# テキスト中の場所表現認識と係り受けに基づく緯度経度推定ツールの開発

大野けやき [1]　西村太一 [1]　亀甲博貴 [2]　森信介 [2]
[1] 京都大学大学院 情報学研究科　[2] 京都大学 学術情報メディアセンター
{ohno.keyaki.57r,nishimura.taichi.43x}@st.kyoto-u.ac.jp
{kameko,forest}@i.kyoto-u.ac.jp

## 概要

文献中の地名の緯度経度を自動推定することは，文献調査を必要とする研究等の場面において有益である．緯度経度を推定する課題は，地名を認識するステップと認識した地名の緯度経度を推定するステップの 2 段階で実現される．この後段のステップについて，単語間の係り受け構造を利用して推定精度向上を図る手法を提案する．本手法を組み込んだ緯度経度推定ツールを鉱山に関する記事に適用し，係り受け構造を利用することで推定精度が向上することを確認した．

## 1　はじめに

行政区画や施設を表す言葉は，ある場所を表す表現＝場所表現として用いられる．文献調査を必要とする研究等において，ある場所について記された文献を探したい場面はしばしばある．ところが，場所表現は時間と共に変わることがあり，また，その付近の人々の間だけで使われることも多いため，想定する文献に合わせて適切な場所表現を使い分けたり複数使ったりしなければならない (表現の一意性がない問題)．また，ある 2 つの場所表現が示す場所が実世界上でどれほど近いかまたは同じなのかが，一見して分からないため，対象付近の情報を探しそびれる可能性がある (関連性が不明瞭な問題)．このような問題を避けるため，文字の場所表現ではなく緯度経度で場所を取り扱うことは有用であり，テキスト中の場所表現が示す緯度経度を推定する課題は従来より研究されてきた [1, 2, 3, 4, 5]．

この推定課題は，場所表現認識と緯度経度推定の 2 ステップからなる．前段の場所表現認識は系列ラベリングの一つである．CRF [6] 等の利用により，効率的に機械学習ができるようになった一方，認識精度の向上のためには膨大な学習データが必要である．後段の緯度経度推定は，文脈を考慮して場所表現が示す緯度経度を推定する作業であり，曖昧性解消タスクの一つである．具体的には，地名や緯度経度，人口等が収録されている地理辞典から適切なデータを選ぶ方法が一般的である．この方法では曖昧性解消アルゴリズムと地理辞典を分けて開発できるため，地理辞典として有志のデータベースを利用したり，利用者専用のデータベースを利用したりでき，応用の幅が広い．しかしながら，地理辞典に同じ表記の異なる場所が収録されている場合やある場所表現が全く収録されていない場合に精度が落ちてしまう問題がある．

本研究では，後段の緯度経度推定タスクに着目し，精度向上を目的として単語間の係り受け構造を利用する手法を提案する．係り受けは言語の構造であり，テキストの内容に依らず存在する．したがって，係り受け構造を利用する手法は，テキストの分野を限定せずに幅広いテキストに利用できる．この普遍性は，ある場所に関するあらゆる文献を集めたい地域研究などにおいて必須の条件である．

本手法を組み込んだ緯度経度推定システムを鉱山に関する英語記事に適用した結果，係り受け構造を利用することで推定精度が向上することが分かった．また，精度向上の要因となった場所表現を調べることにより，テキスト中の場所表現と地理辞典上の表現との間にはそれなりの差があることがあることが示唆された．

## 2　緯度経度推定ツール

### 2.1　課題設定

本研究が対象とする緯度経度推定ツールは，式 1 に示すような 2 段階のタスク構成になっており，入

力テキストに緯度経度が挿入されたテキストを出力する．

$$t' = F_{NER}(t)$$
$$t'' = F_{NEL}(t') \quad (1)$$

ただし，$F_{NER}, F_{NEL}$ はそれぞれ場所表現推定器，緯度経度推定器である．また，$t, t', t''$ はそれぞれ入力テキスト，場所表現かどうかのタグ付きテキスト，タグおよび緯度経度付きのテキストである．テキストは一般にまとまった複数文である．下記に具体例を示す．タグの L は場所表現を，O はそれ以外の表現を表している．

$t$: 岩手の清水寺へ行く．
$t'$: 岩手/L-B の/O 清水/L-B 寺/L-I へ/O 行く/O ．/O
$t''$: 岩手/L-B(39.60000, 141.35000) の/O
清 水/L-B(39.37035, 141.03056) 寺/L-I(39.37035, 141.03056) へ/O 行く/O ．/O

### 2.2 場所表現認識

テキスト中の場所表現認識には，場所表現をアノテーションしたテキストを用いて学習した場所表現推定器を用いた．この推定器は，BERT [7] と CRF [6] と点予測 [8] を組み合わせたものである．推定器の詳細は付録に示した．Stanford NLP Group の Stanza[1] [9] を用いて入力テキストをトーカナイズし，この推定器で各トークンに IOB2 形式のタグを付与した．すなわち各トークンには下記のいずれかのタグが付与される．

L-B :場所表現の最初のトークン
L-I :場所表現の 2 つ目以降のトークン
O :場所表現ではないトークン

### 2.3 緯度経度推定

ベースラインとなる緯度経度推定手法は，原ら [4] の手法を参考にした．

1. 特別に用意した地名-緯度経度辞典がある場合は，場所表現の文字列と一致するデータがあるか検索し，あればその緯度経度を出力する．
2. 場所表現の文字列を地理辞典 GeoNames[2] で検索し，見出し語 (name) または別名 (alternate names) のいずれかに一致するデータがあった場合，それらの中から下記のスコアが最も高いデータの緯度経度を出力する．
   - 文字列が見出し語の場合: 20 点 + 別名の数 × 1 点
   - 文字列が見出し語ではないが別名に含まれる場合: 別名の数× 1 点

   このスコアリングは，重要な地名ほどたくさんの別名が収録されているという考えのもと設定した．
3. 場所表現の文字列と一致するデータが無かった場合は，具体的な緯度経度は出力せず，代わりに推定失敗の旨を表す記号を出力する．

## 3 係り受け構造の利用

本研究では，上記の緯度経度推定ツールに，単語間の係り受け構造を利用する手法を 2 つ追加する．これらの手法は，係り受け木構造上で上下関係のある 2 つの場所表現が実世界上で包含または近接した位置関係にあり，木構造のより頂点に近い場所表現の方が詳細な情報を有している (実世界上でより狭い領域を表している) という仮定に基づいている．3.2 節では地理辞典上の複数候補から適切なデータを選択する工夫を，3.3 節では地理辞典上にデータが収録されていない場合の改善策を述べる．

### 3.1 Universal Dependencies

テキスト中の単語間の係り受けを表すものとして，本研究では Universal Dependencies[3] (以下，UD) を用いる．UD は，特定の言語に依存することなく普遍的な係り受け構造を表そうとする取り組みであり，100 以上の言語に対応している．緯度経度推定ツールは，利用者が詳しくない言語圏のテキストに対しても用いられうることを考え，幅広い言語に対応している UD に着目した．UD 解析ツールとして，Stanza [9] を用いた．

### 3.2 文構造の利用

文の係り受け構造から，ある単語と意味的に関連の強い単語を推測することができる．例えば，「岩手の清水寺から京都の清水寺へ行く．」という文を考えると，1 つ目の清水寺は岩手と関係していて，2 つ目と清水寺は京都と関係している．この関係は UD にも表れ，例ではそれぞれの清水寺から岩手や京都に係っている (図 1)．場所表現間の係り受けの場合，たいていは狭い領域を表す場所表現からそれを補足説明する知名度の高い場所表現に向かって係

1) https://stanfordnlp.github.io/stanza/
2) https://www.geonames.org/
3) https://universaldependencies.org/

り受け構造が存在する．

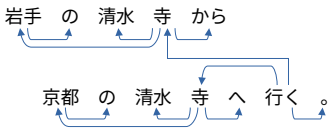


**図 1**　文の係り受け構造

ツールへの実装として，以下の手順をベースライン手法に追加した．まず，ベースラインの手順に先立ち，入力文に対する係り受け木構造を推定し，それぞれの場所表現に対して自身より下層に別の場所表現が存在するか確認しておく．存在すればその別の場所表現を，自身が参照すべき参照表現として設定する．その後，ベースラインの手順に沿って緯度経度推定を行うが，以下の点が異なる．

1. 参照表現が設定さている場合には，その参照表現の緯度経度推定を先に行う
2. その参照表現に対して具体的な緯度経度が推定できた場合には，ベースライン手法のスコアリングに関わらす，参照表現の推定緯度経度に最も近い緯度経度の候補を出力する

例えば図 1 の例では，清水寺の緯度経度推定に先立って岩手や京都の緯度経度推定を行い，それらに近い清水寺の緯度経度を出力する．

### 3.3　表現構造の利用

複数単語からなる一つの場所表現についても，その構造を考えることができる．例えば「京都三条」という場所表現は，京都と三条という 2 つの地名からなり，かつ，三条の場所を示す表現である．このような複合場所表現は，そのままの文字列で地理辞典に収録されていな場合が多いため，この例であれば「京都三条」を「三条」と読み換えて地理辞典を検索する必要がある．この様な構造解析を行う際，対象言語を限定しないためには，文字列の表面的な構造ではなく意味的な構造に着目するのが望ましい．複合場所表現においても，表現内に登場する地名間にそれらが示す領域の包含・近接関係が存在し，前節と同様の係り受け構造が存在する (図 2)．よって，ある場所表現内の係り受け構造の頂点となる単語は，その場所表現の核になる (最も狭い領域を表す地名) 単語だと推測できる．


**図 2**　表現内の係り受け構造

ツールへの実装として，以下の手順をベースライン手法に追加した．複数トークンからなる場所表現が地理辞典に収録されていなかった場合，そのトークン数より 1 少ない部分トークン列で再度ベースラインの手法により緯度経度推定を行う．この際，必然的に複数の部分トークン列で辞書引きを行うことになるが，複数で地理辞典のデータにヒットした場合は，その中から下記で求めるスコアの大きい部分トークン列に対する辞書引き結果を優先して出力を選択する．

1. 元の場所表現に対する係り受け木構造を考え，各要素 (トークン) に対して子孫要素が何層あるかスコアを付ける
2. その部分トークン列に含まれるトークンの上記スコアを合算する (図 3)

いずれの部分トークン列でも地理辞典のデータがヒットしなければ，さらにこの手順を繰り返す．

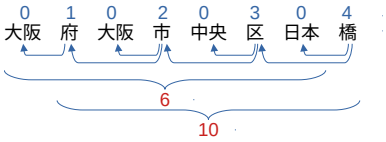


**図 3**　部分トークン列のスコアリング

## 4　実験

提案手法を組み込んだ緯度経度推定ツールを鉱山に関する英語記事に適用し評価を行った．評価対象は 2 つで，場所表現アノテーション済みテキストに対する緯度経度推定と，何もアノテーションされていないテキストに対する場所表現認識 + 緯度経度推定の全自動タスクである．

### 4.1　コーパス

評価には鉱山に関する専門記事を用いた．2.2 節で述べた場所表現推定器の学習には，同様の専門記事に加え，データ不足を補うために一般ニュース記事も用いた (表 1)．AFP 通信と MINING.COM の記事は人手でアノテーションを行い，ロイター通信の記事は CoNLL2003 [10] から LOC タグが付与されているトークンを含む文を抽出して用いた．

### 4.2　評価指標

緯度経度推定については，推定した緯度経度と正解緯度経度の誤差が $X$ km 以内に収まっている場所表現を合格とし，全場所表現中の何件が合格したか

| 学習データ (場所表現認識器) | | |
|---|---|---|
| 一般記事 | AFP 通信 | 982 / 263 |
| 一般記事 | ロイター通信 | 7140 / 1106 |
| 専門記事 | MINING.COM | 181 / 91 |
| 評価データ | | |
| 専門記事 | MINING.COM | 80 / 52 |

**表 1**　コーパスの場所表現数 (のべ / 重複なし)

を表す “$X$ km 正確度”を用いた．全自動タスクについては，上記と同様に合格をもって推定値が正解値と一致したとみなし，精度，再現率，$F_1$ 値を用いて評価した．これらの計算式は付録に示した．

### 4.3　結果

場所表現アノテーション済みテキストに対する緯度経度推定精度を図 4 および表 2 に示す．図の横軸は，左側は線形で右側は対数軸になっている．係り受け構造を利用する手法 (提案手法) はベースラインと比べて精度が向上し，例えば 5 km 正確度が 0.11 向上した．

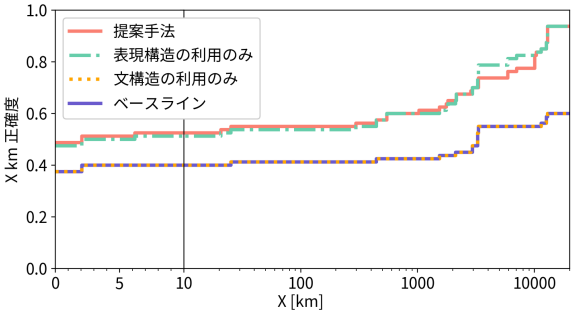


**図 4**　緯度経度推定タスクの精度

| 手法 | 5 km 正確度 |
|---|---|
| ベースライン | 0.40 |
| +文構造の利用のみ | 0.40 |
| +表現構造の利用のみ | 0.50 |
| 提案手法 (両方) | 0.51 |

**表 2**　緯度経度推定タスクの精度

場所表現認識 + 緯度経度推定の全自動タスクの精度 (5 km 合格) を表 3 に示した．提案手法は，わずかながらいずれの指標でもベースラインを上回った．

## 5　考察

文構造の利用と表現構造の利用のうち，推定精度の向上はほぼ後者の寄与であった．これは 2 つの手法がもたらす効果の違いに関係していると考えられる．まず，文構造の利用が効果を発揮するのは，同じ文字列で異なる場所を示しているいわゆる曖昧地名に対してである．例えば実験では，アリゾナ州のエイボンデールとペンシルベニア州のエイボンデールの区別に効果があった．しかしながら，そもそも地理辞典に 0 または 1 件しかデータが収録されていないような場所表現に対しては効果がない．一方，表現構造の利用が効果を発揮するのは，場所表現のそのままの文字列が地理辞典に収録されていない場面である．例えば実験では「New Castle County, Del」といった州，群，市などが併記された複合場所表現に対して効果を発揮していた．文構造の利用が効果を発揮する曖昧地名と表現構造の利用が効果を発揮する複合場所表現の出現頻度を考えると，後者の方が多いため，今回の実験では表現構造の利用による効果が比較的大きかったと考えられる．

また同時に，後者の頻度が多いということは，地理辞典に収録されていない場所表現が多いということを意味している．このことは，地理辞典を利用する緯度経度推定に限界があることを示唆しており，地理辞典への依存度が小さい手法が実用上望ましいと考えられる．

| 手法 | 精度 | 再現率 | $F_1$ 値 |
|---|---|---|---|
| ベースライン | 0.31 | 0.39 | 0.34 |
| 提案手法 | 0.34 | 0.43 | 0.38 |

**表 3**　全自動タスクの精度 (誤差 5 km 以内を合格)

## 6　終わりに

本研究では，テキスト中の場所表現に対する緯度経度推定タスクにおいて，単語間の係り受け構造を利用することでより尤もらしい緯度経度を推定する手法を提案し，その手法を組み込んだ緯度経度推定ツールを開発した．実験により，提案手法は緯度経度推定精度を向上させることを確認した．今後は，地理辞典に収録されていない場所も出力できるように，提案手法に加えて地理辞典を用いない方法も融合したような緯度経度推定手法を検討する．

## 7　謝辞

本研究は JSPS 科研費 JP21H04376 の助成を受けたものです．

## 参考文献

[1] Michael Speriosu and Jason Baldridge. Text-driven toponym resolution using indirect supervision. In **Proceedings of the 51st Annual Meeting of the Association**

**for Computational Linguistics**, 2013.

[2] Takashi Awamura, Daisuke Kawahara, Eiji Aramaki, Tomohide Shibata, and Sadao Kurohashi. Location name disambiguation exploiting spatial proximity and temporal consistency. In **Proceedings of the 3rd International Workshop on Natural Language Processing for Social Media**, 2015.

[3] Davy Weissenbacher, Arjun Magge, Karen O'Connor, Matthew Scotch, and Graciela Gonzalez-Hernandez. SemEval-2019 task 12: Toponym resolution in scientific papers. In **Proceedings of the 13th International Workshop on Semantic Evaluation**, 2019.

[4] Shoichiro Hara, Akira Kubo, Masato Matsuzaki, Hirotaka Kameko, and Shinsuke Mori. Development of methods to extract place names and estimate their places from web newspaper articles. In **Pacific Neighborhood Consortium Annual Conference and Joint Meetings**, 2021.

[5] Maithrreye Srinivasan and Davood Rafiei. Location-aware named entity disambiguation. In **Proceedings of the 30th ACM International Conference on Information & Knowledge Management**, 2021.

[6] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In **Proceedings of the 18th International Conference on Machine Learning**, 2001.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In **Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, 2019.

[8] Tetsuro Sasada, Shinsuke Mori, Tatsuya Kawahara, and Yoko Yamakata. Named entity recognizer trainable from partially annotated data. In **Proceedings of the 14th International Conference of the Pacific Association for Computational Linguistics**, 2016.

[9] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A Python natural language processing toolkit for many human languages. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations**, 2020.

[10] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In **Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL**, 2003.

# A　付録 (Appendix)

## A.1　場所表現推定器

場所表現認識に用いた推定器は，図 5 に示すような構造になっており，下記の手順で場所表現タグを推定する．

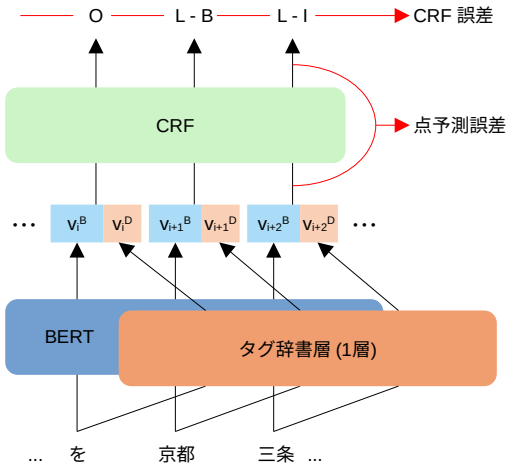


**図 5**　場所表現推定器

1. BERT モデルを用いて，トークン列の各トークン ($w$) を 768 次元のベクトルに埋め込む．学習の際は，テキストのまとまりに関係なく文毎に入力する．推定の際は，段落毎に入力する．$[v_1^B, ..., v_n^B] = BERT([w_1, ..., w_n])$
2. あるトークンが場所表現か否か，すなわちタグとして L-B，L-I がそれぞれありうるかを記したタグ辞書に従い，上記ベクトルに 32 次元増結する．例えば，三条が L-B としても L-I としてもありうると記されていた場合には，L-B タグに対応する 32 次元のベクトルと L-I タグに対応する 32 次元のベクトルの和を (1) に増結する．タグ辞書に登録されていないトークンの場合，もしくは O タグとして登録されていた場合，このベクトルは零ベクトルである．$v_i^D = dictionary(w_i)$
3. CRF モデルを用いて，この 800 次元のベクトル列からタグ列を推定する．学習の際は，CRF モデルおよびその下層の BERT モデル，辞書層を学習する．この際，CRF 損失関数とその中の観測素性部分だけの損失関数を合わせた損失関数を最小化するように学習する．これは，テキストの一部 (特に場所表現) だけをアノテーションした学習データも利用できるようにするためである．$Loss = Loss^{CRF} + Loss^{Pointwise}$

BERT モデルは Hugging Face により提供されている事前学習済みモデル bert-base-uncased[4] を用いた．

## A.2　評価指標の計算式

推定緯度経度と正解緯度経度の誤差が $X$ km 以内に収まっている場所表現を合格として

$$X\text{ km 正確度} = \frac{\text{誤差 } X \text{ km 以内の空間表現数}}{\text{アノテーション空間表現数}}$$

$$\text{精度} = \frac{\text{合格の空間表現数}}{\text{ツールが認識した空間表現数}}$$

$$\text{再現率} = \frac{\text{合格の空間表現数}}{\text{評価データ中の空間表現数}}$$

$$\text{F}_1\text{値} = \left[\frac{(\text{精度})^{-1} + (\text{再現率})^{-1}}{2}\right]^{-1}$$

## A.3　対象とした場所表現

場所表現の基準は原らの基準 [4] を参考にした．この基準では場所表現を，大きく 2 つに分類している．それは，「京都」や「市役所」といったそれ自体がある絶対的な場所を示している表現 (以下，絶対表現) と，「〜の 10 km 東」といった絶対表現が示す場所を参照して別の場所を示している表現 (以下，相対表現) である．絶対表現はさらに，「京都」のような固有名詞と「市役所」のような一般名詞に分けられる．絶対表現が示す場所は文脈によって決定されるべき曖昧性を含んでいる場合もあるが，高々有限個の絶対位置を示しているという点で相対表現とは区別される．一方相対表現は，「(欧州から見た) 極東」といったように絶対表現が省略される場合もあるが，ほとんどは絶対表現に付随して用いられ，絶対表現が示す場所をずらす機能がある．

本研究の緯度経度推定ツールでは，場所表現の内，絶対表現に着目し自動認識を行った．これは原らの基準において，L タグで表される表現である．特に，固有名詞の絶対表現のみを取り扱うことにした．これは，一般名詞の絶対表現が用いられる場合は，そのテキスト内で既に固有名詞により同じ場所が記述されている場合がほとんどであるため，固有名詞の絶対表現さえ把握すれば，テキスト中の絶対表現が示す場所をほぼ全て把握できるからである．

4)　https://huggingface.co/bert-base-uncased