Wikidata に基づく大規模ジオコーディングデータセット

中谷 響 [1]　安井 雄一郎 [2]　若本 亮佑 [2]　石井 昌之 [2]
大内 啓樹 [1,3]　渡辺 太郎 [1]
[1] 奈良先端科学技術大学院大学　[2] 日本経済新聞社 [3] 理化学研究所
{nakatani.hibiki.ni4,hiroki.ouchi,taro}@is.naist.jp
{yuichiro.yasui,ryosuke.wakamoto,masayuki.ishii}@nex.nikkei.com

概要

本稿では，場所を表す言語表現（メンション）を地理データベースの適切なエントリと紐付け，地理座標（緯度・経度）を出力するモデルのための大規模データセットを構築した．具体的には，Wikipedia の各記事に出現するメンションと紐づけられている Wikidata のエントリを収集することによって自動構築した．実際に構築したデータセットで学習したモデルは，異なるドメインのデータに対しても高精度で解析可能であることを示す．

1　はじめに

ジオコーディングは，自然言語で記述された参照表現（メンション）から，現実世界の位置を表す地理座標（主に緯度・経度）に変換する基礎技術である．ジオコーディングの結果は，テキスト内で記述された人間活動や事物・事象の関わる出来事がどこで発生したのかを正確に把握するために有用であり，観光管理，災害管理，疾病監視など，さまざまな応用可能性が秘められている [1]．

ジオコーディング手法は，直接推定法と間接推定法に大別される．直接推定法は，メンションから直接地理座標を推測する．間接推定法は，図 1 のように，メンションを地理データベース (DB) の適切なエントリの ID に紐付け，その ID に付随する地理座標を出力する．ID を介することで入出力が疎結合となり，用途にあわせて異なる属性を利用したり，ID を他の知識ベースと紐付けることも可能となる [2]．このような利点にも関わらず，日本語の大規模なジオコーディングデータセットは直接推定法のためのもののみであり [3]，間接推定法のためのデータセットは小規模なもののみが存在する [4]．

本稿では，現時点で最大規模となる間接推定法のための日本語ジオコーディングデータセットを構築した．世界各国の国名や市区町村といった粗い粒度レベルのものから，学校や地下鉄といった細かな粒度レベルのものまでを広く含む特徴を持つ．このようなデータセットを，Wikipedia の記事の冒頭に出現するメンションおよびメンションに紐づく，地理情報を持つ Wikidata エントリから自動構築した．本データセットの構築過程の概要を図 2 に示す．本データセットで学習したジオコーディングモデルは，別ドメインである新聞記事についても高精度で解析可能であることを示した．本データセットを復元・構築するためのコードは一般に広く公開する[1)]．

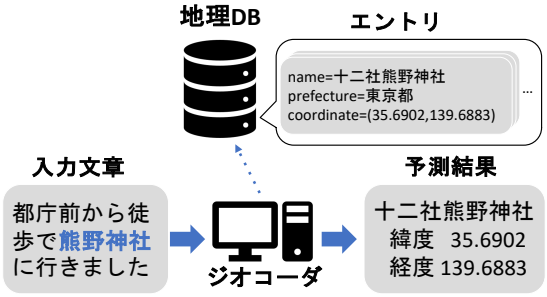

図 1　ジオコーディングの概要

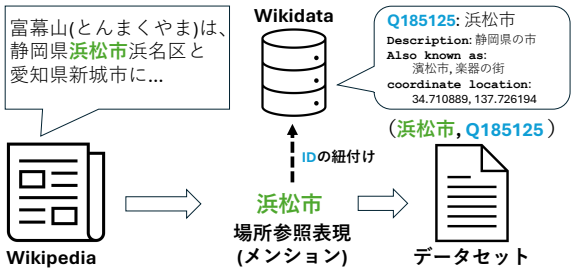

図 2　本データセットの構築過程の概要

1)　https://github.com/naist-nlp/wiki-geocoding-dataset

2 関連研究

間接推定法のためのジオコーディングデータセットとして，Gritta ら [5]，Kamalloo [6] ら，Lieberman ら [7] のものがある．Gritta らは，200 の分野から各 1 記事を選定し，そこに登場するメンションと地図データベース GeoNames[2] の正解エントリを紐付けるアノテーションを施し，GeoWebNews を構築した．Kamalloo らは，曖昧性の高い地名を含めるために多様な地域から計 118 記事を取得し，GeoNames のエントリと紐付けを行って TR-News を構築した．Lieberman は，地名辞書内の地名を曖昧性に基づいてランク付けを行い，その地名付近の新聞から記事を選択してアノテーションを行うことで，意図的に曖昧性の高い地名を含めた LGL データセットを構築した．これらは全て英語のみに対応しており，日本語ジオコーディングには使用できない．

間接推定法のための日本語ジオコーディングデータセットとして，東山らの ATD-MCL データセットがある [4]．本データセットは大内らの歩き方旅行記データセット [8] 内の計 200 記事に対して，OpenStreetMap(OSM)[3] のエントリを割り当てており，都道府県や市区町村などの行政区域だけではなく，施設やランドマークなどの細かい粒度のものにもアノテーションがついている．また，前述したデータセットとは異なり，旅行記という一般の人が記述した文書である都合上，同一の地物を指す言語表現でも，その表層形が大きく異なるという特徴を持つ．前述した全てのデータセットは人手でアノテーションを行っている関係で，データセットの精度としては保証されている反面，規模としては比較的小規模なもの[4] である．

直接推定法のための大規模なジオコーディングデータセットとして，大野らのデータセット[5] がある [3]．Wikipedia 内に登場するメンションに対して，地理座標を自動付与することによって構築された．結果として，日本語の 400 万程度のメンションに地理座標が紐づけられ，既存のジオコーディングデータセットを大きく上回る規模のデータセットとなった．このように直接推定法のデータセットは大規模なものが存在するが，間接推定法における大規模データセットは未だ構築されていない．

そこで，本稿では Wikipedia の冒頭の文書内に出現するメンションに紐づけられた地理情報を持つ Wikidata エントリに着目することで，間接推定法における日本語の大規模なジオコーディングデータセットを自動的に構築することを目指した．既存の間接推定法のデータセットと本データセットの統計情報を表 1 に示す．本稿では，3 章で，実際に構築したデータセットについて説明し，以降の章ではそのデータセットを用いて，既存のジオコーディングモデルを適用，評価を行った結果を報告する．

表 1 間接推定法のデータセットの統計

データセット	メンション数
GeoWebNews	2,401
TR-News	1,274
LGL	4,793
ATD-MCL	6,119
Ours	2,423,134

3 データセット

3.1 地理 DB 構築

Wikidata のダンプデータ[6] から以下の条件を満たす全てのエントリからなるエントリ DB を構築した．ダンプデータは巨大な JSON ファイル形式で，Wikidata 上の各エンティティ（＝エントリ）を要素とするリスト形式で構成される．

- 日本語ラベルを持つ（`labels` フィールドにキーが `ja` となる要素が存在）
- coordinate location (P625) を持つ（`claims` フィールドにキーが `P625` となる要素が存在）

本稿で用いた 2024 年 9 月 2 日のダンプデータには 112,413,055 エントリが収録され，うち日本語ラベルを持つエントリは 3,496,520 件，さらに coordinate location を持つエントリは 435,867 件であった．

3.2 Wikipedia 記事の収集

2024 年 9 月 1 日の日本語 Wikipedia の記事の冒頭を利用したものであり[7]，メンションと，その参

2） https://www.geonames.org/
3） https://www.openstreetmap.org/
4） ATD-MCL では，6,000 程度のメンションについて正解エントリがアノテーションされている．
5） https://www.lsta.media.kyoto-u.ac.jp/resource/data/WHLL/home.html
6） 最新のダンプデータは https://dumps.wikimedia.org/wikidatawiki/entities/latest-all.json.bz2 で公開．
7） Wikipedia ダンプデータからのデータ抽出には Wikipedia2vec[9] を利用した．

表 2　実験データの統計

分類	ページ数	メンション数
訓練	718,398	1,938,507
開発	104,556	242,313
評価	107,044	242,314
日経	28	234

照先に対応するエントリへのリンクが付与された組が複数存在する．本実験では，3.1 節で構築した WikidataDB が持つエントリへのリンク情報が付与されたメンションのみを対象とした．また，該当する記事を表 2 のように訓練，開発，評価データに分割して利用した．

3.3　日経新聞記事の収集

2021-2023 年の日本経済新聞（朝刊，夕刊，電子版），日経産業新聞から記事 28 件をランダムサンプリングし，評価データとして用いた．

4　実験条件

4.1　利用モデル

本実験では，ジオコーディングモデルとして，Nakatani ら [10] が提案したバッチ内の負例だけではなく，BM25 を元にした負例生成による hard-negative を組み込んだ (以下，in-batch-hard と表記)E5 ベースモデルを利用した．モデルケースとしては，提案されていたもののうち，最良のものを利用した．また，本実験では，エントリ表現として，各エントリが持つ日本語ラベルの正式名称，別名および説明を利用した．

4.2　比較手法

文字列類似度に基づいてエントリをランキングするベースラインシステムとして，BM25 を使用した．BM25 は，各クエリとなるメンション文字列に基づいて候補エントリをスコアリングする．この際，エントリは正式名称から構成され，E5 エンコーダのトークナイザによってトークン化される．また，in-batch-hard による有効性を調査するために，バッチ内の負例のみを用いた手法 (以下，in-batch と表記) および，バッチ内の負例だけではなく，一つの正例に対して，全エントリからランダムに一つ負例を生成する手法 (以下，in-batch-random と表記) の 2 つを比較対象として用いる．

4.3　評価手法

地理データベース内のすべてのエントリ（正確にはエントリグループ）を対象としたエントリランキング問題としてジオコーディングタスクを扱い，評価指標として平均逆順位 (MRR) と recall@k (R@k) を使用した．MRR スコアは q 個の例に対して以下の式で計算される：

$$\mathrm{MRR} = \frac{1}{q}\sum_{i=1}^{q}\frac{1}{\mathrm{rank}(m_i, e_i)},$$

ここで，m_i はメンション，e_i はその正解エントリ，$\mathrm{rank}(m_i, e_i)$ はモデルが m_i に対して予測したスコアに基づき e_i がすべてのエントリの中で占める順位を表す．一方で，recall@k は，各メンションに対して予測された k 個のエントリのうち少なくとも 1 つが正解エントリを含んでいれば正解とみなす．BM25 に対する評価指標としては，Recall@k および MRR の期待値を用いた．Recall@k の期待値は [11] に従い計算する．また，メンション m_i におけるメンション単位の $\mathrm{MRR}(\mathrm{MRR}_i)$ に基づいて MRR の期待値を次のように計算する．

$$\begin{aligned}\mathrm{MRR}_i &= \frac{1}{|E_i|}\sum_{j=1}^{|E_i|}\frac{1}{\mathrm{rank}(m_i, e_j)},\\ E_i &= \{e_j \mid s(m_i, e_j) = s(m_i, e_i)\}.\end{aligned}$$

5　実験結果

Wikidata の評価データおよび日経新聞データに対する各手法の精度として，Recall@k（$k \in \{1, 5, 10\}$）および平均逆順位 (MRR) を表 3 に示す（手法 (1)-(3) の精度は 3 回の実行結果の平均を指す）．

同一ドメインの評価セットの結果　手法 (0) に比べて，E5 の手法は Recall@1 において，Wikidata の評価データは最大 0.085 ポイントの向上が見られた．

他ドメインの評価セットの結果　手法 (0) に比べて，E5 の手法は Recall@1 において，Wikidata の評価データは最大 0.459 ポイントの向上が見られた．これより，本データセットを元に学習したモデルは，これは，データセットサイズが少なく，学習データを十分に確保できないデータセットに対して，学習データの代用となることが考えられる．

表 3 Wikipedia 記事の検証データおよび日経新聞記事に対してジオコーディングを行った結果を示す．各データセットの各指標における最良の結果を太字で表す．

手法	モデル	負例生成	Wikipedia				日経新聞			
			Recall@1	Recall@5	Recall@10	MRR	Recall@1	Recall@5	Recall@10	MRR
(0)	BM25	-	0.819	0.944	0.958	0.876	0.387	0.583	0.640	0.481
(1)	E5	in-batch	0.884	0.984	0.991	0.927	0.605	0.890	0.920	0.715
(2)		in-batch-random	0.888	0.985	0.992	0.931	0.664	0.903	0.926	0.777
(3)		in-batch-hard	**0.969**	**0.993**	**0.995**	**0.980**	**0.846**	**0.937**	**0.947**	**0.890**

表 4 Wikipedia の開発データにおける E5 ベースモデルの予測例を示す．メンションは [a]，[b]，[c] それぞれ台北，北国街道，ドイツとなる．トップ予測エントリは各エントリの正式名称部分のみを示す．正解エントリの各行はそれぞれ，日本語ラベルの正式名称，別名，説明を表す．

	正解エントリ	負例生成	トップ予測エントリ	正解エントリの順位
[a]	台北市 台北，臺北，北市，臺北市，北，たいほく 中華民国の直轄市および首都	in-batch in-batch-random in-batch-hard	W 台北 W 台北 台北市	5 2 1
[b]	国道 18 号 国道 18 号線 群馬県から新潟県に至る一般国道	in-batch in-batch-random in-batch-hard	長野県道 12 号 北園街道 北陸道	4,371 4,245 4,219
[c]	ドイツ ドイツ連邦共和国，独国 西ヨーロッパの国	in-batch in-batch-random in-batch-hard	ドイツ国 ドイツ国 ドイツ国	8 10 5

6 定性的分析

表 3 の E5 の手法である (1)，(2)，(3) の予測結果について誤り分析を行った．表 4 は，開発データ上の例に対する分析結果を示す．

例えば，メンション「台北」に関する例 [a] では，in-batch-hard で学習したモデルである手法 (3) は，正解エントリに類似したエントリを負例エントリとして学習している．それが要因で，各エントリの正式名称だけではなく，別名や説明にあたる部分にも着目することができた結果，正解エントリを正しく予測できたと考えられる．対して，バッチ内の負例のみを用いる手法 (1) やランダムに全エントリから選択したエントリを用いる負例（2) では，エントリの正式名称を見るだけで，十分な推測が可能であったため，別名や説明の場所を見る必要がなかったことから誤った予測をしてしまったと考えられる．実際，それらが予測したホテルの名前を指す「W 台北」には，別名が「W タイペイ」と単純な言い換えしか存在しておらず，また，説明には何も記載されていないことが確認できる．

対して，メンション「北国街道」に関する例 [b] では，全く正しく予測できていないことがわかる．このように，現状のモデルとデータセットでは，メンションの文書表現とエントリの正式メンションの表現に乖離があり，別名にメンションの表現が含まれていない場合の推測は困難である．

ちなみに，メンション「ドイツ」に関する例 [c] でも，全てのモデルが推測に失敗してしまっている．しかし，この例の周辺メンションを確認すると，1897–1945 年に実在した軍人が所属する「ドイツ」の話をしている．このように，アノテーションされた Wikipedia エントリが誤っている例も確認できる．これは，前処理の時点で，現代に存在していないエントリやイベントなどを除いていなかったことが原因と考える．

7 結論と考察

本稿では，ジオコーディングタスクに向けて，Wikiedia エントリの紐付けに着目し，ジオコーディングタスクに関連する属性を元にすることで，大規模なデータセットを自動的に構築した．本データセットで学習したジオコーディングモデルは，異なるドメインのデータに関しても高精度で解析可能であることがわかった．今後の方向性として，Wikipedia の記事の冒頭だけでなく本文もデータセットに含め，さらなる大規模化を行う予定である．また，さらに多様なドメインのデータに対する実験を通して，異なるドメインに対する頑健性を包括的に調査したい．

謝辞

本研究は JSPS 科研費 JP23K24904 の助成を受けたものです.

参考文献

[1] Xuke Hu, Zhiyong Zhou, Hao Li, Yingjie Hu, Fuqiang Gu, Jens Kersten, Hongchao Fan, and Friederike Klan. Location reference recognition from texts: A survey and comparison. Vol. arXiv:2207.01683, , 2022.

[2] 久本空海, 西尾悟, 井口奏大, 古川泰人, 大友寛之, 東山翔平, 大内啓樹. 場所参照表現と位置情報を紐付けるジオコーディングの概観と発展に向けての考察. 言語処理学会第 29 回年次大会発表論文集, 2023.

[3] Keyaki Ohno, Hirotaka Kameko, Keisuke Shirai, Taichi Nishimura, and Shinsuke Mori. Automatic construction of a large-scale corpus for geoparsing using Wikipedia hyperlinks. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, editors, **Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)**, pp. 1883–1888, Torino, Italia, May 2024. ELRA and ICCL.

[4] Shohei Higashiyama, Hiroki Ouchi, Hiroki Teranishi, Hiroyuki Otomo, Yusuke Ide, Aitaro Yamamoto, Hiroyuki Shindo, Yuki Matsuda, Shoko Wakamiya, Naoya Inoue, Ikuya Yamada, and Taro Watanabe. Arukikata travelogue dataset with geographic entity mention, coreference, and link annotation. In Yvette Graham and Matthew Purver, editors, **Findings of the Association for Computational Linguistics: EACL 2024**, pp. 513–532, St. Julian's, Malta, March 2024. Association for Computational Linguistics.

[5] Milan Gritta, Mohammad Taher Pilehvar, and Nigel Collier. A pragmatic guide to geoparsing evaluation. **Language Resources and Evaluation**, Vol. 54, No. 3, pp. 683–712, Sep 2020.

[6] Ehsan Kamalloo and Davood Rafiei. A coherent unsupervised model for toponym resolution. In **Proceedings of the 2018 World Wide Web Conference**, WWW '18, p. 1287–1296, Republic and Canton of Geneva, CHE, 2018. International World Wide Web Conferences Steering Committee.

[7] Michael D. Lieberman, Hanan Samet, and Jagan Sankaranarayanan. Geotagging with local lexicons to build indexes for textually-specified spatial data. In **2010 IEEE 26th International Conference on Data Engineering**, pp. 201–212. IEEE, 2010.

[8] Hiroki Ouchi, Hiroyuki Shindo, Shoko Wakamiya, Yuki Matsuda, Naoya Inoue, Shohei Higashiyama, Satoshi Nakamura, and Taro Watanabe. Arukikata travelogue dataset. Vol. arXiv:2305.11444, , 2023.

[9] Ikuya Yamada, Akari Asai, Jin Sakuma, Hiroyuki Shindo, Hideaki Takeda, Yoshiyasu Takefuji, and Yuji Matsumoto. Wikipedia2Vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from Wikipedia. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations**, pp. 23–30. Association for Computational Linguistics, 2020.

[10] Hibiki Nakatani, Hiroki Teranishi, Shohei Higashiyama, Yuya Sawada, Hiroki Ouchi, and Taro Watanabe. A Text Embedding Model with Contrastive Example Mining for Point-of-Interest Geocoding. In **Proceedings of the 31st International Conference on Computational Linguistics**, 2025. To appear.

[11] Shohei Higashiyama, Masao Ideuchi, and Masao Utiyama. Construction of the administrative agency web document corpus for Japanese entity linking [in Japanese]. **IPSJ SIG Technical Report**, Vol. 2024-NL-260, No. 10, pp. 1–15, June 2024.

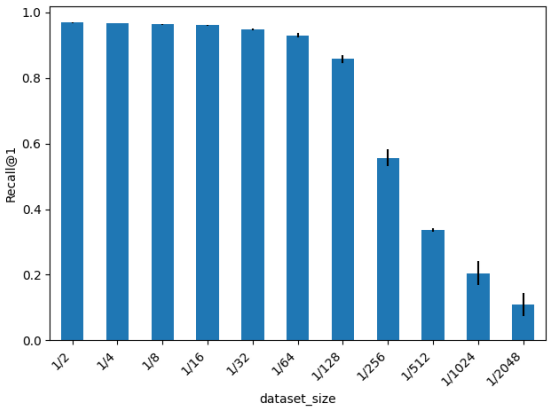

図 3　学習データを変化させた時の，評価データの精度比較

A　学習データの変化による精度の比較

本付録では，訓練データ数の変化による精度の変化を調べた．本実験は，学習データがリークしていないことなどの，不正に学習が行われていないかどうかを確認するために行う．表 2 のメンション数を $\frac{1}{n}$ where $n \in [1, 11]$ にしたものを学習データとして §4 と同様の実験条件で行った．Recall@1 のみに着目したものを図 3 に示す．この結果より，(1) データが少なくなればなるほど順当に低下していることから，データリークなどの不正な学習を行っているわけではなく，非常に大規模なデータであることが，§5 の結果を生み出したと考えられる．(2)$\frac{1}{16}$ まで，Wikipedia の評価データおよび日経データの精度がほとんど低下していないことから $\frac{1}{16}$ までデータを削減して実験を行うことが可能だと推測できる．

B　モデルハイパーパラメータ

表 5 に，Nakatani らのモデルをファインチューニングする際に用いたハイパーパラメータ値を示す．

表 5　モデルのファインチューニングで用いたハイパーパラメータ

Hyperparameter	Value
Training epochs	1
Batch size	16
Weight decay	0.01
Adam β_1	0.9
Adam β_2	0.98
Adam ϵ	1e-6
Learning rate	1e-5
Learning rate scheduler	linear
Warmup ratio	0.06
Optimizer	AdamW

C　複数の地理 DB 横断実験

C.1　異なる地理 DB による精度比較

§5 では Wikidata を地理 DB として用いてモデルを構築した．しかし地理 DB は多様化しており，異なる地理 DB にメンションを紐づけたい場合も少なくない．そこで本節では，Wikidata DB を他の地理 DB と入れ替えた際のモデルの性能の変化を調査する．

C.2　実験条件

地理 DB　日本国内の OpenStreetMap（OSM）エントリ（2,828,654 エントリ）を対象とした．各 OSM エントリに紐づけられた名前を表す *name* 情報以外に逆ジオコーディングを用いて，都道府県および市区町村情報を付与する．OSM エントリには，エントリを表現する属性も付与されているが，他の地理 DB が該当属性を持っているかどうかは不明である点から，本実験では扱わないものとする．

対象メンション　§4 で利用したデータのうち，1 つ以上の地理 DB が内包する OSM のエントリとリンクした Wikidata エントリと紐付けられたメンションのみを対象とする．

比較手法　本実験で用いるデータに対して，従来通り Wikidata のエントリ DB を地理 DB とした場合の精度も算出した．

C.3　評価手法

本実験では，正解となる OSM エントリが複数存在する場合がある．そこで，正解となる OSM エントリ集合のうち，一番順位が高いものを元にした Recall@k を算出した．

C.4　実験結果

表 6　DB の違いによる評価データに対する実験結果

地理 DB	Recall@1	Recall@5	Recall@10
Wikidata	0.994	0.998	0.999
OpenStreetMap	0.764	0.974	0.983

表 3 で最も性能の良かったで (3) の手法における Recall@1，Recall@5，Recall@10 の結果を表 6 に示す．表 6 より，地理 DB を変更にも応用性があること，エントリ数が大幅に増加しても，Recall@5 や Recall@10 では精度があまり落ちず，地理 DB の柔軟性があることがわかる．