2023 年度

卒　業　論　文

題　目　大規模言語モデルを用いた
観光地レビューに基づく
観光地推薦手法に関する研究

学籍番号　20A3009

氏　名　飯塚　柾

提出月日　1 月 31 日

指導教員　二宮　洋

湘南工科大学工学部情報工学科

概要

本研究では，大規模言語モデルを用いて事前に大量の文章で学習した大規模言語モデルに観光地レビューを入力することで観光地をベクトルで表し，観光地を推薦する手法を実装する．

観光地を推薦する手法の例を挙げると，Web 上から観光情報を抽出することで観光地をベクトルで表し，知恵袋・ブログ上での共起キーワード，時系列分布，知恵袋上でのカテゴリ構造，観光地周辺の施設情報，地図画像を基に生成した複数の特徴ベクトルから観光地ベクトルを生成する．そこから得られた観光地ベクトル同士のコサイン類似度を評価することで、観光地を推薦するシステム[1]がある．しかし，この手法では生成された観光地ベクトルが定量的な情報源だけで形成されているという問題点がある．

これらの問題点を解決するために，観光情報サイトであるじゃらんから観光地レビューを抽出し，観光地レビューを学習した Distributed Bag of Words version of Paragraph Vector(PV-DBOW)[2]を用いて観光地の特徴ベクトルを生成する．得られた複数の特徴ベクトルから観光地間のコサイン類似度を評価し，観光地を推薦する手法[3]がある．しかし，PV-DBOW へ予め膨大な量の観光地レビューを学習させなければならない．また，推薦の精度が学習の結果次第となってしまう．さらに，所望する都道府県とは異なる観光地が推薦されてしまうという問題点がある．

提案手法では，先行研究において PV-DBOW の学習が必要であるという問題に対して事前学習済みモデルを用いることで解決している．

実験として，海遊館ベクトルから大阪府ベクトルを引き，沖縄県ベクトルを足した合成ベクトルと全ての観光地ベクトルとのコサイン類似度が最も高い観光地は沖縄美ら海水族館となり，沖縄美ら海水族館を海遊館に似た沖縄県の観光地として推薦している．提案手法においてベクトル演算を行った結果，沖縄美ら海水族館を海遊館に似た沖縄県の観光地を推薦することを確認した．

次に，提案手法と推薦結果の比較を行った．まず，先行研究と提案手法どちらも同じ条件で新宿御苑ベクトルから東京都ベクトルを引き，大阪府ベクトルを足した合成ベクトルと全ての観光地ベクトルとのコサイン類似度を測定した．しかし，等倍の都道府県ベクトルを用いた演算結果では所望する大阪府の公園・庭園カテゴリに属する観光地を表すことができなかった．よって，都道府県ベクトルの要素を定数倍して演算を行った．結果として，所望する大阪府の公園・庭園カテゴリに属する観光地を表すことができた．

このように，提案手法では観光地レビューを入力することで観光地をベクトルで表し，観光地を推薦する．

目次

第 1 章　序論

近年，情報技術の普及により，容易かつ迅速に大量の情報を得ることができ，このようなビッグデータの活用が注目されている．特に，観光情報については，各地の観光施設や公共団体から公開されている情報や，Q&A サイトでのお薦め観光地の質問・回答，ウェブマッピングサービスのレビューなど，多種多様な情報が存在する．しかし，これらから得られる情報量は膨大であり，利用者が自分の好みや嗜好に合った観光地を迅速に見つけることは容易ではない．

このような背景から，観光地推薦システムの開発が盛んに行われている．例を挙げると，Web 上から観光情報を抽出することで観光地をベクトルで表し，知恵袋・ブログ上での共起キーワード，時系列分布，知恵袋上でのカテゴリ構造，観光地周辺の施設情報，地図画像を基に生成した複数の特徴ベクトルから観光地ベクトルを生成する．そこから得られた観光地ベクトル同士のコサイン類似度を評価することで、観光地を推薦するシステム[1]がある．しかし，この手法では生成された観光地ベクトルが定量的な情報源だけで形成されているという問題点がある．

これらの問題点を解決するために，観光情報サイトであるじゃらんから観光地レビューを抽出し，観光地レビューを学習した PV-DBOW[2]を用いて観光地の特徴ベクトルを生成する．得られた複数の特徴ベクトルから観光地間のコサイン類似度を評価し，観光地を推薦する手法[3]がある．しかし，PV-DBOW へ予め膨大な量の観光地レビューを学習させなければならない．また，推薦の精度が学習の結果次第となってしまう．さらに，所望する都道府県とは異なる観光地が推薦されてしまうという問題点がある．

そこで，本研究では事前に大量の観光地レビュー文章で学習した大規模言語モデルに観光地レビューを入力することで観光地をベクトルで表す手法を提案する．[3]と比較すると，推薦の精度は事前学習された大規模言語モデルの精度が高ければ高くなる．また，事前学習された大規模言語モデルを使うだけなので学習させる必要がないという利点がある．

実験として事前に大量の観光地レビュー文章で学習した大規模言語モデルに観光地レビューを入力することで得られた観光地ベクトル同士の演算を行い，全ての観光地ベクトルとのコサイン類似度を測定する．また，[3]と提案手法の観光地推薦結果の比較を行う．

第２章　基盤技術

本章では，本研究で実装する観光地推薦システムの基盤技術となるニューラルネットワーク，Word2vec，PV-DBOW，大規模言語モデルについて説明する．

第２章１節　ニューラルネットワーク[4]

生物の脳内には，何億個もの神経細胞がネットワーク上に接続されている．神経細胞であるニューロンの 1 つ 1 つは，大きく分けて「細胞体」「軸索」「樹状突起」という 3 つのパーツから成り立っている．軸索と樹状突起が接触している部分を「シナプス」と言い，ニューロンは，他の複数のニューロンの「軸索」から発信した信号にシナプスを介して「樹上突起」で受け取る．そして，細胞体で入力が加算され、ある閾値を超えたときに軸索を通じて他のニューロンへ信号を送る．ニューラルネットワークは，このようなシナプスの働きを数式でモデル化されたパーセプトロンを組み合わせることで構成されている．パーセプトロンの構造を(1)，図 1 に示す．

$$y = f\left(\sum_{i=0}^{k} w_i x_i + b\right) \tag{1}$$

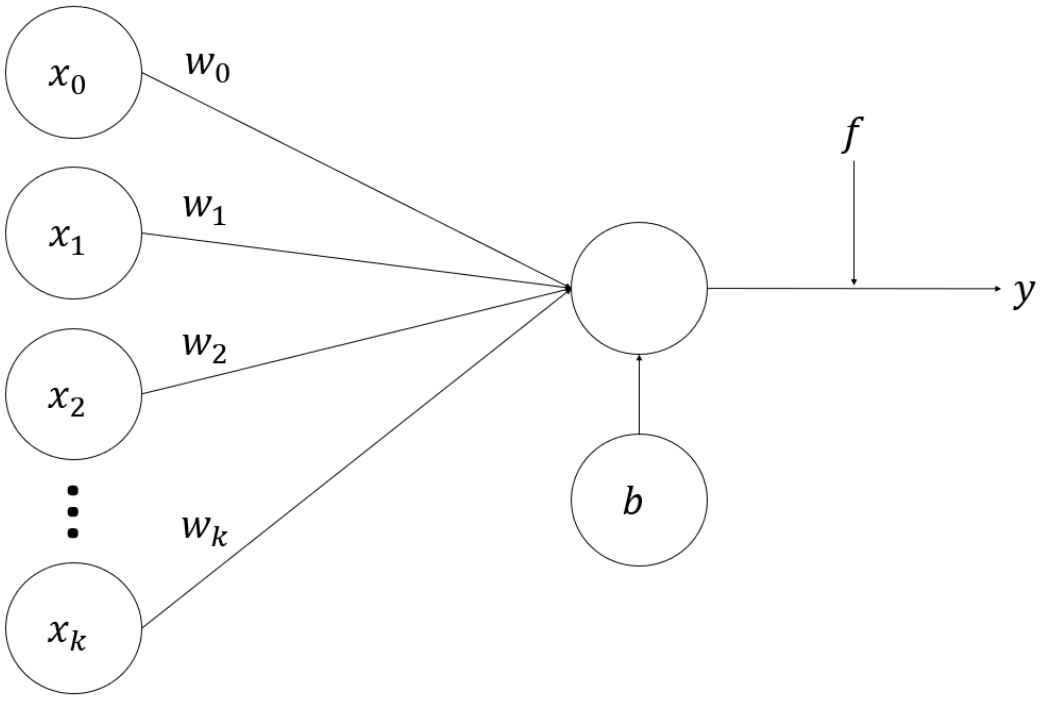

図 1　パーセプトロンの構造

ここで，yはパーセプトロンの出力，kはパーセプトロンへの入力の数，x_iは入力のi番目の要素，w_iは入力のi番目の要素に対応する重み，bは閾値，fは活性化関数である．活性化関数とは，重みづけされた入力値の総和を出力値に変換するための関数となっている．パーセプトロンの実装では入力が 0 を超えたら 1 を出力し，それ以外は 0 を出力するステップ関数が使用されている．ステップ関数を(2)，図 2 に示す.

$$f(x) = step(x) = \begin{cases} 0(x \leq 0) \\ 1(x > 0) \end{cases} \tag{2}$$

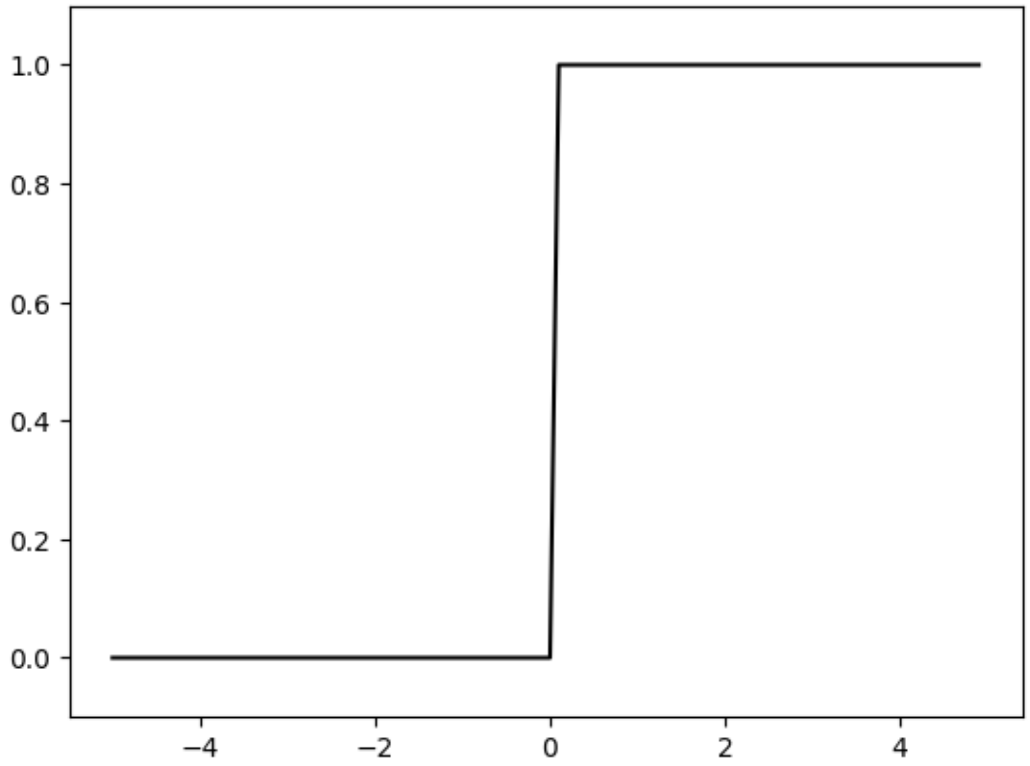

図 2　ステップ関数

パーセプトロンの実装で使用されるステップ関数は 0 を境にして出力が 0 から 1，または 1 から 0 になるため，入力に対して 0 か 1 の二値の信号しか流れない．よって，非線形問題を解くことができない．この問題点を解決するために，ニューラルネットワークでは入力に対して連続的な実数値に出力が変化するシグモイド関数が使用されている．シグモイド関数を(3)，図 3 に示す.

$$f(x) = sigmoid(x) = \frac{1}{1+e^{-x}} \tag{3}$$

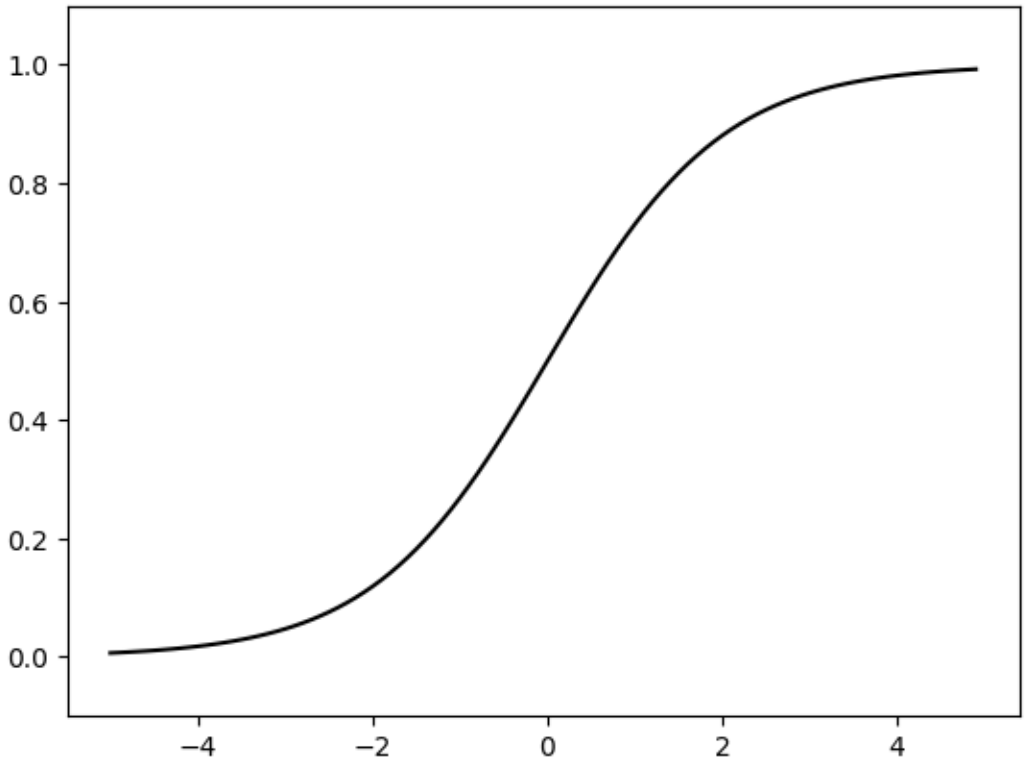

図 3　シグモイド関数

非線形問題を解くために，複数のパーセプトロンを縦方向と横方向に組み合わせることで，ニューラルネットワークを作成することができる．ニューラルネットワークの例を図 4 に示す．(4)にi層目のj番目のニューロンの出力を示す．

$$y_{i,j} = f\left(\sum_{k=0}^{K} y_{i-1,k} \cdot w_{i,j,k} + b_{i,j}\right) \tag{4}$$

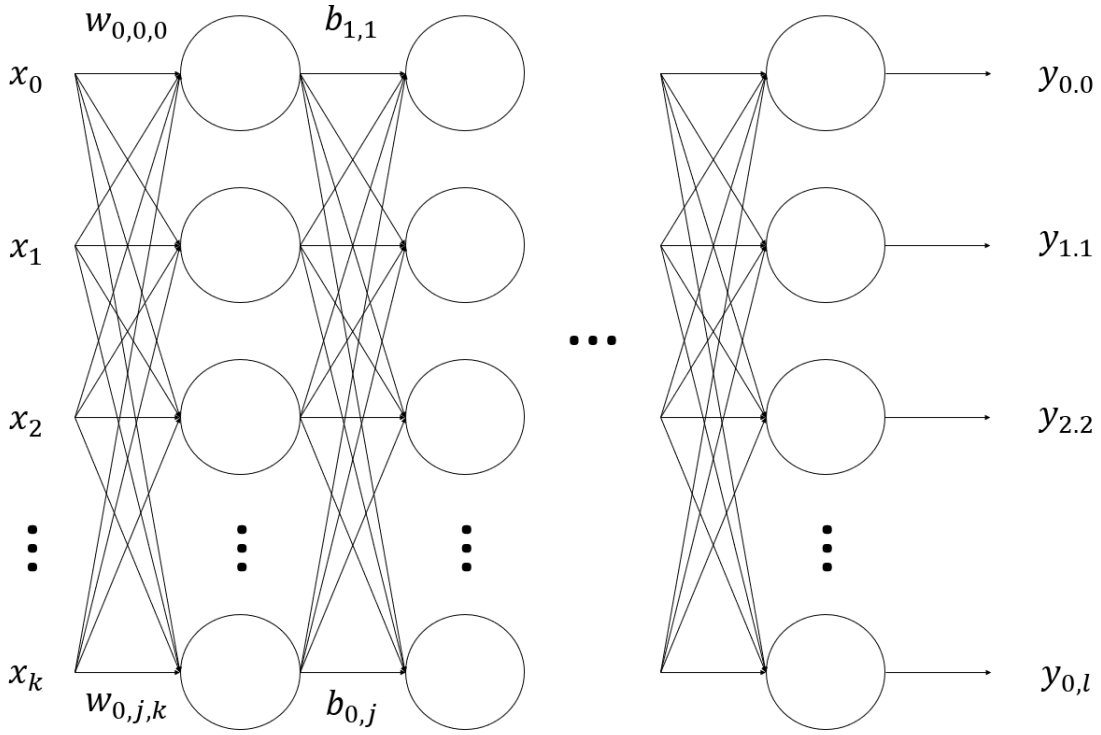

図 4　ニューラルネットワークの例

ここで，$y_{i,j}$はi層目のj番目のニューロンの出力，$w_{i,j,k}$はi層目のj番目のニューロンのk番目の要素に対する重み，$b_{i,j}$はi層目のj番目のニューロンの閾値，Kは$i-1$層目のニューロン数である．また，$y_{0,k}=x_k$である．

学習データが膨大な場合，バッチ学習は学習データ全てを一度に学習するため，計算量やメモリの要件が高くなるという問題がある．この問題を解決するため，学習データ全てをいくつかに分割して学習を行うミニバッチ学習と，学習データを 1 つずつ用いて学習を行うオンライン学習という 3 つの手法が存在する．バッチ学習，ミニバッチ学習，オンライン学習の概要を図 5 に示す．

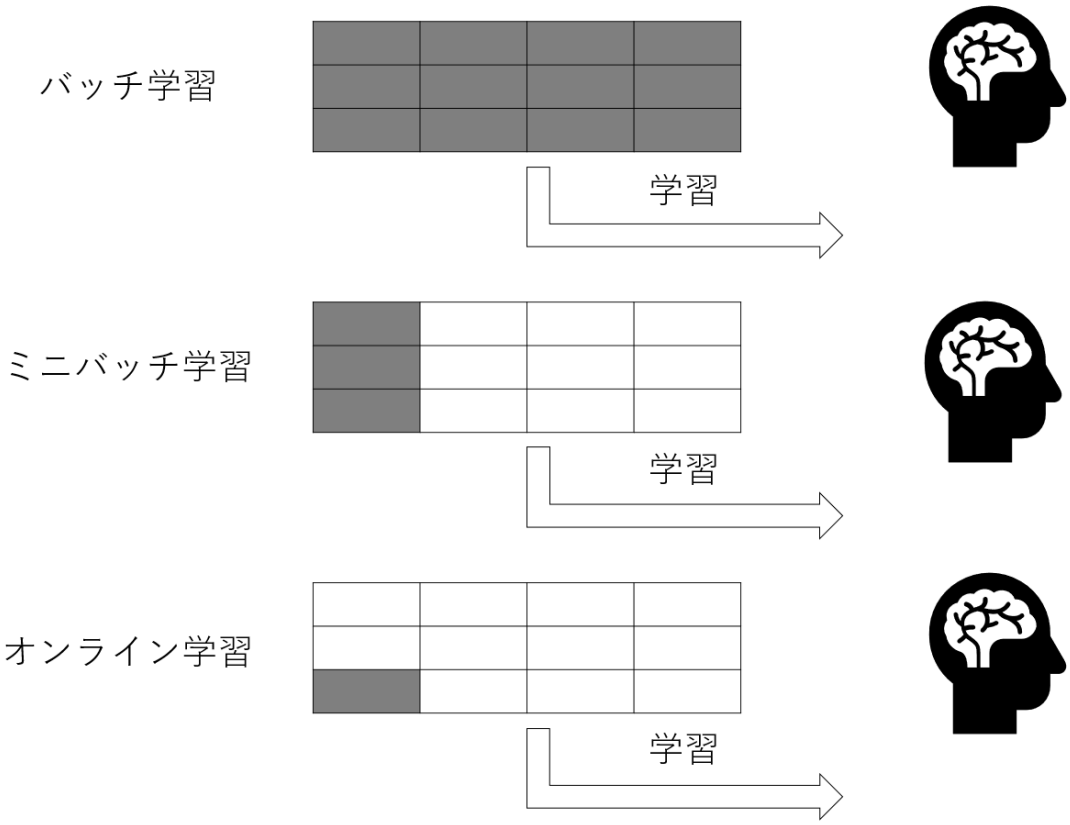

図 5　バッチ学習，ミニバッチ学習，オンライン学習の概要

第２章２節　Word2vec

単語の特徴ベクトルを生成する Word2vec の例を図 6 に示す.

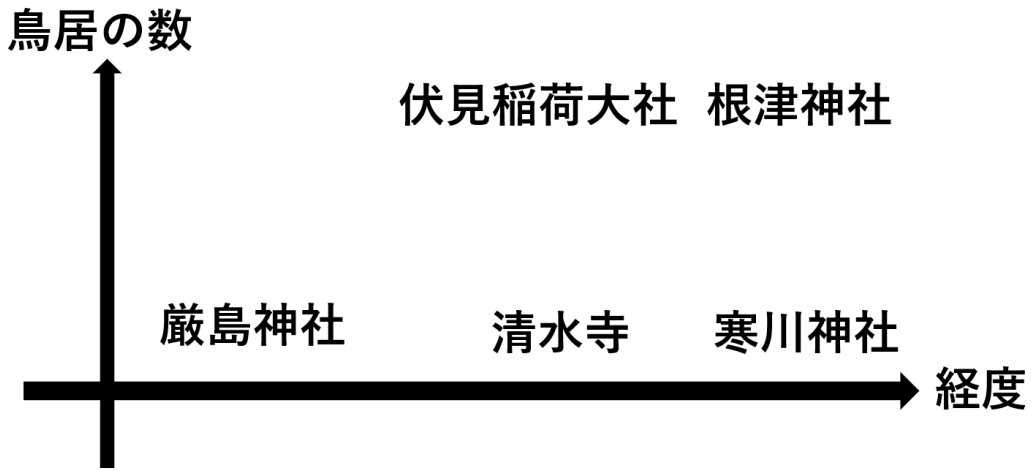

図 6　Word2vec の例

図 6 は，伏見稲荷大社（京都府），根津神社（東京都），寒川神社（神奈川県），清水寺（京都府）厳島神社（広島県）という単語を鳥居の数と経度の 2 次元で表現した例である．縦軸の値が大きければ大きいほど鳥居の数が多く小さければ小さいほど少ない，横軸の値が大きければ大きいほど経度が大きいとした．伏見稲荷大社は清水寺と比べると鳥居の数が多く，どちらも同じ京都府にあるため伏見稲荷大社のベクトルは清水寺のベクトルと比べて鳥居次元の大きさが大きくなっている．根津神社に関しては寒川神社と比べて鳥居の数が多いと言えるため，伏見稲荷大社のベクトルは清水寺のベクトルと比べ鳥居次元の大きさが大きくなっている．また，厳島神社は鳥居が少なく経度が小さい神社であるため，鳥居の数次元と経度次元の大きさが小さくなっている．Word2vec はこのような単語の分散表現を学習により取得する.

Word2vec を実現するニューラルネットワークの構造として Skip-gram と CBOW[5]が提案されている．図 7 及び図 8 は，window size のパラメータを$2c$としたときのニューラルネットワークである．window size は同じ文脈として考慮する前後の単語数を示すパラメータである．図 8 に Skip-gram のニューラルネットワーク構造を示す．このモデルは入力層，中間層，出力層の 3 層で形成されており，文章中のt番目の単語$W(t)$を入力し，その前後の単語$W(t-c)$，$W(t)$，$W(t+c)$を出力となるように学習を行うニューラルネットワークである．学習した Skip-gram に単語$W(t)$を入力した際の中間層の出力は単語$W(t)$を表すベクトルとなる．

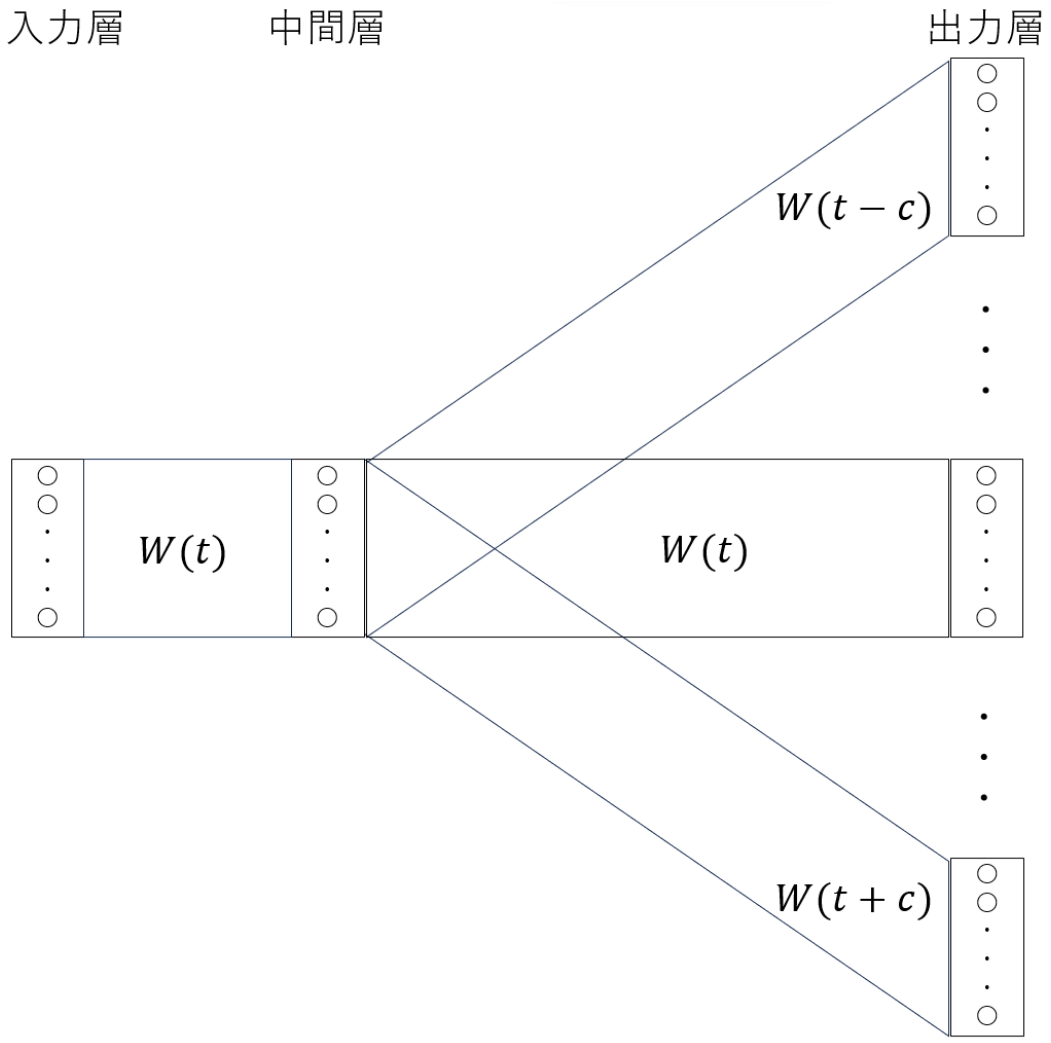

図 7　Skip-gram のニューラルネットワーク構造

図 8 に CBOW のニューラルネットワーク構造を示す．このモデルは Skip-gram と同様に，入力層，中間層，出力層の 3 層からなるが，入出力が Skip-gram の逆となっている．出力は中心の単語$W(t)$であり，入力をその前後の単語$W(t-c)$，$W(t+c)$となるように学習を行うニューラルネットワークである．すなわち，Skip-gram とは反対に，周辺の単語から中心にある単語を推定する問題をニューラルネットワークに学習させるモデルである．学習した CBOW に単語$W(t-c)$，$W(t+c)$を入力した際の中間層の出力は単語$W(t)$を表すベクトルとなる．

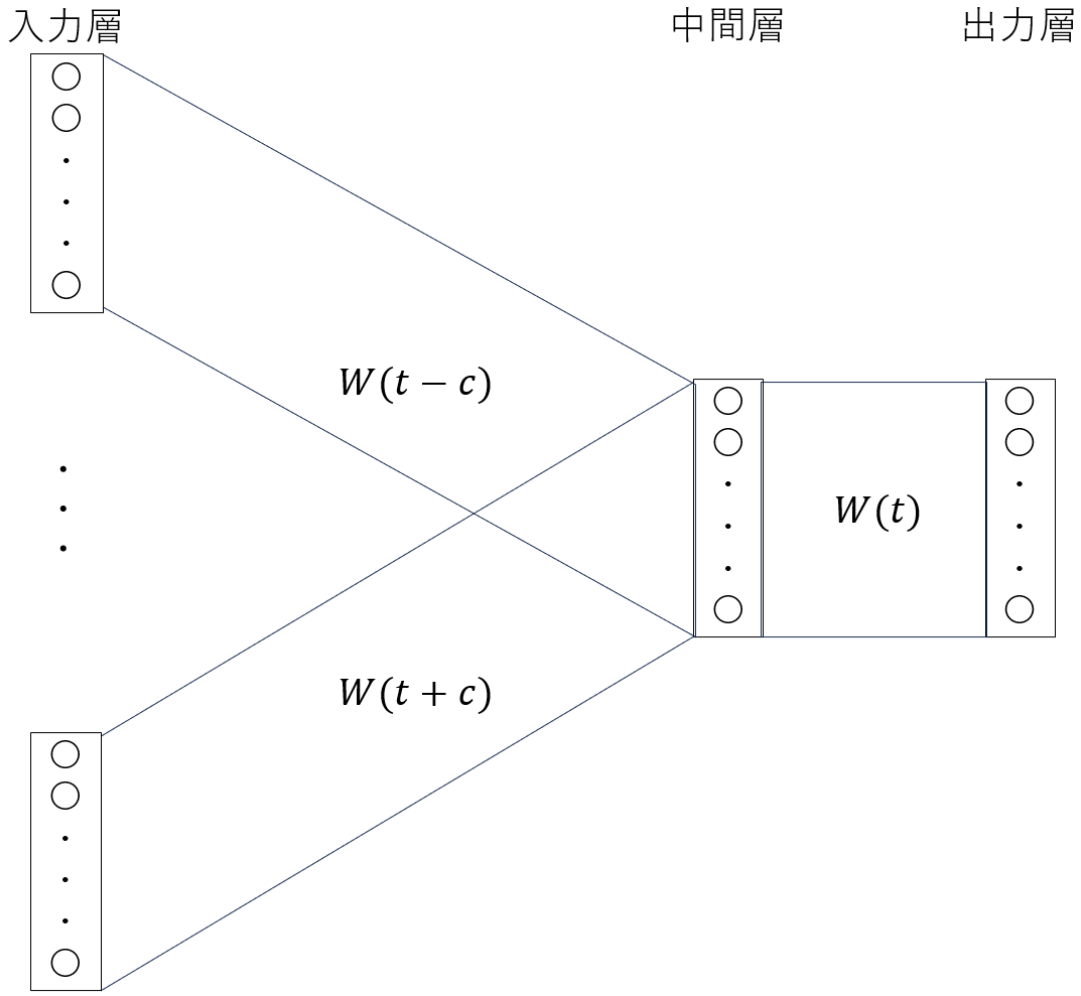

図 8　CBOW のニューラルネットワーク構造

第２章３節　PV-DBOW

Paragraph Vector には PV-DBOW と Paragraph Vector with Distributed Memory という 2 つのニューラルネットワーク構造が存在する．本節では，[3]で用いられている PV-DBOW に絞り説明する．単語の特徴ベクトルを生成する Word2vec に対して，Paragraph Vector は文章の特徴ベクトルを生成できるように拡張されている．図 9 に PV-DBOW のニューラルネットワーク構造を示す．このモデルは word2vec の Skip-gram を拡張したものである．ただし，Skip-gram と異なり，単語$W(t)$ではなく，文章の ID をネットワークの入力とする．ここで，文章の ID とは学習データで使用する各文章につけた番号である．文章の ID に対するニューラルネットワークの重みを Word2vec と同様の方法で学習することで，文章の特徴ベクトルを得ることができる．中間層の出力から得られた特徴ベクトルを PV-DBOW とみなし，文章の特徴ベクトルとなる．

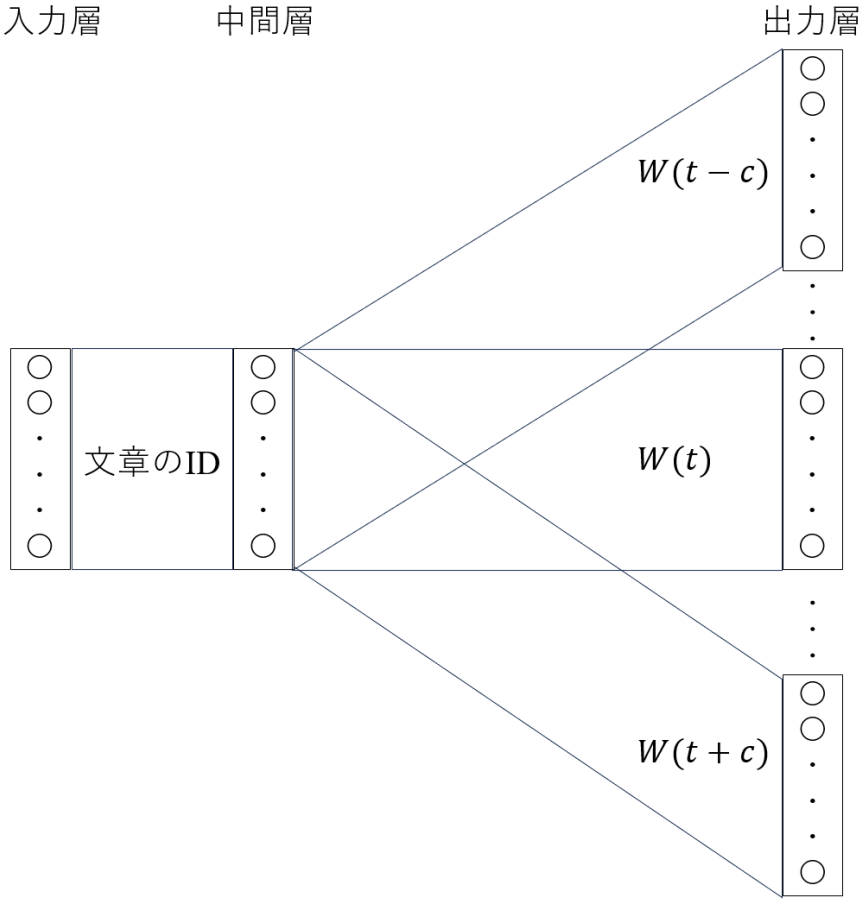

図 9　PV-DBOW のニューラルネットワーク構造

第２章４節　Sentence-LUKE[12]

大規模言語モデルは大量のテキストデータを使ってトレーニングされた自然言語処理のモデルのことである．入力される情報量，コンピューターが処理する計算量，パラメータの量の 3 つが大規模化しており，一般的には大規模言語モデルをファインチューニングすることによって，テキスト分類や文章生成といった様々なタスクに適応できる．大規模言語モデルの代表例としては，2018 年に Google が発表した Bidirectional Encoder Representations from Transformer（BERT）[6]や，2020 年に OpenAI が発表した Generative Pre-trained Transformer[7]などが挙げられる．

BERT はトークンごとにベクトル化を行うのに対して，Sentence-BERT は文章単位でベクトル化を行う．Sentence-BERT の概要を図 10 に示す．

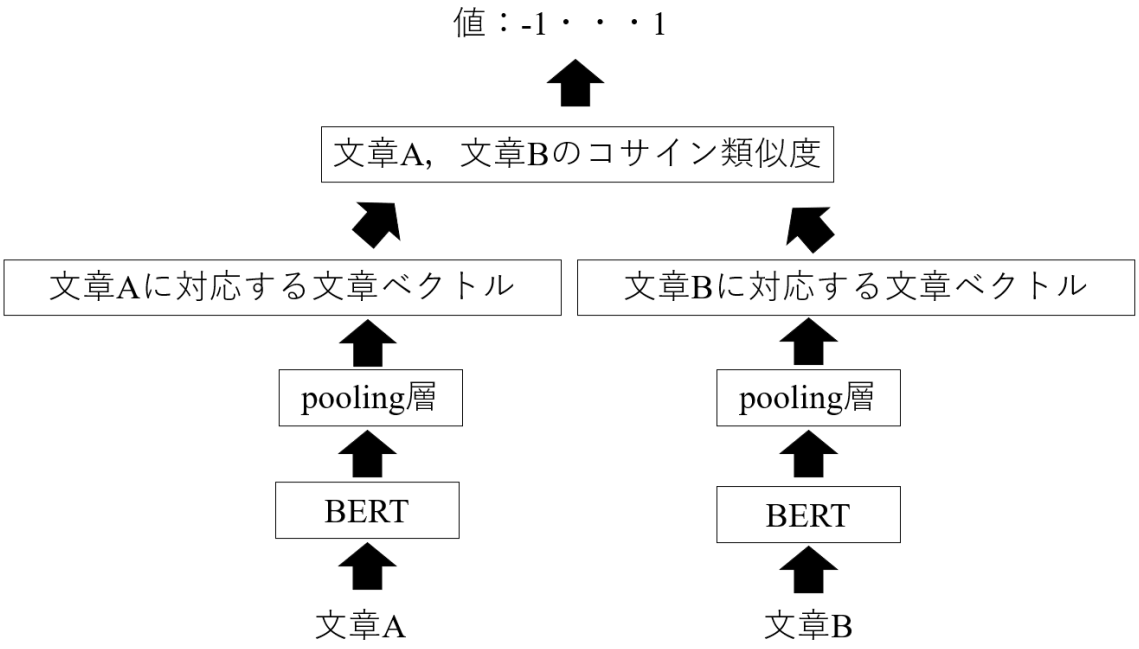

図 10　Sentence-BERT の概要

Sentence-BERT では，文章 A と文章 B を入力としてベクトルに変換する．A，B はコサイン類似度を用いて類似度を算出する．あるカテゴリに属する文章を用意し，文章 1 を A，B とした場合，教師信号を 1 とする．また文章 1 を A，文章 2 を B とした場合，教師信号を-1 とする．Sentence-BERT はこのように学習を行う．例えばレビューA とレビューB が京都タワーのレビューの場合，コサイン類似度の値が1に近づく，レビューA とレビューB がそれぞれ京都タワーと東京タワーのレビューの場合，コサイン類似度の値が−1に近づくようにしてベクトルを生

成している．BERT と異なり，pooling 層を追加している理由としては，コサイン類似度を計算する際，トークンごとのベクトルをまとめて固定長の文章ベクトルにするためである．

Sentence-LUKE は BERT の学習時に文章の固有名詞にエンティティを考慮したモデルであり，従来の LLM と比べ日本語言語理解ベンチマーク等のタスクで最高スコアを有している．エンティティの例として，「金閣寺は衣笠山の近くにある」という文章を用いる．ここで，「金閣寺」は「寺」であり，「衣笠山」は「山」というカテゴリに属している．このように，エンティティとは固有名詞が属するカテゴリを考慮することを指す．続いて，「[MASK]/は/衣笠山/の/[MASK]/に/ある/[SEP]/金閣寺/衣笠山」このように文章の単語の一部をマスクにし，文章の末尾(SEP)の後ろに固有名詞を付ける．Sentence-LUKE の学習方法を図 11，図 12 に示す．

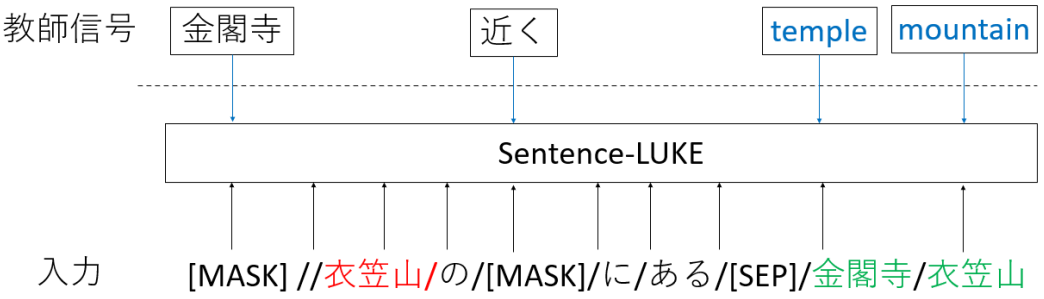

図 11　Sentence-LUKE の学習方法 1

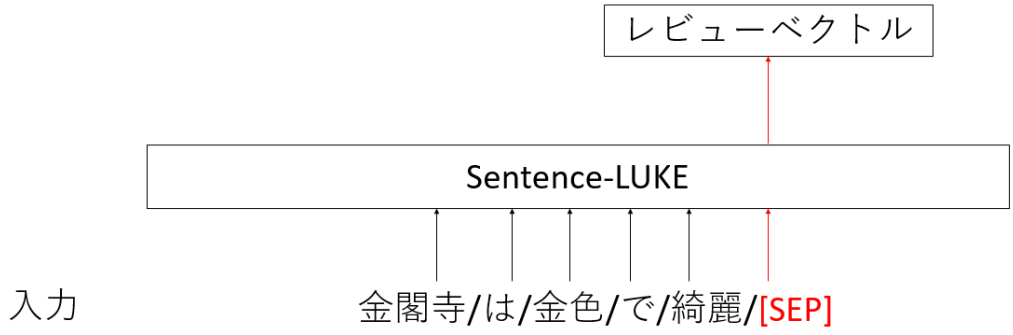

図 12　Sentence-LUKE の学習方法 2

図 11 のように，マスクされた単語と固有名詞のエンティティを予測する．そして，図 12 のように，レビューを入力し[SEP]トークンに対する出力をレビューベクトルとするように学習を行う．このように，Sentence-LUKE は固有名詞のエンティティを考慮することで性能を改善している．

第 3 章　先行研究

第 3 章 1 節　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

[1]では，Web 上から観光情報を抽出し，複数の特徴ベクトルから観光地ベクトルを生成する．得られた観光地ベクトル同士のコサイン類似度を評価することで，観光地を推薦するシステムとなっている．図 13 に[1]の概要を示す．

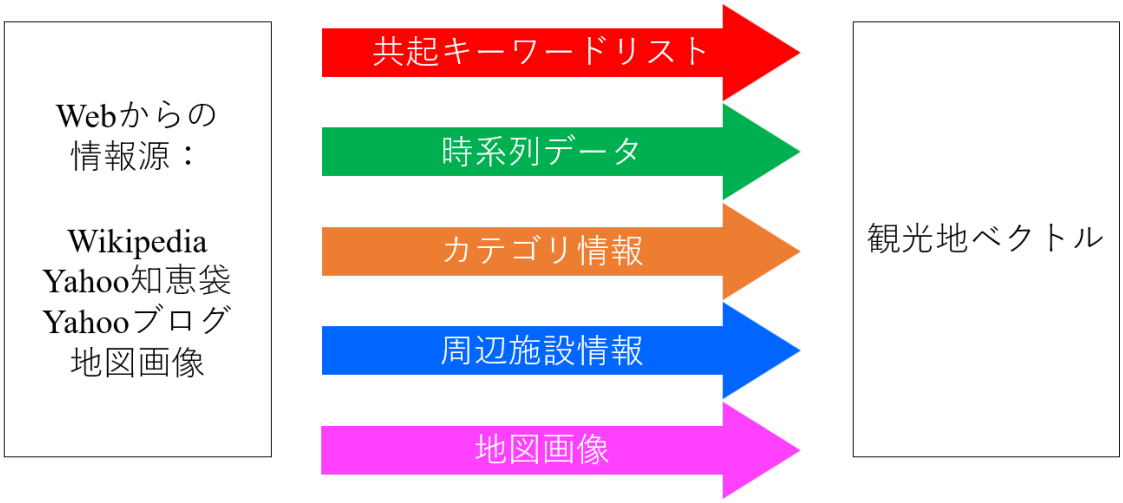

図 13　[1]の概要

Web 上からの情報源として，共起キーワード，時系列データ，カテゴリ情報，周辺施設情報，地図画像を用いる．共起キーワードの例を図 14 に示す．

京都	タワー	東京	きれい	さびれている	高い
0	1	2	3	4	5

京都タワー　＝　[1，1，0，0，1，0]

東京タワー　＝　[0，1，1，0，1，0]

東京スカイツリー　＝　[0，1，1，1，0，1]

図 14　共起キーワードの例

共起キーワードは WEB 上からの情報源として Wikipedia，Yahoo 知恵袋，Yahoo ブログから取得している．単語の出現回数のみを考慮する Bag-of-Words で共起キーワードを用いて京都タワーを表した際，[京都，タワー，さびれている]となる．このように京都タワー，東京タワー，スカイツリーそれぞれの要素を説明できる．

時系列データについて，清水寺と天王寺公園とした時に取得した時系列データの一部を図 15 に示す．

図 15　清水寺（左）と天王寺公園（右）の時系列データの一部[1]

Yahoo 知恵袋・Yahoo ブログで対象となる観光地に関する投稿の時系列データを取得している．清水寺，天王寺公園ともに紅葉スポットとして有名な観光地である．影響が図中のグラフに明確に現れており，10 月～11 月が著しく活性化していることが分かる．このように，時系列データを用いることで，各時期からみた観光地の活性化の度合いを特徴として捉えることができる．

カテゴリ情報について，観光地を東京タワーとした際に，取得したカテゴリ情報を図 16 に示す．

ジャンル	件数		
教養と学問、サイエンス	508	生き方と恋愛、人間関係の悩み	144
スポーツ、アウトドア、車	42	ビジネス、経済とお金	37
エンターテインメントと趣味	424	コンピュータテクノロジー	3
ニュース、政治、国際情勢	402	子育てと学校	17
マナー、冠婚葬祭	24	職業とキャリア	20
暮らしと生活ガイド	114	おしゃべり、雑談	2
健康、美容とファッション	17	インターネット、PCと家電	305

図 16　東京タワーのカテゴリ情報[1]

カテゴリ情報は Yahoo 知恵袋上でのカテゴリを使用する．東京タワーのカテゴリ情報をみると「教養と学問、サイエンス」カテゴリ内でのヒット数が多いことが分かる．つまり，建築学の視点で東京タワーの構造の話題が頻繁に取り上げられていることを示している．

周辺施設情報について，東京スカイツリーの位置情報を対象に，取得した周辺施設情報の一部を図 17 に示す．

和食	4356	ドラッグストア、市販薬	839
洋食	1356	家電、携帯電話	976
バイキング	25	百貨店、ショッピングセンター	132
中華	849	コンビニ、スーパー	1293
アジア料理、エスニック	351	リサイクル、ディスカウントショップ	366
ラーメン	579	生活用品、インテリア	2526
カレー	113	趣味、スポーツ、工芸	642
焼肉、ホルモン、ジンギスカン	515	ファッション、アクセサリー、時計	2738
		食品、食材	1636

図 18　東京スカイツリーの周辺施設情報の一部[1]

Yahoo ローカルサーチ API を用いて取得した地図データから計 58 種類の周辺施設情報を取得する．周辺施設情報とは観光地の周辺 5km 以内にある施設のジャンル情報のことであると定義されている．取得した周辺施設情報から東京スカイツリーの周辺にはレストランが多いということが分かる．

地図画像について，観光地を福岡タワーとした際の地図画像を図 19 に示す．

図 19　福岡タワー（左：標準マップ　右：ナイトマップ）[1]

用いる地図画像としては，標準的な地図画像である標準マップと，ミッドナイトモードの地図画像であるナイトマップの 2 つのパターンである．標準マップでは海や川，山などのように地理的な要素の違いで色合いが大きく変化する．それに対して，ナイトマップでは，周辺の建物・交通機関の数の変化で明暗の差がはっきりと分かれる．ベクトルとして用いる際，地図画像において HTML 色で表現されるカラーコードとその使用回数を指す色情報を抽出する．各要素にはカラーコードの使用回数の値を使用し，マップベクトルとしてベクトルにしている．

このように[1]では Web 上から観光情報を抽出し，複数の特徴ベクトルから観光地ベクトルを生成する．しかし，観光地ベクトルが定量的な情報源だけで形成されているという問題点がある．

例を挙げると，清水寺と金閣寺のカテゴリは同じ寺であり，同じ京都府に位置してるが，人間から見ると明らかに異なる観光地となっているため，改善の余地がある．この問題点を解決するために，人間の情緒を考慮することが挙げられる．

第３章２節　ユーザレビューの分散表現を用いた役割的に類似する観光スポット検索手法

[3]では，観光情報サイトであるじゃらん[13]から観光地レビューを抽出し，観光地レビューを学習した PV-DBOW を用いて観光地をベクトルで表す．得られた複数の特徴ベクトルから観光地間のコサイン類似度を評価し，観光地を推薦するシステムを提案している．観光地レビューを用いることで定量的な部分以外も考慮でき，書かれている文章によって差別化ができる．つまり，[1]の問題点であった人間の情緒を考慮することができる．

[3]では，観光地レビューを入力として，PV-DBOW を用いる．PV-DBOW の内部では，取得した全ての観光地レビューを 1 つのベクトルとする処理が行われ，観光地の特徴ベクトルを生成する．[3]の概要を図 20 に示す．

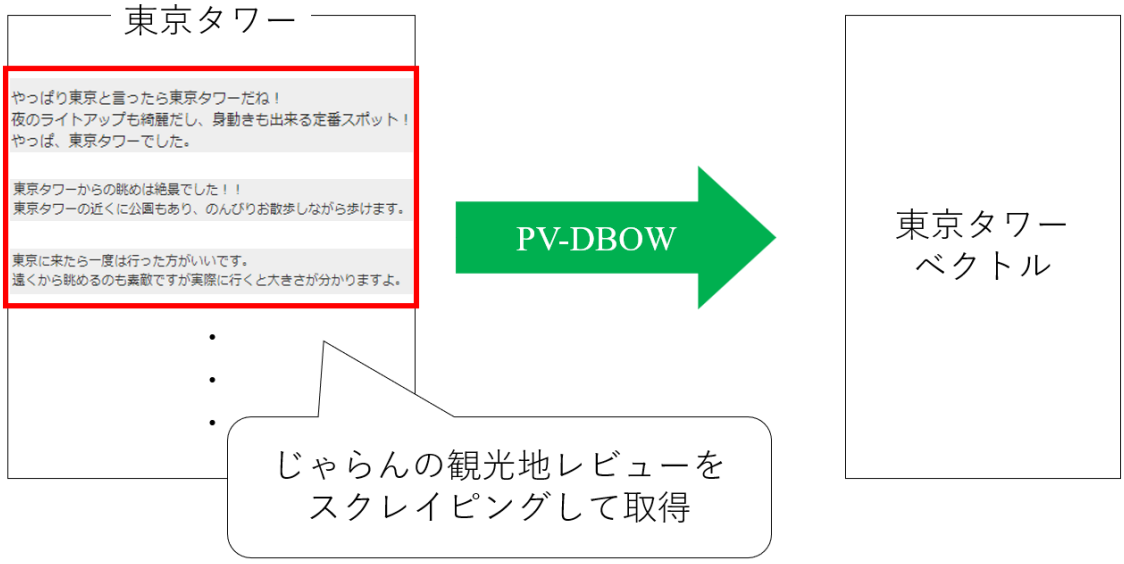

図 20　[3]の概要

PV-DBOW に東京タワーのレビューを入力データとして用いると，Word2vec と同様の方法で学習するため，文章の特徴ベクトルを得ることができる．ここで得られる特徴ベクトルは東京タワーベクトルとなる．そうして生成されたベクトルを基に観光地の推薦を行う．使用するベクトルについては，第 5 章 3 節で述べる．PV-DBOW に観光地レビューの学習を行うには，まず，ある観光地の全てのレビュー文章を 1 つに繋げる．例を図 21 に示す．

図 21　ある観光地の全てのレビュー文章を 1 つに繋げる例

東京タワーのレビュー文章を 1 つに繋げると図のようになる．次に，観光地全てに一意の ID を定める．例として，東京タワーの ID を 0 とする．続いて，ウィンドウサイズごとにレビュー文から単語を取り出す．図 22 にウィンドウサイズ 3 の場合を示す．

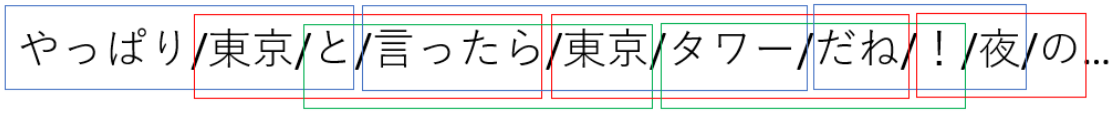

図 22　ウィンドウサイズ 3 の場合

図のように，ウィンドウサイズ 3 の場合，レビュー文の単語を前から 3 文字ずつ取得する．取得した単語と観光地の ID を対応させる．例えば，観光地 ID が東京タワーを示す 0 と「やっぱり/東京/と」ように対応させる．

[1]においてウィンドウサイズは 8，特徴ベクトルの次元数は 300 次元で学習を行っている．学習の例を図 23 に示す．

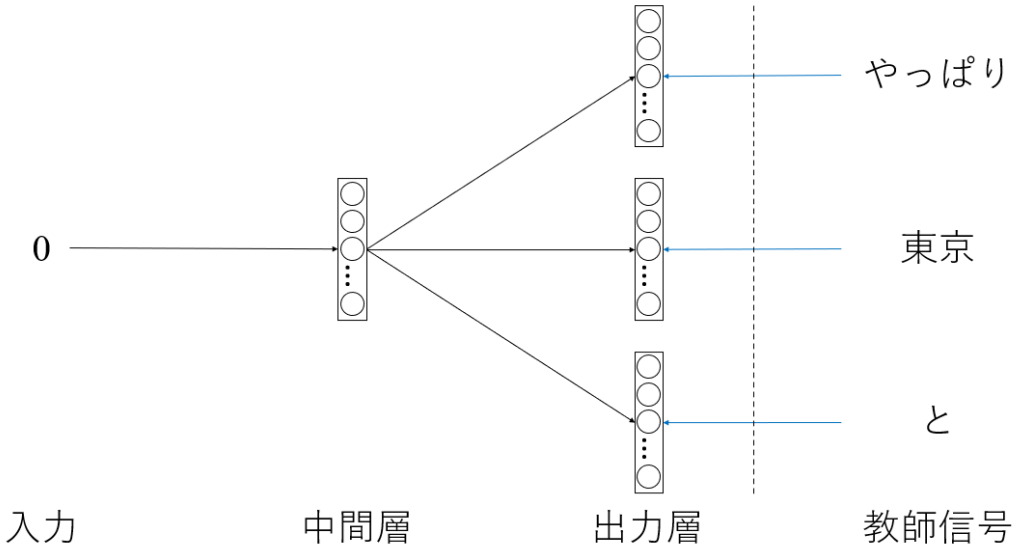

図 23　学習の例

得られた観光地 ID が東京タワーを示す 0 を入力とし，ウィンドウサイズごとにレビュー文から単語を取り出した「やっぱり/東京/と」を教師信号として学習を行う．そうして定められた回数を学習したネットワークに観光地の ID を入れたときの中間層の出力が観光地ベクトルとなる．つまり，東京タワーの ID である 0 を入力とした際の中間層の出力が東京タワーベクトルとなる．

第４章　提案手法

[3]では PV-DBOW に観光地レビューを入力することで観光地をベクトルで表し，複数の特徴ベクトルから観光地間の類似性を評価することで観光地を推薦している．しかし，PV-DBOW へ予め膨大な量の観光地レビューを学習させなければならない．この問題を解決するため，提案手法では，事前学習済みの LLM として日本語を事前学習済みの Sentence-LUKE である sentence-luke-japanese-base-lite に，じゃらんからスクレイピングによって取得した観光地レビューを入力することで観光地をベクトルで表す．得られた観光地ベクトル，カテゴリベクトル，都道府県ベクトル同士を演算するにあたり，観光地の各レビューを大規模言語モデルに入力し，出力されたベクトルを平均する．平均されたベクトルを用いて演算を行い，得られた合成ベクトルと全ての観光地を表すベクトルでコサイン類似度を計算し，類似度が高いものを推薦する観光地とする．提案手法の概要を図 24 に示す．

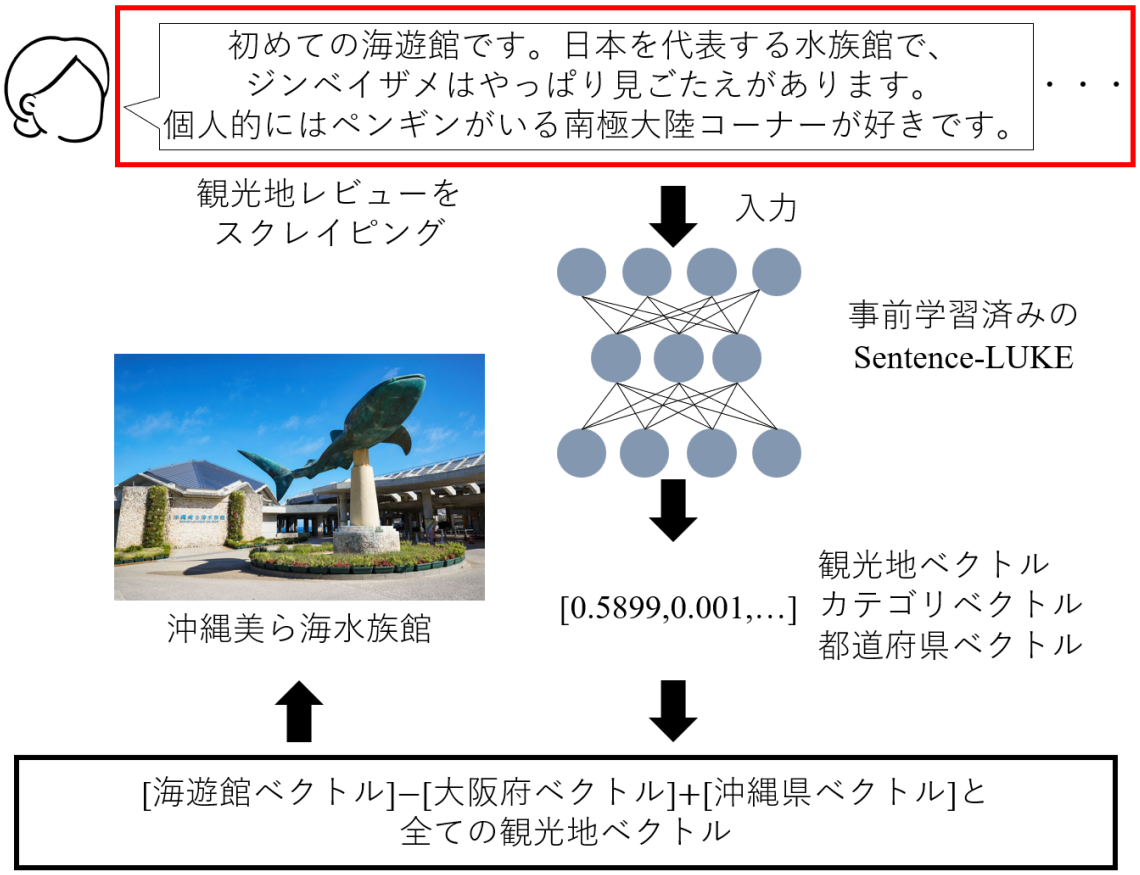

図 24　提案手法の概要

第５章　実験

第５章１節　観光地レビューの概要

提案手法ではじゃらんから取得した観光地レビューを用いる．観光地レビューの概要を図 25 に示す．

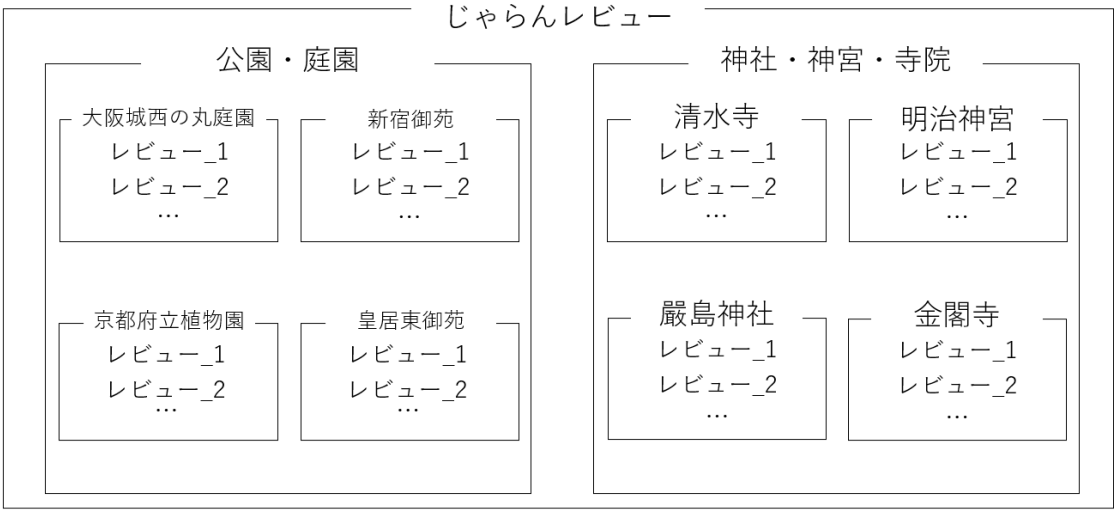

図 25　観光地レビューの概要

じゃらんから取得した観光地レビューがカテゴリ別に公園・庭園，神社・神宮・寺院のように分かれている．他のカテゴリとしては，山岳，ビーチ・海水浴場，展望台・展望施設，スキー・スノーボード，博物館，滝・渓谷，ダム，近代建築，史跡・名所巡り，城郭，水族館，町並み，動物園・植物園，文化史跡・遺跡，歴史的建造物の計 17 カテゴリからレビューを取得した．

例を挙げると，神社・神宮・寺院カテゴリの中に清水寺があり，清水寺のレビューが格納されている．また，各観光地はどの都道府県に存在しているかという情報を保持している．

第５章２節　大規模言語モデルの性能比較

第 2 章 5 節で挙げた日本語を事前学習済みの Sentence-LUKE である sentence-luke-japanese-base-lite [9]，日本語を事前学習済みの Sentence-BERT である日本語 Sentence-BERT[10]，そして新たに SBERT-JSNLI-large[11]の 3 つの大規模言語モデルを用いて，観光地レビュー文章の一致度を比較する実験を行った．入力したレビュー文章 1 を表 1 に示す．

表 1　入力したレビュー文章 1

A	とても見応えのある良い神社でした。引き潮の時間まで島に居たので、 夕陽を背景に鳥居の下迄歩いて行く事が出来ました。皆さん是非 行ってみて下さい。
B	テレビや写真で見ていた厳島神社に、やっと来ることができました。埼玉からは遠いね。日射しが少なく曇りの日だったので、観光地というより 荘厳な雰囲気でした。想像以上に広かったし、参道のお店も、平日なのに外国人観光客や修学旅行生でいっぱいでした。神社自体もよかったけど、潮が満ちて来て、さっきまで立っていた浜辺がみるみる無くなっていき、鳥居が浸されていくのを見ているのは感動モノでした。
C	11 月の終わりに河口湖へ行きました。秋から冬にかけては空気が澄んでいるので、晴れた日は富士山がくっきりと見え何とも素晴らしいです。 やっぱり日本はいいなぁと、何度も見入ってしまいました。 おすすめです。

それぞれ A，B は厳島神社に関するレビュー，C は富士山に関するレビューを用いた．また，レビューの評価に関しては，A が星 5 レビュー，B が星 4 レビュー，C が星 5 レビューとなっている．

A，B，C のレビューを先述した 3 つの大規模言語モデルに入力し，コサイン類似度を算出する．算出するにあたって，A，B は同じ観光地のため，A，B のコサイン類似度が A と C，B と C のコサイン類似度よりも高くなる大規模言語モデルが文意をよく捉えているといえる．表 2 に入力したレビュー文章 1 に対してコサイン類似度を算出した結果を示す．

表 2　入力したレビュー文章 1 に対してコサイン類似度を算出した結果

	A：B	A：C	B：C
SBERT-JSNLI-large	0.6385	0.7523	0.6158
sentence-bert-base-ja-mean-tokens-v2	0.5840	0.4991	0.4614
sentence-luke-japanese-base-lite	0.7448	0.3974	0.4047

表の縦軸は A と B，A と C，B と C でコサイン類似度を計算した結果，横軸は各モデルの数値となっている．同じ観光地のレビューである A と B のコサイン類似度が最も高い数値の場合を赤字，異なる観光地のレビューである A と C，B と C のコサイン類似度が最も高い数値の場合を青字で表した．

結果から，日本語 Sentence-BERT，sentence-luke-japanese-base-lite では文意をよく捉えているといえる．

他にも異なる観光地のレビューを入力して，3 つのモデルの中で最も文意をよく捉えている大規模言語モデルはどのモデルかを調べた．表 3 に入力したレビュー文章 2 を示す．

表 3　入力したレビュー文章 2

A	大アマゾン展と地球館に行きました！大アマゾン展は文句一つない素敵な場所でとても楽しめ、学ぶことができました。地球館は、改装工事が多く見れないところが多かったですが、また夏に来たいねって話ができました。しかし、出口付近にある総合窓口の対応が悪すぎです。閉館直後にお手洗いに忘れものしてしまったことに気付き、戻ったのですが、閉館してるから帰れというかのような表情での対応。閉館して 10 分程なのに、入ることは無理なのは承知ですが、探してもくれませんでした。確実に置いてきてしまった場所や中身を伝えているのにも関わらず、もっと細かく教えろとか、探してくれないくせに言われ、挙句明日以降に電話してくれたら対応するとのこと。 さすがにサービス業ではないにしろ、対応が雑すぎて、楽しかったのに残念に終わりました。また、次の日電話したのですが、まず遺失物届けださせましたか？とのこと。前日にそんな案内されませんでした。また電話対応も本当に社会人なのか？と疑うような喋り方。すいませーん。ありませーん。みたいな。 博物館はよかったのに、もう行きたくない！！！
B	規模が大きすぎです。科学と言う名の下、雑多な展示物と言う感じです。 補助金４０億円で４億円の入場収入なのに、２館必要なのかどうか。 もっと展示物を絞って一回り２時間ぐらいにしないと地方から行きづらい博物館なんですね。こんなに大きくなったのは天下り官僚の予算獲得によってかな？入り口が分からなかったのを含めて、評価を低くしたいと思います。
C	本物の忍者屋敷をくのいちの扮装をした館員がしかけを実践しながら説明してくれ、思った以上に楽しかった。外国人向けに英語の説明文も細かく表示しており感心した。また忍者ショーは笑いあり驚きありで、手裏剣等忍者の戦闘道具の使い方や戦闘を実演してくれる。これも英語での説明もあり、外国人も大うけしていてびっくりした。

それぞれ A，B は国立科学博物館に関するレビュー，C は伊賀流忍者博物館に関するレビューを用いた．また，レビューの評価に関しては，A が星 1 レビュー，B が星 1 レビュー，C が星 5 レビューとなっている．表 4 に入力したレビュー文章 2 に対してコサイン類似度を算出した結果を示す．

表 4　入力したレビュー文章 2 に対してコサイン類似度を算出した結果

	A：B	A：C	B：C
SBERT-JSNLI-large	0.6263	0.6903	0.5718
sentence-bert-base-ja-mean-tokens-v2	0.4484	0.4558	0.2202
sentence-luke-japanese-base-lite	0.5553	0.3411	0.3981

結果から，sentence-luke-japanese-base-lite のみが文意をよく捉えているといえる．

続いて，それぞれのモデルに A，B には京都タワー，C には東京タワーのレビューを入力し，10 通り試した．同様の実験を行い，表 5 に 10 回ランダムにピックアップしたレビューを用いて行った結果を示す．

表 5　10 回ランダムにピックアップしたレビューを用いて行った結果

	精度
SBERT-JSNLI-large	40%
sentence-bert-base-ja-mean-tokens-v2	30%
sentence-luke-japanese-base-lite	80%

以上のことから，sentence-luke-japanese-base-lite の精度が最も良いと示された．よって，sentence-luke-japanese-base-lite を使用し，観光地ベクトルを作成した．

第５章３節　観光地・カテゴリ・都道府県ベクトルの作成

じゃらんから取得した観光地レビューを用いて観光地ベクトルを作成する．ある観光地sの観光地ベクトルの作成方法を図 26，(5)に示す．

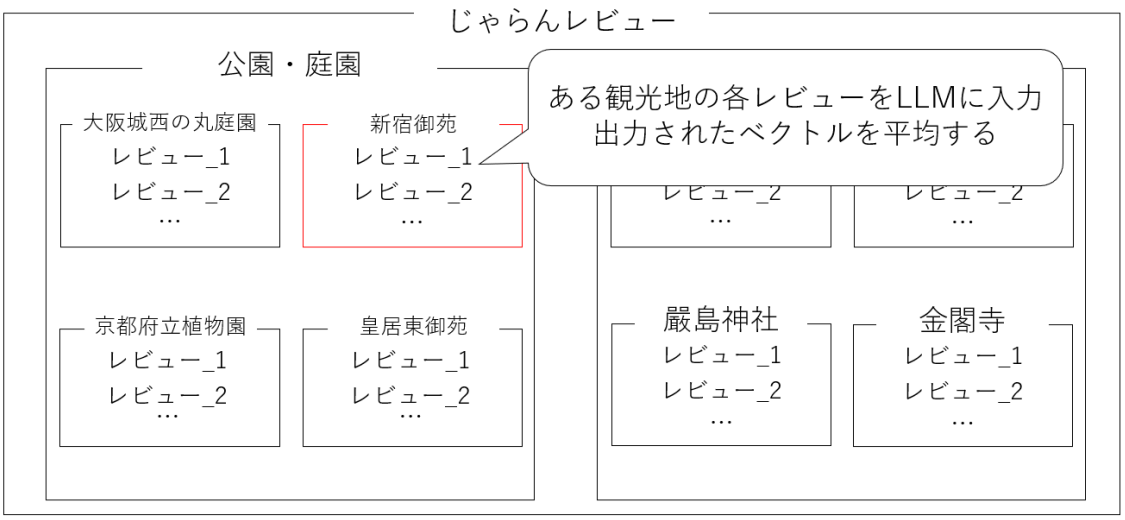

図 26　観光地ベクトルの作成方法

$$\bar{\mathbf{v}_s} = \frac{1}{N_s}\sum_{i=1}^{N_s}\mathbf{v}_{s,i} \tag{5}$$

ここで，N_sは観光地sのレビュー数，$\mathbf{v}_{s,i}$は観光地sのi番目の観光地レビューを sentence-luke-japanese-base-lite に入力して得られたベクトルである．例えば，新宿御苑の各レビューを大規模言語モデルに入力し，出力されたベクトルを平均する．このようにして，新宿御苑ベクトルが得られる．他の観光地も同様にベクトル化を行う．

あるカテゴリcのカテゴリベクトルの作成方法を図 27，(6)に示す．

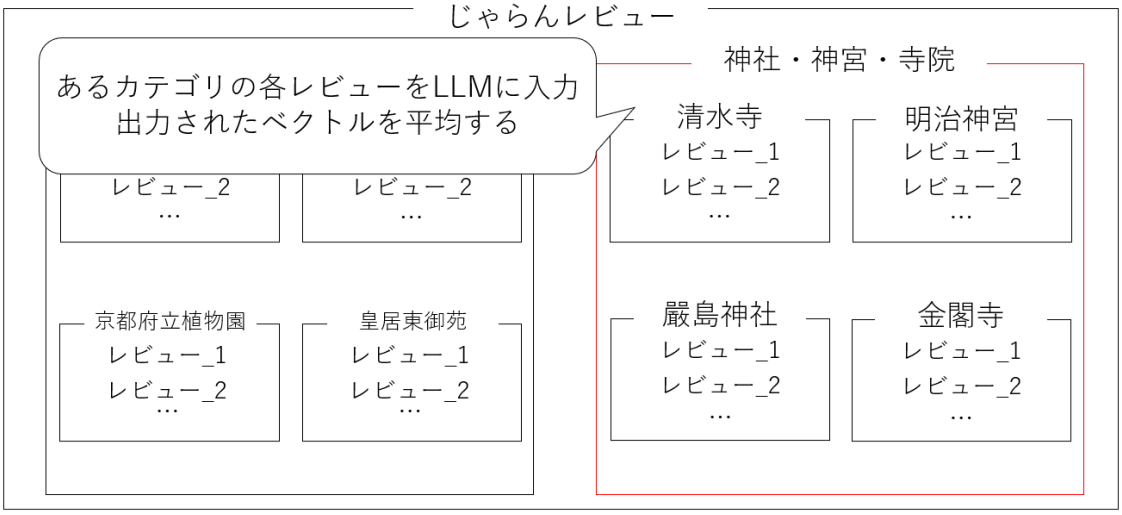

図 27　カテゴリベクトルの作成方法

$$\bar{\mathbf{v}_c} = \frac{1}{N_c}\sum_{i=1}^{N_c}\mathbf{v}_{c,i} \tag{6}$$

ここで，N_cはカテゴリsに属する観光地のレビュー数，$\mathbf{v}_{c,i}$はカテゴリcに属するi番目の観光地レビューを sentence-luke-japanese-base-lite に入力して得られたベクトルである．例えば，神社・神宮・寺院カテゴリの各レビューを大規模言語モデルに入力し，出力されたベクトルを平均する．このようにして，神社・神宮・寺院ベクトルが得られる．他のカテゴリも同様にベクトル化を行う．

ある都道府県pの都道府県ベクトルの作成方法を図 28，(7)に示す．

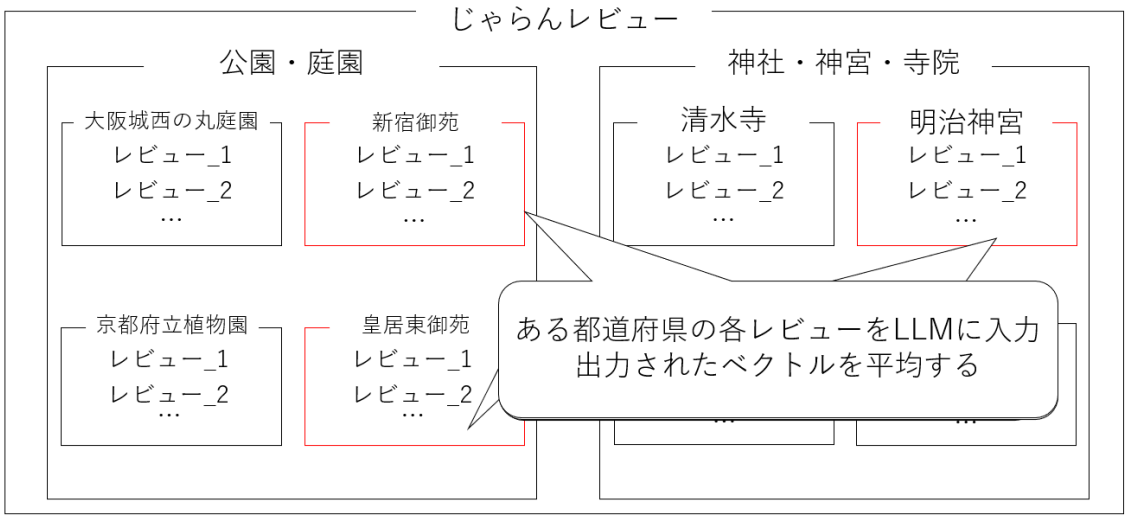

図 28　都道府県ベクトルの作成方法

$$\overline{\mathbf{v}_p} = \frac{1}{N_p}\sum_{i=1}^{N_p}\mathbf{v}_{p,i} \tag{7}$$

ここで，N_pは都道府県pに属する観光地のレビュー数，$\mathbf{v}_{p,i}$は都道府県pに属するi番目の観光地レビューを sentence-luke-japanese-base-lite に入力して得られたベクトルである．例えば，ある都道府県の各レビューを大規模言語モデルに入力し，出力されたベクトルを平均する．図のように，東京都の観光地である新宿御苑，皇居東御苑，明治神宮のじゃらんレビューから，東京都ベクトルが得られる．他の都道府県も同様にベクトル化を行う．

第５章４節　観光地と都道府県に基づく観光地ベクトル類似度実験

観光地ベクトルが正しくできているかを確かめるために，提案手法を用いて実験を行った．

例として大阪府の海遊館がすごい気に入り，沖縄県で同じような観光地を探したい場合，沖縄県において海遊館に似た観光地は沖縄美ら海水族館となる．海遊館であれば人気が高く華やかな水族館であるため，同じく沖縄県で人気が高く華やかな観光地ということで沖縄美ら海水族館が推薦されるのが望ましいと言える．

所望する観光地が大阪府に位置し，水族館カテゴリである海遊館に似た，都道府県が沖縄県である場合の合成ベクトルの計算方法を(8)に示す．

$$\tilde{\mathbf{v}} = \mathbf{v}_{\text{海遊館}} - \alpha\mathbf{v}_{\text{大阪府}} + \alpha\mathbf{v}_{\text{沖縄県}} \tag{8}$$

海遊館は大阪府に位置する観光地であるため，海遊館を表す観光地ベクトルである$\mathbf{v}_{\text{海遊館}}$から，大阪府を表す都道府県ベクトルである$\mathbf{v}_{\text{大阪府}}$を引き，沖縄県を表す都道府県ベクトルである$\mathbf{v}_{\text{沖縄県}}$を足している．ここで，αは都道府県またはカテゴリの影響度を表すパラメータであり，α の値を変更することで推薦される観光地が変わる．得られた合成ベクトルと全ての観光地を表すベクトルでコサイン類似度を計算し，類似度が高いものを推薦する観光地とする．

観光地ベクトル S −都道府県ベクトル P_s ＋都道府県ベクトル P において，カテゴリは観光地 S と同じである C_s，都道府県は足した都道府県である P の観光地を似た観光地とする．実験結果において，基準を満たす観光地を赤字とした．

影響度パラメータ$\alpha = 1$の際の，$\tilde{\mathbf{v}} = \mathbf{v}_{\text{海遊館}} - \alpha\mathbf{v}_{\text{大阪府}} + \alpha\mathbf{v}_{\text{沖縄県}}$と全ての観光地ベクトルとのコサイン類似度を算出した結果を表 6 に示す．

表 6　コサイン類似度を算出した結果

観光地名	コサイン類似度	都道府県	カテゴリ名
沖縄美ら海水族館	**0.9538**	**沖縄県**	**水族館**
のとじま水族館	0.9232	石川県	水族館
いおワールド かごしま水族館	0.9177	鹿児島県	水族館
海中水族館 シードーナツ	0.9162	熊本県	水族館
海遊館	0.9153	大阪府	水族館

表の縦軸は左から順に，推薦結果の上位 5 つの観光地名，コサイン類似度を計算した結果，観光地が位置する都道府県名，観光地が属するカテゴリ名である．横軸は各観光地におけるコサイン類似度の値となっている．1 位に沖縄美ら海水族館という結果となった．結果として，$\mathbf{v}_{海遊館}$から$\mathbf{v}_{大阪府}$を引き，$\mathbf{v}_{沖縄県}$を足すことで，合成ベクトルには沖縄県ベクトルの要素が強く出ているためだと考えられる．また，1 位から 5 位全て水族館カテゴリの観光地となったことから，演算おいて水族館ベクトルが正しく作用していると考えられる．

続いて，所望する観光地が新潟県に位置しスキー・スノーボードカテゴリである斑尾高原スキー場に似た，都道府県が北海道である場合の合成ベクトルの計算方法を(9)に示す．

$$\tilde{\mathbf{v}} = \mathbf{v}_{斑尾高原スキー場} - \alpha\mathbf{v}_{新潟県} + \alpha\mathbf{v}_{北海道} \tag{9}$$

影響度パラメータ$\alpha = 1$の際の(9)と全ての観光地ベクトルでコサイン類似度を算出した．表 7 に提案手法においてコサイン類似度を算出した結果を示す．

表 7　コサイン類似度を算出した結果

観光地名	コサイン類似度	都道府県	カテゴリ名
ルスツリゾート スキー場	**0.9790**	北海道	スキー・スノーボード
ニセコビレッジ スキーリゾート	**0.9760**	北海道	スキー・スノーボード
斑尾高原スキー場	0.9740	新潟県	スキー・スノーボード
ニセコ HANAZONO リゾート	**0.9732**	北海道	スキー・スノーボード
ニセコアンヌプリ国際スキー場	**0.9729**	北海道	スキー・スノーボード

結果から，1 位から 2 位，4 位から 5 位に所望する北海道のスキー・スノーボードカテゴリに属する観光地となった．北海道のスキー・スノーボードカテゴリに属する観光地が 1 位から 5 位を占めたことから，$\mathbf{v}_{北海道}$と$\mathbf{v}_{スキー・スノーボード}$でコサイン類似度を算出した．結果として，値は 0.8598 となった．$\mathbf{v}_{北海道}$と$\mathbf{v}_{スキー・スノーボード}$のコサイン類似度の値が高いため，北海道に占めるスキー・スノーボードカテゴリに属する観光地の割合が大きいと言えるのではないかと考えられる．また，じゃらんにおいて北海道に位置する水族館カテゴリの観光地数は 10 件であるが，北海道に位置するスキー・スノーボードカテゴリの観光地数は 58 件である．よって，他カテゴリと比較してスキー・スノーボードカテゴリに属する観光地が多いことが分かる．このことから，北海道のスキー・スノーボードカテゴリに属する観光地が 1 位から 5 位を占めた要因となり得るのではないかと考えられる．

次に，所望する観光地が鹿児島県に位置し山岳カテゴリである桜島に似た，都道府県が北海道である場合の合成ベクトルの計算方法を(10)に示す．

$$\tilde{\mathbf{v}} = \mathbf{v}_{桜島} - \alpha\mathbf{v}_{鹿児島県} + \alpha\mathbf{v}_{北海道} \tag{10}$$

影響度パラメータ$\alpha = 1$の際の(10)と全ての観光地ベクトルでコサイン類似度を算出した．表 8 に提案手法においてコサイン類似度を算出した結果を示す．

表 8　α = 1としてコサイン類似度を算出した結果

観光地名	コサイン類似度	都道府県	カテゴリ名
桜島	0.9459	鹿児島県	山岳
湯之平展望所	0.8887	鹿児島県	展望台・展望施設
有村溶岩展望所	0.8762	鹿児島県	展望台・展望施設
城山展望所	0.8515	鹿児島県	展望台・展望施設
城山	0.8354	鹿児島県	山岳

結果から，1 位から 5 位全て都道府県が異なる観光地となった。α = 1とした演算結果では所望する北海道の観光地を表せなかったため，都道府県ベクトルの要素を定数倍して演算を行った．実験として(10)において影響度パラメータα = 2の合成ベクトルと全ての観光地ベクトルでコサイン類似度を算出した．影響度パラメータα = 2とした理由については，α = 1からα = 10まで実験して最も北海道の観光地を表せたのがα = 2の場合だったためである．表 9 に提案手法においてα = 2として演算を行った結果を示す．

表 9　α = 2としてコサイン類似度を算出した結果

観光地名	コサイン類似度	都道府県	カテゴリ名
キロロ スノーワールド	0.8253	北海道	スキー・ スノーボード
毛無山展望所	0.8141	北海道	展望台・展望施設
藻岩山	**0.8105**	**北海道**	**山岳**
稚内公園	0.8062	北海道	公園・庭園
朝里川温泉 スキー場	0.8051	北海道	スキー・ スノーボード

結果から，3 位に北海道で桜島と同じカテゴリである山岳という結果になった．α = 1とした演算結果を示す表 9 と比較して，北海道の観光地が表されていることが分かる．しかし，表すことができたのは藻岩山 1 件のみとなった．このように，所望する都道府県とカテゴリの観光地を表すことはできるが，件数が少ない場合もあった．表 9 において 1 位，2 位，4 位，5 位であるキロロスノーワールド，毛無山展望所，稚内公園，朝里川温泉スキー場はいずれも周辺に山岳が位置しており，景色が一望できる観光地であるため，鹿児島県の桜島と似た観光地であることが分かる．よって，カテゴリは異なる観光地であるが，鹿児島県の桜島と似た観光地を結果として出力したのではないかと考えられる．

表 10 から表 17 に(10)においてα = 3からα = 10までを実験してコサイン類似度を算出した結果を示す．

表 10　α = 3としてコサイン類似度を算出した結果

観光地名	コサイン類似度	都道府県	カテゴリ名
キロロ スノーワールド	0.8139	北海道	スキー・ スノーボード
朝里川温泉スキー場	0.8066	北海道	スキー・ スノーボード
札幌国際スキー場	0.8014	北海道	スキー・ スノーボード
富良野スキー場	0.7952	北海道	スキー・ スノーボード
ニセコ東急グラン・ ヒラフスキー場	0.7776	北海道	スキー・ スノーボード

表 11　α = 4としてコサイン類似度を算出した結果

観光地名	コサイン類似度	都道府県	カテゴリ名
札幌国際スキー場	0.7780	北海道	スキー・ スノーボード
朝里川温泉スキー場	0.7727	北海道	スキー・ スノーボード
キロロ スノーワールド	0.7704	北海道	スキー・ スノーボード
富良野スキー場	0.7659	北海道	スキー・ スノーボード
サッポロテイネ	0.7439	北海道	スキー・ スノーボード

表 12　α = 5としてコサイン類似度を算出した結果

観光地名	コサイン類似度	都道府県	カテゴリ名
札幌国際スキー場	0.7438	北海道	スキー・ スノーボード
朝里川温泉スキー場	0.7311	北海道	スキー・ スノーボード
富良野スキー場	0.7278	北海道	スキー・ スノーボード
キロロ スノーワールド	0.7218	北海道	スキー・ スノーボード
さっぽろ ばんけいスキー場	0.7129	北海道	スキー・ スノーボード

表 13　α = 6としてコサイン類似度を算出した結果

観光地名	コサイン類似度	都道府県	カテゴリ名
札幌国際スキー場	0.7102	北海道	スキー・スノーボード
朝里川温泉スキー場	0.6923	北海道	スキー・スノーボード
富良野スキー場	0.6916	北海道	スキー・スノーボード
さっぽろばんけいスキー場	0.6824	北海道	スキー・スノーボード
キロロスノーワールド	0.6781	北海道	スキー・スノーボード

表 14　α = 7としてコサイン類似度を算出した結果

観光地名	コサイン類似度	都道府県	カテゴリ名
札幌国際スキー場	0.6806	北海道	スキー・スノーボード
富良野スキー場	0.6601	北海道	スキー・スノーボード
朝里川温泉スキー場	0.6590	北海道	スキー・スノーボード
さっぽろばんけいスキー場	0.6553	北海道	スキー・スノーボード
サッポロテイネ	0.6450	北海道	スキー・スノーボード

表 15　α = 8としてコサイン類似度を算出した結果

観光地名	コサイン類似度	都道府県	カテゴリ名
札幌国際スキー場	0.6555	北海道	スキー・スノーボード
富良野スキー場	0.6336	北海道	スキー・スノーボード
さっぽろばんけいスキー場	0.6321	北海道	スキー・スノーボード
朝里川温泉スキー場	0.6310	北海道	スキー・スノーボード
サッポロテイネ	0.6200	北海道	スキー・スノーボード

表 16　$\alpha = 9$としてコサイン類似度を算出した結果

観光地名	コサイン類似度	都道府県	カテゴリ名
札幌国際スキー場	0.6342	北海道	スキー・スノーボード
さっぽろばんけいスキー場	0.6125	北海道	スキー・スノーボード
富良野スキー場	0.6114	北海道	スキー・スノーボード
朝里川温泉スキー場	0.6076	北海道	スキー・スノーボード
サッポロテイネ	0.5989	北海道	スキー・スノーボード

表 17　$\alpha = 10$としてコサイン類似度を算出した結果

観光地名	コサイン類似度	都道府県	カテゴリ名
札幌国際スキー場	0.6162	北海道	スキー・スノーボード
さっぽろばんけいスキー場	0.5958	北海道	スキー・スノーボード
富良野スキー場	0.5926	北海道	スキー・スノーボード
朝里川温泉スキー場	0.5879	北海道	スキー・スノーボード
サッポロテイネ	0.5810	北海道	スキー・スノーボード

第５章５節　[3]と提案手法の推薦結果の比較実験

[3]と提案手法と推薦結果の比較を行った．所望する観光地が東京都に位置し公園・庭園カテゴリである新宿御苑に似た，都道府県が大阪府である場合の合成ベクトルの計算方法を(11)に示す．

$$\tilde{\mathbf{v}} = \mathbf{v}_{\text{新宿御苑}} - \alpha\mathbf{v}_{\text{東京都}} + \alpha\mathbf{v}_{\text{大阪府}} \tag{11}$$

[3]と提案手法どちらも影響度パラメータ$\alpha = 1$の際の(11)と全ての観光地ベクトルでコサイン類似度を算出した．表 18 に[3]においてコサイン類似度を算出した結果，表 19 に提案手法においてコサイン類似度を算出した結果を示す．

表 18　[3]においてコサイン類似度を算出した結果

観光地名	コサイン類似度	都道府県	カテゴリ名
新宿御苑の桜	0.6754	東京都	動物園・植物園
京都府立植物園	0.4791	京都府	動物園・植物園
皇居東御苑	0.4564	東京都	公園・庭園
大阪城西の丸庭園	**0.4370**	**大阪府**	**公園・庭園**
大谷山自然公園	0.4123	奈良県	公園・庭園

表 19　提案手法においてα＝1としてコサイン類似度を算出した結果

観光地名	コサイン類似度	都道府県	カテゴリ名
新宿御苑	0.9740	東京都	公園・庭園
青葉の森公園	0.9476	千葉県	公園・庭園
国営昭和記念公園	0.9466	東京都	公園・庭園
大仙公園	**0.9451**	**大阪府**	**公園・庭園**
都立水元公園	0.9397	東京都	公園・庭園

[3]においてコサイン類似度を算出した結果と比較して，提案手法においてコサイン類似度を算出した結果では，4 位のみ大阪府の大仙公園という結果になった．

α＝1とした演算結果では所望する大阪府の観光地を表せなかったため，都道府県ベクトルの要素を定数倍して演算を行った．実験として(11)において影響度パラメータα＝5の合成ベクトルと全ての観光地ベクトルでコサイン類似度を算出した．影響度パラメータα＝5とした理由については，α＝1からα＝10まで実験して最も大阪府の観光地を表せたのがα＝5の場合だったためである．

[3]は等倍の演算結果の場合を比較している．表 20 に提案手法においてα＝5として演算を行った結果を示す．

表 20　提案手法においてα＝5として演算を行った結果

観光地名	コサイン類似度	都道府県	カテゴリ名
浜寺公園	**0.8172**	**大阪府**	**公園・庭園**
久宝寺緑地	**0.8122**	**大阪府**	**公園・庭園**
服部緑地	**0.8114**	**大阪府**	**公園・庭園**
千葉県立柏の葉公園	0.8101	千葉県	公園・庭園
山田池公園	**0.8073**	**大阪府**	**公園・庭園**

結果から，1 位から 3 位と 5 位に大阪府で新宿御苑と同じカテゴリである公園・庭園という結果になった．α＝1とした演算結果を示す表 19 と比較して，大阪府の観光地数が増加していることが分かる．作成した合成ベクトルには大

阪府ベクトルの要素が強く出ているため，所望する大阪府の公園・庭園カテゴリに属する観光地を最も表せたと考えられる．また，表 19，表 20 において，1 位から 5 位全て公園・庭園カテゴリに属する観光地が出力された．このことから，演算おいて公園・庭園ベクトルは正しく作用していると考えられる．

表 21 から表 28 に(11)においてα = 2からα = 4まで，α = 6からα = 10を実験してコサイン類似度を算出した結果を示す．

表 21　提案手法においてα = 2として演算を行った結果

観光地名	コサイン類似度	都道府県	カテゴリ名
国営昭和記念公園	0.9354	東京都	公園・庭園
花博記念公園 鶴見緑地	**0.9350**	**大阪府**	**公園・庭園**
青葉の森公園	0.9295	千葉県	公園・庭園
山田池公園	**0.9292**	**大阪府**	**公園・庭園**
都立小金井公園	0.9277	東京都	公園・庭園

表 22　提案手法においてα = 3として演算を行った結果

観光地名	コサイン類似度	都道府県	カテゴリ名
山田池公園	**0.9014**	**大阪府**	**公園・庭園**
服部緑地	**0.8995**	**大阪府**	**公園・庭園**
花博記念公園 鶴見緑地	**0.8972**	**大阪府**	**公園・庭園**
千葉県立柏の葉公園	0.8985	千葉県	公園・庭園
浜寺公園	**0.8972**	**大阪府**	**公園・庭園**

表 23　提案手法においてα = 4として演算を行った結果

観光地名	コサイン類似度	都道府県	カテゴリ名
浜寺公園	**0.8603**	**大阪府**	**公園・庭園**
服部緑地	**0.8580**	**大阪府**	**公園・庭園**
千葉県立柏の葉公園	0.8569	千葉県	公園・庭園
山田池公園	**0.8566**	**大阪府**	**公園・庭園**
久宝寺緑地	**0.8527**	**大阪府**	**公園・庭園**

表 24　提案手法においてα＝6として演算を行った結果

観光地名	コサイン類似度	都道府県	カテゴリ名
浜寺公園	**0.7747**	**大阪府**	**公園・庭園**
久宝寺緑地	**0.7719**	**大阪府**	**公園・庭園**
淀川河川公園	0.7686	京都府	公園・庭園
服部緑地	**0.7660**	**大阪府**	**公園・庭園**
千葉県立柏の葉公園	0.7647	千葉県	公園・庭園

表 25　提案手法においてα＝7として演算を行った結果

観光地名	コサイン類似度	都道府県	カテゴリ名
浜寺公園	**0.7356**	**大阪府**	**公園・庭園**
久宝寺緑地	**0.7346**	**大阪府**	**公園・庭園**
淀川河川公園	0.7326	京都府	公園・庭園
彩湖・道満 グリーンパーク	0.7307	埼玉県	公園・庭園
服部緑地	**0.7248**	**大阪府**	**公園・庭園**

表 26　提案手法においてα＝8として演算を行った結果

観光地名	コサイン類似度	都道府県	カテゴリ名
久宝寺緑地	**0.7013**	**大阪府**	**公園・庭園**
彩湖・道満 グリーンパーク	0.7009	埼玉県	公園・庭園
浜寺公園	**0.7008**	**大阪府**	**公園・庭園**
淀川河川公園	0.7004	京都府	公園・庭園
矢橋帰帆島公園	0.6922	滋賀県	公園・庭園

表 27　提案手法においてα＝9として演算を行った結果

観光地名	コサイン類似度	都道府県	カテゴリ名
彩湖・道満 グリーンパーク	0.6745	埼玉県	公園・庭園
淀川河川公園	0.6720	京都府	公園・庭園
久宝寺緑地	**0.6719**	**大阪府**	**公園・庭園**
浜寺公園	**0.6703**	**大阪府**	**公園・庭園**
矢橋帰帆島公園	0.6654	滋賀県	公園・庭園

表 28　提案手法においてα = 10として演算を行った結果

観光地名	コサイン類似度	都道府県	カテゴリ名
彩湖・道満 グリーンパーク	0.6511	埼玉県	公園・庭園
淀川河川公園	0.6470	京都府	公園・庭園
久宝寺緑地	**0.6462**	**大阪府**	**公園・庭園**
浜寺公園	**0.6436**	**大阪府**	**公園・庭園**
矢橋帰帆島公園	0.6418	滋賀県	公園・庭園

最後に，東京都に位置し神社・神宮・寺院カテゴリである明治神宮の要素から神社・神宮・寺院カテゴリを減算した場合の合成ベクトルの計算方法を(12)に示す.

$$\tilde{\mathbf{v}} = \mathbf{v}_{\text{明治神宮}} - \alpha \mathbf{v}_{\text{神社・神宮・寺院}} \tag{12}$$

観光地ベクトル S −カテゴリベクトル C_s において，カテゴリは指定せず，観光地 S の都道府県である P_s の観光地を似た観光地とする．実験結果において，基準を満たす観光地を赤字とした.

[3]と提案手法どちらも影響度パラメータα = 1の際の(12)と全ての観光地ベクトルでコサイン類似度を算出した．表 29 に[3]においてコサイン類似度を算出した結果，表 30 に提案手法においてコサイン類似度を算出した結果を示す.

表 29　[3]においてコサイン類似度を算出した結果

観光地名	コサイン類似度	都道府県	カテゴリ名
明治神宮内苑	**0.5389**	**東京都**	**公園・庭園**
熱田神宮	0.4291	愛知県	神社・神宮・寺院
北海道神宮	0.4193	北海道	神社・神宮・寺院
代々木公園	**0.4168**	**東京都**	**公園・庭園**
宮崎神宮	0.4103	宮崎県	神社・神宮・寺院

表 30　提案手法においてα = 1としてコサイン類似度を算出した結果

観光地名	コサイン類似度	都道府県	カテゴリ名
代々木公園	**0.3520**	**東京都**	**公園・庭園**
池袋西口公園	**0.3286**	**東京都**	**公園・庭園**
ハチ公の銅像	**0.3219**	**東京都**	**史跡・名所巡り**
皇居外苑	**0.2929**	**東京都**	**公園・庭園**
新宿中央公園	**0.2915**	**東京都**	**公園・庭園**

東京都に位置する明治神宮の要素から神社・神宮・寺院カテゴリを減算すると，東京都に位置する観光地という要素が残る．よって，結果としては東京都の観光地が出力されるのが望ましい．表 29 と比較して，表 30 では 1 位から 5 位全て東京都の観光地となった．東京都の観光地が 1 位から 5 位を占めたことから，

(12)と$\mathbf{v}_{東京都}$でコサイン類似度を算出した．結果として，値は 0.1215 となった．

よって，東京都の要素が少ないことが分かったため，明治神宮という独自の要素が強く出ているのではないかと考えられる．また，
明治神宮は自然に囲まれた観光地であり，囲まれた自然の中に明治神宮の本殿や様々な施設が位置している．このことから，明治神宮が位置している環境に似た観光地として，1 位と 2 位，4 位と 5 位は公園・庭園カテゴリに属する観光地が出力されたと考えられる．

結論

[1]では，インターネットから取得した観光地周辺の施設情報，地図画像等から観光地をベクトルで表現し，得られた観光地ベクトルと他の観光地ベクトルのコサイン類似度を用いて類似した観光地を推薦する手法を提案している．しかし，この手法では生成された観光地ベクトルが定量的な情報源から形成されており，観光地の情緒的な特徴を考慮していない問題がある．

この問題点に対して[3]では PV-DBOW に観光地レビューを入力することで観光地をベクトルで表し，複数の特徴ベクトルから観光地間の類似性を評価することで観光地を推薦している．しかし，PV-DBOW へあらかじめ膨大な量の観光地レビューを学習させなければならないという問題点があった．

本研究では，[3]の問題点を解決するため，事前に大量の観光地レビュー文章で学習した事前学習済み LLM を用いて観光地のレビュー文章をベクトルに変換することで観光地をベクトルで表現する手法を提案した．提案手法は，従来手法と比べ，レビュー文章をベクトルに変換するモデルを学習する必要が無いという利点がある．

実験では，事前学習済み LLM として日本語を事前学習済みの Sentence-LUKE である sentence-luke-japanese-base-lite に，じゃらんからスクレイピングによって取得した観光地レビューを入力することで観光地をベクトルで表した．得られた観光地ベクトル，カテゴリベクトル，都道府県ベクトル同士を演算するにあたり，観光地の各レビューを sentence-luke-japanese-base-lite に入力し，出力されたベクトルを平均する．平均されたベクトルを用いて演算を行い，得られた合成ベクトルと全ての観光地を表すベクトルでコサイン類似度を計算し，類似度が高いものを推薦した．

また，従来手法と同様の条件で観光地を推薦した結果を比較しており，影響度パラメータ α を適切に設定することで従来手法よりも推薦結果が改善することを確認した．

今後の課題として，定量的に観光地を推薦する結果を評価すること，住所や周辺の観光施設といった定量的な情報源を考慮して観光地をベクトルで表すことが挙げられる．

謝辞

本研究を進めるにあたり，卒業研究を指導していただいた二宮 洋先生，マハブービ シェヘラザード先生，渡辺 重佳先生，佐々木 智志先生，齋藤 友彦先生，鎌塚 明先生には終始ご指導ご鞭撻を賜りました．感謝申し上げます．

また，日頃の議論を通じて知識やアドバイスをいただいた，堀 雄介先輩，山富 龍先輩，学科横断型学修プログラム AI コースの皆様には，深謝の意を表します．

最後に，二宮研究室の皆様には，本研究の遂行にあたり多大なご助言，ご協力いただきました．感謝いたします．

参考文献

[1] 上原尚, 嶋田和孝, and 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." 信学技報, NLC2012-35, Dec (2012).

[2] Le, Quoc, and Tomas Mikolov. "Distributed representations of sentences and documents." International conference on machine learning. PMLR, 2014.

[3] 吉田 朋史，and 北山 大輔．"ユーザレビューの分散表現を用いた役割的に類似する観光スポット検索手法″， 観光情報学会;第 14 回研究発表会講演論文集 pp.78-81 (2016)

[4] 斎藤 康毅（2016），ゼロから作る Deep Learning ―Python で学ぶディープラーニングの理論と実装，オライリージャパン

[5] Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781* (2013).

[6] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).

[7] Brown, Tom, et al. "Language models are few-shot learners." *Advances in neural information processing systems* 33 (2020): 1877-1901.

[8] Hugging Face，”CyberAgent”，https://huggingface.co/cyberagent （最終アクセス:2024 年 1 月 31 日）.

[9] Hugging Face，”sonoisa/sentence-luke-japanese-base-lite”，https://huggingface.co/sonoisa/sentence-luke-japanese-base-lite （最終アクセス:2024 年 1 月 31 日）.

[10] Hugging Face，”sonoisa/sentence-bert-base-ja-mean-tokens-v2”，https://huggingface.co/sonoisa/sentence-bert-base-ja-mean-tokens-v2 （最終アクセス:2024 年 1 月 31 日）.

[11] Hugging Face,”MU-Kindai/SBERT-JSNLI-large”，https://huggingface.co/MU-Kindai/SBERT-JSNLI-large （最終アクセス：2024 年 1 月 31 日）

[12] Yamada, Ikuya, et al. "Luke: deep contextualized entity representations with entity-aware self-attention." arXiv preprint arXiv:2010.01057 (2020)

[13] じゃらん，”観光ガイド”，https://www.jalan.net/kankou/ （最終アクセス:2024 年 1 月 31 日）.