一般論文

Peer-Reviewed

固有表現抽出器学習のための Wikipedia リンク拡張と期待エンティティ率推定

中山　功太[a,d]・栗田　修平[a,e]・馬場　雪乃[b]・関根　聡[a,c]

文章中の固有表現の言及を検出し，人名や地名といったクラスへの分類を行う固有表現抽出は自然言語処理の基礎技術である．近年ではより細分化されたクラスへの分類が求められている．固有表現抽出器の構築には一般的に学習データが必要であるが，特に細分化されたクラスを対象とする場合，人手による学習データ作成は非常にコストが高い．先行研究は Wikipedia のリンク構造を活用して学習データを自動作成することを提案している．Wikipedia のリンクは固有表現抽出器の学習には不十分であるため，先行研究では，固有表現の先頭を大文字にする等の英語等の特徴を活用してリンクを拡張している．しかし，これらの手法は言語依存であり日本語には適用できない．本研究では，Wikipedia のリンク付与ガイドラインの定義を活用することでリンク拡張を行う手法を提案する．加えて，Wikipedia 記事中のエンティティ率を推定する手法を提案し，推定値により学習時に制約をかけることで前者では拡張できないリンクの影響を軽減する．本研究では，拡張固有表現階層の 200 カテゴリーを対象に実際に日本語の固有表現抽出器を構築する．提案手法の評価のため，ウェブニュース記事に対して人手によるラベル付けで評価データを作成し，実験により先行研究より高品質な固有表現抽出器が学習できることを示した．

キーワード：固有表現抽出，拡張固有表現階層，Wikipedia

Wikipedia Link Extension and Expected Entity Rate Estimation for Training Named Entity Recognizer

Kouta Nakayama[a,d], Shuhei Kurita[a,e],
Yukino Baba[b] and Satoshi Sekine[a,c]

Named entity recognition (NER), which detects named entities in text and classifies them as PERSON or LOCATION, is a fundamental technique in natural language processing. Recently, NER systems have been demanded for classification into fine-grained classes. Generally, training data are required to construct an NER system. However, manual labeling is costly, particularly when fine-grained classes are involved. Previous studies proposed utilizing the link structure of Wikipedia to automatically

[a] 理化学研究所 革新知能統合研究センター, RIKEN AIP
[b] 東京大学, The University of Tokyo
[c] 国立情報学研究所 大規模言語モデル研究開発センター, NII LLMC
[d] 現所属：国立情報学研究所 大規模言語モデル研究開発センター, Present affiliation: NII LLMC
[e] 現所属：国立情報学研究所, Present affiliation: NII

create training data for NER. Wikipedia links are insufficient for constructing training data. Therefore, researchers have attempted to extend these links using language-dependent methods that do not apply to Japanese. In this study, we propose a method for extending links using deep learning and a method to estimate the entity rate of Wikipedia articles. The estimated value is used to impose constraints during training, thereby mitigating the effects of links that cannot be extended using the former method. Additionally, we construct a Japanese NER system for 200 categories of an extended named entity hierarchy. For evaluation, we create data by manually annotating web news articles. Experimental results show that the proposed method performs better than previous methods.

Key Words: *Named Entity Recognition, Extended Named Entity Hierarchy, Wikipedia*

1 はじめに

固有表現抽出とは，文章中の固有表現の言及を検出し “人名” や “組織名” といった固有表現クラスへの分類を行う自然言語処理の基礎技術である．本技術は，質疑応答や機械翻訳といった技術への応用が期待される．固有表現抽出は近年の深層学習の発展により飛躍的な性能向上を遂げた (Li et al. 2020; Wang et al. 2021)．しかし，固有表現抽出システムは人手で構築された学習データのもとに成り立っており，データ構築コストの高さが指摘されている (Nothman et al. 2013)．近年では，より細かく固有表現を解析すべく，数種類の固有表現クラスに留まらず数百種類といった固有表現クラスに対して分類を行う細分類固有表現抽出の必要性が議論されている (Mai et al. 2018)．この場合，扱う固有表現クラスの数に伴い，学習データのラベル付けコストも高くなる．

この問題を解決するため，Wikipedia 記事のリンク構造を活用して学習データを自動生成することでコスト削減に取り組む研究が多く行われている (Richman and Schone 2008; Nothman et al. 2008, 2013; Al-Rfou et al. 2015; Pan et al. 2017; Ghaddar and Langlais 2017; Cao et al. 2019; Strobl et al. 2020; Ling and Weld 2021; Tedeschi et al. 2021; Malmasi et al. 2022; Tedeschi and Navigli 2022; Strobl et al. 2022)．Wikipedia は Web 上で閲覧編集が可能な百科事典であり，膨大なエンティティの集合と，各エンティティを説明する記事からなる．記事中では出現した固有表現の言及に対して必要に応じて Wikipedia 内の他のエンティティを示すリンクが付与される．したがって，何らかの手段を用いて各エンティティに対して固有表現クラスを付与することで，リンク構造から固有表現抽出の学習データを自動生成することができる．先行研究では，Wikipedia 記事に付与されているカテゴリーや，外部データにより各 Wikipedia 記事に対して付与されたカテゴリーを固有表現クラスに対応付けることで各エンティティの分類を行っている (Nothman et al. 2013; Al-Rfou et al. 2015; Ghaddar and Langlais 2017)．しかし，カテゴリーの定義と固有表現の定義が異なるため対応付けの際に分類誤りが発生するといった

問題がある．

Wikipedia のリンク構造は，リンク省略や NIL 言及により固有表現抽出器の学習に用いるには不足している．

リンク省略　Wikipedia では，Wikipedia ガイドラインにより多くのリンクが省略されている．本ガイドラインには，リンクが煩雑になることを防ぐために「同一エンティティを指す言及は初出の場合のみリンクする」といったルールが記載されている．例えば，記事中に言及「日本」が出現し，既に『日本』というエンティティにリンクしていた場合，以降の『日本』を指す言及へのリンクは省略される[1]．また，「記事に紐づくエンティティの説明に重要であると判断される言及のみリンクする」といったルールも存在する．つまり，国名のように一般に認知されているエンティティに対するリンクが省略される場合がある．先行研究では，英語等の言語において固有表現の単語の先頭が大文字化されるといった表層的な言語特徴を活用したリンク検出 (Nothman et al. 2008; Richman and Schone 2008; Nothman et al. 2013; Strobl et al. 2020; Ling and Weld 2021; Tedeschi et al. 2021; Malmasi et al. 2022) や，固有表現辞書を用いて検索を行いリンクを拡張する表層マッチ (Nothman et al. 2008; Richman and Schone 2008; Nothman et al. 2013; Al-Rfou et al. 2015; Ghaddar and Langlais 2017; Strobl et al. 2020; Ling and Weld 2021; Tedeschi et al. 2021; Tedeschi and Navigli 2022; Strobl et al. 2022) によりリンク省略に対応している．しかし，前者のような言語依存の特徴のほとんどが日本語には適用できない上，後者のような簡易的な表層マッチのみではカバー率の低下や，精度の低下といった問題が引き起こされる．

NIL 言及　固有表現の言及であるものの，紐づくエンティティが Wikipedia に存在しない場合はリンクは付与されない．例えば，宮沢賢治の母親を指す言及「宮沢イチ」が出現した場合，Wikipedia には紐づく記事が存在しないため，リンクは付与されない．このような行き先のない言及を以下 NIL 言及と呼ぶ．NIL 言及を無視してデータ構築を行った場合，本来ラベルがあるにも関わらずラベル無しと表示される偽陰性ラベルが混在してしまうため，データセットのカバー率が低下し，後続のモデル学習に悪影響を与える．先行研究では，外部辞書を用いた表層マッチ (Richman and Schone 2008; Strobl et al. 2020) や深層学習による偽陰性ラベル検出 (Pan et al. 2017)，言語特徴や深層学習を用いた偽陰性ラベルを含む文のフィルタリング (Tedeschi and Navigli 2022) 等の技術を開発し，解決に取り組んでいる．しかし，他言語で使用されるような辞書や言語特徴は直接日本語に適用できない上，簡易なラベル補完はノイズを増加させる危険性があり，フィルタリングは有用な教師情報を除去してしまうといった問題がある．

本研究では，これらのリンク省略と NIL 言及に対処するための言語非依存な手法を提案する．提案手法とともに Wikipedia から固有表現抽出器を学習する工程を図 1 に示す．

[1] 例外として，記事中で重要と判断された場合，複数回リンクが付与される場合もある．

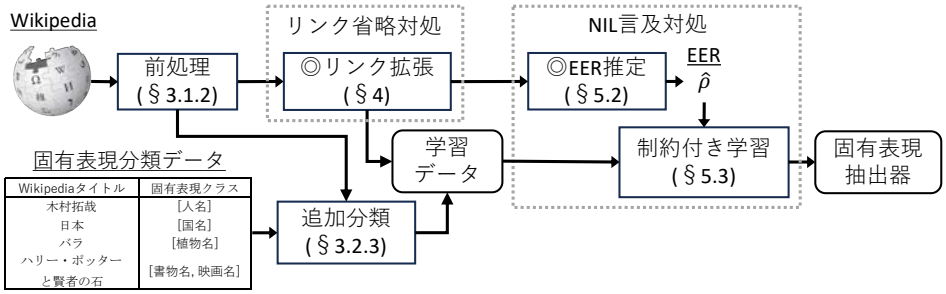

図 1　本研究における Wikipedia からの固有表現抽出器学習工程：◎は提案手法を示す.

リンク省略に対処するため，Wikipedia ガイドラインを活用した深層学習ベースのリンク拡張手法を提案する．本手法では，ガイドラインのうち「同一エンティティに関するリンクは初出の場合のみ付与する」というルールに着目する．これはすなわち，リンク以前の文章には同一エンティティを指す言及が存在しないことを意味している．この性質を利用することで，リンクが示す言及の単語とエンティティを正のペア，リンク以前の単語とエンティティを負のペアとして扱うことができるようになる．これらのペアから単語とエンティティのペアの判定を行う二値分類器を学習し，分類器により全ての単語に対して紐づくエンティティを予測することで省略されたリンクの補完を試みる．

NIL 言及に対処するため，文章中の期待エンティティ率 (Expected Entity Rate; EER) を推定する手法を提案し，固有表現抽出器の学習に推定値による制約を適用する．本手法では，各エンティティのリンク頻度と頻度ごとのエンティティ数の関係を近似することで，文章に含まれる固有表現の言及の割合を示す期待エンティティ率を推定する．エンティティ率を用いて学習時に制約を課し偽陰性ラベルの影響を軽減する手法 (Effland and Collins 2021) に対して本推定値を適用することで，NIL 言及の影響軽減を試みる．

本研究では，リンク省略と NIL 言及によるリンク不足の課題を独立して扱うため，エンティティに対する固有表現クラスの割り当てには人手により作成された正解ラベルを用いる．これにより，Wikipedia のカテゴリー等と固有表現クラスの紐付けにより発生する誤りから，提案手法による誤りを切り分けて検証することが可能になる．Wikipedia の構造化を行う森羅プロジェクト (Sekine et al. 2019) では，日本語の Wikipedia のおよそ 8 割の記事が，拡張固有表現階層 (Sekine et al. 2002) で定義されたカテゴリーに人手で分類されており，本研究ではこちらを用いる．拡張固有表現階層は Wikipedia を階層的に分類するための定義であり，本研究では “名前” 以下の約 200 カテゴリーを固有表現クラスとして扱う.

性能の比較に評価データが必要であるため，本研究では日本語 Wikinews から収集されたニュース記事に対して人手によりラベルを付与することで評価データを作成する．本作業は，

拡張固有表現に対する深い知識が求められるので，前述の Wikipedia 記事分類を行った作業者に対して依頼した．

本研究で，構築された固有表現抽出器は JENER (JENER: Japanese Extended Named Entity Recognizer) として公開している[2]．

本論文の貢献は以下の通りである．

- Wikipedia のリンク省略に対処すべく，Wikipedia ガイドラインを活用した深層学習ベースのリンク拡張手法を提案した．
- Wikipedia 記事中の期待エンティティ率 (EER) の推定手法を提案し，EER により固有表現抽出器の学習に制約をかける既存手法を適用することで，NIL 言及によるラベル欠落の影響を軽減した．
- 日本語 Wikinews から収集されたニュース記事に対して人手でラベルを付与し，評価セットとして公開した[3]．
- 日本語のように固有表現に関する表層的な特徴が少ない言語において，先行研究より高品質な固有表現抽出器の学習が可能であることを示した．
- 予測結果の詳細な分析により Wikipedia のリンク構造を用いて固有表現抽出器を学習する際の更なる課題を示した．

以下に本論文の構成を示す．2 節では，固有表現抽出と Wikipedia を活用した固有表現抽出器学習手法について関連研究を紹介する．3 節では Wikipedia や拡張固有表現の説明と前処理に関しての記述を行う．4 節では，Wikipedia のガイドラインにより省略されたリンクを拡張するための手法を提案する．5 節では，NIL 言及が学習に及ぼす影響を軽減するために，期待エンティティ率の推定を行う手法を提案し，既存手法により学習制約を適用する．6 節では，実験に用いる評価セットの構築方法や，提案手法や比較手法の実験設定について記述し，7 節で実際に固有表現抽出機を学習し，評価，考察する．8 節では，拡張したリンク，推定した期待エンティティ率，固有表現抽出器の予測誤りに関する分析と，学習データにおける頻度をもとにした分析を行い，続く 9 節で Wikipedia から固有表現抽出器を学習する際の課題を整理する．

2 関連研究

本節では，2.1 節で固有表現抽出全体における関連研究，2.2 節で Wikipedia のリンクを活用した固有表現抽出における関連研究を列挙する．

[2] https://github.com/k141303/JENER
[3] https://github.com/k141303/WikinewsENER

2.1 固有表現抽出

固有表現抽出 (Nadeau and Sekine 2007; Li et al. 2022) は，文章中の固有表現の言及を検出し“人名” や “組織名” といった固有表現クラスに分類する技術である．固有表現抽出は，言及検出とクラス分類に分けて考えることができ，後者はエンティティタイピングとも呼ばれる．固有表現という概念は MUC-6 (Grishman and Sundheim 1996) で初めて認識され，同会議において，文章中の “人名”，“組織名”，“地名”，“通貨”，“時間表現”，“割合表現” を検出するタスクが実施された．以降，固有表現抽出は自然言語処理の重要技術として注目されており，IREX (Sekine and Eriguchi 2000)，CoNLL03 (Tjong Kim Sang and De Meulder 2003)，ACE (Doddington et al. 2004) 等の様々なイベントが固有表現抽出を扱うタスクを開催している．

固有表現抽出は主に数クラスを対象としているが，数十，数百といったクラスを扱う場合は，細分類固有表現抽出と呼ばれる．英語の固有表現抽出タグ付きコーパスに着目すると，OntoNotes (Weischedel et al. 2013) は 18 クラス，BBN (Weischedel et al. 2005) は 64 クラス，EER (Ringland et al. 2019) は 114 クラス，FG-NER (Mai et al. 2018) は 200 クラスを対象としている．クラス定義は様々であるが，FG-NER は拡張固有表現階層 (Sekine et al. 2002) で定義された固有表現クラスを用いており，本研究でも同定義を使用する[4]．

日本語においても固有表現抽出の研究は盛んに行われており，タグ付きコーパスに着目すると，IREX が配布している新聞記事を対象とした IREX コーパスや，現代日本語書き言葉均衡コーパス (Maekawa et al. 2014) と新聞記事を対象とした拡張固有表現タグ付きコーパス (橋本他 2008)，現代日本語書き言葉均衡コーパスを対象とした BCCWJ 基本固有表現抽出コーパス (Iwakura et al. 2016)，Wikipedia を対象としたコーパス (近江 2021) などが構築されている．拡張固有表現タグ付きコーパスは拡張固有表現階層をクラス定義として用いているが，過去の定義を用いており最新の定義との互換性が低いことから，本研究では本データは使用せず新しく評価データを作成する．

近年の深層学習の発展により，固有表現抽出の性能は飛躍的に向上している．深層学習を用いて固有表現抽出を解く場合，各単語を分類する系列ラベリングの形に変換して解くことが一般的である．この場合，各ラベルは IOB2 形式や BILOU 形式に変換される．IOB2 形式では，固有表現の開始を示す `B` タグ，固有表現の継続を示す `I` タグ，固有表現以外の範囲を示す `O` タグが定義されており，`B`, `I` タグは言及範囲の固有表現クラスと組み合わせて $\texttt{B}-\texttt{xxx}, \texttt{I}-\texttt{xxx}$ の様に使用される．BILOU 形式では，IOB2 形式に固有表現の終了を示す `L` タグと固有表現が単一のトークンからなる場合に使用される `U` タグが追加される．これらの形式の場合，例えば `O` タグから `I` タグへ遷移することはあり得ない．これらのタグ遷移に対して，深層学習モデルの出力に対して条件付き確率場 (CRF) (Lafferty et al. 2001) により制約をかける場合がある．本

[4] 現時点で最新の定義を用いるため FG-NER で使用された定義と多少異なる．

研究でも先行研究に倣い CRF を使用する．

固有表現抽出の性能向上に貢献している深層学習モデルとして，Bi-LSTM (Graves et al. 2013) などの再帰型ニューラルネットや，BERT (Devlin et al. 2019) などの Transformer (Vaswani et al. 2017) をベースにしたニューラルネットがあげられる．また，性能向上はモデル構造の優位性のみではなく事前学習と呼ばれる技術によっても得られている．事前学習は Wikipedia やウェブデータなどの大規模なコーパスを用いて言語特徴をモデルに教え込む手法であり，BERT の場合は文章中の隠された単語を当てるマスク言語モデリングという手法を用いて事前学習される．その性能の高さから本研究においても事前学習を行った BERT を使用する．

2.2 Wikipedia のリンクを活用した固有表現抽出

固有表現抽出器の学習には多くの場合，ラベル付きデータセットが必要であるが，その構築コストの高さが問題視されている．そのため，Wikipedia のリンクを活用した固有表現抽出器の学習手法が十数年間で多く提案されてきた (Richman and Schone 2008; Nothman et al. 2008, 2013; Al-Rfou et al. 2015; Pan et al. 2017; Ghaddar and Langlais 2017; Cao et al. 2019; Strobl et al. 2020; Ling and Weld 2021; Tedeschi et al. 2021; Malmasi et al. 2022; Tedeschi and Navigli 2022; Strobl et al. 2022)．一覧を表 1 にまとめる．

Wikipedia のリンク構造から固有表現抽出器を学習する際，各リンク先のエンティティに固有表現クラスを付与する必要がある．また，Wikipedia のリンクはリンク省略や NIL 言及により固有表現抽出器の学習に用いるには不足しており，その対策が必要である．

先行研究では，Wikipedia 記事に付与されているカテゴリーや，外部データにより各 Wikipedia 記事に対して付与されたカテゴリーを固有表現クラスに対応付けることで各エンティティの分類を行っている (Nothman et al. 2013; Al-Rfou et al. 2015; Ghaddar and Langlais 2017)．しかし，カテゴリーの定義と固有表現の定義が異なるため対応付けの際に分類誤りが発生するといった問題がある．本研究では，エンティティ分類の問題をリンク不足の問題と切り分けて考えるために，エンティティに対して人手でラベルを付与したデータを用いる．

先行研究は，Wikipedia のリンク省略や NIL 言及に様々な手法で対処しており，最も用いられている手法は表層マッチによるリンク拡張である．Nothman et al. (2008) は，タイトル，エイリアス，リダイレクト等からなる固有名辞書を用いて，表層マッチを適用することで省略されたリンクを補完している．一般的に表層マッチを適用した場合，文章中の固有表現ではない範囲が誤って検出されることにより，多くのノイズが含まれるといった問題がある．Nothman et al. (2008) は，英語では固有表現の各単語はキャピタライズ（つまり，単語の先頭が大文字化）されることに注目し，それらの単語のみをリンク拡張の対象とすることでノイズの混入を抑制している．後続研究の多くが同等の手法を継承している (Nothman et al. 2008; Richman and Schone 2008; Nothman et al. 2013; Al-Rfou et al. 2015; Ghaddar and Langlais 2017; Strobl et al. 2020;

先行研究	リンク拡張手法							言語依存知識			
	表層マッチ	フィルタ	EL	ノイズ分離	オーバーサンプリング	自己学習	外部ツール	キャピタライズ	空白分割	その他	外部資源
N2008	D							✓	✓	✓	✓
R2008	D							✓	✓	✓	✓
N2013	D	D						✓		✓	✓
A2015	D				A				✓		✓
P2017						A					
G2017	D										✓
C2019				A							
S2020	D		D					✓		✓	✓
L2021	D							✓	✓	✓	✓
T2021	D	D				A		✓			✓
M2022		D						✓			
T2022	A	A				A					
S2022	A						D			✓	

表 1 Wikipedia を活用した固有表現抽出器の学習手法一覧．“D” は言語依存 (Language “D”ependent) の手法，“A” は言語非依存 (Language “A”gnostic) の手法を示す．“EL” はエンティティリンキングを示す．スペースの関係上，先行研究は省略表記している．対応する研究は次の通りである．R2008 (Richman and Schone 2008)，N2008 (Nothman et al. 2008)，N2013 (Nothman et al. 2013)，A2015 (Al-Rfou et al. 2015)，P2017 (Pan et al. 2017)，G2017 (Ghaddar and Langlais 2017)，C2019 (Cao et al. 2019)，S2020 (Strobl et al. 2020)，L2021 (Ling and Weld 2021)，T2021 (Tedeschi et al. 2021)，M2022 (Malmasi et al. 2022)，T2022 (Tedeschi and Navigli 2022)，S2022 (Strobl et al. 2022).

Ling and Weld 2021; Tedeschi et al. 2021)．また，Wikipedia のリンクやリンクの表層文字列，タイトルは固有名の完全系を取る場合が多く，別名辞書内に短縮系が含まれないといった問題がある．そのため，タイトルを空白で分割し各単語を固有名辞書へ含める (Nothman et al. 2008; Richman and Schone 2008; Doddington et al. 2004; Ling and Weld 2021)，もしくは WordNet や YAGO KG などの外部知識から得られた人名等のリストを固有名辞書に含める (Richman and Schone 2008; Strobl et al. 2020) 等の手法で対処される場合がある．また，その他の言語依存情報として，Mr. Mrs. 等の敬称 (Nothman et al. 2008; Richman and Schone 2008; Nothman et al. 2013)，所有格 (Nothman et al. 2008)，イニシャル (Richman and Schone 2008) などの言語情報を用い，ルール形式で固有表現クラスラベルを拡張する場合がある．いずれも言語に依存した知識を用いており，言語構造が大きく異なる日本語での適用は難しい．日本語では，Wikipedia から得られた固有名の辞書を固有表現抽出器の学習に用いる研究 (風間，鳥澤 2008) があるが，辞書と表層マッチのみを用いており，リンク構造は使用していない．日本語に依存した言語特徴を用いて Wikipedia のリンク不足への対処を試みた研究は我々の知る限りない．

Strobl ら (Strobl et al. 2022) は，外部の固有表現抽出器である CoreNLP (Manning et al. 2014) を使用することで，ノイズの問題を解決しているがツール自体が言語依存である．また，省略されたリンクやノイズを含む文章をフィルタすることで対応する場合もある (Nothman et al. 2013; Tedeschi et al. 2021; Malmasi et al. 2022; Tedeschi and Navigli 2022)．しかし多くの場合，いずれかの言語依存の情報を用いている．

我々の知る限り，手法全体で言語依存の情報を使用せずに表層マッチを適用しているのは Tedeschi and Navigli (2022) のみである．この場合，表層マッチによるノイズと，短縮形を採用できないことによるカバー率の低下に対処する必要があるが，Tedeschi and Navigli (2022) は自己学習を用いることで，モデル予測と矛盾のあるノイズラベルのフィルタリングと，モデル予測によるラベルの拡張を同時に行い問題解決に取り組んでいる．

表層マッチを用いた場合リンク先候補の衝突が発生する場合があるが，多くの場合，直近でリンクされたエンティティや，出現頻度の高いエンティティを採用することで解決している．Strobl et al. (2020) は，外部のエンティティリンキングツールを用いて衝突を回避している．しかし，同じくエンティティリンキングを用いる本研究とは異なり，NIL 言及は考慮しておらず，候補が全て間違っていた場合もいずれかのエンティティが紐付けられる．また，先行研究の場合は，拡張可能なリンクは表層マッチの辞書内に制限されるため，我々の研究よりも拡張対象範囲が大幅に狭い．

表層マッチを用いず言語非依存に対処している研究も存在する．Pan et al. (2017) は文章に対して割り当てられた固有表現ラベルの信頼度を推定後，全てのラベルが閾値を超えている文章を初期セットとし，自己学習によりラベルを拡張している．Cao et al. (2019) はリンク省略や NIL 言及の影響を含む文を分離し，信頼性の高いセットと，低いセットに分けることでこれらの問題に対処している．具体的には，信頼性の低いセットを用いてラベルが付与された範囲のみからモデルを事前学習したのち，信頼性の高いセットを用いて微調整を行っている．

本研究では，表層マッチ，自己学習，ノイズ分離を先行研究を参考に実装し比較対象とする．

3　データと前処理

本節では主に本研究で使用するデータとその前処理に関しての説明を行う．3.1 節では Wikipedia の説明と Wikipedia ダンプデータの前処理に関して，3.2 節では拡張固有表現階層と Wikipedia 記事分類データ，追加の記事分類に関して説明する．

3.1 Wikipedia

3.1.1 リンク構造

Wikipedia は非営利団体であるウィキメディア財団により運営されている自由に編集と閲覧が可能なオンライン百科事典である．Wikipedia は多くの言語を対象としており，そのうち日本語を対象とした記事は 1,136,514 件存在する[5]．Wikipedia では，記事中に Wikipedia に含まれる他のエンティティに紐づく言及が出現した場合，内部リンクが付与される．例えば，「アメリカの作曲家.」という文章があった場合，「アメリカ」は『アメリカ合衆国』というエンティティにリンクされる．また，Wikipedia は固有名に留まらない幅広い内容を扱っており，「作曲家」も『作曲家』というエンティティにリンクされる．本研究では，これらの内部リンクを活用して固有表現抽出器を学習するが，Wikipedia では多くの内部リンクが省略されており，そのまま学習データとして扱うことは難しい．リンク省略は Wikipedia の編集ガイドラインに起因しており，以下のような言及に対するリンクが付与されない．

出現済みの言及 Wikipedia では同一のエンティティを指すリンクは，基本的に初出の場合にのみ付与される．2 回目以降の出現の場合，多くのリンクが省略される．

重要でない言及 Wikipedia では編集者が記事の理解に重要でないと判断した言及はリンクされない．また，例えば「日本」や「アメリカ」といった国名などの，一般に認知されているエンティティを指す言及へのリンクは省略される場合がある．

自己を参照する言及 Wikipedia では自己記事を参照する言及はリンクされない．例えば，『日本』の記事中に，「日本」という言及が出現した場合，リンクは省略される．

また，当然であるが Wikipedia に存在しないエンティティを指す NIL 言及に対するリンクは基本的には付与されない[6]．これらの省略されたリンクや NIL 言及の影響を軽減した上でモデルを学習する手法を考案する必要がある．

3.1.2 前処理

Wikipedia では，機械的に記事を収集するクローリングは禁止されており，Wikipedia 上の情報を使用したい場合は，定期的に公開されているダンプファイルを使用することが推奨されている．本ダンプファイルは XML 形式で記載されている．また，MediaWiki 形式で記載された CirrusSearch ダンプも公開[7]されている．後者は前者と比較して前処理が容易であることから本研究ではこちらを使用する[8]．

初めに CirrusSearch ダンプに含まれるスクリプトやコメントといった余剰な情報を全て削除

[5] 2019 年 01 月 21 日時点の Wikipedia ダンプファイルによる統計.
[6] 今後の記事作成を見越して，タイトルのみの空のエンティティにリンクされる場合があるが，全体のリンクのうちかなり希少であるため本研究では対象外とする.
[7] https://dumps.wikimedia.org/other/cirrussearch/
[8] 本研究で用いる Wikipedia 分類データに合わせ，2019 年 1 月 21 日時点のダンプを使用する.

し，文章とリストと一部の表[9]のみ残す．

Wikipedia においてリンクは [[アメリカ | アメリカ合衆国]] といった表層文字とリンク先記事のペアで表される．ペアが同一表記の場合は片方は省略される．表層文字とリンク先記事タイトルが異なる場合，これらはタイトルの別名として扱うことができる．これらの表層文字から得られる別名をエイリアスと呼ぶ．エイリアスは本研究で使用しないが，先行研究の再現実験に必要であり，別途辞書として保持しておく．また，Wikipedia では記事タイトルはユニークであるが，他のユニークな文字列でも同一の記事にアクセスすることができる．例えば，『アメリカ合衆国』の記事に対して「USA」でもアクセス可能である．これらはリダイレクトと呼び，こちらも先行研究の再現実験に必要であるので別途保持しておく．

Wikipedia では記事の最初が自己のエンティティを指す言及で始まる．これら言及は太字で強調されており，例えば，『アメリカ合衆国』の記事の場合，「**アメリカ合衆国**（アメリカがっしゅうこく，[[英語]]: United States of America）は，北アメリカに位置し...」のように表記される．最初に出現した太字強調を自己記事へのリンクとすることで，自己記事を参照する言及が省略される問題は，出現済みの言及が省略される問題として扱うことができる．また，エンティティの別の表記が存在する場合，上記例のように自己言及に続く括弧内に併記される場合がある．これらも機械的に自己リンクに変換する．

3.2　拡張固有表現階層

3.2.1　階層定義

拡張固有表現階層 (Sekine et al. 2002) は，固有名を階層的に分類するためのカテゴリー定義である．拡張固有表現階層は，ニーズに合わせて都度更新されており，階層構造の変更やカテゴリーの細分化などが行われている．後述する分類データが現時点で最新であるバージョン 9.0 を採用していることから，本研究でも同バージョンを使用する．本定義では，大枠として，“名前”，“時間表現”，“数値表現” を対象としている．その他に，Wikipedia のような知識ベースのエンティティを全て分類するために，一般名詞等を表すエンティティが割り当てられる “CONCEPT” や，曖昧性回避ページ等のメタ的な記事が割り当てられる “IGNORED” といった区分も存在する．本研究では “名前” 以下の末端カテゴリーのみを対象とし，その他の区分はすべて単一の “OTHER” カテゴリーとして扱う．名前以下の末端カテゴリーは 198 件存在し，表 2 に “名前” 以下のカテゴリー階層を示す．4 階層目は括弧内に表記している．4 階層目が存在しない場合は 3 階層目が，3 階層目以下が存在しない場合は 2 階層目が末端カテゴリーとなる．“生物呼称名__その他”，“アドレス__その他”，“電子メイル” カテゴリーを示すリンクが Wikipedia に存在しないことから，本研究ではこれら 3 カテゴリーを除いた 195 カテゴリーを対象とする．

[9] <table>タグで囲まれた部分のみ残す．

2 階層目	3 階層目（4 階層目）
名前__その他	
人名	
神名	
生物呼称名	生物呼称名__その他，動物呼称名（動物呼称名__その他，競走馬名），植物呼称名
組織名	組織名__その他，国際組織名，公演組織名，家系名，民族名（民族名__その他，国籍名），競技組織名（競技組織名__その他，競技連盟名，競技リーグ名，競技団体名），法人名（法人名__その他，非営利団体名，企業名，企業グループ名），政治的組織名（政治的組織名__その他，政府組織名，政党名，内閣名，軍隊名）
地名	地名__その他，ＧＰＥ（ＧＰＥ__その他，市区町村名，都道府県州郡名，国名），地域名（地域名__その他，大陸地域名，国内地域名），地形名（地形名__その他，温泉名，山地名，島名，河川名，湖沼名，海洋名，湾名），天体名（天体名__その他，天体部分名，銀河名，恒星名，惑星__衛星名，星座名），アドレス（アドレス__その他，郵便住所）
施設名	施設名__その他，施設部分名，ダム名，遺跡名（遺跡名__その他，墳墓名），ＦＯＥ（ＦＯＥ__その他，軍事基地名，城名，宮殿名，公共機関名，宿泊施設名，医療機関名，学校名，研究機関名，取引所名，発電所名，公園名，商業施設名，競技施設名，美術博物館名，動植物園名，遊園施設名，劇場名，宗教施設名），交通施設名（交通施設名__その他，停車場名，鉄道駅名，空港名，港名，道路施設名，鉄道施設名），路線名（路線名__その他，道路名，鉄道路線名，航路名，運河名，トンネル名，橋名）
プロダクト名	プロダクト名__その他，株名，便名，識別番号，サービス名，ブランド名，ソフトウェア名，情報機器名，玩具名，楽器名，衣類名，医薬品名，キャラクター名，作品名（作品名__その他，絵画名，番組名，映画名，公演名，音楽名，映像作品名），出版物名（出版物名__その他，新聞名，雑誌名，書物名），ゲーム名（ゲーム名__その他，電子ゲーム名），食べ物名（食べ物名__その他，料理名），武器名（武器名__その他，火器名），乗り物名（乗り物名__その他，車名，列車名，飛行機名，船名，宇宙船名，軍用車両名，軍用機名，艦艇名），主義方式名（主義方式名__その他，罪名，等級名，賞名，勲章名，貨幣名，技術名，規格名，制度名，試験名，主義思想名，文化名，宗教名，学問名，理論名，流派名，競技名，政策計画名），規則名（規則名__その他，条約名，法令名），称号名（称号名__その他，地位職業名），言語名（言語名__その他，国語名），単位名（単位名__その他，通貨単位名），バーチャルアドレス名（バーチャルアドレス名__その他，チャンネル名，電話番号，電子メイル，ＵＲＬ）
イベント名	イベント名__その他，自然現象名，自然災害名（自然災害名__その他，地震名，水害名），催し物名（催し物名__その他，祭礼名，選挙名，競技会名，展示会名，会議名），事故事件名（事故事件名__その他，交通事故名，社会事件名，戦争名）
自然物名	自然物名__その他，元素名，化合物名，鉱物名，生物名（生物名__その他，空想生物名，細菌__ウイルス名，真菌類名，軟体動物名，節足動物名，昆虫類名，魚類名，両生類名，恐竜名，爬虫類名，鳥類名，哺乳類名，植物名），生物部位名（生物部位名__その他，動物部位名，植物部位名）
病気名	病気名__その他，動物病気名
色名	

表 2　拡張固有表現階層カテゴリー一覧

3.2.2 Wikipedia 分類データ

Wikipedia から固有表現抽出器の学習を行うには，Wikipedia の各エンティティが固有表現クラスに分類されている必要がある．本研究は，リンク省略や NIL 言及からなるリンク不足の解消のみを目的とするため，人手で Wikipedia 記事に固有表現クラスを付与したデータを用いる．Wikipedia の構造化を目標とする森羅プロジェクト (Sekine et al. 2019) という取り組みにおいて，日本語の Wikipedia が拡張固有表現階層の末端カテゴリーに分類されている．本データは公開されており，本研究ではこちらを使用する．以下，本データを Wikipedia 分類データと呼ぶ．Wikipedia 分類データは，エンティティが他のエンティティの記事からリンクされている数を示す被リンク数が 5 以上のエンティティのみを対象としており，全件 1,136,514 件中 920,444 件が分類されている．ラベル付けは，人手でラベル付けされた少量のデータから学習した機械学習モデルにより対象データを分類した上で，再度人手で確認する形で行われている．基本的には，Wikipedia の各記事に対して単一のカテゴリーが割り当てられるが，一意に定まらない場合は複数のカテゴリーが付与される．例えば，『ハリー・ポッターと賢者の石』といった記事には，“映画名”，“書物名” カテゴリーが付与されている．全体のうち，18,246 件 (1.98%) の記事に複数カテゴリーが付与されている．

3.2.3 追加分類

Wikipedia の各記事が分類されている場合，記事中のリンク先が指す固有表現クラスを参照することで，リンクを固有表現ラベルに変換可能である．その場合，Wikipedia の記事は全件分類されていることが望ましいので，深層学習モデルを用いて，未分類の 216,070 件に対して分類を行う．この時，すでに分類されているデータのうち 500 件を開発データとしてランダムにサンプリングし，残りを学習データとして切り分ける．分類モデルには RoBERTa (Liu et al. 2020) を採用する．使用する RoBERTa は 6.2 節で説明するものと同じである．分類には各エンティティの記事の先頭 510 トークンを使用し，前後にそれぞれ特殊トークンである [CLS]，[SEP] を結合した計 512 トークンを RoBERTa に入力する．この際，入力長が 512 トークンに満たない場合は，特殊トークンである [PAD] をトークン列の末尾に追加することで長さを合わせる．学習は 10 エポック行う．各エポックごとに開発データを用いて評価を行い，評価値が最も高かったモデルを最終的な予測に使用する．評価は，各エンティティに対する予測カテゴリーが完全一致していたものを正解として扱い，その正解率を評価値とする．本問題は，各インスタンスに対して，複数のラベルが付く多クラス多ラベル分類であるため，各カテゴリーごとの多ラベル分類タスクとして扱い，損失関数には二値交差エントロピーを使用する．バッチサイズには 200 を使用し，最適化関数には Adam を使用する[10]．学習の結果，開発データにおける評価値

[10] 学習率には 5.0×10^{-5} を使用し，その他のパラメーターは $\epsilon = 1.0 \times 10^{-8}$，$\beta_1 = 0.9$，$\beta_2 = 0.999$ 使用する．

は 99.1%であり，予測のうち 5,839 件 (2.70%) に複数のラベルが付与された．

4　提案手法 1：深層学習によるリンク拡張

本節では，編集ガイドラインにより省略されている Wikipedia のリンクを拡張するための手法を説明する．リンク拡張の対象は，紐づくエンティティが Wikipedia に存在する言及のみであり，紐づくエンティティが Wikipedia に存在しない NIL 言及に関しては 5 節で扱う．

提案するリンク拡張手法では，図 2 に示すような深層学習モデルを用いる．本モデルは，検索器，符号器，修正器からなる．検索器は入力トークンに対して最も類似するエンティティを検索する．図 2 の場合，入力トークンである $\{x_1(\text{キム}), x_2(\text{タク}), x_3(\text{は})\}$ に対して，それぞれ $\{e_1(\text{木村拓哉}), e_1(\text{木村拓哉}), e_2(\text{日本})\}$ が予測されている．符号器はこれらのトークンとエンティティのペアの一致率を求める．図 2 の場合，それぞれのペアに対して $\{0.9, 0.9, 0.1\}$ といった一致率が得られている．ここで，一致率 0.5 以上のトークンとエンティティを紐づけた場合，同一のエンティティを指す連続したトークン $\{x_1(\text{キム}), x_2(\text{タク})\}$ をエンティティ $e_1(\text{木村拓哉})$ へのリンクとして扱うことができる．しかし，検索器と符号器はリンクの言及範囲に関する情報を直接学習していないため，予測したリンクの言及範囲が誤っている場合がある．そのため，リンクの言及範囲を学習した修正器により，最終的なリンク範囲を決定する．図 2 の場合，修正器により $\{\text{B}, \text{L}, \text{U}\}$ タグが予測されており，$\{x_1(\text{キム}), x_2(\text{タク})\}$ と，$\{x_3(\text{は})\}$ をそれぞれ言及範囲として扱うことができる．符号器の結果と統合し，一致率が 0.5 以上である前者のみが有効なリンクとして出力される．

以下では，4.1 節で使用する記号を定義したのち，4.2 節で検索器，4.3 節で符号器，4.4 節で

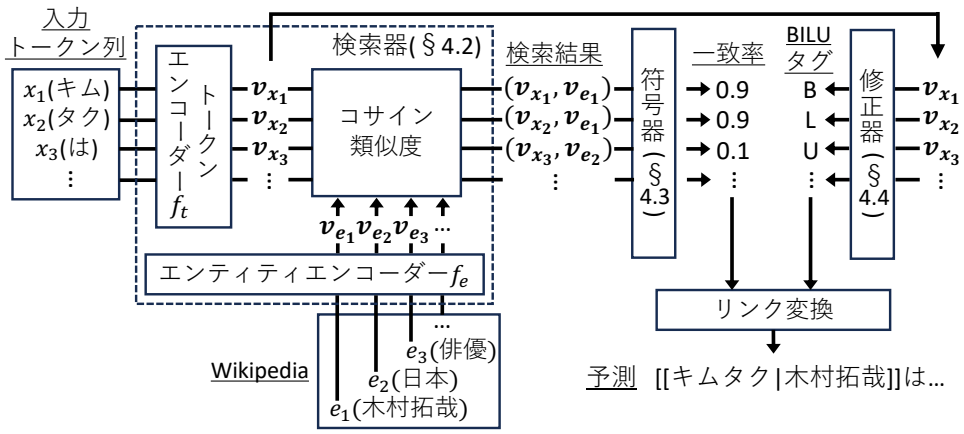

図 2　リンク拡張モデル概要

修正器のモデル構造と学習方法について解説を行う．最後に，4.5 節でリンク拡張モデル全体の学習とリンク拡張の流れについて説明する．

4.1 記号の定義

本小節では，リンク拡張の説明に用いる記号を定義する．各記号の一覧とその説明を表 3 に示す．

Wikipedia 記事の集合を $\mathcal{A} = \{a_1, ..., a_{|\mathcal{A}|}\}$ と表す．ここで，$|\mathcal{A}|$ は対象とする言語の Wikipedia の記事数である．各記事は単語もしくはより細かく分割された単位であるトークンの列 $a = \{x_1, ..., x_{|a|}\}$ で構成されており，ここで，$|a|$ は各記事のトークン長である．Wikipedia に記事が存在するエンティティの集合を $E = \{e_1, ..., e_{|\mathcal{A}|}\}$ とし，Wikipedia に記事が存在しないエンティティ（NIL エンティティ）の集合を $\overline{E}$ と表す．Wikipedia の各エンティティ $e_i \in E$ は，対応する説明文 $a_i \in \mathcal{A}$ によって一意に定義される．

拡張固有表現階層の末端カテゴリーを c，その集合を C とする．Wikipedia のエンティティ e に付与されたカテゴリーの集合を $C_e \subset C$ と表す．ここで，$|C_e| \geq 1$ である．

記号	説明
$\mathcal{A} = \{a_1, ..., a_{\|\mathcal{A}\|}\}$	Wikipedia 記事集合
$\|\mathcal{A}\|$	対象とする言語の Wikipedia 記事数
$a = \{x_1, ..., x_{\|a\|}\}$	Wikipedia 記事
$\|a\|$	記事のトークン長
x	Wikipedia 記事のトークン
$E = \{e_1, ..., e_{\|\mathcal{A}\|}\}$	Wikipedia のエンティティ集合
$\overline{E}$	Wikipedia に記事が存在しない (NIL) エンティティ集合
e_i	記事 a_i によって定義されるエンティティ
C	拡張固有表現の末端カテゴリー集合
$c \in C$	拡張固有表現の末端カテゴリー
$C_e \subset C$	エンティティ e に付与された拡張固有表現カテゴリー
$M = \{m_1, ..., m_{\|M\|}\}$	既にリンクされている言及集合
$\overline{M} = \{m_{\|M\|+1}, ..., m_{\|M\|+\|\overline{M}\|}\}$	リンクが付与されていない言及集合
$m = \{x_o, ..., x_{o+\|m\|-1}\}$	言及
$\|m\|$	言及長
o	言及の開始オフセット
$U = \{(m, e)\}$	Wikipedia のオラクルリンク集合
$S = \{(m, e) \in U \| m \in M\}$	既に付与されているリンクの集合
$\overline{S} = \{(m, e) \in U \| m \in \overline{M}\}$	付与されていないリンクの集合
$\overline{S}_{(\overline{M} \to E)} = \{(m, e) \in \overline{S} \| e \in E\}$	$\overline{S}$ のうち Wikipedia に存在するエンティティを指すリンクの集合
$\overline{S}_{(\overline{M} \to \overline{E})} = \{(m, e) \in \overline{S} \| e \in \overline{E}\}$	$\overline{S}$ のうち NIL エンティティを指すリンクの集合

表 3　リンク拡張に使用する記号一覧

Wikipedia記事全体$\mathcal{A}$で既にリンクが付与されている言及集合を$M = \{m_1, ..., m_{|M|}\}$, リンクが付与されていない言及集合を$\overline{M} = \{m_{|M|+1}, ..., m_{|M|+|\overline{M}|}\}$と表す．ここで$m = \{x_o, ..., x_{o+|m|-1}\}$は言及に含まれるトークン列を示しており，$o$は言及の開始オフセット，$|m|$は言及トークン列長を表している．各リンクは言及と対応するエンティティのペア(m, e)で表す．Wikipediaに含まれるすべての固有表現の言及$|M| \cup |\overline{M}|$とそれらに紐づくエンティティからなるリンクの集合を$U = \{(m, e)\}$とする．ここで，Uをオラクルリンク集合と呼ぶ．

集合Uは，既に付与されているリンクの集合$S = \{(m, e) \in U | m \in M\}$と付与されていないリンクの集合$\overline{S} = \{(m, e) \in U | m \in \overline{M}\}$からなる．$\overline{S}$は，Wikipediaに存在するエンティティ$e \in E$を指すリンクの集合$\overline{S}_{(\overline{M} \to E)} = \{(m, e) \in \overline{S} | e \in E\}$と，NILエンティティ$e \in \overline{E}$を指すリンクの集合$\overline{S}_{(\overline{M} \to \overline{E})} = \{(m, e) \in \overline{S} | e \in \overline{E}\}$からなる．リンク拡張の目的は，リンク$\overline{S}_{(\overline{M} \to E)}$を全て検出することである．

4.2　検索器

検索器は，エンティティリンキングの手法を用いており，入力シーケンス$X = \{x_1, ..., x_{|X|}\}$の各トークン$x$に最も類似するWikipediaのエンティティ$\hat{e} \in E$を検索する．検索器は，$x$と$E$を受け取り$\hat{e}$を返す関数$f_{\text{search}}$として定義される．

$$\hat{e} = f_{\text{search}}(x, E)$$

xが$\hat{e}$と完全に一致するかどうか，つまりxが省略されたリンク$(m, e) \in \overline{S}_{(\overline{M} \to E)}$であるかどうかは，4.3節で説明する符号器により判定する．

検索器はエンティティリンキング手法の一つであるデュアルエンコーダー (Gillick et al. 2019) のスキームを参考にしている．デュアルエンコーダーは，言及mをベクトル$\boldsymbol{v}_m$に埋め込む言及エンコーダーf_{m}とエンティティeをベクトル$\boldsymbol{v}_e$に埋め込むエンティティエンコーダーf_{e}からなり，ベクトル間類似度を用いて言及に紐づくエンティティの検索を行う技術である．各エンコーダーはトークン列$X = \{x_1, ..., x_{|X|}\}$を受け取り，埋め込みベクトル列$V_X = \{\boldsymbol{v}_1, ..., \boldsymbol{v}_{|X|}\}$を出力するようなモデルをベースにしている．近年はTransformer (Vaswani et al. 2017) で構成されたBERT (Devlin et al. 2019) などのモデルが用いられており，本研究でもBERTを使用する．また，本研究では後述する理由により言及エンコーダーの代わりにトークンエンコーダーを用いる．

以下では，4.2.1節でトークンエンコーダー，4.2.2節でエンティティエンコーダーのモデル構造について説明し，4.2.3節ではベクトル間類似度の計算方法について，4.2.4節では検索器の学習方法について説明する．

4.2.1 トークンエンコーダー

初めに，言及エンコーダーの説明を行い，エンティティリンキングと検索器の設定の違いを述べた後，トークンエンコーダーを導入する．

言及エンコーダーは言及を受け取り，その埋め込みベクトルを出力するモデルである．言及エンコーダーは言及とその周辺文章からなるトークン列

$$X_m = \{[\texttt{CLS}], x_{o-\omega}, ..., x_{o-1}, [\texttt{START}], x_o, ..., x_{o+|m|-1}, [\texttt{END}], x_{o+|m|}, ..., x_{o+|m|+\omega}, [\texttt{SEP}]\}$$

を入力に取り，以下の様に言及ベクトル $\boldsymbol{v}_m \in \mathbb{R}^{d_{\text{emb}}}$ を出力する．

$$\boldsymbol{v}_m = \text{red}(f_{\text{m}}(X_m)) \tag{1}$$

$[\texttt{CLS}], [\texttt{SEP}]$ はそれぞれ入力列の先頭と末尾に付与される特殊トークン，$[\texttt{START}], [\texttt{END}]$ はそれぞれ言及の先頭と末尾に付与される特殊トークンを示す．d_{emb} は埋め込み空間の次元サイズ，ω は使用する周辺コンテキストの範囲を示す．$\text{red}(\cdot)$ は，f_{m} の出力ベクトル列 V からベクトル $\boldsymbol{v}_m$ を抽出する関数を表す．多くの場合，$\boldsymbol{v}_m$ には，$[\texttt{CLS}]$ に対応する出力ベクトル $\boldsymbol{v}_{[\texttt{CLS}]}$ や，$[\texttt{START}], [\texttt{END}]$ に対する出力ベクトル $\boldsymbol{v}_{[\texttt{START}]}, \boldsymbol{v}_{[\texttt{END}]}$ の要素平均などが使用される．

エンティティリンキングの設定では，学習時と予測時の両方において，エンティティ検索対象の言及範囲が与えられるため，その範囲のみエンコードすれば良い．しかし，検索器の設定では入力トークン全てに対して検索を行う必要があり，言及エンコーダーにより各トークンごとにエンコードすると計算コストが膨大になってしまう．本研究では代わりにトークンエンコーダーを使用する．

トークンエンコーダーは各入力トークンに対する埋め込みベクトルを一度に生成する．トークンエンコーダーはトークン列

$$X = \{[\texttt{CLS}], x_1, ..., x_{l_{\text{t}}-2}, [\texttt{SEP}]\}$$

を入力に取り，各単語ごとのベクトル列 $V_X = \{\boldsymbol{v}_x\} \in \mathbb{R}^{l_{\text{t}} \times d_{\text{emb}}}$ を出力する．

$$V_X = f_{\text{t}}(X) \tag{2}$$

ここで，l_{t} は入力シーケンス長である．

4.2.2 エンティティエンコーダー

エンティティエンコーダーはエンティティを受け取り，その埋め込みベクトルを出力するモデルである．エンティティエンコーダーはエンティティの説明文からなるトークン列

$$X_e = \{\texttt{[CLS]}, x_1, ..., x_{l_e-2}, \texttt{[SEP]}\}$$

を入力に取り，エンティティベクトル $\boldsymbol{v}_e \in \mathbb{R}^{d_{\text{emb}}}$ を出力する．

$$\boldsymbol{v}_e = \text{red}(f_{\text{e}}(X_e)) \tag{3}$$

ここで，l_{e} は入力シーケンス長である．$\boldsymbol{v}_e$ には，各トークンの埋め込みベクトルの平均等も考えられるが，実装の簡素化のため本研究では [CLS] トークンに対する出力ベクトル $\boldsymbol{v}_{\texttt{[CLS]}}$ を用いる．

4.2.3 ベクトル類似度

トークンとエンティティの紐付けは，トークンベクトルとエンティティベクトル間の類似度 $\psi_{\text{search}}(x, e)$ により行われる．類似度関数には内積やコサイン類似度が使用される場合が多く，本研究ではコサイン類似度を用いる．

$$\psi_{\text{search}}(x, e) = \frac{\boldsymbol{v}_x \cdot \boldsymbol{v}_e}{\|\boldsymbol{v}_x\| \|\boldsymbol{v}_e\|} \tag{4}$$

4.2.4 学習手法

検索器は，Wikipedia のリンク $(m, e) \in S$ から学習する．この際，検索器は言及単位ではなくトークン単位で入力を扱うため，リンク (m, e) はトークンレベルのペア $\{(x_o, e), .., (x_{o+|m|-1}, e)\}$ に分解する．検索器は，トークンと紐づくエンティティの正例ペア (x, e^+) の類似度を最大化，トークンと紐づかないエンティティの負例ペアの類似度を最小化することで学習を行う．この時，正例ペアはリンクから得られるが，負例ペアは得られない．Wikipedia に含まれる e^+ 以外のエンティティの集合 $\{e \in E | e \neq e^+\}$ を全て負例として使用することが考えられるが，これは計算コストの観点から現実的ではない．そのため，本研究では負例ペアの選択にバッチ内サンプリングとハードサンプリングを適用する．

バッチ内サンプリングは，学習データからサンプリングされたバッチに含まれるエンティティ集合を負例として扱う手法である．$\mathcal{A}$ からランダムにサンプリングされたバッチを $\mathcal{A}' \subset \mathcal{A}$ とし，バッチ中に出現するリンクの集合を $S' \subset S$ とする．また，S' を構成する言及集合を $M' = \{m | (m, e) \in S'\}$，エンティティ集合を $E' = \{e | (m, e) \in S'\}$ と表す．この時の，E' をバッチ内サンプルと呼び，正例 e^+ 以外のエンティティ $E^- = \{e | e \neq e^+, e \in E'\}$ を負例として扱う．

ハードサンプリングは，正例ペア (m, e^+) に対してモデルが判定を誤りやすい負例 e^- を学習

時のサンプル E' に追加することで，学習中にモデルの判定誤り修正を効率化し，学習の高速化を図る手法である．本研究では，言及 m のベクトル $\boldsymbol{v}_x$ と全てのエンティティベクトル V_E の類似度を求め，e^+ 以外で最も類似度が高かったエンティティ e^- を E^- に追加する．

$$e^- = \underset{e \neq e^+, e \in E}{\operatorname{argmax}} \ \psi_{\text{search}}(x, e) \tag{5}$$

類似度の最適化は，以下の交差エントロピー誤差を最小化することで行う．,

$$\mathcal{L}_{\text{search}} = -\frac{1}{|X_{M'}|} \sum_{x \in X_{M'}} \sum_{e \in E'} p(e|x) \log \left(\frac{\exp(\alpha \cdot \psi_{\text{search}}(x, e))}{\sum_{e' \in E'} \exp(\alpha \cdot \psi_{\text{search}}(x, e'))} \right) \tag{6}$$

ここで，$X_{M'} = \{x | x \in m, m \in M'\}$ は言及内のトークンの集合を表す．$p(e|x)$ は真の確率分布であり，トークン x とエンティティ e が紐づく場合は 1，それ以外は 0 となる．また，α はスケール用の変数である．交差エントロピー誤差を用いるには，コサイン類似度の値の範囲 $[-1, 1]$ は狭く学習に時間がかかるため，α により範囲 $[-\alpha, \alpha]$ に拡張する．

学習後の検索器では以下のようにして，入力シーケンスの各トークンに最も類似するエンティティを得ることができる．

$$f_{\text{search}}(x, E) = \underset{e \in E}{\operatorname{argmax}} \ \psi_{\text{search}}(x, e) \tag{7}$$

4.3 符号器

符号器は，検索器により得られたトークンとエンティティのペアの一致率を出力する．符号器の学習は，二値交差エントロピー誤差の最小化により，正例ペアの一致率を 1，負例ペアの一致率を 0 に近づけることで行われる．Wikipedia のリンクから正例ペアは得られるが，負例ペアは得られないため，Wikipedia のガイドライン等を活用した負例選択方法を提案する．

符号器は，トークンとエンティティのペア (x, e) を受け取りその一致率 $q(e|x)$ を出力する．符号器では，トークンベクトル $\boldsymbol{v}_t$ とエンティティベクトル $\boldsymbol{v}_e$ から以下のように一致率 $q(e|x)$ を計算する．

$$\boldsymbol{v}_c = \frac{\boldsymbol{v}_x}{\|\boldsymbol{v}_x\|} \odot \frac{\boldsymbol{v}_e}{\|\boldsymbol{v}_e\|} \tag{8}$$

$$\boldsymbol{h}_1 = \text{dropout}(\tanh(W_1^{\mathsf{T}} \boldsymbol{v}_c + \boldsymbol{b}_1)) \tag{9}$$

$$\boldsymbol{h}_2 = W_2^{\mathsf{T}} \boldsymbol{h}_1 + \boldsymbol{b}_2 \tag{10}$$

$$q(e|x) = \frac{\exp(h_{2,1})}{\exp(h_{2,0}) + \exp(h_{2,1})} \tag{11}$$

ここで，$\odot$ はベクトルの要素積を表す．$W_1 \in \mathbb{R}^{d_{\text{emb}} \times d_{\text{match}}}$，$W_2 \in \mathbb{R}^{d_{\text{match}} \times 2}$ は線形層の重み行

列，$\boldsymbol{b}_1 \in \mathbb{R}^{d_{\text{match}}}$，$\boldsymbol{b}_2 \in \mathbb{R}^2$ は線形層のバイアスベクトルであり，学習可能なパラメーターである．活性化関数として tanh 関数を，過学習の抑制のため dropout 関数を使用している．

符号器の学習は正例ペアの一致率を 1，負例ペアの一致率を 0 に近づけることで行われる．学習に使用する正例ペア $(x,e)^+$ は Wikipedia の各リンク $(m,e) \in S$ から得られるが，負例ペア $(x,e)^-$ は得られない．入力シーケンス X におけるリンク範囲以外のトークンとエンティティのペア集合 $\{(x,e) \in X \times \{e\} | x \notin m\}$ を負例として扱った場合，e に関する省略されたリンク $(m,e) \in \overline{S}$ により偽陰性ラベル $\{(x,e) \in X \times \{e\} | x \in m, (m,e) \in \overline{S}\}$ が発生してしまう．

本研究では，Wikipedia ガイドラインやエンティティの特性，ハードサンプリングを活用し，表 4 に示すような負例を選択する．Wikipedia ガイドラインの「同一のエンティティを指す言及に関しては基本的に初出の場合のみリンクする」という定義は，「リンクより後方にはリンクと同一のエンティティを指す言及が含まれる可能性があるが，リンクより前方の文章には同様の言及は含まれない」と解釈できる．つまり，初出のリンク $\{(x_o,e)^+, ..., (x_{o+|m|-1},e)^+\}$ より前のトークンとエンティティのペア集合 $\{(x_i,e)^- | i < o\}$ を負例として扱うことができる．例えば，表 4 のペア $(x_2(\text{は}), e_3(\text{和光市}))^-$ などである．

同一のエンティティを指すリンクの隣接はないと考えられるため，リンクの左右には同一エンティティを指す言及は出現しないと言える．そのため，リンクの左右をそれぞれ負例ペア $\{(x_{o-1},e)^-, (x_{o+|m|},e)^-\}$ として扱う．例えば，表 4 のペア $(x_6(\text{和光市}), e_2(\text{埼玉県}))^-$ などである．

学習の効率化のため，検索器と同様に式 5 のハードサンプリングで得られた難しい例 e^- も負例ペアとして扱う．例えば，表 4 のペア $(x_1(\text{理研}), \text{HN}(x_1(\text{理研})))$ などである．

上記の手順で $\mathcal{A}'$ から得られたペア (x,e) の集合を Z' とすると，符号器は，以下の二値交差

	x_1(理研)	x_2(は)	x_3(日本)	x_4(の)	x_5(埼玉県)	x_6(和光市)	x_7(に)
リンク	✓				✓	✓	
初出	✓					✓	
e_1(理化学研究所)	T	F	—	—	—	—	—
e_2(埼玉県)	—	—	—	F	T	F	—
e_3(和光市)	F	F	F	F	F	T	F
HN(x_1(理研))	F	—	—	—	—	—	—
HN(x_5(埼玉県))	—	—	—	—	F	—	—
HN(x_6(和光市))	—	—	—	—	—	F	—

表 4　符号器の学習における負例選択の例．$x_i(\cdot)$ はトークンとその中身，$e_j(\cdot)$ はエンティティとそのタイトルを示している．“リンク” はトークンにリンクが付与されているかどうかを示し，“初出” はそのエンティティに関するリンクが最初の出現であるかを示している．T は正例ペア，F は負例ペアを示す．HN(·) は式 5 と同等であり，言及に対する難しい負例を取得する関数を示す．

エントロピー誤差を最小化することで学習可能である．

$$\mathcal{L}_{\text{match}} = \frac{1}{|Z'|} \sum_{(x,e)\in Z'} -p(e|x)\log(q(e|x)) - (1-p(e|x))\log(1-q(e|x)) \tag{12}$$

ここで，$p(e|x)$ は真の確率分布であり，トークン x とエンティティ e が紐づく場合は 1，それ以外は 0 となる．

4.4 修正器

修正器は，リンクの言及範囲から学習することで，検索器と符号器によって得られたリンクの範囲を修正する．

本研究では，修正器による範囲修正を系列ラベリング問題として解く．修正器は言及の範囲のみから学習できれば良いため，系列ラベルとして BILOU タグから O タグを除いた BILU タグを使用する．タグセットを $\mathcal{T}$，各タグを $t \in \mathcal{T}$ として表す．言及範囲 $\{x|x \in m, m \in M, x \in \mathcal{A}\}$ のみから学習した場合，言及範囲以外 $\{x|x \notin m, m \in M, x \in \mathcal{A}\}$ のモデルの挙動が予測不可能であるが，誤検出は検索器と符号器によって棄却が期待できるため問題はない．

修正器では，トークンベクトル x に対するタグ t の予測確率 $q(t|x)$ は線形層を使用し，以下のように得る．

$$\boldsymbol{h}_3 = \text{dropout}(\tanh(W_3^{\mathsf{T}}\boldsymbol{v}_x + \boldsymbol{b}_3)) \tag{13}$$

$$\boldsymbol{h}_4 = W_4^{\mathsf{T}}\boldsymbol{h}_3 + \boldsymbol{b}_4 \tag{14}$$

$$q(t|x) = \frac{\exp(h_{4,t})}{\sum_{t^*\in\mathcal{T}}\exp(h_{4,t^*})} \tag{15}$$

ここで，$W_3 \in \mathbb{R}^{d_{\text{emb}}\times d_{\text{modif}}}$，$W_4 \in \mathbb{R}^{d_{\text{modif}}\times|\mathcal{T}|}$ は線形層の重み行列，$\boldsymbol{b}_3 \in \mathbb{R}^{d_{\text{modif}}}$，$\boldsymbol{b}_4 \in \mathbb{R}^{|\mathcal{T}|}$ は線形層のバイアスベクトルである．両者とも学習可能なパラメーターである．

バッチ $\mathcal{A}'$ に含まれる言及集合を M'，言及を指すトークン列を $X' = \{x|x \in m, m \in M'\}$，トークン x に付与されたタグを t_x とする．修正器の学習は，以下の交差エントロピー誤差を最小化することで行う．

$$\mathcal{L}_{\text{modify}} = -\frac{1}{|X'|}\sum_{x\in X'}\log(q(t_x|x)) \tag{16}$$

4.5 リンク拡張モデル

リンク拡張モデルでは，検索器，符号器，修正器を同時に学習する．リンク拡張を行う際は，処理の単純化のため修正器，検索器，符号器の順で拡張を行う．

リンク拡張モデルは以下の誤差 $\mathcal{L}$ を最小化することで検索器，符号器，修正器を同時に学習する．

$$\mathcal{L} = \mathcal{L}_{\text{search}} + \mathcal{L}_{\text{ene}} + \mathcal{L}_{\text{match}} + \mathcal{L}_{\text{modify}} \tag{17}$$

リンク拡張は，処理の単純化のため，先に入力シーケンスに対して修正器によりリンクの言及検出を行う．その後，検索器により言及に対してエンティティを紐付けることでリンクを生成し，符号器により有効なリンクのみを残すことで達成される．

修正器と，BILU タグ列を受け取り言及に変換する関数 offset$(\cdot)$ を用いて，入力シーケンス X に対する言及集合 $\hat{M} = \{\hat{m}\}$ を以下のように得る．

$$\hat{M} = \text{offset}(\{\underset{t \in \mathcal{T}}{\text{argmax}}\ q(t|x) | x \in X\}) \tag{18}$$

この時，既に Wikipedia 上で付与されているリンクに重複する言及は削除する．

$$\hat{M} = \{\hat{m} | \hat{m} \cap m = \{\phi\}, \hat{m} \in \hat{M}, m \in M\}$$

検索器 f_{search} と符号器の予測確率 $q(e|x)$ を用いて，予測言及 $\hat{m}$ を受け取り $\hat{e}$ を返す f_{link} を以下のように定義する．

$$\hat{e} = f_{\text{link}}(\hat{m}) \tag{19}$$

$$f_{\text{link}}(\hat{m}) = f_{\text{search}}(\underset{x \in \hat{m}}{\text{argmax}}\ q(f_{\text{search}}(x, E)|x), E) \tag{20}$$

$$q(\hat{e}|\hat{m}) = \max_{x \in m} q(f_{\text{search}}(x, E)|x) \tag{21}$$

複数トークンからなる言及にエンティティを付与する際，エンティティが競合する可能性が考えられる．この場合，最も高い一致率 $q(\hat{e}|\hat{m})$ を持つエンティティを採用する．

最後に修正器による一致率 $q(\hat{e}|\hat{m})$ が 0.5 未満のリンクを棄却することで，最終的な予測リンク集合 $\hat{S}_{(\overline{M} \to E)}$ を以下のように得ることができる．

$$\hat{S}_{(\overline{M} \to E)} = \{(\hat{m}, \hat{e}) | q(\hat{e}|\hat{m}) \geq 0.5, \hat{e} = f_{\text{link}}(\hat{m}), \hat{m} \in \hat{M}\} \tag{22}$$

5　提案手法 2：期待エンティティ率 (EER) 推定と固有表現抽出器学習

本研究では，Wikipedia に対応するエンティティが存在しない NIL 言及に対処するため，Wikipedia における期待エンティティ率を推定する手法を提案する．推定値を既存手法である固有表現抽出の学習制約に適用することで，NIL 言及が固有表現抽出器の学習に与える影響を軽減する．

期待エンティティ率 (Expected Entity Rate; EER) とは，文章中における固有表現の言及を示すトークンの割合であり，不完全なデータセットから固有表現抽出器を学習するため，Effland and Collins (2021) により導入された．しかし，Effland and Collins (2021) は正しいラベルが付与された外部データセットから期待エンティティ率を測定した上で，不完全なデータセットに転用しており，言語やドメイン，固有表現クラスの定義が変わった場合には適用できないといった問題がある．

本研究では既に付与されている Wikipedia のリンクから，Wikipedia 全体の期待エンティティ率を推定する手法を提案する．得られた推定値を Effland and Collins (2021) の手法に適用することで NIL 言及によるラベル欠落の影響軽減を試みる．

以下では，5.1 節で本節で用いる記号を定義したのち，5.2 節で期待エンティティ率の推定手法，5.3 節で推定された期待エンティティ率を用いた固有表現抽出器の学習制約手法と深層学習モデル構造について説明する．

5.1　記号の定義

特に記述のない場合，使用する記号は 4.1 節から引き継ぐ．

初めに，観測可能なリンク $S \cup \hat{S}_{(\overline{M} \to E)}$ をエンティティごとに集計し行列 G を得る．この時，行列の要素 $G_{i,j}$ はエンティティ e_i を説明する記事 $a_i \in \mathcal{A}$ から，エンティティ e_j へのリンク数を示している．ここで，エンティティ e_j を受け取り，e_j が他の記事から参照された回数である被リンク数を返す関数 f_{in} は以下の様に定義される．

$$f_{\text{in}}(e_j) = \sum_{i \neq j}^{|\mathcal{A}|} G_{i,j} \tag{23}$$

この際，参照元と参照先が同じエンティティである自己リンク数 $G_{i,i}$ は特異であるため除き，エンティティ e_i の自己リンク数を求める関数 f_{self} として以下のように別途定義しておく．

$$f_{\text{self}}(e_i) = G_{i,i} \tag{24}$$

被リンク数が k であるエンティティ数を返す関数 $f_{\text{count}}(k)$ を以下の様に定義する．

$$f_{\text{count}}(k) = \sum_{e \in E} \mathbb{1}(f_{\text{in}}(e) = k) \tag{25}$$

5.2　期待エンティティ率推定

期待エンティティ率 ρ は，オラクルリンク集合のトークン数（オラクルトークン数）を集計し，Wikipedia の総トークン数で割ることで導出できる．

$$\rho = \frac{\sum_{(m,e) \in U} |m|}{\sum_{a \in \mathcal{A}} |a|} \tag{26}$$

提案手法では，いくつかの仮定のもと，被リンク数ごとのエンティティ頻度の関係を関数近似することで，オラクルトークン数を推定する．推定したオラクルトークン数を用いることで，期待エンティティ率が導出可能となる．

オラクルトークン数の推定のため，以下の仮定を置く．

仮定 1　被リンク数が多いエンティティは Wikipedia ドメイン内で有名であると考え，エンティティ e の被リンク数 $f_{\text{in}}(e)$ を e の有名度として扱うことができるとする．

仮定 2　Wikipedia では一定以上に有名なエンティティは記事として追加されていると考え，以下の式が成り立つとする．

$$\forall e \in E, f_{\text{in}}(e) \geq \tilde{k} \tag{27}$$

有名度がある値 $\tilde{k}$ 以上のエンティティであれば，対応する記事が必ず Wikipedia に存在することを意味する．言い換えると，有名度が $\tilde{k}$ 以上の区間に NIL エンティティ $e \in \overline{E}$ が存在しないことを意味する．

仮定 3　被リンク数 k のエンティティに紐づくリンクの平均言及長は，k のみに依存するとする．つまり，観測可能なリンク $S \cup \hat{S}_{(\overline{M} \to E)}$ から，観測不可能なリンク $\hat{S}_{(\overline{M} \to \overline{E})}$ における k ごとの平均言及長を予測できるとする．ある被リンク数 k を受け取り，平均言及長を返す関数 $f_{\text{m\_ave}}(k)$ を以下のように定義する．

$$f_{\text{m\_ave}}(k) = \frac{\sum_{(m,e) \in S \cup \hat{S}_{(\overline{M} \to E)}} [\mathbb{1}(f_{\text{in}}(e) = k) \cdot |m|]}{k \cdot f_{\text{count}}(k)} \tag{28}$$

仮定のもと，被リンク数 k のエンティティに紐づくオラクル言及トークン数は以下のように計算できる．

$$\sum_{f_{\text{in}}(e)=k, (m,e) \in U} |m| = k \cdot f_{\text{count}}(k) \cdot f_{\text{m\_ave}}(k) \tag{29}$$

だが，オラクルリンク集合 U は NIL エンティティに関するリンクを含むため，一部が観測不可能である．しかし，仮定により，被リンク数 $\tilde{k}$ 以上の区間には NIL エンティティを指すリンクは含まれないことから，$\tilde{k}$ 以上の区間では式 29 によりオラクル言及トークン数を計算可能である．被リンク数 $\tilde{k}$ 未満の区間を扱うため，NIL エンティティ集合 $\overline{E}$ を使用し，被リンク数が k のエンティティ頻度を返す関数 $\dot{f}_{\text{count}}(k)$ を以下のように定義する．

$$\dot{f}_{\text{count}}(k) = \sum_{e \in E \cup \overline{E}} \mathbb{1}(f_{\text{in}}(e) = k) \tag{30}$$

関数 $\dot{f}_{\text{count}}(k)$ を用いることで，被リンク数 $\tilde{k}$ 未満の区間における被リンク数 k のオラクル言及トークン数を以下のように計算できる．

$$\sum_{f_{\text{in}}(e)=k,(m,e)\in U} |m| = k \cdot \dot{f}_{\text{count}}(k) \cdot f_{\text{m\_ave}}(k) \tag{31}$$

最終的な Wikipedia 全体のオラクルトークン数は，自己リンクのトークン数も考慮した上で，以下のように計算できる．

$$\sum_{(m,e)\in U} |m| = \sum_{k=\tilde{k}}^{\max(k)} (k \cdot f_{\text{count}}(k) \cdot f_{\text{m\_ave}}(k)) + \sum_{k=1}^{\tilde{k}-1} (k \cdot \dot{f}_{\text{count}}(k) \cdot f_{\text{m\_ave}}(k)) + \sum_{e\in E} f_{\text{self}}(e) \tag{32}$$

本研究では，$\tilde{k}$ 以上の区間での関数 f_{count} の振る舞いを何らかの関数 $\hat{f}_{\text{count}}$ で近似することで，$\tilde{k}$ 未満での関数 $\dot{f}_{\text{count}}$ の振る舞いを予測し，オラクルトークン数を推定する．関数 f_{count} は，被リンク数とその頻度の関係を表しているが，Wikipedia では，この関係が Zipf の法則に従うことが知られている (Voß 2005)．Voß (2005) は，各記事が持つリンク，無効なリンクに関しても Zipf 則に従うことを示している．Zipf の法則とは，単語の出現頻度とその順位の積が定数になることを示した経験則であり，言語にとどまらない様々な頻度関係が Zipf の法則に従うことが知られている．Zipf の法則を一般化した累乗関数に，k と k_{count} の関係を適用すると以下のようになる．

$$\hat{f}_{\text{count}}(k) = \alpha \cdot k^{-\beta} \tag{33}$$

非リンク数が 30 以上のエンティティは全て Wikipedia に追加されているとし，$\tilde{k} \geq 30$ の区間を調べたところ，リンク拡張を施した Wikipedia においても被リンク数 k と頻度 $f_{\text{count}}(k)$ の関係が Zipf の法則に当てはまることが確認された．

実際に，$k = [30, 1000]$ の区間から非線形回帰を行ったところ，$\alpha = 3516323.8, \beta = 1.80$ となった．実測値と近似した関数 $\hat{f}_{\text{count}}$ のグラフを図 3 に示す．

Zipf の法則に基づき近似した関数 $\hat{f}_{\text{count}}$ を使用して，式 32 により期待エンティティ率を計算したところ，$\hat{\rho} = 0.238$ であった．本研究ではこの値を使用する．元々の Wikipedia のリンクにおけるエンティティ率と，リンク拡張後のエンティティ率を調べたところ，それぞれ $\hat{\rho}_{\text{orig}} = 0.101$，$\hat{\rho}_{\text{ext}} = 0.209$ であった．

5.3　固有表現抽出器学習

本研究では，NIL 言及の影響を軽減するため，期待エンティティ率による学習制約を適用する．学習制約には先行研究 (Efland and Collins 2021) の手法を用いる．しかし，先行研究と本研究では対象とする問題設定が異なるため，固有表現抽出モデルと損失関数に対して修正を加

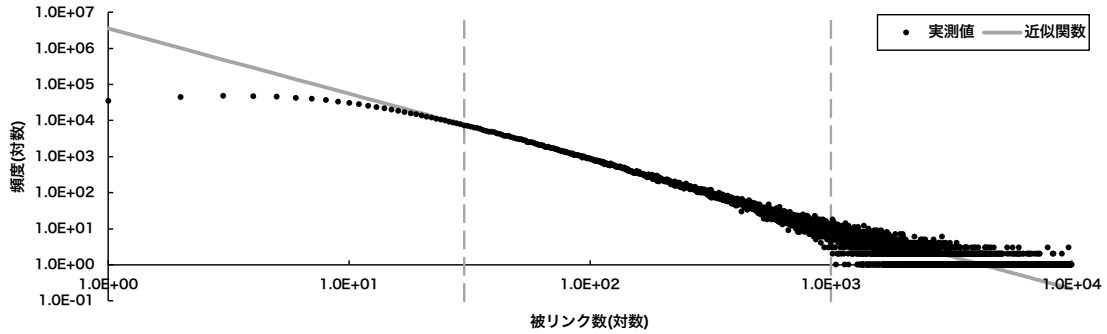

図 3　被リンク数ごとのエンティティ頻度の関係と Zipf の法則に基づく近似関数：被リンク数は 10,000 まで描画しており，点線は近似に使用した被リンク数の区間 $[30, 1000]$ を示す．

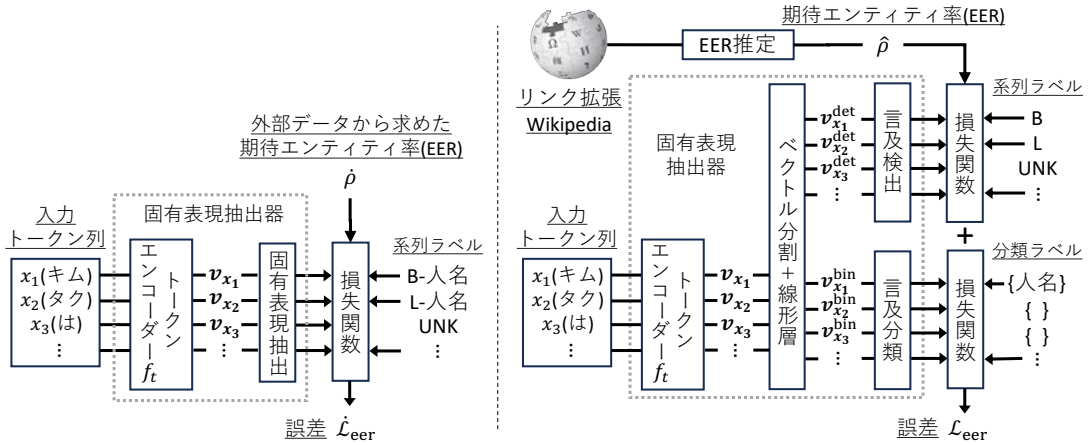

図 4　固有表現抽出モデルと損失計算工程:左は先行研究 (Effland and Collins 2021)，右は本研究を示す．

える．具体的には，先行研究は言及に対して単一の固有表現クラスが付与されることを想定しているが，本研究は複数の固有表現クラスが付与される場合があるため直接適用できない．そのため，本研究では固有表現抽出を言及の検出と分類に分けて考えることで解決している．先行研究と本研究における固有表現抽出モデルと損失計算の概要を図 4 に示す．以下では，初めに先行研究をタスク設計，モデル構造，制約手法に分けて解説した後，問題設定の違いについて述べ，本研究のモデル構造と損失関数を解説する．

先行研究では固有表現抽出を入力トークン列 X に対して BILOU 形式のタグ列 Y を予測するタスクとして扱っている．Effland and Collins (2021) は，不完全ラベルから学習するため，BILOU 形式のタグ集合 $\mathcal{Y} = (\{\mathtt{B}, \mathtt{I}, \mathtt{L}, \mathtt{U}\} \times \mathcal{C}) \cup \{\mathtt{O}\}$ に，ラベルが未知であることを示すタグ $\mathtt{UNK}$ を加えたタグ集合 $\mathcal{Y}_{\mathcal{O}} = \{\mathtt{UNK}\} \cup \mathcal{Y}$ を用いている．ここで，$\mathcal{C}$ は対象としている固有表現クラス

の集合である．$\mathcal{Y}_\mathcal{O}$ からなるタグ列を $Y_\mathcal{O}$ と表す．

先行研究ではモデルに BERT (Devlin et al. 2019) と CRF (Lafferty et al. 2001) を用いている．入力トークン列

$$X = \{[\texttt{CLS}], x_1, ..., x_{l_\mathrm{n}}, [\texttt{SEP}]\}$$

を受け取り，BERT により得られた各トークンごとの埋め込みベクトル列 $V_X = \{\boldsymbol{v}_x\} \in \mathbb{R}^{l_\mathrm{n} \times d_\mathrm{emb}}$ を出力する関数を f_n とする．

$$V_X = \mathrm{red}(f_\mathrm{n}(X)) \tag{34}$$

ここで，l_n は入力シーケンス長，d_emb は埋め込み次元である．また，$\mathrm{red}(\cdot)$ は f_n の出力ベクトル列から特殊トークン [CLS] と [SEP] に対するベクトルを除外する関数である．CRF は学習可能な重み $W = \{\boldsymbol{w}_y\} \in \mathbb{R}^{|\mathcal{Y}| \times d_\mathrm{emb}}$ と遷移スコア $T \in \mathbb{R}^{|\mathcal{Y}| \times |\mathcal{Y}|}$ により構成され，以下の様に入力系列 X に対するタグ列 Y の予測確率 $q(Y|X)$ を出力する．

$$q(Y|X) = \frac{\exp\left(\sum_{i=1}^{l_\mathrm{n}-1} \phi(i, y_i, y_{i+1}) + \phi(l_\mathrm{n}, y_{l_\mathrm{n}})\right)}{Z(\phi)} \tag{35}$$

$$Z(\phi) = \sum_{Y' \in \mathcal{Y}} \exp\left(\sum_{i=1}^{l_\mathrm{n}-1} \phi(i, y'_i, y'_{i+1}) + \phi(l_\mathrm{n}, y'_{l_\mathrm{n}})\right) \tag{36}$$

$$\phi(i, y, y') = \phi(i, y) + T_{y,y'} \tag{37}$$

$$\phi(i, y) = \boldsymbol{w}_y^\top \boldsymbol{v}_i \tag{38}$$

ここで $Y' \in \mathcal{Y}$ は，タグ集合 $\mathcal{Y}$ の順列により表現される全てのタグ遷移からタグ列 Y' を順に取り出す操作を示す．

先行研究における学習制約は損失関数により行われる．一般に CRF では，トークン列 X と固有表現タグ列 Y のペアからなるバッチ $\mathcal{B} = \{(X, Y)\}$ に対して，以下の損失関数により得られる誤差 $\mathcal{L}_\mathrm{p}$ を最小化することで学習を行う．

$$\mathcal{L}_\mathrm{p} = -\frac{1}{|\mathcal{B}|} \sum_{(X,Y) \in \mathcal{B}} \log\, q(Y|X) \tag{39}$$

先行研究では，バッチ $\mathcal{B} = \{(X, Y_\mathcal{O})\}$ に対して誤差 $\dot{\mathcal{L}}_\mathrm{eer} = \dot{\mathcal{L}}_\mathrm{p} + \dot{\mathcal{L}}_\mathrm{u}$ を最小化することで学習制約を行う．$\dot{\mathcal{L}}_\mathrm{p}$ はラベルが付与された範囲のみから学習を行う誤差で $\mathcal{L}_\mathrm{p}$ を元に以下の損失関数により計算される．

$$\dot{\mathcal{L}}_\mathrm{p} = -\frac{1}{|\mathcal{B}|} \sum_{d} (X, Y_\mathcal{O}) \in \mathcal{B} \log\, q(Y_\mathcal{O}|X) \tag{40}$$

$$\log q(Y_{\mathcal{O}}|X) = \log \sum_{Y' \models Y_{\mathcal{O}}} q(Y'|X) \tag{41}$$

ここで $Y' \models Y_{\mathcal{O}}$ は，タグ列 $Y_{\mathcal{O}}$ においてあり得る全てのタグ遷移からタグ列 Y' を順に取り出す操作を示す．具体的には，実際のタグが未知であるタグ UNK は $y \in \mathcal{Y}$ のいずれでもあり得るとし，その全ての遷移を考慮する．$\dot{\mathcal{L}}_{\mathrm{u}}$ は期待エンティティ率による制約を行う誤差で，以下の損失関数により計算される．

$$\dot{\mathcal{L}}_{\mathrm{u}} = \max(0, \mathrm{abs}(\hat{\rho} - \rho) - \gamma) \tag{42}$$

$$\rho = -\frac{\sum_{(X,Y_{\mathcal{O}})\in B} \sum_{x \in X} \sum_{y \in \mathcal{Y}\setminus\{\mathrm{O}\}} q(y|x)}{\sum_{(X,Y_{\mathcal{O}})\in B} |X|} \tag{43}$$

$\hat{\rho}$ は 5.2 節で求められた期待エンティティ率，ρ はモデルの予測エンティティ率を示し，γ は，期待エンティティ率と本来のエンティティ率との差異を許容するマージンを示す．

一般的な固有表現抽出タスクでは固有表現の言及に対して単一の固有表現クラスが割り当てられるため，CRF で扱うことは容易である．しかし，拡張固有表現階層では，言及に対して複数の固有表現クラスが割り当てられる場合があり，出力として単一のタグ列を想定する CRF で直接扱うことは難しい．そのため，本研究では図 4 に示すように固有表現の言及検出と検出された言及の分類を分けて考える．言及検出は，入力トークン列 X に対して BILOU 形式のタグ列 Y を予測するタスクとして解く．タグ集合は $\mathcal{Y} = \{\mathtt{B}, \mathtt{I}, \mathtt{L}, \mathtt{O}, \mathtt{U}\}$ と，UNK を追加した $\mathcal{Y}_{\mathcal{O}} = \{\mathtt{UNK}\} \cup \mathcal{Y}$ である．この時，“名前” 以下のカテゴリーが付与されていない，例えば “コンセプト” のみ付与された記事等へのリンクは UNK ではなく，O タグとして扱う．言及分類は言及区間の先頭トークンを各カテゴリーに分類する多ラベル分類タスクとして解く．

言及検出と多ラベル分類を独立して解くために，トークンベクトル $\boldsymbol{v}_x \in \mathbb{R}^{d_{\mathrm{emb}}}$ を $d_{\mathrm{emb}}/2$ 次元ずつに分割する．その後，線形層により再度 d_{emb} へと拡張し，言及検出用ベクトル $\boldsymbol{v}_x^{\mathrm{det}} \in \mathbb{R}^{d_{\mathrm{emb}}}$ と，多ラベル分類用ベクトル $\boldsymbol{v}_x^{\mathrm{bin}} \in \mathbb{R}^{d_{\mathrm{emb}}}$ を得る．

$$\boldsymbol{v}_x^{\mathrm{det}} = \mathrm{dropout}(\tanh(W_5^{\mathsf{T}} \boldsymbol{v}_{x[:d_{\mathrm{emb}}/2]} + \boldsymbol{b}_5)) \tag{44}$$

$$\boldsymbol{v}_x^{\mathrm{bin}} = \mathrm{dropout}(\tanh(W_6^{\mathsf{T}} \boldsymbol{v}_{x[d_{\mathrm{emb}}/2:]} + \boldsymbol{b}_6)) \tag{45}$$

ここで，$W_5 \in \mathbb{R}^{d_{\mathrm{emb}/2} \times d_{\mathrm{emb}}}$，$W_6 \in \mathbb{R}^{d_{\mathrm{emb}/2} \times d_{\mathrm{emb}}}$ は線形層の重み行列，$\boldsymbol{b}_5 \in \mathbb{R}^{d_{\mathrm{emb}}}$，$\boldsymbol{b}_6 \in \mathbb{R}^{d_{\mathrm{emb}}}$ は線形層のバイアスベクトルであり，学習可能なパラメーターである．言及検出は $\boldsymbol{v}_x$ の代わりに $\boldsymbol{v}_x^{\mathrm{det}}$ を入力とし，前述の CRF を適用する．

言及分類では，各トークン x があるカテゴリー c に属する確率 $q(c|x)$ とカテゴリー集合 C_x に属する確率 $q(C_x|x)$ はトークンベクトル $\boldsymbol{v}_x^{\mathrm{bin}}$ を使用して次のように計算する．

$$\boldsymbol{h}_7 = W_7^{\mathsf{T}} \boldsymbol{v}_x^{\mathrm{bin}} + \boldsymbol{b}_7 \tag{46}$$

$$q(c|x) = \frac{1}{1 + e^{-h_{7,c}}} \tag{47}$$

$$q(C_x|x) = \frac{\sum_{c \in C_x} \exp(h_{7,c})}{\sum_{c' \in C} \exp(h_{7,c'})} \tag{48}$$

ここで，$W_7 \in \mathbb{R}^{d_{\text{emb}} \times |C|}$ は線形層の重み行列，$\boldsymbol{b}_7 \in \mathbb{R}^{|C|}$ は線形層のバイアスベクトルであり，学習可能なパラメーターである．バッチ $\mathcal{B}$ に含まれる言及集合を M'，言及の先頭トークンのみで構成されるトークン列を $X'' = \{x | x = m_0, m \in M'\}$ とし，各トークン x に付与されるカテゴリー集合を C_x とする．クラス分類では以下の誤差 $\mathcal{L}_{\text{b}}$ と $\mathcal{L}_{\text{m}}$ を対象とする．

$$\mathcal{L}_{\text{b}} = -\frac{1}{|X''|} \sum_{x \in X''} w_c \cdot \{p(c|x) \cdot \log\, q(c|x) + (1 - p(c|x)) \cdot \log(1 - q(c|x))\} \tag{49}$$

$$p(c|x) = \mathbb{1}(c \in C_x) \tag{50}$$

$$\mathcal{L}_{\text{m}} = -\frac{1}{|X''|} \sum_{x \in X''} \log(q(C_x|x)) \tag{51}$$

ここで，w_c は各カテゴリー c ごとの重みであり，ロングテールなカテゴリー分布に対処するために導入される．w_c は学習データ全体におけるカテゴリー c の出現数を求める関数 $\text{count}(c)$ を用いて以下の様に計算される．

$$w_c = \frac{\max\limits_{c' \in C} \text{count}(c')}{\text{count}(c)} \tag{52}$$

固有表現クラスに拡張固有表現階層を用いる場合，各言及 m の先頭トークン x に対して 1 つ以上のカテゴリーが必ず付与される．最終的な予測カテゴリー $\hat{C}_x$ は次のように得る．

$$\hat{C}_x = \begin{cases} \{\underset{c \in C}{\text{argmax}}\ p(c|x)\} & (\text{bin}(x) = \{\phi\}) \\ \text{bin}(x) & (\text{bin}(x) \neq \{\phi\}) \end{cases} \tag{53}$$

$$\text{bin}(x) = \{c | p(c|x) \geq 0.5, c \in C\} \tag{54}$$

最終的なモデルの学習は以下の誤差 $\mathcal{L}_{\text{eer}}$ の最小化により行われる．

$$\mathcal{L}_{\text{eer}} = \lambda_{\text{p}} \dot{\mathcal{L}}_{\text{p}} + \lambda_{\text{u}} \dot{\mathcal{L}}_{\text{u}} + \mathcal{L}_{\text{b}} + \mathcal{L}_{\text{m}} \tag{55}$$

λ_{u} は先行研究でも使用される期待エンティティ率の制約を調節する重みである．トークンレベルの確率の負の対数尤度 $\mathcal{L}_{\text{b}}, \mathcal{L}_{\text{m}}$ と比較して，トークン列レベルの確率の負の対数尤度 $\mathcal{L}_{\text{p}}, \dot{\mathcal{L}}_{\text{p}}$ は非常に大きな値を取る．これらのバランスの調節に，本研究では重み $\lambda_{\text{p}} < 1$ も導入する．

6　実験

本節では，実際に日本語 Wikipedia から固有表現抽出器を学習し評価するための，様々な設定について説明する．以下では，6.1 節で評価用データセットの構築について，6.2 節で本実験で用いる深層学習モデルについて，続く 6.3 節と 6.4 節でそれぞれリンク拡張手法，固有表現抽出器学習手法の実験設定について記述する．

6.1　評価データセット

日本語拡張固有表現抽出器の評価を行うべく，評価セットを構築した．拡張固有表現階層を固有表現クラスとして採用した評価セットとして拡張固有表現タグ付きコーパス (橋本 他 2008) が存在するが，採用している階層定義が古く，本研究で使用する最新の定義と互換性が低いため，直接転用することは難しい．本研究では，ウェブ上で自由に編集と閲覧が可能なニュースサイトである日本語版 Wikinews の記事に対して人手でラベルを付与することで評価セットを構築した．Wikinews はライセンスに CC BY 2.5 を採用しており，有償の文章を対象とする前述のコーパスとは異なり，無償で配布可能であるといった利点がある．利点を活かすべく，本研究では評価データを無償公開している．

Wikinews では各記事に対して階層構造を持つカテゴリーが付与されている．表 5 に日本語 Wikinews のトップカテゴリーとサブカテゴリーの一覧を示す．評価セットには，幅広い文章が含まれていることが望ましいので，各サブカテゴリーに属する記事から 2 件ずつサンプリングした．Wikinews ではトップカテゴリーを跨いで複数のサブカテゴリーを付与することが許可さ

トップカテゴリー	サブカテゴリー
ひと	人事，人権，出産と誕生，結婚，訃報
スポーツ	アマチュアレスリング，アメリカンフットボール，オリンピック，ゴルフ，サッカー，スキー，スケート，スノーボード，ソフトボール，テニス，ハンドボール，バイアスロン，バスケットボール，バドミントン，バレーボール，フィギュアスケート，プロレス，ホッケー，ボクシング，モータースポーツ，ラグビー，体力，体操，公営競技，卓球，囲碁，将棋，柔道，格闘技，水泳，競馬，野球，陸上競技，馬術
学術	コンピュータ，ノーベル賞，人文学，医学，宇宙，歴史，社会科学，科学技術，考古学，自然科学
政治	各国の政治，国際政治，地方自治，核兵器，法律，紛争，選挙
文化	アニメ，メディア，宗教，文学，映画，演劇，漢字，漫画，美術，芸能，音楽
気象	地震，季節，梅雨，火山，熱帯低気圧，秋雨，竜巻，雪
社会	事件，事故，交通，健康，募金，団体，娯楽，情報セキュリティ，教育，新聞，火災，災害，環境，皇室，統計，行事，裁判，観光
経済	企業，倒産，労働，原油，商品，株式，為替，産業，財政，金融

表 5　Wikinews カテゴリー一覧

れており，通常のサンプリングを行った場合，例えば訃報といった記事数の多いサブカテゴリーに偏る可能性がある．従って，本研究ではサンプリングの際に単一のサブカテゴリーに属する記事を優先的に選択した．

ラベル付けは，拡張固有表現階層の定義書を参考に行った．しかし，定義書は主に Wikipedia 記事を固有表現カテゴリーに分類するために定義されており，固有表現抽出に対しては十分ではない．それゆえ，以下の新たな定義を導入した．

(1) 言及範囲が曖昧な場合全体が固有名として認められるかどうかを考慮し決定する．例えば「東京五輪」は全体が固有名として認められるため，まとめて “競技会名” が付与されるが，対象的に「日本代表」は全体が固有名として認められず「日本」のみに “国名” が付与される．

(2) 固有表現に対して複数のカテゴリーが考えられる場合は最も文脈に合致するカテゴリーのみを選択する．優劣が付けられない場合のみ，複数のカテゴリー付与が認められる．

(3) 文脈から判断されるカテゴリーと固有名が持つカテゴリーの両方が考えられる場合は固有名として新しく定義されているかを考慮し決定する．例えば，野球における日本代表が「侍ジャパン」として表記されている場合は新規に固有名として定義されているため “競技団体名” が付与されるが，「日本」として表記されている場合は新規に固有名として定義されているとはいえず，元々の固有名が持つ “国名” が付与される．

ラベル付けは，拡張固有表現階層を熟知している必要がある．従って，3.2.2 節で説明した拡張固有表現階層の分類データ作成に携わっているメンバーにラベル付けを依頼した．作業は，各記事に対して 2 人割り当て，片方が記事全体に対してラベル付けを行った後，片方が確認を行う形で進めた．ラベル付けを行った結果，198 件の記事に対して 5,984 個の固有表現ラベルが付与された．

6.2 深層学習モデル

本研究では全ての実験で同一の BERT-base と CRF を使用する．base サイズでは，全ての埋め込み次元は $d_{\mathrm{emb}} = 756$ となる．BERT は，RoBERTa (Liu et al. 2020) と同様の手法で，日本語 Wikipedia を使用して事前学習を行う．トークナイザーには MeCab の Python ラッパーである janome と BPE を使用する．BPE は日本語 Wikipedia からマージ数 10,000 で学習する．BERT の語彙数は 24,000 とする．学習時のバッチサイズは 1,024 とし，最適化関数には Adam を使用する．学習率は 4.0×10^{-4}，その他のパラメーターは $\epsilon = 1.0 \times 10^{-6}$, $\beta_1 = 0.9$, $\beta_2 = 0.98$ を使用し，パラメーターの更新回数は 500,000 回とする．学習は，fairseq[11] を使用して行う．事前学習済み BERT は別途公開[12] している．

[11] https://github.com/facebookresearch/fairseq
[12] https://github.com/k141303/liat_ml_roberta

6.3 リンク拡張手法の実験設定

本研究では深層学習によるリンク拡張を用いて固有表現抽出データセットの構築を行う．提案手法の比較対象として，リンク拡張を使用しないベースラインと表層マッチによるリンク拡張を実装する．以下に，表層マッチと提案手法の実験設定について明記する．

6.3.1 表層マッチ

省略されたリンクを表層マッチにより補完することで，固有表現抽出データセットを作成する．表層マッチを行うために，3.1.2 節で説明した Wikipedia のタイトルとリダイレクト辞書，エイリアス辞書を使用する．この際，エイリアスはノイズを含む場合があるため，2 回以上使用されたエイリアスのみエイリアス辞書に追加する．日本語では人名が苗字もしくは名前のみといったように省略表記される場合がある．そのため，次のようにエイリアス辞書の拡張を試みる．Wikipedia では人名記事の場合，「木村 拓哉」のように記事の先頭に姓名が空白で分割され表記される．これらをエイリアス辞書に追加することで拡張を行い省略表記に対処する．

リンク省略は多くの場合，既にリンクしたエンティティに関するリンクの省略により発生する．誤ったマッチによるノイズを軽減するため，リンク拡張の対象は文章中で既にリンクされているエンティティのみとし，リンク以後の文脈に対してマッチを行う．この場合，「日本」などの周知のエンティティを指す言及のリンク省略に対処できないため，Strobl ら (Strobl et al. 2020) を参考に，被リンク数の多い記事上位 1,000 件のタイトル，リダイレクト，エイリアスを特例として全ての記事でマッチ対象とする．優先順位は，(1) 記事内リンクのタイトルとリダイレクト，(2) 記事内リンクのエイリアス，(3) 上位記事のタイトルとリダイレクト，(4) 上位記事のエイリアスとする．表層マッチによる候補の衝突は，優先順位が高い候補を採用し，同順位の場合は最長一致の候補を採用する．

6.3.2 提案手法

リンク拡張に使用するトークンエンコーダー f_{t} とエンティティエンコーダー f_{e} にはそれぞれ個別の BERT を用いる．トークンエンコーダーの入力長は $l_{\mathrm{t}} = 256$，エンティティエンコーダーの入力長は $l_{\mathrm{e}} = 128$ とする．符号器と修正器で用いる線形層の次元にはそれぞれ $d_{\mathrm{match}} = d_{\mathrm{emb}} \times 2$，$d_{\mathrm{modif}} = d_{\mathrm{emb}}$ を用いる．学習時のバッチサイズは 160 とし，コサイン類似度のスケール変数は $\alpha = 32$ とする．最適化関数には Adam を使用し，学習率は 5.0×10^{-5}，その他のパラメーターは $\epsilon = 1.0 \times 10^{-8}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ を使用する．パラメーターの更新は学習データ 3 周分行う．ハードサンプリングを行うためには全エンティティの埋め込みが必要であるが，各ステップにおいて全エンティティのエンコードを行うことは計算コストの点から現実的ではない．そのため基本的にはステップごとのエンコードを行わず，エンティティ埋め込みには事前に計算したキャッシュが用いられる．本研究ではキャッシュの更新頻度は 500 ステップとする．また，

学習初期はハードサンプリングを用いる必要がないため，4,000 ステップ以降に開始する．

6.4 固有表現抽出器学習手法の実験設定

本研究では NIL 言及によりラベルが欠落したデータセットから学習するため，期待エンティティ率の推定を行いその推定値を用いてモデルを学習する．比較手法として，通常の学習を行うベースラインと自己学習，ノイズ分離の実装を行う．モデルは BERT-base (Devlin et al. 2019) と CRF (Lafferty et al. 2001) を使用する．特に明示しない場合モデルの入力長は $l_{\mathrm{n}} = 512$ とする．学習に期待エンティティ率を使用しない場合，モデルの学習は誤差 $\mathcal{L}_{\mathrm{crf}}$ を最小化することで行われる．

$$\mathcal{L}_{\mathrm{crf}} = \lambda_{\mathrm{p}}\mathcal{L}_{\mathrm{p}} + \mathcal{L}_{\mathrm{b}} + \mathcal{L}_{\mathrm{m}} \tag{56}$$

全ての実験で，$\lambda_{\mathrm{p}} = 0.1$ を用いる．以下では，自己学習，ノイズ分離の解説を行ったのち，提案手法の実験設定について説明する．

6.4.1 自己学習

自己学習は，学習済みのモデルの予測を活用して再度モデルを学習する手法である．研究では，Tedeschi ら (Tedeschi et al. 2021; Tedeschi and Navigli 2022) の手法を参考に実装する．この手法では，モデルの予測を用いて繰り返しデータセットを修正することでラベルの欠落に対処する．手法の流れをアルゴリズム 1 に示す．2 行目と 7 行目はデータセットから重複なくランダムにサブセットを抽出する操作を示す．3 行目，8 行目 train(·) は，サブセットを受け取り学習済みモデルを返す関数である．7 行目 modify(·) は，学習済みモデルとサブセットを受け取り，ラベル補完とフィルタリングを適用した修正済みサブセットを返す関数である．ラベル補完は，元のデータにない固有表現がモデルの予測に含まれた場合に適用される．その場合はラベルの欠落があったとして，データにラベルを追加する．フィルタリングは，元データに存在する固有表現がモデルの予測に含まれなかった場合に適用される．その場合は，元データが誤っているとして，特定の範囲を修正後のサブセットから除外する．先行研究ではフィルタリングを文章中の文単位に適用しているが，計算コストが膨大になることから，本研究ではモデルの入力長を 128 トークンとし，入力全体をフィルタリング範囲とする．本実験では，全体の学習回数を 5 回とし，$n/5$ 件のデータが各ステップでサンプリングされる．モデルはベースラインと同様に事前学習済み BERT と CRF を用いる．最適化関数には Adam を使用し，学習率は 3.0×10^{-5}，その他のパラメーターは $\epsilon = 1.0 \times 10^{-6}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ を使用する．バッチサイズは 800 とし，各ステップの学習は 10 エポック行う．この際，開発データでの評価値が最良のモデルを採用する．

アルゴリズム 1 自己学習

入力: 固有表現抽出データセット $\mathcal{W}$，学習回数 t
出力: 学習済みモデル θ

$n \leftarrow |\mathcal{W}|/t$
$\mathcal{S} \leftarrow \{w_1, ..., w_n | w_i \in \mathcal{W}\}$
$\theta \leftarrow \text{train}(\mathcal{S})$
for $i \leftarrow 2, ..., t$ **do**
　$\dot{\mathcal{S}} \leftarrow \{w'_1, ..., w'_n | w'_i \in W, w'_i \notin \mathcal{S}\}$
　$\mathcal{S} = \mathcal{S} \cup \dot{\mathcal{S}}$
　$\mathcal{S}_\theta = \text{modify}(\mathcal{S}; \theta)$
　$\theta = \text{train}(\mathcal{S}_\theta)$
end for
return θ

6.4.2 ノイズ分離

ノイズ分離は，データセットにおけるノイズや欠落の少ない信頼できるサブセットを推定し，モデルの学習を行う手法である．本研究では，ラベル精度と文章中のラベルカバー率を用いてサブセットを推定する Cao et al. (2019) の手法を参考に実装する．本手法は Wikipedia のカテゴリー情報を用いて各記事を半自動的に固有表現クラスに分類しており，その過程で多くの分類誤りが含まれる可能性がある．そのため，結果としてデータセットに誤った固有表現ラベルが含まれることとなる．そのため，彼らはラベルの精度として，言及中のトークン x がある固有表現クラス y に属する確率 $p(y|x)$ を導入している．しかし，本研究では多くの記事は人手により分類されており分類誤りに起因するラベルノイズが少ないことからラベル精度は用いず，ラベルカバー率のみを用いてサブセットを推定する．ラベルカバー率 ξ は，ラベルが多く付与されているデータはラベル欠落が少なく信頼性が高いと考えられるため導入されており，ラベルが未知であることを示す UNK タグを含むラベル列 $Y_\mathcal{O}$ から以下のように求められる．

$$\xi = \frac{\sum_{y \in Y_\mathcal{O}} \mathbb{1}(y \neq \texttt{UNK})}{|Y_\mathcal{O}|} \tag{57}$$

Cao らは 2 段階の学習を採用しており，初めに信頼性の低いサブセットを用いてラベルが付与された範囲のみからモデルを事前学習した後，信頼性の高いサブセットでモデルを再学習している．本研究では ξ をデータ信頼性として考え，信頼性の低い 9 割のデータを事前学習に，残りの 1 割のデータを通常学習に使用する．事前学習では，式 55 から $\dot{\mathcal{L}}_\text{u}$ の項を除いた誤差を最小化する．最適化関数には Adam を使用し，学習率は 3.0×10^{-5}，その他のパラメーターは $\epsilon = 1.0 \times 10^{-6}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ を使用する．バッチサイズは 400 とし，各学習は 10 エポック行う．この際，開発データでの評価値が最良のモデルを採用する．

6.4.3　提案手法

期待エンティティ率による制約を調節する重みは，先行研究 (Effland and Collins 2021) において経験的に検証された値である $\lambda_u = 10$ を用いる．推定された期待エンティティ率に対して，より厳密な制約をかけるため，本来のエンティティ率との差を許容するためのマージン用変数 γ は用いない．つまり，$\gamma = 0.0$ とする．最適化関数には Adam を使用し，学習率は 3.0×10^{-5}，その他のパラメーターは $\epsilon = 1.0 \times 10^{-6}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ を使用する．バッチサイズは 400 とし，学習は 10 エポック行う．この際，開発データでの評価値が最良のモデルを採用する．

7　結果

本節では，提案手法や比較手法の評価とその考察を行う．以下では，7.1 節で評価データ全体に対する評価と考察を行ったのち，更なる比較のため 7.2 節と 7.3 節では拡張固有表現の階層ごととカテゴリーごと，7.4 節では評価データのドメインごとに評価し考察を行う．

7.1　全体評価

リンク拡張手法と学習手法はそれぞれ独立して使用することができるため，全ての組み合わせで学習と評価を行う．Wikinews データセットでの評価結果を表 6 に示す．ここで，評価値はマイクロ平均 F1 であり，精度 (P) と再現率 (R) も併記している．

表 6 より，深層学習によるリンク拡張手法と，期待エンティティ率 (EER) の推定による学習手法を組み合わせた場合の評価値が最も高く，提案手法の優位性が見て取れる．学習手法を EER 推定に固定し，深層学習による拡張を拡張無しと比較すると評価値が 15.7 ポイント向上しており，ノイズの少ないリンク拡張が実現できていることが分かる．リンク拡張手法を深層学習に固定し，EER 推定をベースラインと比較すると評価値が 3.2 ポイント向上しており，リンク拡張手法で補完できなかったリンクの影響を学習手法により軽減できていることがわかる．自己学習もベースラインと比較して評価値で 1.2 ポイントの向上を得ているが，EER 推定の向

		リンク拡張手法								
		拡張無し			表層マッチ			深層学習		
		P	R	F1	P	R	F1	P	R	F1
学習手法	ベースライン	84.4	25.2	38.8	30.7	59.2	40.4	74.4	56.3	64.1
	自己学習	79.9	34.6	48.3	27.9	52.6	36.5	72.2	59.6	65.3
	ノイズ分離	68.3	35.9	47.1	33.2	63.4	43.6	59.7	50.8	54.9
	EER 推定	80.2	38.0	51.6	30.8	62.0	41.1	70.2	64.7	**67.3**

表 6　Wikinews データセットにおける固有表現抽出評価結果 [%]

上には及ばない．また，自己学習は学習と予測を繰り返すため計算コストが高く，特に今回のように大規模なコーパスを用いる場合は，計算コストの面でも EER 推定の方が優位である．表層マッチを拡張無しと比較した場合，全体的に再現率の向上は見られるものの精度が大幅に低下しており，ベースライン以外の学習手法との組み合わせにおいて評価値が低下している．これは，周辺文章を無視した表層文字列のみでのリンク拡張が誤ったラベルを多く生成しており，後続の学習手法へ大きな負の影響を与えているためと考えられる．ノイズ分離は，表層マッチと組み合わせた場合に他の学習手法と比較して最も良い評価値であるが，深層学習による拡張と組み合わせた場合は最も低い評価値となっている．これは，手法自体にノイズ耐性があることを示しているが，深層学習によるリンク拡張で得られるようなノイズの少ないデータセットを対象とした場合は性能を発揮できないことを示している．

7.2 階層別評価

拡張固有表現階層はその名の通り階層構造を持っており，本研究では末端のカテゴリーを固有表現クラスとして採用している．末端カテゴリーから上位のカテゴリーを辿ることが可能なため，実用時に詳細なカテゴリー分類が必要ない場合，固有表現抽出器の予測を上位のカテゴリーに集約して使用することが可能である．例えば末端カテゴリーの一つである “都道府県郡州名” カテゴリーは “名前”>“地名”>“GPE”>“都道府県郡州名” のように上位カテゴリーを持ち，2 階層目のカテゴリーを用いる場合は “地名” に集約することができる．

実際に上位階層のカテゴリーに集約した場合の性能を評価するため，本節では階層別の評価を行う．末端カテゴリーは 4 階層目まであり，より上位の 1, 2, 3 階層目における評価を行った結果を表 7 に示す[13]．階層が上位であるほどカテゴリーは抽象的になり，分類対象のカテゴリー数が減るため解くべき問題は容易になる．また，1 階層目では根ノードである “名前” カテゴリーに全て集約されるため，言及分類の必要がなく，言及検出のみの評価となる．

提案手法同士の組み合わせは他の組み合わせと比較していずれの階層でも最良の評価値を保持しており，提案手法の一貫した優位性が表れている．提案手法同士の組み合わせに着目すると，1 階層目での評価値は 76.7%，4 階層目の評価値は 67.3%であり，その差分は 9.4 ポイントである．本差分は末端階層までの言及誤分類による評価値の低下を示している．評価値は 1 階層目と 2 階層目の間の区間で 4.5 ポイント，2 階層目と 3 階層目の間で 3.5 ポイント，3 階層目と 4 階層目の間で 1.4 ポイント低下している．これらの言及誤分類に関しては 8.3 節で分析する．また，1 階層目での提案手法の再現率は 73.6%である．つまり，全ての固有表現のうち 73.6%を検出できており，残りの 26.4%が言及検出漏れとなる．この言及検出漏れに関しては 8.4 節で分析する．

[13] 4 階層目での評価は表 6 の通常評価と同じであるためそちらを参照する．

		リンク拡張手法									階層
		拡張無し			表層マッチ			深層学習			
		P	R	F1	P	R	F1	P	R	F1	
学習手法	ベースライン	86.1	25.7	39.5	31.5	60.7	41.5	75.7	57.2	65.2	3
	自己学習	81.8	35.5	49.5	29.3	55.2	38.3	73.9	61.1	66.9	
	ノイズ分離	69.8	36.8	48.2	33.9	64.5	44.4	60.9	51.9	56.0	
	EER 推定	81.9	38.8	52.7	31.7	63.8	42.3	71.7	66.0	**68.7**	
	ベースライン	89.3	26.6	41.0	33.0	63.8	43.5	79.4	60.1	68.4	2
	自己学習	86.1	37.3	52.1	31.8	59.9	41.5	77.7	64.2	70.3	
	ノイズ分離	73.4	38.7	50.6	35.6	67.9	46.8	64.0	54.4	58.8	
	EER 推定	85.6	40.5	55.0	33.4	67.3	44.7	75.2	69.3	**72.2**	
	ベースライン	91.9	27.4	42.2	36.4	70.3	48.0	84.1	63.6	72.4	1
	自己学習	89.7	38.9	54.3	38.6	72.7	50.4	82.1	67.8	74.3	
	ノイズ分離	77.9	41.0	53.7	39.2	74.7	51.4	68.6	58.4	63.1	
	EER 推定	88.7	42.0	57.0	36.8	74.1	49.2	79.9	73.6	**76.7**	

表 7　Wikinews データセットにおける階層別固有表現抽出評価結果

7.3　カテゴリー別評価

本節では，カテゴリー別に手法を評価する．評価は 2 階層目のカテゴリーを対象とし，マイクロ平均 F1 を算出する．この際，7.2 節の階層別評価とは異なり上位カテゴリーへの置き換えは行わない．評価値を表 8 に示す．評価データ頻度は各カテゴリーに属する固有表現が評価データに出現した回数を示す．“名前__その他”，“色名” カテゴリーは評価データに出現しないため省略されている．

提案手法同士の組み合わせは，“組織名”，“プロダクト名” において最も優れた性能を示している．また，“地名” や “組織名” も最良の組み合わせと同等の性能である．これらのカテゴリーでは特に提案手法を用いる意義があると言える．リンク拡張手法に着目すると，表層マッチは “人名”，“生物呼称名”，“病気名” において深層学習による拡張より優れている．これらのカテゴリーの固有表現は表層文字列が特異的であると考えられ，表層マッチ時に正しいラベルのみを特定しやすいため優れた結果となっていると推測される．しかし，“人名” の場合は，表層マッチは姓名分割といった言語依存の処理を使用しているため，言語非依存に処理を行う提案手法より有利であることに注意する必要がある．上記のカテゴリー以外では，表層マッチは深層学習による拡張より大きく劣っていることから，表層文字列といった言語依存特徴に強く依存する手法の限界が見える．また，拡張無しは，“神名” において深層学習による拡張より優れているが，評価データに十分な量のラベルが存在しないため評価が難しい．学習手法に着目すると，EER 推定は “イベント名”，“自然物名”，“病気名” でその他の学習手法より劣っている．これらのカテゴリーは，8.5 節で学習データにおける頻度的な視点から分析する．

リンク拡張手法	学習手法	人名	神名	生物呼称名	組織名	地名
拡張無し	ベースライン	46.6	**100.0**	16.7	39.4	50.6
	自己学習	58.6	50.0	16.7	51.1	60.2
	ノイズ分離	53.2	40.0	40.0	47.9	59.1
	EER 推定	59.5	33.3	30.8	49.7	66.8
表層マッチ	ベースライン	72.7	33.3	**77.8**	39.0	38.5
	自己学習	77.4	25.0	44.4	33.6	40.1
	ノイズ分離	78.9	16.7	63.6	42.4	39.9
	EER 推定	**79.8**	15.4	**77.8**	36.3	39.3
深層学習	ベースライン	64.1	25.0	53.3	65.5	**78.6**
	自己学習	69.3	20.0	53.3	67.7	77.7
	ノイズ分離	57.7	0.0	47.6	57.5	66.7
	EER 推定	76.0	22.2	70.0	**68.3**	78.4
評価データ頻度		761	1	11	900	1,080

リンク拡張手法	学習手法	施設名	プロダクト名	イベント名	自然物名	病気名
拡張無し	ベースライン	52.8	28.1	42.1	28.9	34.8
	自己学習	61.1	36.2	55.9	29.2	44.0
	ノイズ分離	61.6	37.7	41.2	41.9	37.7
	EER 推定	63.4	40.5	54.2	37.8	62.1
表層マッチ	ベースライン	52.5	36.8	38.2	10.0	55.3
	自己学習	40.1	27.0	43.1	7.3	62.5
	ノイズ分離	50.1	40.6	36.0	23.1	**71.4**
	EER 推定	50.7	37.7	37.4	10.3	61.7
深層学習	ベースライン	71.1	57.1	60.2	**61.0**	62.7
	自己学習	**71.3**	58.0	**65.8**	49.7	61.7
	ノイズ分離	58.5	50.2	41.2	57.4	46.7
	EER 推定	71.2	**60.4**	62.2	56.8	60.7
評価データ頻度		296	2,576	297	64	38

表 8 Wikinews データセットにおけるカテゴリー別固有表現抽出評価結果 [%] ：各評価値はマイクロ平均 F1 を示す.

7.4 ドメイン別評価

本節では，表 5 に示す Wikinews のトップカテゴリーをドメインとして扱いドメイン別評価を実施する．評価結果を表 9 に示す．各評価値は，その他の実験と同様にマイクロ F1 平均である．

表 9 よりいずれのドメインにおいても提案手法同士の組み合わせが最良値を得ており，提案手法の優位性が伺える．特に気象や学術，政治において評価値が高く，構築された固有表現抽出器はこのような専門性の高いドメインで特に活用できる可能性がある．対照的に社会やスポーツ，ひとといったドメインでは評価値が低いが，これは 8.4 節で分析する “地位職業名” や “称

リンク拡張手法	学習手法	ひと	スポーツ	学術	政治	文化	気象	社会	経済
拡張無し	ベースライン	32.3	37.9	47.0	37.9	40.7	51.4	35.1	34.3
	自己学習	41.7	47.5	56.4	46.1	47.8	58.7	47.1	44.8
	ノイズ分離	32.2	46.3	58.8	42.7	50.0	59.3	43.6	45.3
	EER 推定	48.8	49.7	59.1	50.5	52.7	68.3	47.1	47.8
表層マッチ	ベースライン	27.0	46.6	43.0	40.2	41.4	49.5	34.8	34.8
	自己学習	25.7	44.2	39.5	38.5	39.1	41.8	28.6	26.6
	ノイズ分離	28.3	49.1	47.3	40.0	46.9	52.6	38.9	38.5
	EER 推定	30.6	46.4	43.4	40.1	41.9	51.5	34.7	37.3
深層学習	ベースライン	59.4	60.1	74.0	61.6	62.1	79.5	63.1	65.7
	自己学習	64.3	62.3	73.6	64.0	64.2	80.2	62.4	65.9
	ノイズ分離	43.4	51.6	64.9	52.4	58.6	71.1	51.0	59.6
	EER 推定	**65.2**	**64.7**	**74.2**	**71.1**	**67.2**	**80.7**	**63.8**	**66.4**

表 9　Wikinews データセットにおけるドメイン別評価結果 [%]：各評価値はマイクロ平均 F1 を示す．

号名__その他”，“人名” カテゴリーにおける言及検出漏れに関連している．最良値の次に良い値と比較した場合，政治カテゴリーでは評価値が 7.1 ポイント向上，文化カテゴリーでは評価値が 3.0 ポイント向上しており，特に提案手法を用いる意義があると言える．

8　分析

本章ではより詳細な提案手法の分析を行う．以下では，8.1 節で拡張されたリンクに関して，8.2 節で任意の期待エンティティ率を使用した場合の評価値の推移に関して，8.3 節で提案手法により言及を拡張固有表現階層に分類する際の誤りに関して，8.4 節で提案手法における言及の検出漏れに関して，8.5 節で評価値と学習データのラベル頻度の関係に関して分析する．

8.1　リンク拡張手法

本小節ではリンク拡張手法の分析を行う．分析のため，拡張したリンクから 100 件をランダムにサンプリングし人手で評価を行った．結果を表 10 に示す．各手法によるリンク拡張後のエンティティ率も併記している．

表 10 より，提案手法は非常に高い精度でリンクを拡張できていることが分かる．固有表現クラス精度は 91.0%であり，非常に高い精度で固有表現抽出器の学習データを生成できていることがわかる．誤りを分析したところ，マイナーなエンティティが Wikipedia に存在しない場合に，拡大解釈されたエンティティが紐づけられる例が目立った．例えば，「交通アクセス：岐阜バス「岐阜保健短大」バス停留所で下車」の「岐阜保健短大」はバス停を示すエンティティだが，短期大学を示すエンティティ『岐阜保健短期大学』に紐づいている．また，「1929 年・『汗』:

	リンク精度 [%]	固有表現クラス精度 [%]	エンティティ率
拡張無し	—	—	0.101
表層マッチ	16.0	18.0	0.286
提案手法	86.0	91.0	0.209

表 10　リンク拡張人手評価結果とエンティティ率：リンク精度は検出範囲と紐づくエンティティが正しいリンクの割合，固有表現クラス精度は検出範囲と紐づくエンティティの固有表現クラスが正しいリンクの割合を示す．

監督 内田吐夢」の「汗」は映画を示すエンティティだが，哺乳類の分泌液を示す『汗』に紐づいている．本問題の解決のため，より厳密な符号器を考案する必要がある．

表 10 より，表層マッチのリンク精度が非常に低いことがわかる．誤りを分析したところ，「1978 年 4 月録音」の「月」が衛星を示す『月』に紐づく，「1984 年 4 月 23 日-」の「日」が国を示す『日本』に紐づく，短い固有名による誤りが特に目立った．これらは品詞情報を用いることで改善できるが，品詞解析には言語依存のツールが必要である．

エンティティ率を見ると，表層マッチは提案手法よりも多くのリンクを拡張している．だが，誤ったリンクを含まないエンティティ率をリンク精度から推測すると表層マッチは 0.131 であり，これは提案手法の 0.194 よりもかなり低い数値である．提案手法は，表層マッチよりも高い精度で，より多くの正しいリンクを拡張できることが示されている．

8.2　期待エンティティ率の影響

本研究では，Wikipedia の被リンク数を活用してデータセットの期待エンティティ率を推定し，先行手法 (Effland and Collins 2021) を用いて学習中に制約をかけることで NIL 言及の影響軽減を試みている．本推定値の妥当性を確認するため，期待エンティティ率を手動で操作し，先行手法を適用した場合の結果について調査する．期待エンティティ率 $\hat{\rho}$ を $\{0.05, 0.10, 0.15, 0.20, 0.25, 0.30\}$ のいずれかの数値に設定し学習を行った結果を表 11 に示す[14]．実験で得られた推定値 $\hat{\rho} = 0.238$ を使用した結果も表記している．

表 11 より，最良値は $\hat{\rho} = 0.2$ を用いた場合であるものの，提案手法による推定値 $\hat{\rho} = 0.238$ 以下で評価値に大きな差はなく，推定値より大きい場合に大きく評価値が下がり始めていることが分かる．先行研究 (Effland and Collins 2021) でも，オラクルエンティティ率[15]以下の値を用いれば性能がほぼ変わらないことが報告されている．本実験でも同様の現象が起きていることから，推定した期待エンティティ率の妥当性は高いと考えられる．しかし，得られた結果か

[14] 入力にエンティティが含まれない状況は想定していないため $\rho = 0.0$ は実験していない．
[15] 正解データから計測された値．

期待エンティティ率	P	R	F1
0.05	71.4	63.7	67.3
0.1	71.4	63.7	67.3
0.15	71.4	63.7	67.3
0.2	71.9	63.5	**67.4**
0.238	70.2	64.7	67.3
0.25	69.4	64.5	66.9
0.3	64.5	65.2	64.9

表 11　提案手法における期待エンティティ率を変化させた場合の評価値の推移 [%]：下線は実験で用いた推定値を示す．

		予測カテゴリー									
		人名	神名	生物呼称名	組織名	地名	施設名	プロダクト名	イベント名	自然物名	病気名
正解カテゴリー	人名	—	—	—	2	2	—	2	—	—	—
	神名	—	—	—	—	—	—	—	—	—	—
	生物呼称名	—	—	2	—	—	—	—	—	—	—
	組織名	1	—	—	45	8	12	50	4	—	—
	地名	—	1	—	34	51	2	—	2	—	—
	施設名	—	2	—	5	16	22	—	4	—	—
	プロダクト名	8	—	—	40	1	1	150	13	2	—
	イベント名	—	—	—	7	1	5	6	10	—	—
	自然物名	—	—	—	—	1	—	1	—	1	2
	病気名	—	—	—	—	—	—	—	—	5	—
予測総数		569	4	9	759	925	247	1,613	236	59	29

表 12　提案手法予測の混同行列：0 の場合はハイフン (—) で示し，対角成分には下線を付与している．

らはエンティティ率推定による性能向上は得られない可能性が示唆されている．推定値の有効性を示すためには，制約手法におけるハイパーパラメーターの妥当性の検証や，新たな制約手法を考案する必要がある．

8.3　言及誤分類

本節では，検出した固有表現の言及を拡張固有表現階層のカテゴリーに分類する際に発生した誤りを分析する．2 階層目のカテゴリーにおける誤分類のみの混同行列を表 12 に示す．通常の混同行列と異なり，対角成分は 3 階層目以降で分類を誤った固有表現の総数のみを表す．分類だけに着目し分析するため，表には 1 階層目時点で誤っている，つまり言及検出の時点で誤っている固有表現の予測は含まない．詳細な分析のため，2 階層目で発生した誤分類の例を表 13，3 階層目以降で発生した誤分類の例を表 14 に示す．それぞれ表 12 における誤分類の多い上位

正解カテゴリー	予測カテゴリー	件数	表層文字列例
組織名 >法人名 >企業名	プロダクト名 >バーチャルアドレス名 >チャンネル名	14	NHK (6) CNN (4) メガポート放送 (2)
組織名 >法人名 >企業名	プロダクト名 >出版物名 >新聞名	13	産経新聞 (3) 産経 (3) 毎日新聞 (2)
組織名 >法人名 >企業名	プロダクト名 >サービス名	9	ライブドア (9)
プロダクト名 >バーチャルアドレス名 >チャンネル名	組織名 >法人名 >企業名	17	日テレ (2) 日本テレビ (2) ITpro (1)
プロダクト名 >出版物名 >新聞名	組織名 >法人名 >非営利団体名	5	アジアネット (2) ASCII (1) ASCII.jp (1)
プロダクト名 >バーチャルアドレス名 >チャンネル名	組織名 >法人名 >非営利団体名	5	アジアネット (2) ASCII (1) ASCII.jp (1)
地名 >ＧＰＥ >国名	組織名 >競技組織名 >競技団体名	30	日本 (9) ドイツ (8) アルゼンチン (5)
地名 >ＧＰＥ >国名	組織名 >政治的組織名 >政府組織名	2	フランス (1) 日本 (1)
地名 >ＧＰＥ >市区町村名	組織名 >競技組織名 >競技団体名	2	福岡 (2)

表 13　2 階層目における分類誤り例:表層文字列例には上位 3 件を表記し，括弧内はその件数を示す.

3 カテゴリーを対象に，上位 3 つの例を表示している．誤分類の発生した表層文字列の例も上位 3 件まで示している．以下に，上記に対する分析により判明した誤分類の原因を 5 件示す．

周辺文脈の情報の少なさに起因　表 13 上段では，モデルがメディア運営組織をメディア自体として誤って解釈したことにより誤分類が発生している．表 13 中段はその反対である．表層文字例に示されるようなエンティティはメディアとメディア運営組織の両方の特性を持つ場合が多く，周辺文脈からそのどちらであるか推定できない場合に誤分類が発生している．評価データをウェブニュース記事から作成している都合上，出典元としてこれらエンティティが出現する場合が多いため，その総数が多くなっている．

正解カテゴリー	予測カテゴリー	件数	表層文字列例
プロダクト名 >バーチャルアドレス名 >チャンネル名	プロダクト名 >サービス名	43	ウィキニュース (18) ウィキペディア (12) ウィキソース (2)
プロダクト名 >出版物名 >新聞名	プロダクト名 >サービス名	19	ウィキニュース (18) ITmedia (1)
プロダクト名 >主義方式名 >競技名	プロダクト名 >作品名 >公演名	6	シングルス (6)
地名 >ＧＰＥ >都道府県州郡名	地名 >ＧＰＥ >市区町村名	11	島根県 (1) アテネ (1) 広島県 (1)
地名 >ＧＰＥ >都道府県州郡名	地名 >地形名 >島名	11	北海道 (11)
地名 >地名__その他	地名 >ＧＰＥ >市区町村名	10	長滝 (2) 浅内 (1) ダーバン (1)
組織名 >法人名 >非営利団体名	組織名 >法人名 >企業名	11	日本赤十字社 (5) AP 通信 (2) 全国農業協同組合連合会 (1)
組織名 >競技組織名 >競技リーグ名	組織名 >競技組織名 >競技連盟名	3	V リーグ (3)
組織名 >政治的組織名 >政治的組織名__その他	組織名 >政治的組織名 >政府組織名	3	チベット亡命政府 (3)

表 14　3 階層目以下における分類誤り例：表記は表 13 に従う．

固有表現抽出ラベル付け定義と学習特徴間の齟齬に起因　13 の下段より，全体として “国名” や “市区町村名” が “競技団体名” として扱われている例が目立つ．本誤分類は 8.3 節で導入した固有表現抽出ラベル付け定義の (3) をモデルが考慮できていないため発生している．

類似した文脈特徴に起因　表 14 上段より，“プロダクト名” 以下では「ウィキペディア」,「ウィキニュース」といった言及を “サービス名” に誤分類している．ここで，Wikipedia 分類データ中の対応するエンティティを調べたところ，それぞれに “サービス名” カテゴリーは付与されていないことが分かった．評価データにおける対象言及の周辺文脈が，学習データ中の “サービス名” の言及が登場する周辺文脈と類似していたため誤分類が発生した可能性がある．同様に，表 14 下段の「V リーグ」も Wikipedia 分類データにおいて “競技連盟名” カテゴリーは付与さ

れていない．これらの誤分類は，エンティティリンキング等を用いて再度エンティティに紐づく固有表現カテゴリーを参照することで解決できる可能性がある．

Wikipedia 分類データ自体の誤りに起因　表 14 中段の誤分類に着目すると，「島根県」，「アテネ」，「広島県」，「北海道」といった有名な固有名の誤分類が目立つ．Wikipedia 分類データを調べたところ，前者 3 つには “市区町村名” が，後者には “島名” が付与されており，これらは全て “都道府県郡州名” が正しいため Wikipedia 分類データ自体が誤っている．「ダーバン」も “市区町村名” カテゴリーが付与されているが，正しくは “地名__その他” である．また，表 14 下段の「日本赤十字社」，「AP 通信」，「全国農業協同組合連合会」も “企業名” が付与されているが，正しくは “非営利団体名” である．

固有名に対する知識不足に起因　表 14 中段では，“地名__その他” を正解とする「長滝」や「浅内」といった言及が誤って “市区町村名” に分類されている．“市区町村名” と “地名__その他” はその区画に向けた行政組織が存在するかどうかで区別されるが，文脈で判断することが難しい．評価時は，学習時に獲得したエンティティの知識を用いて分類することになるが，これらのエンティティは Wikipedia に専用記事が存在せず，誤分類に繋がっている．表 14 下段の「チベット亡命政府」も “政府組織名” と “政治的組織名__その他” のどちらに属するかは国家の政府であるかどうかにより決定されるが，チベット亡命政府に対する Wikipedia の専用記事がないため情報が少なく誤分類が発生している．

8.4　言及検出漏れ

7.2 節の階層別評価において，26.4%の言及検出漏れが発生していることが判明した．より詳細な分析のため，言及検出漏れが発生しているカテゴリーを集計し，上位 10 件を表 15 に示す．以下に，上記に対する分析により判明した検出漏れの原因を 4 件示す．

リンク表層文字列による誤学習に起因　表 15 より “地位職業名” カテゴリーでは，特に表層文字列が「選手」の場合に多く検出漏れが発生している．Wikipedia 分類を調べたところ「選手」には “地位職業名” カテゴリーが付与されており問題はない．実際の予測結果を調べたところ「○○選手」のように “人名” の後に「選手」が続く場合に多く検出漏れが発生していた．具体

正解カテゴリー	件数	表層文字列例
プロダクト名>称号名>地位職業名	341	選手 (175)・天皇 (14)・首相 (12)
プロダクト名>称号名>称号名__その他	239	さん (120)・氏 (99)・陛下 (12)
人名	198	エンドン (23)・沢尻 (8)・篠田 (7)
プロダクト名>主義方式名>賞名	83	メダル (36)・金メダル (21)・銅メダル (7)
地名>ＧＰＥ>国名	60	日本 (23)・中国 (4)・日 (3)

表 15　提案手法における言及検出漏れ例：上位 5 カテゴリーを表示している．表記は表 13 に従う．

的には，「選手」に関する検出漏れ 175 件のうち 154 件が “人名” に連続していた．Wikipedia のリンクを調べたところ “人名” に続く「選手」に対してリンクが付与されているパターンは 1 件のみであり，その後のリンク拡張においても “人名” に続く「選手」のリンクが 27 件しか補完できていなかった．“称号名__その他” カテゴリーでは，「さん」や「氏」といった敬称の検出漏れが多く発生している．Wikipedia を調べたところ「さん」や「氏」を敬称として説明する専用の記事がなく，「敬称」という記事内にリスト形式でまとめられていることが分かった．このように言及に対して専用の記事がない場合は NIL 言及として扱われており，リンク拡張では対応できない．期待エンティティ率を用いた学習手法を用いているが，今回のパターンでは適切に対処できていない．“人名” カテゴリーでは突出している表層文字列はないものの，検出漏れの総数が多い結果となっている．実際の予測結果を確認したところ，“人名” に “地位職業名” や “称号名__その他” が続く場合に結合して “人名” として誤検出される例が多いことがわかった．具体的には，“人名” と “地位職業名” が結合して “人名” として誤検出された例が 51 件，“人名” と “称号名__その他” が “人名” として誤検出された例が 54 件存在した．Wikipedia のリンクを調べたところ，“人名” へのリンクのうち表層文字列が「○○選手」のパターンが 91 件，「○○さん」のパターンが 1,380 件，「○○氏」のパターンが 6,601 件存在した．この中には「ぐっさん」や「メリーさん」のように全体で固有名として認められる場合や，「足利尊氏」や「大舘尚氏」のように名前が氏で終わる例も含まれている．しかし「イチロー選手」や「石田さん」，「野口氏」といった拡張固有表現階層において全体で固有名とは認められないリンクが多く散見されることから，記事の編集者によってリンク付与範囲が統一されていないことが分かる．上記のように “地位職業名” や “称号名__その他” を取り込んで “人名” として扱うリンクが多いことが，“人名” の検出漏れの要因，「選手」等の “地位職業名” に対してリンク拡張が機能しない要因，「さん」や「氏」等の “称号名__その他” に対して期待エンティティ率を用いた学習手法が機能しない要因となっている．リンクの表層文字列がリンク先のタイトルもしくはリダイレクトと大きく異なる場合は，タイトルやリダイレクトで置き換えることでこれらの影響を軽減できる可能性がある．

短縮形に関する教師信号不足に起因　“人名” カテゴリーにおいて苗字のみといった短縮形の検出漏れが目立つ．リンク拡張ではこれらの短縮形に完全に対処できておらず，固有表現抽出器への教師信号として不十分であることから，検出漏れが発生していると考えられる．

Wikipedia 分類が “コンセプト” の記事に起因　“賞名” カテゴリーでは「メダル」，「金メダル」，「銅メダル」の検出漏れが多い．Wikipedia 分類データではこれらは “賞名” ではなく，固有表現以外を指す “コンセプト” のみが付与されている．本研究では，“名前” 以下カテゴリーが付与されていない記事へのリンクは未知（UNK タグ）ではなく，固有表現以外（O タグ）として扱われる都合，誤った学習が行われ言及漏れが発生している．それらを全て未知（UNK タグ）として扱った場合，固有表現以外に対する貴重な教師信号が失われることになる．

Wikipedia 分類定義と固有表現抽出ラベル付け定義間の齟齬に起因　“国名” カテゴリーでは「日本」の検出漏れが 23 件発生している．実際の結果を分析したところ，「日本代表」を “競技団体名” に，「日本ラグビー」を “競技名” といったように「日本」を含み誤検出している例が 13 件存在した．最も多い「日本代表」に関して Wikipedia のリンクを調べたところ，表層文字が「日本代表」のリンクが合計で 3,702 件存在した．リンク先として最も多い 3 記事は，「サッカー日本代表」へのリンクが 1,644 件，「ラグビー日本代表」へのリンクが 730 件，「日本代表」へのリンクが 193 件であった．これらリンク先は Wikipedia 分類データにおいて全て “競技団体名” カテゴリーに分類されている．それゆえ，学習データにおいて「日本代表」に対して “競技団体名” が付与されている例が多く存在している．しかし，8.3 節でも述べたように，今回は評価データ作成時に全体が新しい固有名として定義されているかどうかを考慮しており，「日本代表」はそれ全体を固有表現とは認められず誤検出扱いとなる．

8.5　学習データ頻度

7.3 節のカテゴリー別評価では，“イベント名”，“自然物名”，“病気名” カテゴリーにおいて EER 推定を用いた学習手法の性能が大きく劣ることが分かった．本原因を探るべく，学習データにおける固有表現ラベル数を集計し，提案手法の性能との関係性を調べた結果を図 5 に示す[16]．縦軸は提案手法の評価値と提案手法以外の最良評価値の差分を示し，横軸は対数軸で学習データのラベル数を示す．評価値差分とラベル数の間の相関係数は 0.866 であり，強い相関があることがわかる．つまり，“イベント名”，“自然物名”，“病気名” における EER 推定を用いた学

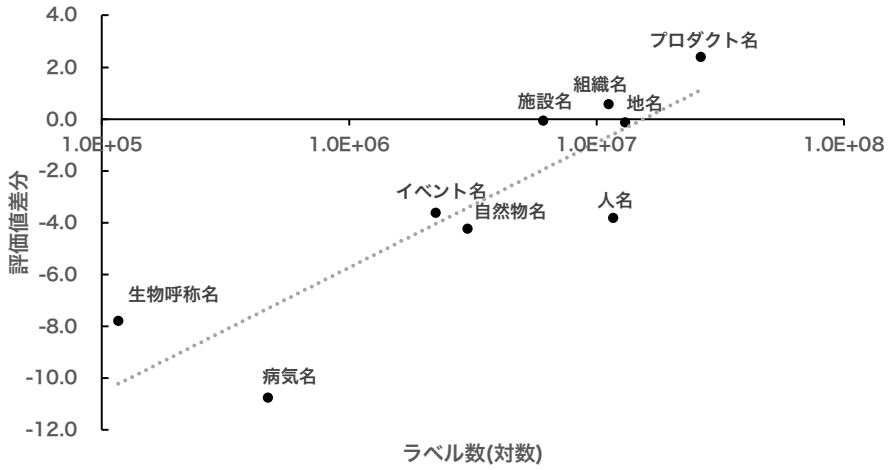

図 5　最良評価値との差分と学習データにおけるラベル数の相関

[16] 神名は評価データの数が少ないため除外している．

習手法の評価値の低さは学習データのラベル数の少なさに起因している可能性がある．EER 推定による学習手法は，全てのカテゴリーに対する予測数をまとめて制限しているため，上記のような教師信号の少ないカテゴリーは予測数を減少させる方向に強い影響を受けており，性能が低下していると考えられる．本問題は，上位のカテゴリーごとに EER を推定し，予測数に対して個別に制約をかけることで解決できる可能性がある．

9 議論

8.3 節，8.4 節での詳細分析により，Wikipedia から固有表現抽出器を学習する際の様々な障壁が明らかになった．課題を整理し以下に示す．

省略形に関する課題　苗字などの短縮形に対する Wikipedia のリンク不足が明らかになった．本問題は，Wikipedia では初出の場合のみリンクが付与されるが，省略形は二回目以降の出現で使用されるため発生している．6.3.1 節では人名の短縮形を収集する方法を導入したが，言語依存である上，その他カテゴリーでの短縮形には対応できていない．今後，Wikipedia の情報から短縮形を言語非依存に予測する手法の開発が必要である．

固有表現抽出定義に関する課題　Wikipedia の記事分類やリンク構造に関する定義と，固有表現抽出に求められる定義の齟齬が明らかになった．固有表現抽出器を実応用するにあたって定義は明確であった方が良いが，Wikipedia の膨大さから前者定義を後者の定義に合わせることは難しい．今後，Wikipedia から学習された固有表現抽出器の振る舞いを紐解いた上で，適した定義を再考する必要がある．

拡張固有表現階層に関する課題　Wikipedia 記事の分類上では “コンセプト” であるが，文脈によっては固有表現として扱われる言及が存在する事が明らかになった．拡張固有表現階層では幅広いカテゴリーを採用しており，一般名詞との境界に近いカテゴリーも存在している．その影響範囲を特定する必要があるため，拡張固有表現階層のカテゴリーから一般名詞に近いカテゴリーを洗い出した上で，固有表現抽出器の振る舞いを個別に分析する必要がある．

固有表現抽出全般における課題　周辺文脈情報の少なさや知識不足による予測誤りが発生している事が明らかになった．特に，今回のように学習データと評価データで分野や分布が異なる場合，同様の問題が発生しやすい．評価対象に合わせてその都度学習データを構築することは現実的ではないので，本問題は今回の実験設定に依存せず，固有表現全般における課題であると言える．地名等の外部知識を言語非依存に活用する手法の考案や，カテゴリーが曖昧な場合に複数候補の提示を許可するといった緩和策が考えられる．

より実用性の高い固有表現抽出器を Wikipedia から学習するには，少なくともこれらの課題は解決される必要がある．

10 おわりに

Wikipedia から固有表現抽出器を学習する場合，ガイドラインに起因するリンク省略や NIL 言及が問題となる．固有表現の特徴をもとにリンクやラベルの補完に取り組む研究が多く存在するが，その多くが英語を対象としており，固有表現に関する表層的な特徴の少ない日本語には適用ができないといった問題がある．本研究では，深層学習を用いたリンクの拡張手法を提案し，省略されたリンクの補完を行った．また，文章中のエンティティの割合を示す期待エンティティ率の推定手法を提案し，NIL 言及により学習データに混在する偽陰性ラベルの影響軽減を試みた．実際に，日本語 Wikipedia から固有表現抽出器の学習を行い評価を行ったところ，提案手法は比較手法と比べ非常に高い性能を示した．詳細な分析により，Wikipedia のリンク構造を用いて言語非依存に固有表現抽出器の学習を行うための更なる課題を示した．

謝　辞

本研究は，JST，ACT-X，JPMJAX20AI の支援を受けたものである．

参考文献

Al-Rfou, R., Kulkarni, V., Perozzi, B., and Skiena, S. (2015). *POLYGLOT-NER: Massive Multilingual Named Entity Recognition*, pp. 586–594.

Cao, Y., Hu, Z., Chua, T.-s., Liu, Z., and Ji, H. (2019). “Low-Resource Name Tagging Learned with Weakly Labeled Data.” 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. 261–270, Hong Kong, China. Association for Computational Linguistics.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, L., Strassel, S., and Weischedel, R. (2004). “The Automatic Content Extraction (ACE) Program – Tasks, Data, and Evaluation.” In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC’04)*, pp. 837–840, Lisbon, Portugal. European Language Resources Association

(ELRA).

Effland, T. and Collins, M. (2021). "Partially Supervised Named Entity Recognition via the Expected Entity Ratio Loss." *Transactions of the Association for Computational Linguistics*, **9**, pp. 1320–1335.

Ghaddar, A. and Langlais, P. (2017). "WiNER: A Wikipedia Annotated Corpus for Named Entity Recognition." In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 413–422, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Gillick, D., Kulkarni, S., Lansing, L., Presta, A., Baldridge, J., Ie, E., and Garcia-Olano, D. (2019). "Learning Dense Representations for Entity Retrieval." In Bansal, M. and Villavicencio, A. (Eds.), *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pp. 528–537, Hong Kong, China. Association for Computational Linguistics.

Graves, A., Mohamed, A.-r., and Hinton, G. (2013). "Speech Recognition with Deep Recurrent Neural Networks." *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, **38**.

Grishman, R. and Sundheim, B. (1996). "Message Understanding Conference- 6: A Brief History." In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

橋本泰一，乾孝司，村上浩司 (2008). 拡張固有表現タグ付きコーパスの構築. 情報処理学会研究報告, **2008** (113), pp. 113–120. [T. Hashimoto et al. (2008). Constructing Extended Named Entity Annotated Corpora. IPSJ SIG Technical Report, 2008 (113), pp. 113–120.].

Iwakura, T., Komiya, K., and Tachibana, R. (2016). "Constructing a Japanese Basic Named Entity Corpus of Various Genres." In *Proceedings of the 6th Named Entity Workshop*, pp. 41–46, Berlin, Germany. Association for Computational Linguistics.

風間淳一，鳥澤健太郎 (2008). Web 上の資源から構築した複数の固有表現辞書を用いた日本語固有表現認識. 言語処理学会第 14 回年次大会発表論文集, pp. 813–816. [J. Kazama and T. Torisawa (2008). Web Zyou no Shigen kara Kochikushita Hukusu no Koyuhyogen Jisho wo Motiita Nihongo Koyuhyogen Ninshiki. The 14th Annual Meeting of the Association for Natural Language Processing, pp. 813–816.].

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data." In *Proceedings of the 18th International Conference on Machine Learning*, ICML '01, pp. 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Li, J., Sun, A., Han, J., and Li, C. (2022). "A Survey on Deep Learning for Named Entity Recognition." *IEEE Transactions on Knowledge and Data Engineering*, **34** (1), pp. 50–70.

Li, X., Sun, X., Meng, Y., Liang, J., Wu, F., and Li, J. (2020). "Dice Loss for Data-imbalanced NLP Tasks." In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. (Eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 465–476, Online. Association for Computational Linguistics.

Ling, X. and Weld, D. (2021). "Fine-Grained Entity Recognition." In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 26 (1), pp. 94–100.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2020). "RoBERTa: A Robustly Optimized BERT Pretraining Approach.".

Maekawa, K., Yamazaki, M., Ogiso, T., Maruyama, T., Ogura, H., Kashino, W., Koiso, H., Yamaguchi, M., Tanaka, M., and Den, Y. (2014). "Balanced Corpus of Contemporary Written Japanese." *Language Resources and Evaluation*, **48** (2), pp. 345–371.

Mai, K., Pham, T.-H., Nguyen, M. T., Nguyen, T. D., Bollegala, D., Sasano, R., and Sekine, S. (2018). "An Empirical Study on Fine-Grained Named Entity Recognition." In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 711–722, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Malmasi, S., Fang, A., Fetahu, B., Kar, S., and Rokhlenko, O. (2022). "MultiCoNER: A Large-scale Multilingual Dataset for Complex Named Entity Recognition." In *Proceedings of the 29th International Conference on Computational Linguistics*, pp. 3798–3809, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., and McClosky, D. (2014). "The Stanford CoreNLP Natural Language Processing Toolkit." In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Nadeau, D. and Sekine, S. (2007). "A Survey of Named Entity Recognition and Classification." *Lingvisticae Investigationes*, **30**, pp. 3–26.

Nothman, J., Curran, J. R., and Murphy, T. (2008). "Transforming Wikipedia into Named Entity Training Data." In *Proceedings of the Australasian Language Technology Association Workshop 2008*, pp. 124–132, Hobart, Australia.

Nothman, J., Ringland, N., Radford, W., Murphy, T., and Curran, J. R. (2013). "Learning Multilingual Named Entity Recognition from Wikipedia." *Artificial Intelligence*, **194**, pp. 151–175.

近江崇宏 (2021). Wikipedia を用いた日本語の固有表現抽出のデータセットの構築. 言語処理学会第 27 回年次大会発表論文集, pp. 350–352. [T. Omi (2021). Wikipedia wo Mochiita

Nihongo no Koyuhyogen Chushutu no Deta Setto no Kochiku. The 27th Annual Meeting of the Association for Natural Language Processing, pp. 350–352.].

Pan, X., Zhang, B., May, J., Nothman, J., Knight, K., and Ji, H. (2017). "Cross-lingual Name Tagging and Linking for 282 Languages." In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Richman, A. E. and Schone, P. (2008). "Mining Wiki Resources for Multilingual Named Entity Recognition." In *Proceedings of ACL-08: HLT*, pp. 1–9, Columbus, Ohio. Association for Computational Linguistics.

Ringland, N., Dai, X., Hachey, B., Karimi, S., Paris, C., and Curran, J. R. (2019). "NNE: A Dataset for Nested Named Entity Recognition in English Newswire." In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5176–5181, Florence, Italy. Association for Computational Linguistics.

Sekine, S. and Eriguchi, Y. (2000). "Japanese Named Entity Extraction Evaluation: Analysis of Results." In *Proceedings of the 18th Conference on Computational Linguistics - Volume 2*, COLING '00, pp. 1106–1110, USA. Association for Computational Linguistics.

Sekine, S., Kobayashi, A., and Nakayama, K. (2019). "SHINRA: Structuring Wikipedia by Collaborative Contribution." In *Conference on Automated Knowledge Base Construction*.

Sekine, S., Sudo, K., and Nobata, C. (2002). "Extended Named Entity Hierarchy." In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain. European Language Resources Association (ELRA).

Strobl, M., Trabelsi, A., and Zaiane, O. (2020). "WEXEA: Wikipedia EXhaustive Entity Annotation." In *Proceedings of the 20th Language Resources and Evaluation Conference*, pp. 1951–1958, Marseille, France. European Language Resources Association.

Strobl, M., Trabelsi, A., and Zaïane, O. (2022). "Enhanced Entity Annotations for Multilingual Corpora." In *Proceedings of the 13th Language Resources and Evaluation Conference*, pp. 3732–3740, Marseille, France. European Language Resources Association.

Tedeschi, S., Maiorca, V., Campolungo, N., Cecconi, F., and Navigli, R. (2021). "WikiNEuRal: Combined Neural and Knowledge-based Silver Data Creation for Multilingual NER." In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 2521–2533, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tedeschi, S. and Navigli, R. (2022). "MultiNERD: A Multilingual, Multi-Genre and Fine-Grained Dataset for Named Entity Recognition (and Disambiguation)." In *Findings of the Association for Computational Linguistics: NAACL 2022*, pp. 801–812, Seattle, United States.

Association for Computational Linguistics.

Tjong Kim Sang, E. F. and De Meulder, F. (2003). “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*, pp. 142–147.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). “Attention is All you Need.” In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (Eds.), *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc.

Voß, J. (2005). “Measuring Wikipedia.” In *Proceedings International Conference of the International Society for Scientometrics and Informetrics: 10th.*

Wang, X., Jiang, Y., Bach, N., Wang, T., Huang, Z., Huang, F., and Tu, K. (2021). “Automated Concatenation of Embeddings for Structured Prediction.” In Zong, C., Xia, F., Li, W., and Navigli, R. (Eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 2643–2660, Online. Association for Computational Linguistics.

Weischedel, R., Pradhan, S., Ramshaw, L., Kaufman, J., Franchini, M., El-Bachouti, M., Belvin, R., and Houston, A. (2013). “OtoNotes Release 5.0.” *LDC2013T19, Philadelphia, Penn.: Linguistic Data Consortium.*

Weischedel, R. M., Brunstein, A., and Linguistic Data Consortium (Eds.) (2005). *BBN Pronoun Coreference and Entity Type Corpus.* Linguistic Data Consortium, Philadelphia, PA. Title from index.html on CD-ROM. “LDC2005T33.”.

略歴

中山　功太：大規模言語モデル研究開発センター特任研究員．2020 年豊橋技術科学大学情報・知能工学系修士号．理化学研究所革新知能統合研究センター・言語情報アクセスチームリサーチアソシエイトを経て，現職．専門は深層学習，自然言語処理．特にアンサンブル学習，固有表現抽出，情報抽出の研究に従事．言語処理学会会員．

栗田　修平：2013 年京都大学理学部卒業．2015 年同大学院理学研究科物理学教室修士課程修了．2019 年京都大学大学院情報学研究科にて博士（情報学）取得．2019 年より国立研究開発法人理化学研究所革新知能統合研究センター特別研究員．2023 年より同研究員．2020 年よりニューヨーク大学訪問研究員．2024 年より国立情報学研究所コンテンツ科学研究系助教ならびに理化学研究

所客員研究員．深層学習を用いた自然言語処理ならびにコンピュータビジョンの研究に従事．言語処理学会会員．

馬場　雪乃：2012 年東京大学大学院情報理工学系研究科博士課程修了．博士（情報理工学）．同年東京大学特任研究員，2014 年国立情報学研究所特任助教，2015 年京都大学大学院情報学研究科助教，2018 年筑波大学システム情報系准教授を経て，2022 年より東京大学大学院総合文化研究科広域科学専攻准教授．機械学習，ヒューマンコンピュテーションの研究に従事．

関根　　聡：理化学研究所革新知能統合研究センター・言語情報アクセスチームチームリーダー，情報学研究所大規模言語モデル研究開発センター特任教授，合同会社ランゲージ・クラフト主任研究員．1992 年英国マンチェスター大学計算言語学部修士号．1998 年ニューヨーク大学コンピューターサイエンス学部博士号．その後，ニューヨーク大学研究准教授に就任．松下電業産業株式会社（現パナソニック），SONY CSL，マイクロソフトリサーチ，楽天技術研究所ニューヨークなどでの研究職を歴任．専門は自然言語処理．特に情報抽出，固有表現抽出，質問応答，情報アクセス，知識構築の研究に従事．情報処理学会自然言語処理研究会主査，その他役職多数．複数の企業の技術顧問なども兼任．

（2024 年 2 月 1 日　受付）
（2024 年 5 月 1 日　再受付）
（2024 年 6 月 18 日　採録）