メンション文脈とエントリ属性を考慮した Transformer Bi-Encoder によるジオコーディング

中谷響 [1] 寺西裕紀 [2,1] 東山翔平 [3,1] 大内啓樹 [1,2,4] 渡辺太郎 [1]
[1] 奈良先端科学技術大学院大学 [2] 理化学研究所 [3] 情報通信研究機構 [4] 国立国語研究所
{nakatani.hibiki.ni4,hiroki.ouchi,taro}@is.naist.jp
hiroki.teranishi@riken.jp shohei.higashiyama@nict.go.jp

概要

ジオコーディングは，地名や施設名など，実世界の特定の場所を指す言語表現（メンション）に対し，経緯度あるいは地理データベース上のエントリを推定する基盤技術である．本研究では，類似の名前を持つエントリの曖昧性の問題に対処するため，メンションが出現する入力テキストの文脈情報および候補エントリの属性情報を捉えるジオコーディング手法を提案する．実験により，これらの情報を考慮することの有効性を確認した．

1 はじめに

ジオコーディングは，地名や施設名など，実世界の特定の場所を指す言語表現（場所参照表現またはメンション）の経緯度を推定する基盤技術である．ジオコーディングの代表的なユースケースとして，POI（Point of Interests；関心のある場所）を表す表現や住所などのメンション文字列に適用して位置情報を取得する状況が挙げられ，地図検索などの応用で実用化されている．ジオコーディング技術を更に発展させ，文章を解析対象とし，文章中に出現するメンションに対して適用可能にすることで，新たな応用の実現も見込める．たとえば，SNS をはじめとするインターネット上の文章から，被災地に関連する災害情報を迅速に把握したり，感染症のクラスター発生箇所を特定したりすることも可能になると期待できる．なお，文章中のメンションの抽出（ジオタギング）と，メンションの位置情報の推定（ジオコーディング）を合わせたタスクはジオパージング [1, 2] と呼ばれているが，本研究ではジオコーディングに焦点を当てる．

ジオコーディングの主流な方法には，(i) 位置情報を直接的に推定する方法と，(ii) 地理データベース（DB）の検索を通して間接的に位置情報を推定する方法の 2 つがある．前者は，システム（ジオコーダ）に緯度・経度に対応する 2 つの数値を出力させる．後者は，図 1 のように，まず地理 DB に問い合わせて適切なエントリを取得し，経緯度も含むそのエントリに付随する属性情報全体を取得する [3]．この方法には，入出力が疎結合になることで，モデルが地理 DB の変更に対して頑健となる利点や，エントリに付随する様々な属性情報が利用可能となる利点があり，本研究ではこの方法を採用する．

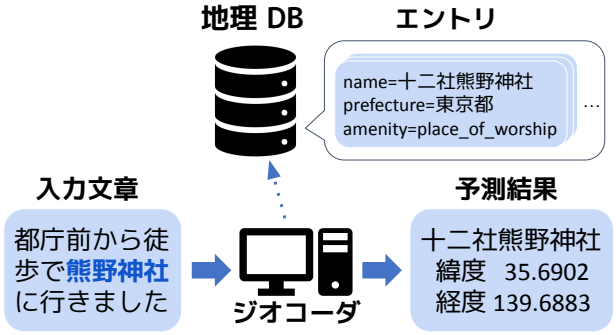

図 1 ジオコーディングの概要

ジオコーディングの主な課題として，類似または同一の名前を持つエントリの曖昧性を解消し，適切なエントリを特定する必要があることが挙げられる．たとえば，「熊野神社」は全国各地に存在するため，メンション文字列のみでは一意に特定できない．しかし，入力が「都庁前から徒歩で熊野神社に行きました」という文であった場合，人間が読むと都庁前駅付近にある「熊野神社」の可能性が高いと推測できる．こうした推論をシステムにより実現するには，曖昧性解消に有効な情報を含んだエントリの表現を用いることも必要である．具体的には，地理 DB 上で定義されエントリに割り当てられている施設種別などの属性情報や，エントリに紐づく経緯度を住所に変換した情報などが有効と考えられる．

本研究では，類似の名前を持つエントリに対処するために，(i) メンションが含まれる入力テキストの文脈情報と (ii) 地理 DB におけるエントリの属性情報を考慮するジオコーディング手法を提案する．旅行記テキストを用いたジオコーディングの実験の結果，文脈情報・属性情報ともに曖昧性解消の精度向上に有効であることを確認した．

2 提案手法

Wu ら [4] のエンティティリンキングモデルを参考に，入力テキストと DB エントリのベクトル化に異なるエンコーダを用いる Bi-Encoder 型のモデルを採用する．Wu らはエントリのランキングに Cross-Encoder も用いているが，本研究では Bi-Encoder の出力ベクトルのみを用いる簡易なランキング方法を採用する．後述するエンコーダには事前学習済日本語 BERT モデル[1)]を使用し，ジオコーディングタスクでの fine-tuning を行う．

メンションのベクトル化 メンションが出現する文のトークン列 $x = (x_0, \ldots, x_N)$（$x_0 =$ [CLS]）を入力とし，Transformer エンコーダによりベクトル表現の系列 $\mathbf{H}^{(\mathrm{t})} = (\mathbf{h}_0^{(\mathrm{t})}, \ldots, \mathbf{h}_N^{(\mathrm{t})})$ に変換する．そして，トークン系列 x 中のメンション $m = (x_i, \ldots, x_j)$（$1 \le i \le j \le N$）に対し，その内部トークンのベクトル表現 $(\mathbf{h}_i^{(\mathrm{t})}, \ldots, \mathbf{h}_j^{(\mathrm{t})})$ を平均することでメンションのベクトル表現 $\mathbf{h}_{\mathrm{m}}^{(\mathrm{t})}$ を得る．つまり，$\mathbf{h}_{\mathrm{m}}^{(\mathrm{t})} = \frac{1}{j-i+1} \sum_{k=i}^{j} \mathbf{h}_k^{(\mathrm{t})}$ である．メンションの内部トークンのベクトル表現として文脈化埋め込みを用いることで，外部の周辺文脈も考慮したメンションのベクトル表現が得られると期待できる．

エントリのベクトル化 DB の各エントリを，その属性名と属性値を連結したエントリ文字列で表す．エントリの属性には特に制限を設けないが，住所の一部を表す文字列や，地形・施設の種別を表すようなカテゴリ値を値に持つ属性を想定する．たとえば，図 1 に示すような「十二社熊野神社」のエントリがあった場合，“[CLS] name=十二社熊野神社 [SEP] prefecture=東京都 [SEP] amenity=place_of_worship [SEP]” のように，先頭が [CLS] トークンで始まり，各属性が [SEP] トークンで区切られたエントリ文字列で表されるものとする．メンション用のエンコーダとは異なるエントリ用の Transformer エンコーダにより，エントリ文字列のトークン列 $e = (e_0, \ldots, e_M)$（$e_0 =$ [CLS]）をベクトル表現の系列 $\mathbf{H}^{(\mathrm{d})} = (\mathbf{h}_0^{(\mathrm{d})}, \ldots, \mathbf{h}_M^{(\mathrm{d})})$ に変換する．そして，[CLS] トークンに該当するベクトル表現 $\mathbf{h}_0^{(\mathrm{d})}$ をエントリのベクトル表現 $\mathbf{h}_{\mathrm{e}}^{(\mathrm{d})}$ とする．

エントリのランキング 各メンション m と候補エントリ集合 E の各エントリ $e \in E$ について，それらのベクトル表現間の内積 $f(m, e) = \mathbf{h}_{\mathrm{m}}^{(\mathrm{t})} \cdot \mathbf{h}_{\mathrm{e}}^{(\mathrm{d})}$ により，メンションに対する候補エントリのランキングを行う．

損失関数 ミニバッチ B の各事例はメンション m_b とその正解エントリ（エントリ文字列）e_b のペアから構成され，$B = \{(m_b, e_b)\}_{b=1}^{|B|}$ と表す．各ミニバッチ B について，以下の損失関数 L によりエンコーダのパラメタ更新を行う．

$$L(B) = \frac{1}{|B|} \sum_{b=1}^{|B|} \ell(m_b, e_b)$$

$$\ell(m_b, e_b) = -f(m_b, e_b) + \log \sum_{k=1}^{|B|} \exp(f(m_b, e_k))$$

ここで，メンション m_b の正例エントリは e_b であるが，それに対する負例として，同一ミニバッチ内のエントリ e_k が用いられている（In-batch random negatives [5]）．

3 実験設定

提案手法の有効性を確認するため，ジオコーディングの実験を行った．

3.1 実験データ

実験データとして，日本語旅行記ジオパージングデータセット ATD-MCL[2)] [6] を使用した．本データセットは，「地球の歩き方旅行記データセット」[3)] [7, 8] の日本国内旅行記に対し，メンションと，その参照先に対応する地理データベース OpenStreetMap（OSM）[4)] 上のエントリへのリンク情報が人手で付与されたものである．本データセットのメンションには，固有名の表現と，一般名詞句や指示表現などその他の表現が含まれるが，本研究では固有名の表現[5)]のみ対象とした．本研究では，リンク情報が付与されている 100 記事を，表 2 のように訓練・開発・評価データに分割して用いた．その

1) https://huggingface.co/cl-tohoku/bert-base-japanese-whole-word-masking

2) https://github.com/naist-nlp/atd-mcl
3) https://www.nii.ac.jp/dsc/idr/arukikata/
4) https://www.openstreetmap.org/
5) 種別ラベルが LOC_NAME，FAC_NAME，LINE_NAME のいずれかであるメンション．

表 1 テストデータにおけるジオコーディング精度．手法 (0′) は “name=” を除外したエントリ文字列を用いる方法．

手法	Fine-tuning	文脈	属性	Recall@1	Recall@5	Recall@10	Recall@100	MRR
(0)				0.043	0.126	0.169	0.358	0.088
(0′)				0.174	0.403	0.464	0.562	0.281
(1)	✓			0.285 (±0.018)	0.617 (±0.035)	0.722 (±0.029)	0.900 (±0.011)	0.432 (±0.022)
(2)	✓		✓	0.325 (±0.011)	0.653 (±0.017)	0.770 (±0.014)	**0.909** (±0.004)	0.472 (±0.009)
(3)	✓	✓		0.281 (±0.010)	0.605 (±0.007)	0.727 (±0.010)	0.902 (±0.012)	0.429 (±0.005)
(4)	✓	✓	✓	**0.352** (±0.032)	**0.674** (±0.024)	**0.771** (±0.017)	0.896 (±0.009)	**0.495** (±0.025)

表 2 実験データの記述統計

	記事数	メンション数
訓練	70	1,555
開発	10	223
評価	20	461

他の実験設定の詳細は付録 B に示す．

DB として，OSM の原データ[6]に対し，実質的に同一の場所を指すエントリを一つのグループにまとめ上げる前処理を行ったもの（グループ数：1,828,617 件）を使用した．つまり，本実験では個々の DB エントリの代わりに，エントリのグループを単位として予測する設定を採用し，全件を候補エントリとした．前処理の詳細は付録 A に記載する．

3.2 比較手法

入力文の文脈情報と DB エントリの属性情報を考慮する提案手法に対し，文脈または属性を考慮しない手法も比較対象に加え，「文脈あり」または「文脈なし」と，「属性あり」または「属性なし」を組み合わせた 4 手法を比較する．

- 文脈なし：入力文全体の代わりにメンションのみを入力する方法．
- 属性なし：エントリ名（name 属性）のみからなる DB エントリ文字列（例：“[CLS] name=十二社熊野神社 [SEP]”）を用いる方法．

また，文脈なしのメンションと属性なしのエントリ文字列に，事前学習済日本語 BERT モデルをそれぞれ適用する fine-tuning なしのベースライン手法（エントリ文字列に “name=” を含める場合と含めない場合の 2 通りの設定を適用）も比較する．

3.3 モデル学習

最適化関数 AdamW を使用し，学習率を $2e-5$ とした．スケジューラとして，warmup_steps を全体の 10%とする warmupLinearSchedule を用いた．バッチサイズを 128 として，20 エポック学習を繰り返した．モデル選択の指標には後述する平均逆順位（MRR）を使用し，開発データで MRR が最大となった学習エポックのパラメタを用いた．

4 実験結果

テストデータにおける各手法の精度として，Recall@k（$k \in \{1, 5, 10, 100\}$）および平均逆順位（MRR）を表 1 に示す（手法 (1)–(4) の精度は 5 回の実行結果の平均 ± 標準偏差）．

ベースラインとの比較 ベースライン手法 (0), (0′) と比べ，fine-tuning を行った (1)–(4) の手法では，いずれの評価尺度でも精度が向上した．なお，ベースライン手法においてエントリ文字列に “name=” を含める設定 (0) と含めない設定 (0′) では，設定 (0) での精度が著しく低く，fine-tuning を行わない状況では，メンションとの照合にこの文字列が悪影響を与えていることがわかる．

エントリ属性の有効性 エントリ属性情報については，手法 (1) と (2) の比較および手法 (3) と (4) の比較から，Recall@1 において最大 0.07 ポイントの向上，他のほとんどの尺度でも向上を達成しており，属性情報の利用が有効であることがわかった．

メンション文脈の有効性 メンション文脈情報については，属性情報を使用しない手法 (1) と (3) の比較からは必ずしも有効性が確認できない．一方，属性情報を使用する手法 (2) と (4) の比較では，手法 (4) で Recall@1 において 0.03 ポイント程度の向上が見られ，属性情報を考慮する条件下では文脈情報の利用が有効であった．この点は，エントリが名前以外の属性情報（例：“prefecture=東京都”）を持たない状況では，メンションの文脈情報（例：「都庁前から徒歩で」）をエントリとの照合に役立てられないという理由が考えられる．

総括 Recall@1 では，最良の結果でも 0.35 程度であり，改善の余地が大きい．一方，fine-tuning を

6） http://download.geofabrik.de/asia/ にて公開されていた japan-230601.osm.bz2 を使用した．

行った各手法において，Recall@10 では 0.72–0.77，Recall@100 では 0.9 程度であり，正解エントリの大部分を拾えている．したがって，候補エントリの適切な絞り込みには成功していると言える．将来的には，我々の提案手法に加え，候補エントリの詳細なリランキングを行うステップを導入することで，より高精度なジオコーディングが実現できる可能性がある．

5 誤り分析

開発データにおける提案手法の予測誤りを分析したところ，メンション文字列を重視した推論を行い，類似した名前を持つエントリからの適切な曖昧性解消が行えていない事例が見られた．例として，「三仏堂の裏手にある護摩堂では新年の護摩法要が行われていました。」という文内のメンション「護摩堂」を挙げる．同メンションには栃木県の寺院の堂を表す「大護摩堂」のエントリが正解として割り当てられていたのに対し，モデルは，富山県のランドマークを表す「護摩堂」のエントリを予測していた．この種の問題への対応策として，複数文の広い周辺文脈を入力としつつ，他のメンションが指す場所と候補エントリの間の地理的距離を考慮する方法が考えられる．前述の例では，周辺文に「日光東照宮」や「陰陽門」など栃木県の場所を表す表現が出現していることから，「護摩堂」も，これらの場所に近い栃木県の場所であるという予測が可能になると期待できる．

6 関連研究

英語テキストを対象としたジオコーディングの従来研究を以下に挙げる．Gritta ら [9] は，入力文中の注目するメンションとその他の単語を別の畳み込み層でベクトル化し，さらにメンションに関する人口分布を表すベクトル（MapVec）を素性に加え，全結合層によってメンションが位置する地球表面上のタイルを予測する手法 CamCoder を提案した．Wang ら [10]，Zhang ら [11] は，メンションとの文字列類似性に基づくルールベースの候補エントリ抽出と，メンション・候補エントリ対のスコア付けによるランキングを行う 2 段階の手法を提案した．Wang らは，ランキングで LightGBM を使用し，素性として，人口などの候補エントリの数値的属性や，メンションの周辺文脈と候補エントリに対応する Wikipedia 記事先頭パラグラフとの各 Bag-of-Words 表現間の類似度などを使用している．Zhang らは，ランキングで fine-tuned BERT を使用し，メンションと候補エントリの名前と別名を [SEP] トークンで連結した文字列を BERT でベクトル化し，さらに候補エントリの DB 上での種別[7] を表す one-hot ベクトルと人口の対数値を素性に加えてスコアを計算している．

日本語テキストに対するジオコーディング研究の例として，本研究における事前学習済日本語 BERT モデル (0′) と同様の方法を用いた大内ら [8] の研究の他に，Awamura [12]，関ら [13] の研究がある．関らは，メンションの周辺文脈に出現した地名を候補とし，関連文書中での候補地名の出現頻度をスコアとして，メンションの位置情報を大字以上の地名の粒度で同定するルールベースの手法を提案した．Awamura らは，SNS 投稿に含まれる地名の曖昧性解消のため，投稿に含まれる他のメンションと候補エントリとの地理的距離と，一定時間内の過去の投稿中のテキストを素性として，SVM で候補エントリを推定する手法を提案した．

以上のように，従来研究でもメンションの周辺文脈を利用したものは多い．本研究では，メンションの文脈化埋め込みを用いることで周辺文脈を考慮しつつ，任意の数のカテゴリ的属性をベクトル化したエントリ属性情報を使用し，両者の有効性を実験的に確認した．従来研究で用いられた人口のような数値的なエントリ属性や，メンション・エントリ間の地理的距離などの情報も，更に追加することが有効であると考えられる．

7 まとめと今後の展開

本稿では，曖昧性を持つメンションの解消を目的として，メンションの周辺文脈とエントリの属性情報を利用したジオコーディング手法を提案した．実験により，周辺文脈およびエントリ属性を用いる有効性を示した．

今後の展望として，(i) リランキング機構の追加，(ii) より広範なメンション文脈の考慮，(iii) メンション・エントリ間の地理的関係性の考慮により，より上位の出力エントリに正解が含まれる精度を向上させる点，(iv) 指示表現・一般名詞句などの非固有名で言及されたメンションにもジオコーディング対象を広げる点を挙げる．

7） 具体的には GeoNames における “type” (https://download.geonames.org/export/dump/featureCodes_en.txt)であり，属性情報に相当する．

謝辞

本研究は JSPS 科研費 JP22H03648 の助成を受けたものです．ATD-MCL を用いた実験において，「地球の歩き方旅行記データセット」を利用しました．

参考文献

[1] Jochen L Leidner. An evaluation dataset for the toponym resolution task. **Computers, Environment and Urban Systems**, Vol. 30, No. 4, pp. 400–417, 2006.

[2] Milan Gritta, Mohammad Taher Pilehvar, and Nigel Collier. A pragmatic guide to geoparsing evaluation: Toponyms, named entity recognition and pragmatics. **Language resources and evaluation**, Vol. 54, pp. 683–712, 2020.

[3] 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.

[4] Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. Scalable zero-shot entity linking with dense entity retrieval. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, 2020.

[5] Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. Learning dense representations for entity retrieval. In Mohit Bansal and Aline Villavicencio, editors, **Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)**, pp. 528–537, Hong Kong, China, November 2019.

[6] 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. arXiv:2305.13844, 2023.

[7] Arukikata. Co., Ltd. Arukikata travelogue dataset, 2022. Informatics Research Data Repository, National Institute of Informatics. https://doi.org/10.32130/idr.18.1.

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

[9] Milan Gritta, Mohammad Taher Pilehvar, and Nigel Collier. Which Melbourne? Augmenting geocoding with maps. In **Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 1285–1296, 2018.

[10] Xiaobin Wang, Chunping Ma, Huafei Zheng, Chu Liu, Pengjun Xie, Linlin Li, and Luo Si. DM_NLP at SemEval-2018 task 12: A pipeline system for toponym resolution. In **Proceedings of the 13th International Workshop on Semantic Evaluation**, pp. 917–923, 2019.

[11] Zeyu Zhang and Steven Bethard. Improving toponym resolution with better candidate generation, Transformer-based reranking, and two-stage resolution. In **Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)**, pp. 48–60, 2023.

[12] Takashi Awamura, Daisuke Kawahara, Eiji Aramaki, Tomohide Shibata, and Sadao Kurohashi. Location name disambiguation exploiting spatial proximity and temporal consistency. In **Proceedings of the third International Workshop on Natural Language Processing for Social Media**, pp. 1–9, Denver, Colorado, June 2015.

[13] 関龍, 乾孝司. 局所文脈と関連文書を用いた地名に対する地理的位置の同定. 2018 年度人工知能学会全国大会, 2018.

[14] 東山翔平, 大内啓樹, 寺西裕紀, 大友寛之, 井手佑翼, 山本和太郎, 進藤裕之, 渡辺太郎. 日本語旅行記ジオパージングデータセット ATD-MCL. 言語処理学会第 30 回年次大会発表論文集, 2024.

A　OpenStreetMap 前処理の詳細

実験に使用した OpenStreetMap（OSM）の前処理方法および前処理済みデータは，文献 [14] で用いたものと同一である．前処理の内容は次の通りである．まず，日本国内の場所を表す 2.8B エントリから，name タグに値を持つ（つまりエントリ名を持つ）2.8M エントリを抽出した．次に，name のタグ名・値に加えて，後述する属性の名前および値を連結することでグループ ID を表す文字列を各エントリに割り当てた．そして，グループ ID が同一のエントリを同一グループとすることで，1.8M 件のグループに集約した．たとえば，「スターバックスコーヒー 奈良猿沢池店」に相当する OSM エントリ[8]のグループ ID は，次の文字列である．

> name=スターバックス|branch=None|prefecture=奈良県|city=奈良市|quarter=樽井町|road=猿沢遊歩道|amenity=cafe

グループ ID 文字列の構成要素に含める属性として，OSM エントリの経緯度や郵便番号情報を基に取得・作成した表 3 上段の住所に関する属性[9][10]と，主に地形や施設の種類を表すのに適度な内容のカテゴリ値を持つことから採用した表 3 下段の OSM タグ[11]を用いた．

表 3　エントリのグループ ID 文字列に含めた属性（上段：住所関連属性，下段：OSM タグに相当する属性）

prefecture	city	suburb	quarter
neighbour	road		
aerialway	aeroway	amenity	artwork_type
attraction	branch	building	craft
castle_type	highway	historic	leisure
man_made	museum	natural	office
place	public_transport	railway	route
route_master	ruins	shop	tourism
tower:type	water	waterway	

8）https://www.openstreetmap.org/node/8341902317

9）OSM エントリが郵便番号に関するタグ（addr:postcode または postal_code）を持つ場合は郵便番号データ（https://www.post.japanpost.jp/zipcode/dl/kogaki-zip.html）を使用し，他の場合は経緯度を基にした逆ジオコーディング API（https://nominatim.org/release-docs/latest/api/Reverse/）を利用し，住所情報を取得した．

10）エントリが特定の OSM タグ（admin_level，place，highway，railway，route，warterway）を持つ場合，その値に基づいてグループ ID 文字列に含める住所関連属性を調整するルールを適用した．たとえば，admin_level=7（市町村・特別区に相当）のエントリでは prefecture のみ含め，railway のタグ（値は任意）を持つエントリではいずれの住所関連属性も含めないこととした．

11）OSM のタグの一覧は https://wiki.openstreetmap.org/wiki/JA:Map_Features から参照可能．

B　実験設定の詳細

ATD-MCL では，同一の場所を指すメンションの共参照クラスタ単位で DB エントリ（主に OSM で，一部は他のサイト）へのリンク情報が付与されている．本実験では，リンク情報として，OSM エントリの URL が付与され，かつ best_ref_type=PARTOF でないもの（best_ref_type=OSM であるもの）に限定し，共参照クラスタ中の各メンションにリンク情報を割り当て，メンション単位でのジオコーディングを実施した．なお，PARTOF とは，メンション（の共参照クラスタ）が指す場所に直接対応する DB エントリがなく，当該場所を地理的に包含するような場所にあたる DB エントリが割り当てられていることを表す情報である．