地理的エンティティ情報が与えられた文書ジオロケーションモデルの有効性検証

山本祐耶
筑波大学情報学群
s2012003@s.tsukuba.ac.jp

乾孝司
筑波大学大学院 システム情報工学研究群
inui@cs.tsukuba.ac.jp

概要

SNS 投稿文書などの投稿位置を推定する文書ジオロケーション課題では、文書内に含まれる地名（「茨城」）やランドマーク（「ディズニーランド」）への言及が有力な手がかりになることが多いが、これら言及のみから常に十分な情報が得られるとは限らない。本研究では、これらの言及情報をより効果的に活用するため、各言及からそれらが指し示す実世界上の実体（エンティティ）を同定し、同定されたエンティティに関する情報を利用することを考える。評価実験を通して、エンティティ情報のうち、エンティティの所在情報に注目して文書ジオロケーションモデルに情報を取り込むことで、文書ジオロケーションの性能が改善することが確認できた。

1 はじめに

位置情報が付与された SNS 投稿文書は、ソーシャルセンシングに欠かせない情報源であるものの、SNS 投稿文書のうち位置情報が付与された文書はその一部に過ぎないという問題がある。この問題に対して、位置情報が付与されていない SNS 投稿文書が与えられた時に、その文書に対応する位置情報を推定する文書ジオロケーションに関する研究が進められている [1, 2, 3]。

文書ジオロケーション課題では、文書内に含まれる地名やランドマークなどへの言及が有力な手がかりになることが多いが、言及のみから常に十分な情報が得られるとは限らない。例えば、千葉県にある東京ディズニーランドを訪れた旅行者が『ディズニーランドに到着！』と SNS に投稿したと仮定する。この投稿の投稿位置は言及「ディズニーランド」に関連することが予想されるものの、言及「ディズニーランド」が手がかりとして十分に機能するには、これが千葉県に所在する東京ディズニーランドへの言及であるのか、あるいは米国カリフォルニアに所在する Disneyland への言及であるのかを文書ジオロケーションモデルが把握できていることが望ましい。

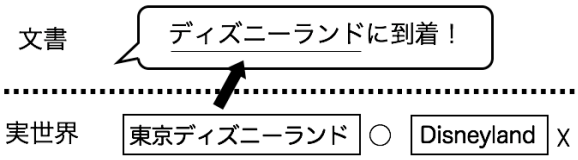

図 1 実世界上の実体を同定しモデルで利用

文書ジオロケーションに外部知識を利用する研究は既にあるものの [4, 5]、実世界上の実体（エンティティ）を同定し、その情報の利用に注目した議論はなされていない。そこで本研究では、文書内に含まれる言及からそれらのエンティティを同定し、同定されたエンティティの情報を文書ジオロケーションに利用することに注目する（図 1）。具体的には、エンティティ情報を文書ジオロケーションモデルに取り込むにあたり、エンティティ情報のうちどのような情報を、どのように埋め込み表現に変換し、どのようにモデルに取り込むかについて議論する。

2 構成要素

本稿の主要な内容に入る前に、本研究の構成要素について説明する。

地理的エンティティ：文書ジオロケーションでは地名やランドマークなどの地理的位置に関する地理的エンティティが特に重要であると考えられる。そこで、本研究では地理的エンティティに対象を絞って検討する。具体的には、東北大学で公開されている日本語 Wikipedia エンティティベクトル[1] に収録されているエンティティのうち、森羅プロジェクト [2] の拡張固有表現ラベル [3] で組織名、地名、施設

1） https://www.cl.ecei.tohoku.ac.jp/~m-suzuki/jawiki_vector/
2） https://2022.shinra-project.info/
3） http://ene-project.info/

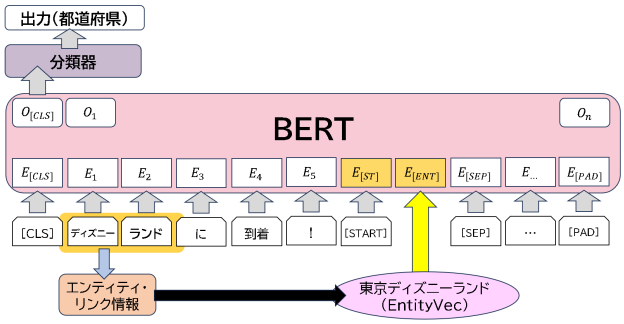

図 2　エンティティ情報の取り込み例

名、イベント名に割り当てられるエンティティを地理的エンティティとして用いた。

文書ジオロケーション：入力文書に対して 47 クラスの都道府県を出力する都道府県レベルの文書ジオロケーション課題を扱う。さきほどの例『ディズニーランドに到着！』の場合、千葉県が期待される出力クラスとなる。また、文書ジオロケーションのモデルには BERT に基づく文書分類モデル [6] として、Huggingface [4] で公開されている BertForSequenceClassification を基本モデルとして採用した。本モデルの詳細設定を付録 A.1 節に示す。

エンティティ・リンキング：文書内のある言及（メンション）と実世界のエンティティを結びつける課題はエンティティ・リンキング課題と呼ばれ、特にエンティティとして Wikipedia ページを仮定した Wikification[7] に関する研究活動が盛んである。本研究でも Wikification を前提にして、Wikipedia ページの情報をエンティティ情報として文書ジオロケーションモデルに取り込む。使用した Wikipedia データ [5] は 2023 年 8 月に取得したダンプデータである。

3　エンティティ情報の取り込み

文書ジオロケーションモデルにエンティティ情報を取り込む例を図 2 に示す。この図は、Wikification によって言及「ディズニーランド」から東京ディズニーランドのエンティティ情報が得られた場合の取り込み例である。本研究では、言及に対するエンティティが与えられた時、エンティティ情報から埋め込み表現を獲得する手法（図 2 の黒矢印に相当）として、以下の 4 種類を検討する。図 3 に埋め込み表現獲得手法の違いの概略をまとめる。この内、EntityVec と MentionVec は既存研究で採用されている方法である。一方、ConvertedEntityVec と

4）https://huggingface.co/
5）https://dumps.wikimedia.org/other/cirrussearch/

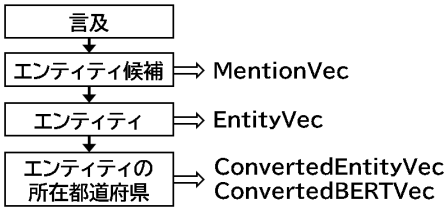

図 3　埋め込み表現獲得手法の違い

ConvertedBERTVec は、文書ジオロケーション課題向けに本研究で提案する新規な方法である。

- EntityVec[8]：エンティティ（Wikipedia ページ）の見出し語から埋め込み表現を獲得する。実装としては、日本語 Wikipedia エンティティベクトルを用いる。これは、Wikipedia のリンク情報を考慮した word2vec[9] によって埋め込み表現を学習したものである。
- ConvertedEntityVec：予備調査から、都道府県名を含む文書であれば、エンティティ情報を取り込むことなく良好な分類性能を達成できることを確認した。そこで、EntityVec のようにエンティティの見出し語を使うのではなく、エンティティの所在都道府県のエンティティ情報（都道府県エンティティ）を利用する。元エンティティを都道府県エンティティに変換したあとの処理は EntityVec と同様である。元エンティティの所在都道府県は、Okajima ら [10] を参考に、元エンティティである Wikipedia ページの本文に初めて登場する都道府県とした。
- ConvertedBERTVec：ConvertedEntityVec と同様にエンティティの所在都道府県情報を利用する。ただし、ConvertedEntityVec のように埋め込み表現の獲得に EntityVec を用いるのではなく、BERT への元入力テキストへ所在都道府県名を挿入する。この操作によって、挿入された都道府県名は元テキストとあわせて BERT の学習過程を通して埋め込み表現に変換される。
- MentionVec[11]：エンティティ情報を利用することの有効性を検証するための比較手法として、同定されたエンティティの情報ではなく、エンティティ候補群の情報から埋め込み表現を獲得する。具体的には、エンティティ候補ごとに EntityVec で埋め込み表現を獲得し、その平均ベクトルを用いる。

つぎに、上記のいずれかの方法で獲得された埋め込み表現の取り込み位置（図 2 の黄矢印に相当）と

して、以下の２種類を検討する。

- concat：トークン列の末尾に特殊トークン [START] を加え、その後ろにエンティティ情報を取り込む。これは中本ら [12] を参考にした方法である。もし、エンティティ情報が複数ある場合は言及の出現順に並べて取り込む。
- infuse：トークン列に対して、言及の直前に特殊トークン [MENTION] を挿入、かつ言及の直後に [START] と [END] で挟む形でエンティティ情報を取り込む。これは Faldu ら [13] を参考にした方法である。図 2 は EntityVec を infuse で取り込む例である。

4 評価実験

4.1 実験設定

第 2 節で述べた BERT に基づく文書分類モデルに対して、前節の方法でエンティティ情報を取り込んだモデルをそれぞれ構築し、それらの性能を比較する。事前学習およびファインチューニングの詳細設定は付録 A.2 節に示す。

文書ジオロケーション用の文書データには、観光ドメイン日本語 Twitter 投稿文書データ [14] を用いた。このデータセットは、2014 年から 2015 年にかけて 47 都道府県から投稿された日本語ツイートを基に構成されている。すべての投稿に付与されたジオタグを逆ジオコーディングすることで得られた投稿位置の都道府県情報を正解ラベルとして用いた。データセット中の文書数は、学習データが 197,741 件、検証データが 4,000 件、そして評価データが 7,000 である。

エンティティ情報の取得対象メンションは文書を GiNZA [6] で解析した際、場所を表す固有表現クラス [7] と認識されたフレーズとした。次に、あるメンション m に対して、陰山ら [11] に倣い、Wikipedia ページ内でメンション m がアンカー文字列としてリンクしているエンティティ e_i の集合 $E(m)$ を m のエンティティ候補とした。ただし、以下の条件を満たすものは、ノイズとなる可能性が高いため、候補から削除した。

1. e_i の見出し語と m の間に文字列的な包含関係がない。
2. e_i への総リンク回数に対して、m から e_i へのリンク回数が 1%未満である。

結果として、エンティティ候補数が 0 になる場合がある。この場合は、続くエンティティ同定の処理はおこなわない。また、後述の評価実験でメンション数を数える際、エンティティ候補数が 0 になるメンションは数えないことにした。

エンティティ情報の有効性を検証するためには、可能な限り正確な情報を利用することが望ましい。そこで、一部のメンションに対しては、人手で正確にエンティティを同定した。ただし、作業負荷の観点から全てのメンションへの人手作業は困難であるため、評価データに対しては人手で同定し、学習データに対しては自動的に同定することにした。人手同定では、候補生成時に得たリンク回数で並べられた候補の中からエンティティをひとつ選択する作業をおこなった。この際、必要に応じて該当する Wikipedia ページが参照できる作業環境を用意した。自動同定では、候補生成時に得たリンク回数が最も多いエンティティ候補を自動的に選択した。

評価尺度には以下の分類正解率を用いた。

$$\text{分類正解率} = \frac{\text{正しく分類できた文書数}}{\text{入力文書数}}$$

表 1 実験結果

	concat	infuse
MentionVec	74.71	74.80
EntityVec	75.34^{+}	75.30^{+}
ConvertedEntityVec	75.41^{+}	75.46^{++}
ConvertedBERTVec	76.06^{++}	75.86^{++}

4.2 実験結果

実験結果を表 1 に示す [8]。MentionVec とそれ以外の手法との間で符号検定を実施し、有意水準 5% で有意差があった結果に「+」、有意水準 1% で有意差があった結果に「++」を付けている。

6） https://megagonlabs.github.io/ginza/

7） すなわち、Airport,Amusement Park Archaeological Place Other Bay,Bridge,Canal,Car Stop,City,Company,Continental Region Corporation Other,Country,County,Domestic Region Earthquake,Facility Other,Facility Part,Game,Geological Region Other GOE Other,Government,GPE Other,International Organization Island,Lake,Location Other,Mountain,Museum,Occasion Other Organization Other,Park,Port,Postal Address,Pro Sports Organization Province,Public Institution,Railroad,Religious Festival,Research Institute,River,Road,School,Sea,Show Organization,Spa,Sports Facility,Sports League,Sports Organization Other Station,Theater,Tumulus,Tunnel,War,Water Route,Worship Place,Zoo.

8） 参考情報として、エンティティ情報を取り込まない純粋な BERT 文書分類モデルによる分類正解率は、74.33 であった。

表 2　文書内に含まれている言及数ごとの結果 (concat)

#言及	MentionVec	EntityVec	ConvertedEntityVec	ConvertedBERTVec	事例数 (割合)
0	45.89	45.44	45.44	46.85	1,556 (22.23)
1	73.50	74.69	74.89	74.79	2,023 (28.90)
2	86.50	87.36	87.36	88.40	1,733 (24.76)
3	91.25	91.82	92.01	92.39	1,051 (15.01)
4 ≤	89.64	90.58	90.42	90.89	637 (9.10)

表 3　文書内に含まれている言及数ごとの結果 (infuse)

#言及	MentionVec	EntityVec	ConvertedEntityVec	ConvertedBERTVec	事例数 (割合)
0	44.79	45.76	45.57	45.63	1,556 (22.23)
1	73.90	74.15	75.14	75.38	2,023 (28.90)
2	86.56	87.13	86.79	87.77	1,733 (24.76)
3	91.82	92.01	91.91	92.39	1,051 (15.01)
4 ≤	90.89	91.37	91.52	91.52	637 (9.10)

表 1 から、concat と infuse のどちらにおいても、MentionVec に比べてそれ以外の手法では性能が向上しており、文書ジオロケーションモデルに地理的エンティティ情報を与えることが有効であることが確認できる。埋め込み表現の獲得手法間で比較すると、都道府県名への変換をおこなう ConvertedEntityVec および ConvertedBERTVec が EntityVec よりも分類正解率が高い。また、都道府県変換をおこなう両者の比較では、BERT を通して埋め込み表現を獲得する ConvertedBERTVec が良い結果となっていた。今回の設定では、BERT へ外部知識を取り込む際、BERT とは独立に獲得された埋め込み表現を利用するよりも外部知識を表層的に入力文書内に取り込んだ方が効果が高いことが示唆される。なお、取り込み位置に関しては、concat と infuse で明確な違いは見られなかった。

次に、文書内に含まれている言及数ごとの結果を表 2 および表 3 に示す。これらの表から、まず、言及数が 0 の場合は性能が著しく低くなることが確認できる。このことからも、言及情報が文書ジオロケーションにおいて有力な手がかりになっていることがわかる。文書内に言及が含まれている場合は言及数が増えるにつれて分類正解率が向上する傾向が伺える。ただし、言及数が 4 以上になると、分類正解率は下降している。言及数がある程度多い文書では、複数地点間を移動する内容であったり、あるいは　複数地点間の比較に関する内容であったりと、内容が複雑化することが多く、このような原因から分類正解率が下降していると考えられる。

ConvertedBERTVec と concat によってエンティティ情報を取り込んだモデルでの分類結果の例を以下に示す。事例（**c1**）は、エンティティ情報を利用することで正解に変化した事例である。この例では、「成田」からエンティティ「成田国際空港」の情報を経由してその所在地である「千葉県」の情報が与えられたことで正しく分類できるようになっていた。一方、事例（**w1**）は、EntityVec では正解できていたが、都道府県名への変換処理を挟むことで誤りに変化した例である。蔵王連峰は山形県と宮城県の県境に所在するが、ConvertertedBERTVec では宮城県として埋め込み表現を獲得したことで、誤りに繋がっていた。この例のように、都道府県名への変換が悪い影響を及ぼす事例の存在も確認できた。

- （**c1**）【正解：千葉県／出力：千葉県】
  飛行機乗れなくて成田成田国際空港（千葉県）で酔っ払うなう Sapporo ⇒Tokyo東京都（東京都）
- （**w1**）【正解：山形県／出力：宮城県】
  雪だ□□#蔵王蔵王連峰（宮城県）#寒いわけだ...

5　おわりに

文書ジオロケーションモデルへの地理的エンティティ情報の取り込みについて述べた。評価実験の結果から、地理的エンティティの有効性を示した。特に、エンティティの所在情報に注目して獲得した埋め込み表現が有効に機能することがわかった。今後の課題としては、Wikipedia 等のエンティティ情報を拡充することや、LUKE [15] の枠組みによるエンティティ情報の埋め込み表現学習の検討等がある。

謝辞

本研究は JSPS 科研費 21K12137 の助成を受けたものです。

参考文献

[1] Han Bo, Cook Paul, and Baldwin Timothy. Geolocation prediction in social media data by finding location indicative words. In Proceedings of COLING 2012, pp. 1045–1062, 2012.

[2] Jey Han Lau, Lianhua Chi, Khoi-Nguyen Tran, and Trevor Cohn. End-to-end network for twitter geolocation prediction and hashing. In Proceedings of the Eighth International Joint Conference on Natural Language Processing, pp. 744–753, 2017.

[3] Binxuan Huang and Kathleen Carley. A hierarchical location prediction neural network for twitter user geolocation. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 4732–4742, 2019.

[4] Taro Miyazaki, Afshin Rahimi, Trevor Cohn, and Timothy Baldwin. Twitter geolocation using knowledge-based methods. In Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text, pp. 7–16, 2018.

[5] 平川冬尉, 乾孝司. 地理的知識グラフを取り込んだニューラル文書ジオロケーションモデル. 情報処理学会論文誌, Vol. 63, No. 12, pp. 1870–1883, dec 2022.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the North American Chapter of the Association for Computational Linguistics, pp. 4171–4186, 2019.

[7] Rada Mihalcea and Andras Csoma. Wikify! linking documents to encyclopedic knowledge. In Proceedings of the sixteenth ACM conference on Conference on Information and Knowledge Management, pp. 233–242, 2007.

[8] 鈴木正敏, 松田耕史, 関根聡, 岡崎直観, 乾健太郎. Wikipedia 記事に対する拡張固有表現ラベルの多重付与. 言語処理学会第 22 回年次大会, pp. 797–800, 2016.

[9] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Proceedings of the 1st International Conference on Learning Representations, 2013.

[10] Seiji Okajima and Tomoya Iwakura. Japanese place name disambiguation based on automatically generated training data. In Proceedings of the 19th International Conference on Computational Linguistics and Intelligent Text Processing, 2018.

[11] 陰山宗一, 乾孝司. 言及に対する地理的特定性指標の提案と文書ジオロケーションへの適用. 情報処理学会自然言語処理研究会 (NL-253-19), 2022.

[12] 中本裕大, 瀬在恭介, 元川凱喜, 麻生英樹, 岡崎直観. 日本語大規模言語モデルにおける知識グラフを活用した意味理解性能の向上. 言語処理学会第 29 回年次大会, pp. 2140–2145, 2023.

[13] Keyur Faldu, Amit Sheth, Prashant Kikani, and Hemang Akbari. Ki-bert: Infusing knowledge context for better language and domain understanding, 2021.

[14] 平川冬尉, 乾孝司. 日本語地理的位置推定課題におけるインジケータ付 deepgeo 法の提案と評価. 人工知能学会全国大会論文集, Vol. JSAI2020, pp. 3Rin473–3Rin473, 2020.

[15] Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. Luke: Deep contextualized entity representations with entity-aware self-attention. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 6442–6454, 2020.

[16] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In Proceedings of International Conference on Learning Representations, 2017.

[17] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification? In Proceedings of Chinese Computational Linguistics, pp. 194–206, 2019.

[18] 笹沢裕一, 岡崎直観. 属性情報を追加した事前学習済みモデルのファインチューニング. 言語処理学会第 27 回年次大会, pp. 765–770, 2021.

A 付録

A.1 ベースとなる BERT 文書分類モデルの詳細

第 2 節で述べた本研究のベースとなる BERT 文書分類モデルの詳細設定について述べる。事前学習済み BERT モデルとして、東北大学が公開している bert-base-japanese-v3 （2023 年 5 月公開）[9] を使用した。文書ジオロケーションのためのファインチューニングには、4 節で述べた学習データを用いた。最適化手法には AdamW [16] を使用し、損失関数として Cross Entropy Loss を使用した。その他のハイパーパラメータの設定を表 4 に示す。BERT の Encoder Layer は、分類に用いる最終 4 層のみ学習対象としており、Sun ら [17] を参考に複数の学習率を使用している。Twitter 投稿文書はメタ情報を持っているため、入力文書として、投稿テキストを 1 文目、投稿者プロフィールの居住地情報を 2 文目として BERT への入力とした。

表 4 ベースモデルの設定

ネットワーク	パラメータ	値
全体	バッチサイズ	32
	エポック数	5
BERT	最大入力トークン長	512
	語彙数	32,768
	隠れ層サイズ	768
	ドロップアウト率	0.1
	Encoder Layer (9) 学習率	5e-6
	Encoder Layer (10) 学習率	1e-5
	Encoder Layer (11) 学習率	2e-5
	Encoder Layer (12) 学習率	5e-5
分類器	入力次元	768
	出力次元	47
	学習率	5e-5

A.2 エンティティ情報を取り込んだモデルの詳細

エンティティ情報を取り込んだ文書ジオロケーションモデルの詳細設定について述べる。基本的な設定は上述のベースモデルと共通であるが、エンティティ情報を取り込むため、中本ら [12] にならい全結合層を知識埋め込み層としてベースモデルに追加したネットワーク構成としている。エンティティ情報の埋め込み表現に加える Position Embeddings と Token Type Embeddings は事前学習済みの Word Embeddings と同じものを用いる。その他のハイパーパラメータのうち、ベースモデルと値が異なるものを表 5 に示す。ここで、エンティティ情報を取り込んだ文書ジオロケーションモデルではモデルを知識埋め込みに適応させるため、BERT の最終 4 層の Encoder Layer 以外の層も学習対象とした。BERT 学習の学習率は、笹沢ら [18] を参考にし、埋め込み表現獲得手法ごとに変更した。各手法の学習率を表 6 に示す。知識埋め込み層や分類器の学習率と比べて小さな値である 1e-5 から 1e-6 までの間で精度が高かった値を採用している。

表 5 エンティティ情報を取り込んだモデルの設定

ネットワーク	パラメータ	値
全体	バッチサイズ	16
	エポック数	4
知識埋め込み層	入力次元	200
	出力次元	768
	学習率	1e-3
	ドロップアウト率	0.1

表 6 埋め込み表現獲得手法ごとの BERT 学習率

埋め込み表現	情報の取り込み	値
MentionVec	concat	1e-6
	infuse	1e-6
EntityVec	concat	5e-6
	infuse	2e-6
ConvertedEntityVec	concat	1e-5
	infuse	1e-6
ConvertedBERTVec	concat	5e-6
	infuse	1e-5

A.3 エンティティ情報の自動同定での実験結果

評価実験において、評価データに対しては人手で正確なエンティティを同定し、学習データに対しては自動的に同定した。参考情報として、評価データと学習データの双方に対してエンティティを自動同定した場合の実験結果を表 7 に示す。

表 7 エンティティ自動同定データでの実験結果

	concat	infuse
EntityVec	74.83	74.91
ConvertedEntityVec	75.40	75.30
ConvertedBERTVec	75.94	75.74

9） https://huggingface.co/cl-tohoku/bert-base-japanese-v3