/単語クラスタリング結果をwのベクトル/単語クラスタリング結果とする

2 点目は，「横浜で大仏を見てから，鎌倉，東京で観光したい.」のような複雑な文にたいして正しくレコメンドすることができない点である．解決策として，自由記述文における係り受け解析を行い，係り受け関係にある単語間のベクトル距離が離れすぎている場合に無関係な話題，あるいは矛盾する内容としてクエリベクトル算出から除外するなどが考えられる．

3 点目は，本システムが日本語しか入力できない点である．観光スポットレコメンドシステムは特に外国人観光客向けに必要と考えている．解決策は，英語のスポットベクトル，スポットレコメンド値を算出することである．これらの算出方法は，日本語に依拠しないため，言語を切り替えるなどの web ページ上のインターフェースを実装することで，英語の場合も同様の手順で観光スポットをレコメンドすることが可能である．

今後，これらの改善を行うと共に，大規模なユーザー評価実験を実施し，より良い観光レコメンドシステムを開発していきたい．

参考文献

[1] Laura Martínez García. A user modeling approach to personalized sightseeing tours. Proceeding Interacción '17 Proceedings of the XVIII International Conference on Human Computer Interaction, Article No. 47.
[2] Konomy AI travel consultant | WOW U - EXest. https://www.exest.jp/konomy, (参照 2019-12-17).
[3] 開地亮太，檜垣泰彦．観光地推薦システムへの単語分散表現の適用．信学技報, vol. 116, no. 488, LOIS2016-85, pp. 129-134, 2017 年 3 月.
[4] 観光ガイドアプリ『鎌倉 NAVITIME Travel』提供開始｜ナビタイム. http://corporate.navitime.co.jp/topics/pr/201702/21_4061.html, (参照 2019-12-17).
[5] MeCab. https://taku910.github.io/mecab/, (参照 2019-12-17)
[6] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, pages 3111–3119, 2013.
[7] Google Maps API. https://developers.google.com/maps/?hl=ja, (参照 2019-12-17)