卒業論文

観光分野における単語の分散表現のモデル構築に用いた情報源の差異がもたらす影響の分析

関谷　虎汰郎

2025 年 3 月 28 日

岐阜大学 工学部 電気電子・情報工学科 情報コース
鈴木研究室

本論文は岐阜大学工学部に
学士（工学）授与の要件として提出した卒業論文である.

関谷　虎汰郎

指導教員：
鈴木 優 特任准教授

観光分野における単語の分散表現のモデル構築に用いた情報源の差異がもたらす影響の分析*

関谷　虎汰郎

内容梗概

役割が同じ POI を提示することで旅行などのヒントになると考えた．役割が同じ POI の提示に意味演算を用いる．事実に基づく情報が記述されていると考えられる Wikipedia を情報源とし単語の分散表現のモデル構築を行うことで事実に基づく情報が一致した POI を提示することができると考えた．また，意見や感想が記述されていると考えられる Twitter を情報源とし単語の分散表現のモデル構築を行うことで意見や感想が一致した POI を提示することができると考えた．本研究では，Wikipedia と Twitter を情報源として単語の分散表現のモデル構築を行い，その出力を分析することで単語の分散表現のモデル構築に用いた情報源の差異がもたらす影響を示す．

キーワード

分散表現，Word2vec，fastText，観光，Twitter，Wikipedia

*岐阜大学 工学部 電気電子・情報工学科 情報コース 卒業論文, 学籍番号: 1173033083, 2025 年 3 月 28 日.

目次

図目次 iv

表目次 v

第 1 章 はじめに 1

第 2 章 基本的事項 2

Word2vec 2

fastText 2

第 3 章 関連研究 3

第 4 章 実験手法 4

4.1 異なる情報源からの収集 4

4.1.1 手法 1:情報源を Twitter とした学習データ 4

4.1.2 手法 2:情報源を Wikipedia とした学習データ 5

4.2 単語の分散表現のモデル構築 5

4.2.1 Word2vec 5

4.2.2 fastText 6

4.3 役割が同じ POI 順序リストの出力 6

4.3.1 所在地を演算するクエリ 7

4.3.2 所在地と注目したい情報を演算するクエリ 7

第 5 章 評価実験 8

5.1 手順 8

5.1.1 データセット 8

5.1.2 評価基準 8

Wikipedia を情報源とした際の評価基準 8

Twitter を情報源とした際の評価基準 9

5.1.3 クエリ 9

所在地を演算するクエリ 9
所在地と注目したい情報を演算するクエリ 10
5.1.4 実験結果 10
5.1.5 所在地を演算させるクエリの結果 11
5.1.6 所在地と注目したい情報を演算させるクエリの結果 12

第 6 章 おわりに 15

謝辞 16

参考文献 17

発表リスト 18

図目次

4.1 役割が同じ POI 出力の流れ 7

5.1 所在地を演算させるクエリの結果 13
5.2 所在地と注目したい情報を演算させるクエリの結果 13

表目次

5.1 所在地を演算させるクエリの結果を Wikipedia 基準で評価したときの平均精度．W は所在地だけを演算させるクエリ結果を Wikipedia 基準で評価したときの平均精度，T は同じ結果を Twitter 基準で評価したときの平均精度，W’ と T’ はそれぞれ所在地に加えて注目したい情報を考慮した場合を表す． 14

第 1 章　はじめに

旅行などで他府県へ赴く場合を考える．この時，赴いた先にあるショッピングセンターや城といった場所に何があるのかやどのような歴史があるのかといった事実に基づく情報が分からないとき Web 検索システムを用いて調べることで理解を得ることが出来る．しかし，その場所に対する意見や感想といった情報を調べるには多くの人の個人的な意見や感想を集め，総合的に判断する必要がある．これら両方の情報を調べるには多くの手間が必要となる．そこで，両方の情報が対象となる場所と一致している場所を提示することで対象となる場所の理解を促すシステムを考える．これにより，対象となる場所への理解が容易に得られるのではないかと考えた．一致した場所を提示する手法として意味演算を用いる．単語をベクトルで表現した単語の分散表現を用いることで“king - woman + man” のベクトル演算は“queen”のベクトルに近くなるという意味の演算が出来る．対象となる場所と所在地で演算を行うことで役割が同じ場所を提示することができると考えた．また，単語の分散表現はニューラル言語モデルを構築することで取得できる．モデルの構築に用いるデータの収集元を情報源とする．事実に基づく情報が記述されている情報源から単語の分散表現のモデルを構築することで事実に基づく情報が一致した場所が出力されると考えた．また，意見や感想といった情報が記述されている情報源から単語の分散表現のモデルを構築することで意見や感想といった情報が一致した場所が出力されると考えた．本研究では，異なる情報源を用いて単語の分散表現のモデル構築を行い，対象とする場所と同じ役割を持つ場所を提示するシステムを構築する．このシステムの出力結果を情報源ごとに比較することで情報源の差異による影響を分析した．

本論文の構成は以下の通りである．2 章では基礎事項ついて述べる．3 章では先行研究について述べる．4 章では実験手法について述べる．5 章では評価実験の結果と考察を述べる．最後に 6 章では本論文のまとめと今後の課題について述べる．

第 2 章　基本的事項

Word2vec

Word2vec とは Mikolov ら [1] によって提唱された二層のニューラルネットワークモデルを用いて単語の分散表現を獲得する手法である．「類似した文脈で出現する単語は類似した意味を持つ」という分布仮説に基づいている．Word2vec のモデルとして Skip-gram と CBOW が存在する．Skip-gram はある単語の周辺文脈を予測するタスクを行うことで分散表現を獲得するモデルであり，CBOW は周辺の単語からある単語を推測するタスクを行うことで分散表現を獲得するモデルである．得られた分散表現は，類似した意味を持つ単語のベクトル間の距離は少なくなる．また，“king - woman + man” の単語のベクトル演算を行った結果は “queen” に近くなる．

fastText

fastText は Word2vec を考案した Mikolov ら [2][3] によって提唱された手法である．fastText は Word2vec を拡張した手法である，単語の構成要素であるサブワードを用いて学習を行い，同じサブワードを持つ単語は類似した意味を持つ単語であるとすることで活用形を考慮することができる．未知語に対してもサブワードを用いることでベクトルを表現することが出来る．fastText のモデルは Word2vec と同じ Skip-gram と CBow が存在する．fastText は Word2vec と比べ短時間で学習を行うことが出来る．

第 3 章　関連研究

単語の分散表現を用いて類似したものを提供する手法として様々な技術が研究されている．高橋ら [4] は飲食店の店舗ごとのベクトルを定義し，キーワードと演算することで店舗を検索する研究を行った．飲食店のレビュー文から単語の分散表現を取得し，店舗を表す単語のベクトルと店舗ごとのカテゴリを表すベクトルを合わせて店舗ベクトルを作成する．作成したベクトルにキーワードのベクトルを演算させることで細かい要求を検索結果に反映させるシステムを構築した．赤木ら [5] は類似した報告書を提示する研究を行った．職業支援システムの報告書から単語の分散表現を取得する．クエリとして入力した単語の類似単語の出現回数とコサイン類似度を基準に類似した報告書を提示するシステムを提示した．　類似した観光スポットを提示する研究も複数存在する．住友ら [6] は口コミ情報に含まれている感情語を感情スコアデータベースを用いてスコア化し，感情毎に合計したスコアの割合を基準に類似した観光スポットを抽出する研究を行った．長谷川ら [7] は Twitter から地域の特徴を表す特徴語を抽出することで特徴語辞書を作成し，これを用いて観光体験を検索する研究を行った．分散表現を観光分野に用いた研究として，野島ら [8] はある地域から次の地域への移動のみを学習し，分散表現のモデルを構築することで移動先の地域が類似している地域を発見する研究を行った．開地ら [9] は観光に関する質問，回答文を用いて単語の分散表現のモデルを構築し，観光地に対するイメージなどを表す単語のベクトルを加算することで別の観光地を推薦する研究を行った．土田ら [10] は Twitter から収集したテキストデータから単語の分散表現のモデルを構築し，都市・地域とランドマークの意味演算を行うことで関係性を抽出する研究を行った．

本研究では事実に基づく情報が記述されている情報源と意見や感想といった情報が記述されている情報源を用いること両方の情報が一致している場所を提示することが出来ると考え，情報源の差異がもたらす影響を示す．

第 4 章　実験手法

図 4.1 にシステムの全体図を示す．以降，施設名などの本研究で対象としている場所を POI(Point-of-Interest) と記述する．この実験では，異なる情報源を用いて単語の分散表現のモデル構築を行い，出力された POI を比較することで情報源の差異がもたらす影響を示す．

4.1　異なる情報源からの収集

異なる情報源として Twitter と Wikipedia を用いる．Wikipedia には POI の所在地や歴史といった事実に基づく情報が記述されている．また，Twitter には POI へ赴いた際の意見や感想といった情報が記述されていると考えられる．これらの異なる情報源から収集したテキストデータを用いて単語の分散表現のモデル構築を行うことで事実に基づく情報が一致した POI と意見や感想といった情報が一致した POI を出力として得られると考えられる．以下の二つの手法に従って作成した学習データを用いて単語の分散表現のモデル構築を行う．

- 手法 1：情報源を Twitter とした学習データ
- 手法 2：情報源を Wikipedia とした学習データ

4.1.1　手法 1:情報源を Twitter とした学習データ

(1) TwitterStreamingAPI を用いて Tweet を収集し，本文テキストを抽出する．
(2) 収集した Tweet 本文のテキストデータから正規表現を用いて記号と数字を削除し，ひらがな，カタカナ，漢字を抽出することで日本語以外を取り除く．
(3) Mecab を用いて分かち書きを行うことで学習データとする．辞書は IPA 辞書と NEologd 辞書とする．

4.1.2　手法 2:情報源を Wikipedia とした学習データ

(4) Wikipedia が提供している日本語版のダンプデータ*から最新日本語記事の本文のテキストデータを xml 形式でダウンロードする.
(5) ダウンロードデータから WikiExtractor†を用いて xml タグや Wikipedia がデータ管理の為に用いるタグを取り除く.
(6)　残ったタグや空行を分散表現を用いて取り除く.
(7) 手順 1 の (3) と同様に, MeCab を用いて分かち書きを行ったものを学習データとする.

4.2　単語の分散表現のモデル構築

単語の分散表現のモデル構築として，以下の二つの手法を用いる.

- 手法 1：Word2vec
- 手法 2：fastText

4.2.1　Word2vec

Word2vec は「類似した文脈で出現する単語は類似した意味を持つ」という分布仮設に基づき，二層のニューラルネットワークを用いてある単語とその単語の周辺の文脈情報から単語の分散表現を獲得する手法である．今回の実験では，文脈中で出現するある単語から，その前後に出現する単語を予測するタスクを通して単語の分散表現を獲得する Skip-gram を用いる．POI を表す単語の周辺には POI の情報が出現すると考えられるため，役割が同じ POI は周辺単語も類似したものとなる．したがって，得られた単語の分散表現を用いることで役割が同じ POI が出力されると考えられる．オープンソースライブラリである gensim の Word2vec を用いて Python で実装する．使用するモデルは Skpi-gram で，単語ベクトルの次元数を

*https://dumps.wikimedia.org/jawki/
†https://github.com/attardi/wikiextractor

300 次元，前後 10 単語内に出現する単語を予測するタスクを行う．4.1 章で作成した学習データで学習を行い，入力層と隠れ層間の重みを用いて単語の分散表現を獲得する．

4.2.2 fastText

fastText は Word2vec の学習手法に加え，対象とする単語の構成要素であるサブワードを用いた学習により，活用形や表記ゆれを考慮に入れた単語の分散表現を獲得することが出来る．そのため，Twitter といった表記方法が統一されていないデータを扱うのに適していると考えられる．Fecebook が開発したライブラリである fastText[‡]を用いて実装する．単語ベクトルの次元数を 300 次元，前後 10 単語内に出現する単語を予測するタスクを行う．4.1 章で作成した学習データで学習を行う．

4.3 役割が同じ POI 順序リストの出力

単語の分散表現は，意味的に類似した単語はベクトル空間上の近い位置に配置される．また，単語のベクトルを用いて“king - woman + man”というベクトル演算を行うと “queen”のベクトルと近くなるという意味演算を行うことが可能である．この演算を用いることで役割が同じ POI 順序リストを出力する．対象とする POI を表す単語と考慮したい情報を表す単語をクエリとして入力することでベクトル演算を行い，演算結果のベクトルとコサイン類似度の高いベクトルを持つ単語リストがコサイン類似度の降順で出力される．今回の実験では以下の二つのクエリを実行する．

- 所在地を演算させるクエリ
- 所在地と注目したい情報を演算させるクエリ

[‡]https://github.com/facebookresearch/fastText

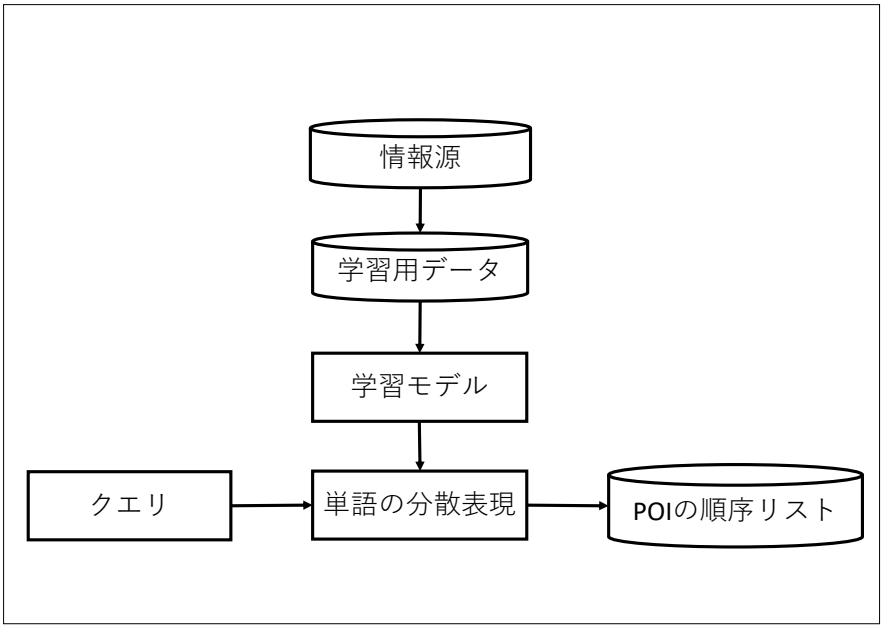


図 4.1　役割が同じ POI 出力の流れ

4.3.1　所在地を演算するクエリ

対象となる POI を一つ決める．その POI の単語ベクトルからその POI の所在地の単語のベクトルを減算し，指定する所在地を加算することで指定した所在地にあり，また，対象とした POI と同じ役割を持つ POI を出力することが出来ると考えられる．今回の実験では，所在地を都道府県として行った．岐阜城-岐阜県 + 愛知県をクエリとして入力することで名古屋城といった愛知県にある岐阜城と同じ役割を持つ場所が高いコサイン類似度で出力されると考えられる．

4.3.2　所在地と注目したい情報を演算するクエリ

所在地を演算するクエリに加え，注目したい情報の単語ベクトルを加算することにより，出力される POI が注目したい情報を重視した出力になると考えられる．岐阜城-岐阜県＋愛知県 + 織田信長とすることで小牧山城といった愛知県のお城でかつ織田信長が関わっている場所が高いコサイン類似度で出力されると考えらる．

第 5 章　評価実験

5.1　手順

5.1.1　データセット

単語の分散表現のモデル構築のための学習データとして Twitter から 2020 年 1 月上旬から 11 月上旬までの期間で収集した 175,767,614 件のツイートを使用する．また，Wikipedia から 11 月上旬の最新全日本語記事の本文データを使用した．

5.1.2　評価基準

Wikipedia と Twitter を情報源として構築した単語の分散表現が出力する結果は異なる基準で役割が同じ POI が出力されると考えられる．そのため，出力結果に対する評価基準を情報源ごとに定める必要がある．

Wikipedia を情報源とした際の評価基準

(1) 公園である．
(2) 指定した所在地にある．
(3) 薔薇がある．

Wikipediak から収集したテキストを用いて単語の分散表現のモデル構築を行うことで事実に基づく情報が一致する POI が出力されると考えた．事実に基づく情報として今回の実験では，POI の分類と所在地，POI の設備を対象とする．fastText の出力はサブワードの影響によりクエリに用いた単語の文字列と共通項が多いほど類似度が高くなる．駿府城をクエリとして入力とした場合，駿府城を含む単語である駿府城公園が類似度が高くなり，リストの上位に出力されると考えられる．公園には名称に公園という文字列を含むものが多く，公園であるという条件を満たす POI が出力されやすいと考えた．公園について記述された Wikipedia のページには所在地と公園内設備について記述されているものが多く，公園内設備の中で今回は薔薇という情報に注目して評価した．Twitter を情報源としたモデルと比べ，

Wikipedia を情報源としたモデルを用いたほうが基準を満たす POI が出力されやすいと考えられる．以降，この基準を Wikipedia 基準とする．

Twitter を情報源とした際の評価基準

(1) 景観がきれいな POI

Twitter には POI に対する意見や感想といった情報が記述されていると考えられる．Twitter を情報源として単語の分散表現のモデル構築を行うことで意見や感想といった情報が一致した POI が出力されると考えた．今回の実験では感想として景観がきれいな場所とした．第一著者が景観がきれいな POI かを判別し今回の実験の評価基準とした．Wikipedia を情報源としたモデルと比べ，Twitter を情報源としたモデルのほうが基準を満たす POI が出力されやすいと考えられる．以降，この基準を Twitter 基準とする．

5.1.3 クエリ

5.1.2 章で定めた基準を全て満たす POI として神奈川県にある山下公園を例とする．指定する所在地を東京都，愛知県，兵庫県，広島県，鹿児島県とした．今回用いたクエリは以下の通りである．

所在地を演算するクエリ

山下公園から山下公園の所在地を減算し，指定する所在値を加算することで，指定した所在地にあり，所在地が異なる役割が同じ POI が出力されると考えた．

(1) 山下公園-神奈川県 + 東京都
(2) 山下公園-神奈川県 + 愛知県
(3) 山下公園-神奈川県 + 兵庫県
(4) 山下公園-神奈川県 + 広島県
(5) 山下公園-神奈川県 + 鹿児島県

所在地と注目したい情報を演算するクエリ

所在地を演算するクエリの出力は所在地以外の情報が一致していない POI が出力されると考えられる．そこで，注目したい情報として薔薇を加算する．これにより，薔薇に関連のある POI のコサイン類似度が高くなると考えた．

は

(6) 山下公園-神奈川県 + 東京都 + 薔薇
(7) 山下公園-神奈川県 + 愛知県 + 薔薇
(8) 山下公園-神奈川県 + 兵庫県 + 薔薇
(9) 山下公園-神奈川県 + 広島県 + 薔薇
(10) 山下公園-神奈川県 + 鹿児島県 + 薔薇

5.1.4 実験結果

Wikipedia と Twitter から作成した学習データから Word2vec と fastText を用いて四種類の単語の分散表現を獲得する．各クエリごとに 50 件をコサイン類似度の降順で出力する．出力した POI 順序リストのうち閾値以上の順位であった POI を役割が同じ POI とし，閾値を変化させていきながら五府県の結果の精度を計算する．二種類のクエリと用いた単語の分散表現毎に平均したものを横軸を再現率，縦軸を精度である PR 曲線にプロットする．また，平均精度を表**??**に示す．PR 曲線において，Wikipedia を情報源として fastText でモデル構築を行った単語の分散表現を”Wikipedia，fastText”と記述し，Twitter を情報源，fastText をモデル構築を行った単語の分散表現を”Twitter，fastText”と記述し，Wikipedia を情報源とし，Word2vec でモデル構築を行った単語の分散表現を”Wikipedia，Word2vec”と記述し，Twitter を情報源とし，Word2vec でモデル構築を行った単語の分散表現を”Twitter，Word2vec”と記述する．

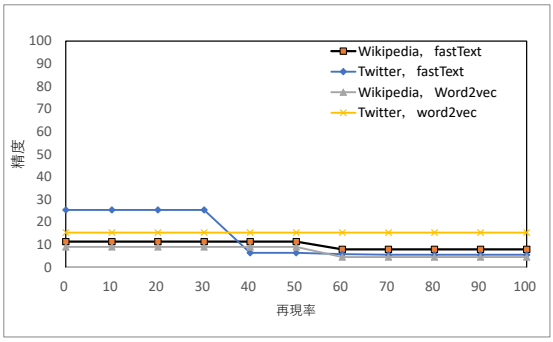
5.1.5　所在地を演算させるクエリの結果

クエリの (1)〜(5) の結果を単語の分散表現毎に平均を算出する．Wikipedia 基準で評価した PR 曲線を 5.1 の (a) に示す．また，Twitter 基準で評価した PR 曲線を 5.1 の (a) に示す．Wikipedia 基準で評価したときの平均精度で比較した際，最も高いのは Twitter を情報源とし，Word2vec でモデルを構築した単語の分散表現のときとなった．最も低いのは Wikipedia を情報源とし，Word2vec でモデルを構築した単語の分散表現のときとなった．また，fastText を用いた単語の分散表現は情報源の差異で大きな違いは見られなかった．Twitter 基準で評価したときの平均精度で比較した際，最も高いのは Twitter を情報源とし，fastText でモデルを構築した単語の分散表現のときとなった．次に高いのは Twitter を情報源とし，Word2vec でモデルを構築した単語の分散表現のときとなった．Wikipedia を情報源とした単語の分散表現は fastText と Word2vec で大きな差は見られなかった．Twitter を情報源とした際，Wikipedia k を情報源にしたときと比べ，両方の基準で平均精度が同じか少し上回る結果となった．これは今回 Wikipedia 基準として用いた条件について Twitter にも記載があるためと考えた．Wikipedia を情報源とし，fastText でモデル構築を行った単語の分散表現のとき，出力したリストの上位には加算した所在地の影響がみられた．加算した所在地が広島県とすると，広島南が最初に出力されており，以降広島とついた単語が複数出力されている．これは fastText のサブワードによる影響であると考えられる．また，公園である POI の出力が多くみられたが，どちらの基準も満たしていないことが多かった．こちらもサブワードによる影響であり，公園であるだけで他の情報を考慮せずにコサイン類似度が高くなっていると考えられる．埠頭や港が出力の中に幾つか見ることが出来た．これは山下公園の薔薇以外の情報である海を埋め立てて出来た公園である，あるいは海に面している場所であるという情報に注目して出力された結果であると考えられる．Twitter を情報源とし，fastText でモデル構築を行った単語の分散表現のとき，Wikipedia のときと同様にサブワードの影響を受けた出力が多く見られた．Wikipedia を情報源としたときと比べ，指定した所在地以外の公園が多く見られた．また，fastText のときと同様に埠頭や港が出力に見られた．Wikipedia を情報源とし，Word2vec でモデル構築を行った単語の分散表現のとき，加算した所在地と関係のある地名や道路の名前が出力に多く，fastText を用いた単語の分散表現

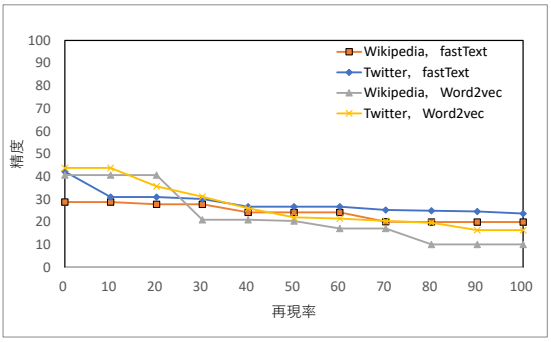
と比べ公園の出力が少なかった．Twitter を情報源とし，Word2vec でモデル構築を行った単語の分散表現のとき，出力される公園は少ないが，Wikipedia 基準を満たす POI は 10 番目以内で出力されており，平均精度が上がる原因となった．

5.1.6　所在地と注目したい情報を演算させるクエリの結果

クエリの (6)〜(10) の結果を単語の分散表現毎に平均を算出する．Wikipedia 基準で評価した PR 曲線を 5.2 の (a) に示す．また，Twitter 基準で評価した PR 曲線を 5.1 の (a) に示す．Wikipedia 基準で評価したときの平均精度で比較した際，最も高いのは Wikipedia を情報源とし，Word2vec でモデル構築を行った単語の分散表現のときとなった．最も低いのは Twitter を情報源とし，Word2vec でモデル構築を行った単語の分散表現のときとなった．Twitter 基準で評価したときの平均精度を比較した際，fastText で構築した単語の分散表現が両方の情報源で大きな差はなく，共に高かった．最も低いのは Twitter を情報源とし，Word2vec でモデル構築を行った単語の分散表現のときとなった．所在地を演算させるクエリと比べ，全体的に公園が出力されることが少なくなった．また，さくら広場といった薔薇ではなく花に関する出力が多くなった．Wikipedia を情報源とし，fastText でモデル構築を行った単語の分散表現のとき，出力したリストはサブワードの影響により加算した所在地に関する出力が多かったが，薔薇に関する出力はあまり見られなかった．しかし，埠頭や港といった出力が見られなくなった．これにより，クエリを変更することで別の情報を考慮した出力にすることが出来るが，どのような形で出力に影響を与えるかを確認することが出来なかった．Twitter を情報源とし，fastText でモデル構築を行った単語の分散表現のとき，Wikipedia を情報源としたときとは異なり，加算した所在地に関する出力が少なくなった．しかし，紫陽花や色とりどりといった花に関する出力が多く見られた．Wikipedia を情報源とし，Word2vec でモデル構築を行った単語の分散表現のとき，桜といった花に関する出力が多くなり，公園の出力が少なくなった．しかし，一部の Wikipedia 基準を満たす公園のコサイン類似度が高くなったため Wikipedia 基準では平均精度が上がり，Twitter 基準では減少した．Twitter を情報源とし，Word2vec でモデル構築を行った単語の分散表現のとき，出力に花に関する出力が多く，POI があまり見られなかった．そ

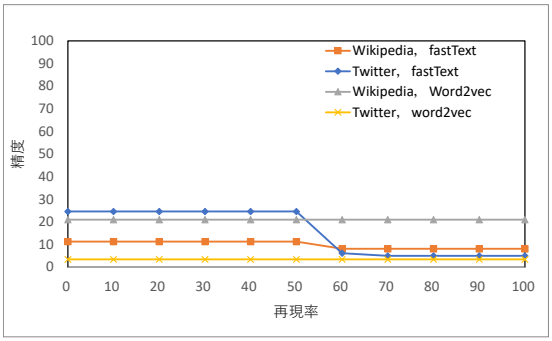


(a) wikipedia 基準で評価した場合

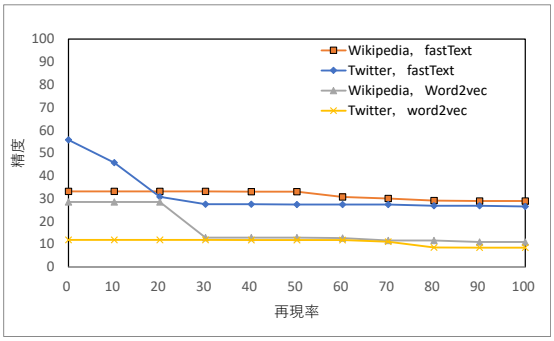


(b) Twitter を基準で評価した場合

図 5.1　所在地を演算させるクエリの結果



(a) wikipedia 基準で評価した場合



(b) Twitter を基準で評価した場合

図 5.2　所在地と注目したい情報を演算させるクエリの結果

れにより所在地を演算させるクエリと比べ両方の基準で平均精度が下がった.

表 5.1　所在地を演算させるクエリの結果を Wikipedia 基準で評価したときの平均精度．W は所在地だけを演算させるクエリ結果を Wikipedia 基準で評価したときの平均精度，T は同じ結果を Twitter 基準で評価したときの平均精度，W’ と T’ はそれぞれ所在地に加えて注目したい情報を考慮した場合を表す．

	W	T	W’	T’
Wikipedia，fastText	12.9	21.1	13.1	26.2
Twitter，fastText	12.5	25.3	14.4	26.5
Wikipedia，Word2vec	6.6	21.4	21.0	13.6
Twitter，Word2vec	15.4	23.5	3.1	9.3

第 6 章　おわりに

本研究では，単語の分散表現のモデル構築に情報源として Twitter と Wikipedia を用いることでその差異がもたらす影響についてと用いる単語の分散表現のモデル構築手法と入力するクエリについての比較を行った．実験により，今回用いた評価指標とクエリにおいて，精度が高い情報源を示した．また，クエリとして加算した単語に出力が大きな影響を受け，加算部を増やすほど対象とする POI の情報が考慮されにくくなるといった問題が分かった．さらに，fastText を用いることで対象とする POI と同じ分類の POI が出力されやすいが，加算した単語と文字の並びが類似している単語のコサイン類似度が高くなり，平均精度が下がるという問題が分かった．　今後の研究として，クエリの影響を分析することでタスクに適したシステムの構築手法を提示することが考えられる．

謝辞

本論文の執筆にあたり，最後まで多くのご指導していただいた指導教授である鈴木優特任准教授．様々な事務的な手続きを手伝っていただいた秘書の井尾さん．研究についての多くの助言をいただいた研究室の皆さん．そして，最後まで支えていただいた家族．皆さんのご助力により研究を行うことができました．上手に研究を進めることができず，大きく遅れることとなりましたが一応の終わりを迎えることが出来ました．心よりの感謝を申し上げます．

参考文献

[1] Kai Chen Greg S Corrado Tomas Mikolov, Ilya Sutskever, Jeff Dean. Distributed representations of words and phrases and their compositionality. *In Advances in neural information processing systems*, pp. 3111–3119, 2013.

[2] Piotr Bojanowski Tomas Mikolov Armand Joulin, Edouard Grave. Bag of tricks for efficient text classification. *Facebook AI Research*, 2016.

[3] Piotr Bojanowski, Edouard Grave, Armand Joulin, Tomas Mikolov. Enriching word vectors with subword information. *Facebook AI Research*, pp. 3111–3119, 2016.

[4] 高橋輝, 北山大輔. 店舗の分散表現に対する意味演算を用いた飲食店検索手法. 第 12 回データ工学と情報マネジメントに関するフォーラム, 2020.

[5] 赤木里騎, 徐海燕. キャリアポートフォリオデータの活用による推薦精度の向上. 第 10 回データ工学と情報マネジメントに関するフォーラム, 2018.

[6] 住友千将, 石野拓哉, 久保洸貴, 岳五一. 口コミ情報に含まれる感情語に基づく類似スポット推薦システムの構築と実証実験. パーソナルコンピューター利用技術学会論文誌, pp. 29–35, 2020.

[7] 長谷川馨亮, 馬強, 吉川正俊. Twitter からの地域特徴語辞書の構築とその観光情報検索への応用. 第 6 回データ工学と情報マネジメントに関するフォーラム, 2014.

[8] 野島僚太, 廣田雅春, 石川博. 位置情報による分散表現を用いたユーザの移動の分析. 第 11 回データ工学と情報マネジメントに関するフォーラム, 2019.

[9] 開地亮太, 檜垣泰彦. 単語の分散表現を使用した観光地推薦システムの構築. 信学技報, Vol. 115, No. 486, pp. 45–50, 2016.

[10] 土田崇仁, 遠藤雅樹, 加藤大受, 江原遥, 廣田雅春, 横山昌平, 石川博. Word2vec を用いた地域やランドマークの意味演算. 第 8 回データ工学と情報マネジメントに関するフォーラム, 2016.

発表リスト

[1] 関谷虎汰郎，鈴木優『Tweet の分類による観光情報の取得』第 19 回情報科学技術フォーラム, 2020.
[2] 関谷虎汰郎，鈴木優『Tweet の分類による観光情報の取得』東海関西データベースワークショップ, 2020.