

ラベルの構成性を考慮した系列ラベリング

加藤 拓真¹ 阿部 香央莉^{1,2} 大内 啓樹^{2,1} 乾 健太郎^{1,2}¹ 東北大学 ² 理化学研究所¹{takuma.kato, abe-k, inui}@ecei.tohoku.ac.jp ²hiroki.ouchi@riken.jp

1 はじめに

系列ラベリングは、入力文の各単語に対してラベルを付与する問題である。品詞タグ付与や固有表現抽出も系列ラベリング問題として解くことができる。例えば、固有表現抽出では、各単語に対して B-Person や I-Person といったラベルを付与する。これらのラベルは、固有表現のスパン（境界）を表す B, I という要素と、固有表現のタイプを表す Person（人間）という要素からなる。さらに、拡張固有表現 [12] などのデータセットでは、固有表現タイプが階層構造を持つ。例えば、図1中の固有表現タイプ Facility/GOE/School は三階層からなり、第一階層が Facility, 第二階層が GOE, 第三階層が School である。このように、ひとつのラベルは複数の構成要素のラベルから成り立っている。この特徴をラベルの構成性と呼ぶ。

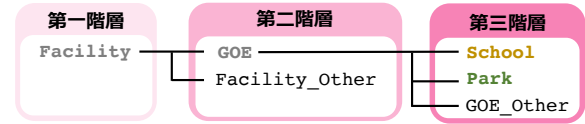
多くの既存研究では、ラベルの構成性を特に考慮せず、各ラベルに紐づく重みベクトルと単語ベクトルの内積に基づいて予測を行っている [3, 5, 7]。しかしながら、ラベルはいくつかの構成要素を共有することが多いため、共有情報を考慮する方が学習の観点から望ましいと考えられる。例えば、図1に示した2つの固有表現ラベル B-Facility/GOE/School と B-Facility/GOE/Park は、前者のラベルは訓練データ中に高頻度で出現する一方、後者のラベルはほとんど出現しない*1。これらのラベルを独立のクラスとして分類器を学習する既存の方法では、出現頻度の低いラベルについて適切に学習することが難しい。ここで、各ラベルを、B, Facility, GOE といった構成要素に分解し、構成要素ごとの重みベクトルをモデルが学習するようにすれば、B-Facility/GOE/Park のような低頻度ラベルに対しても、コーパスに高頻度で出現する構成要素の重みベクトルを共有できるため、汎化性能の向上が期待できる。

本稿では、ラベルの構成要素を考慮した系列ラベリングの手法を提案する。この手法では、各ラベルを構成要素に分解し、それぞれの構成要素に対してベクトル表現を定義して学習する。これらのベクトル表現を足し合わせたベクトルを各ラベルの重みベクトルとして予測に用いる。評価実験を通して、提案モデルが既存モデルを上回る性能を達成し、特に低頻度ラベルに対する顕著な性能改善が見られた。また、提案手法モデルでは共通の上位階層を持つラベル同士が近くにまとまったベクトル空間が観測された。本研究の主な貢献は以下の二点である。

貢献 1: ラベルの構成性を考慮し、ラベルに紐づく重みベクトルを構成的に計算する手法の提案と定式化。

貢献 2: 日本語拡張固有表現抽出の実験を通して、提案手法が低頻度ラベルに対して顕著な精度改善の効果があることを実証。

*1訓練データ中には、B-Facility/GOE/School は 372 件出現するのに対し、B-Facility/GOE/Park はわずか 22 件しか出現しない。



例1:	東北	大学	に行く
	B-Facility/GOE/School	I-Facility/GOE/School	o o
例2:	青葉山	公園	に行く
	B-Facility/GOE/Park	I-Facility/GOE/Park	o o

図1: ラベルの構成性を持つ系列ラベリング問題の一例（ラベルに階層構造を持つ拡張固有表現抽出）。

2 手法

2.1 既存モデル

入力文 $X = (x_1, x_2, \dots)$ の各単語 x に対して、最大の確率となるラベル \hat{y} を出力する。

$$\hat{y} = \arg \max_{y \in \mathcal{Y}} P(y|x, X) \quad (1)$$

ここで、 \mathcal{Y} はデータセットごとに定義されたラベルセットである。例えば、図1では、「東北」と「大学」に対して B-Facility/GOE/School と I-Facility/GOE/School というラベルを付与し、「東北大学」という文字範囲が固有表現 Facility/GOE/School であることが認識できれば正解となる。固有表現ではない「に」、「行く」に対しては Outside を表す 0 のラベルを付与する。

式1中の確分布率 $P(y|x, X)$ は次のように求められる。

$$P(y|x, X) = \frac{\exp(\mathbf{W}[y] \cdot \mathbf{f}(x, X))}{\sum_{y' \in \mathcal{Y}} \exp(\mathbf{W}[y'] \cdot \mathbf{f}(x, X))} \quad (2)$$

ここで、 $\mathbf{W} \in \mathbb{R}^{|\mathcal{Y}| \times \dim}$ はラベルセット \mathcal{Y} に対する重み行列である*2。各行が各ラベル $y \in \mathcal{Y}$ に紐づき、 $\mathbf{W}[y]$ は y 行目のベクトルを表す、 $\mathbf{f}(x, X)$ はニューラルネットワークによってエンコードされたベクトルを表す。このふたつのベクトル間の内積をもとに確率分布が計算される。

2.2 ラベル重み行列の構成的計算

一般的に、系列ラベリングで使用されるラベルは複数の要素から構成される。例えば、ラベル B-Facility/GOE/Park は、エンティティスパンの始まりを表す B ラベルと、エンティティのタイプを表す Facility/GOE/Park ラベルから構成されている。さらに拡張固有表現ラベルの場合、タイプの部分は階層構造を表し、各階層の Facility と GOE と Park の要素から構成される。このような構成要素を考慮して、式2中のラベル重

*2 dim は各重みベクトルの次元数を表す。

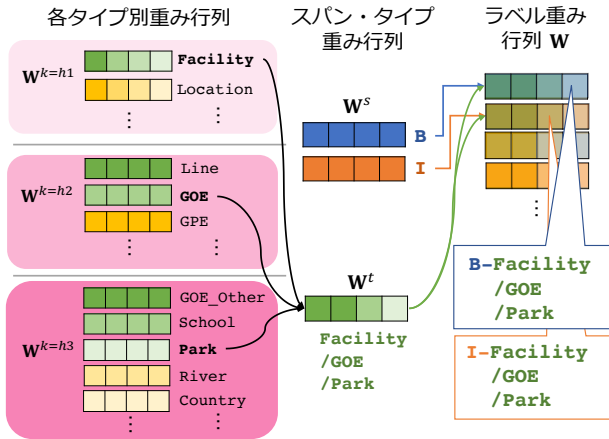


図2: 提案手法の概略. ひとつのラベルに対する重みベクトルを, 構成要素 (スパン・タイプ) から計算している.

み行列 \mathbf{W} を次のように定義する.

$$\mathbf{W}[y] = \mathbf{W}^s[\phi^s(y)] + \mathbf{W}^t[\phi^t(y)] \quad (3)$$

ここでは, スパンとタイプに対する重み行列がそれぞれ $\mathbf{W}^s \in \mathbb{R}^{|\mathcal{V}^s| \times \dim}$ と $\mathbf{W}^t \in \mathbb{R}^{|\mathcal{V}^t| \times \dim}$ として定義され, $\phi^s(y)$ 行目と $\phi^t(y)$ 行目のベクトルが足し合わされている. $\phi^s(y)$ と $\phi^t(y)$ はそれぞれ, ラベル y の持つスパンとタイプのインデックス番号を返す. 例えば, ラベル $y = \text{B-Facility/GOE/Park}$ であり, スパンラベル集合 $\mathcal{V}^s = \{\text{B}, \text{I}\}$ である場合, \mathcal{V}^s におけるスパンラベル B のインデックス番号 1 を返す. 同様に, タイブラベル集合 $\mathcal{V}^t = \{\text{Facility, Facility/GOE, Facility/GOE/Park, \dots}\}$ である場合, \mathcal{V}^t におけるタイブラベル Facility/GOE/Park のインデックス番号 3 を返す. 以上のように, スパンとタイプのそれぞれの重みから構成的にラベル重み行列 \mathbf{W} を計算する.

さらに, 本稿ではタイプに対する重み \mathbf{W}^t も構成的に計算する. 複数の考慮したいタイプがあることを想定し, それぞれのタイプ $k \in \mathcal{K}$ に対して重み行列を定義する.

$$\mathbf{W}^t[\phi^t(y)] = \sum_{k \in \mathcal{K}} \mathbf{W}^k[\phi^k(y)] \quad (4)$$

ここで, $\mathbf{W}^k \in \mathbb{R}^{|\mathcal{V}^k| \times \dim}$ はタイプ k のラベル集合 \mathcal{V}^k に対する重み行列である. $\phi^k(y)$ はラベル y の持つタイプ k のラベルインデックスを表す. 例えば, 図2中の三階層からなるタイブラベルセットを考える. 第一・二・三階層のラベルセットをそれぞれ, $h1 \cdot h2 \cdot h3$ と表し, 各階層 $k \in \{h1, h2, h3\}$ のラベル集合 \mathcal{V}^k にそれぞれ重み行列 \mathbf{W}^k を定義する. ラベル Facility/GOE/Park の場合, Facility と GOE と Park のそれぞれに該当する重みベクトルを足し合わせる. したがって, Facility/GOE/Park と $\text{Facility/GOE/School}$ のような共通の階層ラベルを持つラベルに対しても, Facility と GOE の重みベクトルを共有することが可能となる. 式 4 は, 階層構造に限らず任意の種類タイプ^{*3}を構成的に計算できる点で, Shimaoka ら [6] の階層ラベル行列計算の一般化と見なすことができる.

^{*3}例えば, 性別 (gender) や有生性 (animacy) などのタイプも統一的に扱うことができる.

表1: 拡張固有表現コーパスの詳細

	文数	固有表現数
訓練	34,784	72,318
開発	7,009	11,954
評価	6,783	11,669

表2: 固有表現ラベルの各頻度の詳細

頻度分類 (頻度数)	事例数	
	開発	評価
低頻度ラベル群 (1~29)	1,109	1,037
中頻度ラベル群 (30~99)	2,654	2,358
高頻度ラベル群 (100~)	8,191	8,274

3 実験

3.1 実験設定

データ 本実験では, 拡張固有表現コーパス [12]^{*4}を用いた日本語拡張固有表現抽出を行う. 学習・開発・評価データの分割は, Universal Dependencies Japanese BCCWJ [1]^{*5}にしたがう. データの詳細を表1に示す. このデータ中の固有表現には, 関根の拡張固有表現階層 [10] で定義されている 200 種類のラベルが付加されている. しかし, これらのラベルの間では, 出現頻度にかかなり大きな差が生じている. 我々は, 今回実験を行う上で, 開発データおよび評価データにおけるラベルの頻度別の事例 (固有表現) 数を事前に調査した. 表2にその結果を示す. 表における頻度数は, 頻度分類 (低・中・高) の基準となった頻度の数値幅を表している. 例えば, 低頻度ラベル群は, 各データにおいて, 出現頻度が 1~29 回であったラベル群を表す^{*6}. この表より, 各ラベルの頻度は, 低頻度から高頻度の間では確かに大きな差があることが読み取れる. 実験では, この頻度分類 (低頻度, 中頻度, 高頻度) に基づき, 各頻度レベルでの F_1 値の比較も行う.

モデルのセットアップ 2.1節の式2のニューラルネットワーク $f(x, X)$ として BERT [4] を用いる. 既存モデルでは, BERT で計算したベクトルと, 構成的演算を施さない通常のラベル重み行列との積に基づいて確率分布を計算する^{*7}. 提案モデルでは, ラベルの構成要素に基づいて構成的に計算したラベル重み行列 (2.2節) を利用する. したがって, 両モデル間の違いはラベル重み行列のみであるため, 実験結果として観測される性能差はこの点に帰着できる.

ハイパーパラメータ ハイパーパラメータの設定は既存モデルと提案モデルにおいて同じである. 日本語 Wikipedia で事前訓練済みの BERT [11] を初期値として, 固有表現の学習中

^{*4}<https://www.gsk.or.jp/catalog/gsk2014-a/>

^{*5}https://github.com/UniversalDependencies/UD_Japanese-BCCWJ

^{*6}各ラベル群のラベル種類数は, 開発が低頻度: 107, 中頻度: 47, 高頻度: 24, 評価が低頻度: 104, 中頻度: 45, 高頻度: 24 となっていた.

^{*7}オープンソースの BERT を利用した固有表現抽出モデルをベースにモデルを実装した: <https://github.com/kamalkraj/BERT-NER>.

表3: 既存モデルと提案モデルの比較実験結果

	開発	評価
既存モデル	64.04±0.19	66.18±0.54
提案モデル	64.43±0.14	66.77±0.31

表4: 頻度ごとの f1 値

		開発	評価
(a) 低頻度	既存モデル	36.81±0.73	39.23±0.70
	提案モデル	37.33±0.48	41.19±0.54
(b) 中頻度	既存モデル	47.71±0.65	50.99±0.34
	提案モデル	48.80±0.20	51.96±0.63
(c) 高頻度	既存モデル	68.39±0.27	69.52±0.79
	提案モデル	68.63±0.29	69.90±0.44

に fine-tune を行う。fine-tune の epoch 数は 100 で、パラメータの最適化には Adam[2] を用いる。損失関数にはクロスエントロピー損失関数を用いる。バッチサイズは 32、学習率は 5.0×10^{-5} である。BERT の隠れ層とラベル重み行列の次元数 (dim) は 768 である。ドロップアウト率は $\{0.1, 0.3, 0.5\}$ の中から、開発データの F_1 値を基準に選ぶ。

3.2 結果

表3は、開発データおよび評価データに対する既存モデルと提案モデルの F_1 値の比較実験結果である。提案モデルは、既存モデルを上回る性能を記録した。この結果は、ラベルの構成要素からラベル重み行列を計算する手法の有効性を示唆している。

次に、各頻度ごとの F_1 値の結果を表4に示す。どの頻度のラベル群においても一貫して性能の向上が見られた。特に、低頻度ラベル群に対して、評価データにおいて 1.96 ポイントの向上が見られ、他の頻度のラベル群よりも性能向上に顕著な差が現れた。この結果は、我々が期待した通り、ラベル間で共有される構成要素ごとにラベル重みを学習することによって、低頻度ラベルに対する汎化性能が向上したことを示唆している。

3.3 予測結果の分析

提案モデルの予測正解例 表5に、提案モデルによって改善した低頻度ラベルの出力例を示す。例 (a) では、既存モデルが固有表現である「清水トンネル」を認識できなかったのに対し、提案モデルは正しく認識できている。例 (b) では、両モデルともに「関空」が固有表現であることは認識できているが、既存モデルは誤ったタイプラベルを予測している。その一方で、提案モデルは正しいタイプラベルを予測している。これらの例に見られるように、Facility (施設) などの事例数が多い上位階層タイプラベルを持つラベルにおいて、多くの事例にモデル予測の改善が見られた。

提案モデルの予測誤り例 表6に、既存モデル・提案モデルともに誤った事例を示す。例 (c) では、正解ラベルが B-Product/Material (材料名) であるのに対し、提案モデル

表5: 低頻度ラベルが改善した例

例 (a)	… 越後 湯沢 で、 <u>清水 トンネル</u> を …	
固有表現	清水	トンネル
正解ラベル	<u>B-Facility/Line/Tunnel</u>	<u>I-Facility/Line/Tunnel</u>
既存モデル	0	0
提案モデル	<u>B-Facility/Line/Tunnel</u>	<u>I-Facility/Line/Tunnel</u>
例 (b)	… <u>関空</u> 経由 の マツタケ …	
固有表現	関空	
正解ラベル	<u>B-Facility/GOE/Airport</u>	
既存モデル	<u>B-Product/Food/Food_Other</u>	
提案モデル	<u>B-Facility/GOE/Airport</u>	

表6: 既存モデル・提案モデル共に間違った例

例 (c)	… に <u>ジュラルミン</u> の 機械 が …	
固有表現	ジュラルミン	
正解ラベル	<u>B-Product/Material</u>	
既存モデル	<u>B-Organization/Corporation/Company</u>	
提案モデル	<u>B-Product/Product_Other</u>	
例 (d)	… を <u>1年</u> から 6月 に 短縮 …	
固有表現	1	年
正解ラベル	<u>B-Periodx/Period_Year</u>	<u>I-Periodx/Period_Year</u>
既存モデル	<u>B-Timex/Date</u>	<u>I-Timex/Date</u>
提案モデル	<u>B-Timex/Date</u>	<u>I-Timex/Date</u>

は B-Product/Product_Other (製品名_その他) と誤っている。しかし、第一階層のタイプラベル Product (製品名) は正しく予測できており、末端のタイプラベルで誤ってしまう惜しい予測結果となっている。対照的に、既存モデルはラベル B-Organization/Corporation/Company を出力しており、第一階層のタイプラベルから誤っていることがわかる。このように、提案モデルの予測誤りとして、ラベルの上位階層は正解しているが、末端のタイプラベルで誤っているような惜しい事例が多数見られた。その一方で、既存モデル・提案モデルともに第一階層までもが誤りである予測をしてしまった事例も見られた。例 (d) では、正解タイプラベルが Periodx/Period_Year であるが、両モデルともに Timex/Date と予測している。これらのラベルの第一階層は Periodx (期間) と Date (日付表現) であり、どちらも数値表現に関連する。このように、数値表現に関連するラベルは、モデルにとって区別するのが難しい傾向が観測された。

3.4 ラベルベクトルの可視化

本節では、提案モデルがラベルの構成性を考慮できているか、実際に学習されたラベルベクトルの可視化を通して分析する。

もし、提案モデルにおけるラベルの重みベクトルが効果的に学習できていれば、共通の上位階層タイプラベルを持つラベルはベクトル空間上での位置が近くなるはずである。この仮説を

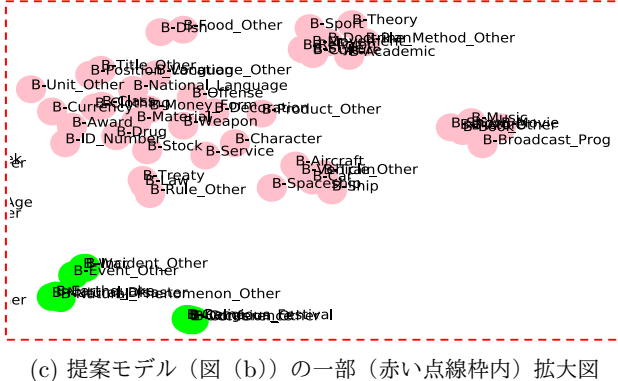
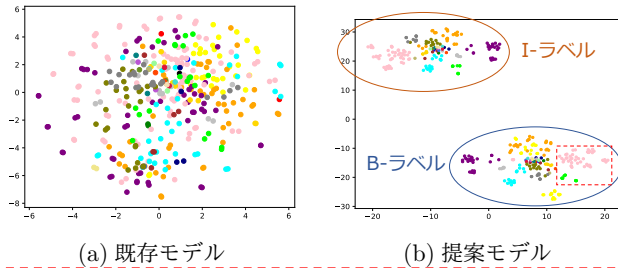


図3: 学習により得られたラベルベクトル空間の t-SNE 可視化
同じ色は上位ラベル (第一階層のラベル) が同じであることを表している。また、「B-ラベル」と「I-ラベル」として囲まれている部分は、それぞれ B から始まるラベル、I から始まるラベルがクラスタを形成していることを表す。

検証するため、各ラベルベクトルを二次元にマップして可視化を行う。

図3に既存モデルと提案モデルのラベルベクトル空間を t-SNE によって可視化したものを示す。既存モデル (図3(a)) のラベルベクトル空間では、共通の上位階層を持つラベルベクトルも散らばっており、近くにまとまっていない。一方で、提案モデルのラベルベクトル空間 (図3(b)) は、B と I のスパンラベルごとに二つの明瞭なクラスタが形成されていることがわかる。各クラスタ内では、共通の上位階層を持つラベル (同じ色) ごとに、近くにまとまっていることが読み取れる。

さらに、図3(b)の拡大図を図3(c)に示す。図3(c)を見ると、確かに同じ上位階層ごとにまとまっていることがわかる。このことから、提案モデルのラベルベクトルはスパンの情報やタイプの階層構造という各構成要素を有効に学習できていると考えられる。

4 関連研究

Mai ら [8] は日本語拡張固有表現抽出に取り組み、辞書情報を用いることによってモデルの性能が向上したことを報告している。彼らの手法は我々の提案手法と直交する関係にあり、ふたつの手法を組み合わせることが可能である。島岡ら [6] は Entity Typing において、ラベルの階層構造を考慮してラベルベクトルを計算する手法を提案し、モデルの性能が向上したことを報告している。本研究では、階層構造に限られていた彼らの手法を、任意の構成要素の集合に適用可能なように一般化し、

系列ラベリングに導入した。本研究と最も近い関連研究として、Matsubayashi ら [9] の研究が挙げられる。彼らは意味役割付与において、ひとつの意味役割ラベルを複数のラベルの集合として表現することによって効率的な学習を可能とした。彼らは最大エントロピーモデルを利用し、各ラベルを離散的な素性として表現していた。一方で、我々はニューラルモデルを利用し、各ラベルをベクトルの演算によって定義している点異なる。

5 終わりに

本稿において、我々はラベルの構成性を考慮した系列ラベリングのモデルを提案した。また、その構成性を考慮したモデルを利用して、約 200 種類の固有表現ラベルを付加する日本語拡張固有表現抽出のタスクに取り組んだ。結果として、構成性を考慮した提案モデルは、固有表現抽出の精度向上に繋がり、特に階層構造を考慮したことによって低頻度ラベルの精度の向上を達成することができた。詳細な分析を通して、提案モデルでも未だ解けていない事例の傾向も明らかになった。特に区別が難しい数値等を含む固有表現などに対して、どのようにアプローチすべきか、という点に関しては、今後改善しなければならない課題である。

謝辞

本研究は JSPS 科研費 JP19H04162, JP19K20351 の助成を受けたものである。

参考文献

- [1] Masayuki Asahara et al. “Universal Dependencies Version 2 for Japanese”. In: *Proceedings of the Eleventh International Conference on the LREC*. Miyazaki, Japan: European Language Resources Association (ELRA), May 2018.
- [2] Yoshua Bengio and Yann LeCun, eds. *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. 2015.
- [3] Jason P.C. Chiu and Eric Nichols. “Named Entity Recognition with Bidirectional LSTM-CNNs”. In: *Transactions of the Association for Computational Linguistics* 4 (2016), pp. 357–370.
- [4] Jacob Devlin et al. “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)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186.
- [5] Guillaume Lample et al. “Neural Architectures for Named Entity Recognition”. In: *Proceedings of NAACL-HLT*. 2016, pp. 260–270.
- [6] Mirella Lapata, Phil Blunsom, and Alexander Koller, eds. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*. Association for Computational Linguistics, 2017.
- [7] Xuezhe Ma and Eduard Hovy. “End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF”. In: *Proceedings of ACL*. 2016, pp. 1064–1074.
- [8] Khai Mai et al. “An Empirical Study on Fine-Grained Named Entity Recognition”. In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 711–722.
- [9] Yuichiro Matsubayashi, Naoaki Okazaki, and Jun’ichi Tsujii. “A Comparative Study on Generalization of Semantic Roles in FrameNet”. In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. Suntec, Singapore: Association for Computational Linguistics, Aug. 2009, pp. 19–27.
- [10] Satoshi Sekine, Kiyoshi Sudo, and Chikashi Nobata, eds. *Proceedings of the Third International Conference on Language Resources and Evaluation, LREC 2002, May 29-31, 2002, Las Palmas, Canary Islands, Spain*. European Language Resources Association, 2002.
- [11] 柴田 智秀, 河原 大輔, and 黒橋 禎夫. “拡張固有表現タグ付きコーパスの構築”. In: 言語処理学会第 25 回年次大会. 2019.
- [12] 橋本 泰一, 乾 孝司, and 村上 浩司. “拡張固有表現タグ付きコーパスの構築”. In: 情報処理学会研究報告. 自然言語処理研究会報告 188 (Nov. 2008), pp. 113–120.