

文字分散表現に基づく単語分類情報を用いた レシピ固有表現抽出

平松 淳^{1,a)} 若林 啓^{2,b)} 原島 純^{3,c)}

概要: 固有表現抽出は自然言語処理の基本的なタスクの1つであり、活発に研究が行われている。固有表現の抽出を行うためには、テキストに対して固有表現を付与した教師データが必要である。しかし、ドメインごとに教師データを構築することはコストが大きい。そこで、本研究では教師データだけではなく、ドメインに関連する言語資源を利用する固有表現抽出器を提案する。具体的には、文字分散表現に基づいて文中の単語を言語資源中で定めたカテゴリに分類し、分類情報を固有表現抽出器の入力として利用する。このモデルについて料理ドメインのデータを用いて実験し、その結果を報告する。

キーワード: レシピ, 固有表現抽出

Recipe Named Entity Recognition based on Word Classification Results using Character-based Distributed Representations

MAKOTO HIRAMATSU^{1,a)} KEI WAKABAYASHI^{2,b)} JUN HARASHIMA^{3,c)}

Abstract: Named entity recognition is a fundamental task in natural language processing and has been widely studied. The construction of a recognizer requires training data that contain annotated named entities. However it is expensive to construct such training data for each domain. In this paper, we propose a recognizer that uses not only the training data but also a domain-specific language resource. Our recognizer first uses character-based distributed representations to classify words into categories in the language resource. It then uses the classification results as an input of the recognition. We report the results of experiments conducted to evaluate our method in the cooking domain.

Keywords: Recipe, Named Entity Recognition

1. はじめに

固有表現抽出は自然言語処理の基盤的なタスクの1つであり、活発に研究が行われている。特に、ニュース記事などの一般的なテキストに限らず、現実世界に存在している

様々なドメインにそれぞれ特化した用語の自動抽出は重要なタスクである。

固有表現抽出は、入力文を単語列 $\mathbf{X} = (x_1, x_2, \dots, x_n)$ としたときに、ラベル列 $\mathbf{y} = (y_1, y_2, \dots, y_n)$ を予測するタスクとして定式化できる。このとき、 y_k は人名や組織名などの固有表現を系列タグ (Begin; B, Inside; I, Other; O) の形に変換したものが用いられる。例えば、 $\mathbf{X} = (\text{日本}, \text{を}, \text{愛}, \text{する})$ という単語列に対しては、 $\mathbf{y} = (\text{B-Location}, \text{O}, \text{O}, \text{O})$ というラベル列が正解データとなる。

固有表現抽出を行うためには、テキストに対して固有表現を付与する必要があるが、ドメインごとに教師データを構築することはコストが大きい。この課題に対し、ドメイ

¹ 筑波大学大学院 図書館情報メディア研究科
茨城県つくば市春日 1-2, 305-8550

² 筑波大学 図書館情報メディア系
茨城県つくば市春日 1-2, 305-8550

³ クックパッド株式会社
東京都渋谷区恵比寿 4-20-3 恵比寿ガーデンプレイスタワー 12F,
150-6012

a) himkt@klis.tsukuba.ac.jp

b) kwakaba@slis.tsukuba.ac.jp

c) jun-harashima@cookpad.com

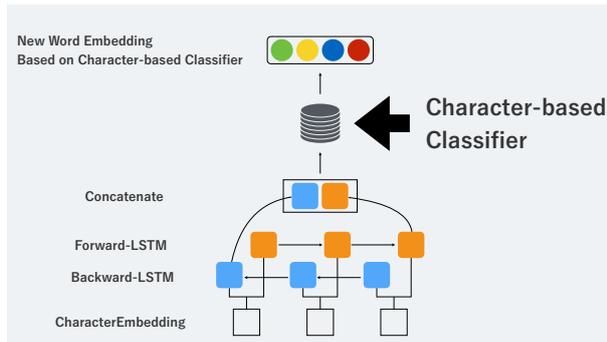


図 1 文字ベースの単語分類器の概要

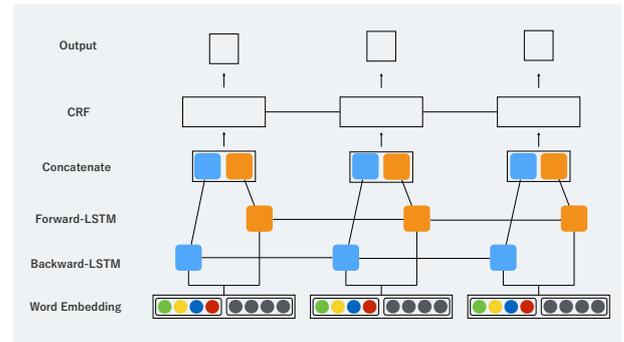


図 2 固有表現抽出器の概要

ンに関連する言語資源を活用することが考えられる。例えば、Sato ら [1] や Pham ら [2] は、辞書情報を特徴量として利用することで、抽出器の性能向上を試みている。しかし、これらの研究では、辞書に含まれない単語の特徴量が得られない。

本研究では、レシピドメインの固有表現抽出に取り組み、辞書として料理オントロジーデータ [3] の情報を活用する。文中の単語について、オントロジー中で付与されている属性ラベルを予測する文字ベースの分類器を学習し、固有表現抽出器の特徴量に組み込む。これにより、辞書には直接含まれない単語に対しても、辞書情報を活用した単語特徴量を獲得できる。得られた単語特徴量の情報を用いることで、固有表現抽出器はより多くの情報を用いて学習ができる。

2. 関連研究

一般ドメインの固有表現抽出では、CoNLL2003 [4] コーパスが教師データとして使われることが多い。CoNLL2003 コーパスは新聞記事データに対して固有表現 (人名, 地名, 組織名など) が付与されたデータである。CoNLL2003 コーパスで良い性能が報告されている Lample らの手法 [5] では、単語情報を特徴量に変換するために Long Short-Term Memory (LSTM) [6] を順方向・逆方向に適用し、2つの出力を結合する手法である Bidirectional-LSTM (BiLSTM) を用いている。Lample らは単語単位の入力と文字単位の入力を組み合わせた特徴量を設計している。これは単語列を $\mathbf{X} = (x_1, x_2, \dots, x_n)$, 単語列の t 番目の単語に含まれる文字列を $C_t = (c_1, c_2, \dots, c_m)$, ラベル列を $\mathbf{y} = (y_1, y_2, \dots, y_n)$ として、

$$\mathbf{c}_t = \text{Bi-LSTM}_{char}(C_t), \quad (1)$$

$$\mathbf{x}_t = [\mathbf{w}_t; \mathbf{c}_t], \quad (2)$$

$$\mathbf{h}_t = \text{Bi-LSTM}(\mathbf{x}_t). \quad (3)$$

と表せる。ここで、 \mathbf{w}_t は x_t の単語分散表現である。

さらに、Lample らは最適なラベル列を求めるために、得られた特徴ベクトルに条件付き確率場 (Conditional Random Field; CRF) [7] を適用している。CRF を用いるため

には、先程得られた \mathbf{h}_t を、

$$\mathbf{z}_t = \mathbf{W}\mathbf{h}_t + \mathbf{b} \quad (4)$$

のように変換する。ここで、 \mathbf{W} は (ラベルの種類数) \times (隠れ層の次元数) の重み行列であり、 \mathbf{h}_t をラベル次元のベクトル \mathbf{z}_t に変換する。 \mathbf{z}_t を用いて、条件付き確率場に基づくラベル列 \mathbf{y} の確率は、 $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n)$ として、

$$P(\mathbf{y} | \mathbf{z}; \mathbf{W}, \mathbf{b}) = \frac{\prod_{i=1}^n \psi_i(y_{i-1}, y_i, \mathbf{z})}{\sum_{y' \in \mathcal{Y}(\mathbf{z})} \prod_{i=1}^n \psi_i(y'_{i-1}, y'_i, \mathbf{z})}, \quad (5)$$

と書ける。ここで、 $\psi_i(y', y, \mathbf{z}) = \exp(W_{y', y}^T \mathbf{z}_i + \mathbf{b}_{y', y})$ である。最後に、最適パス $\hat{\mathbf{y}}$ は

$$\hat{\mathbf{y}} = \underset{\mathbf{y} \in \mathcal{Y}}{\operatorname{argmax}} P(\mathbf{y} | \mathbf{z}; \mathbf{W}, \mathbf{b}) \quad (6)$$

と表せる。 $P(\mathbf{y} | \mathbf{z}; \mathbf{W}, \mathbf{b})$ を最大化する \mathbf{y} は動的計画法の一種である Viterbi アルゴリズムを用いて効率的に計算できる。

文字分散表現の他にも、辞書情報が得られる場合には、それらの情報を活用することでより高性能な抽出器の構築を目指す研究が存在する [1], [2]。Sato ら [1] は単語が辞書に含まれているかどうかを示すバイナリの値を持つベクトルを素性に加えることで、辞書特徴量を活用している。Pham ら [2] は単語があらかじめ定められたカテゴリに含まれる確率を計算し、抽出器の素性ベクトルに追加している。

これらのアプローチに対して、本研究では文字単位の分散表現を BiLSTM に入力し、得られた特徴量を用いてカテゴリを予測するモデルを構築し、モデルの出力を抽出器の素性ベクトルに追加している。

レシピドメインの固有表現抽出では、笹田らがレシピ NE コーパスを整備している [8]。同時に、彼らはレシピ NE コーパスに対する抽出器を提案している [9]。この研究では、文字 n-gram, 文字種 n-gram, 単語 n-gram を用いたロジスティック回帰モデルを用いて固有表現抽出を行っている。さらに、ロジスティック回帰によって得られた系列に対して、動的計画法を用いて起こりえないラベルの遷移を除去することで、ラベル列の最適化を行っている。笹

表 1 レシピ NE コーパス

レシピ数	436
文数	3,317
延べ語数	60,542
異なり語数	3,390
延べ文字数	91,560
異なり文字数	1,130

表 2 Wikipedia コーパス

記事数	1,114,896
文数	18,375,840
延べ語数	600,890,895
異なり語数	2,306,396

表 3 クックパッドコーパス

レシピ数	1,715,589
手順数	8,849,850
文数	12,659,170
延べ語数	216,248,517
異なり語数	221,161

田らの手法は、ラベル列の各ラベルを独立に予測する点推定に基づく手法であり、部分アノテーションコーパスを利用できるという特性を持っている。

3. 提案手法

Pham らの手法では、辞書に含まれない単語に対しては、辞書に基づく特徴ベクトルはゼロベクトルとなる。しかし、入力される単語の中には (i) 辞書中には含まれないが (ii) 辞書中の用語から単語のクラスを予測できる単語が存在する可能性がある。このような単語に対して、辞書中の情報に基づいた特徴ベクトルを割り当てることで、既存手法の抽出性能の向上を期待できる。

本研究では、図 1 に示すニューラルネットワークを用いて単語の特徴量を獲得する。文字分散表現を BiLSTM に入力することで単語特徴量を抽出し、全結合層へ入力し活性化関数を適用することで、単語が辞書中のどのクラスに属するかを予測する。得られた確率ベクトルは、単語レベルの特徴量として固有表現抽出器の入力に追加される。

提案する固有表現抽出器を図 2 に示す。Lample らの手法と提案手法の違いは \mathbf{z}_t の構成方法である。Lample らの手法では式 (1) から式 (4) を用いて \mathbf{z}_t を構成する。これに対して、提案手法では、 \mathbf{w} , \mathbf{c}_t , \mathbf{h}_t を用いて、

$$\mathbf{v}_t = \text{Word-Classifier}(\mathbf{c}_t), \quad (7)$$

$$\mathbf{x}_t = [\mathbf{w}_t; \mathbf{c}_t; \mathbf{v}_t], \quad (8)$$

$$\mathbf{h}_t = \text{Bi-LSTM}(\mathbf{x}_t), \quad (9)$$

$$\mathbf{z}_t = \mathbf{W}\mathbf{h}_t + \mathbf{b}, \quad (10)$$

によって \mathbf{z}_t を構成する。ここで、Word-Classifier は単語分類器である。Word-Classifier は 1 層の全結合層とソフトマックス関数からなり、辞書中のクラスの種類の次元のベクトルを出力する。

提案手法で使用する単語の分類器は、文字分散表現をもとに単語のクラスを予測するため、(i) 辞書中には含まれないが (ii) 辞書中の情報に基づいた単語の特徴を獲得できると期待できる。

4. 実験

4.1 実験データ

4.1.1 レシピ NE コーパス

本研究では、固有表現抽出器の学習と評価のためにレシ

ピ NE コーパス [8] を利用した。レシピ NE コーパスはレシピサービスであるクックパッド*1 に投稿されたレシピの手順データに対して、料理ドメインに則した固有表現を付与したデータセットである。1 つの手順には少なくとも 1 つの文が含まれており、それぞれの文は KyTea [10] を用いて単語単位に分割されている。レシピ NE コーパスの統計情報を表 1 に示す。

4.1.2 ラベルなしコーパス

学習済み分散表現はラベルなしコーパスから学習できる。ニューラルネットワークを用いた固有表現抽出器では、学習済みの分散表現が単語の埋め込み層の初期値として有用であることが知られている [5]。本研究では、2 種類のラベルなしコーパスを用いて単語の分散表現を学習し、埋め込み層の重みの初期値として利用する。それぞれのコーパスには、以下の前処理を行う。

文分割 手順データを文単位に分割

単語分割 KyTea を用いて文を単語単位に分割

本研究で使用する 2 種類のコーパスについて、その詳細を説明する。

百科事典サービスである Wikipedia は、Wikipedia のデータベースのダンプファイルを公開している*2。ダンプファイル (2018 年 08 月 01 日のもの) をダウンロードし、前処理を行ったものを Wikipedia コーパスと呼ぶ。Wikipedia コーパスの統計情報を表 2 に示す。

クックパッドデータセット [11] はクックパッド株式会社が提供するデータセットであり、クックパッドに投稿された献立とレシピのデータが含まれている。ここからレシピの手順に関するデータを取得し、前処理を行ったデータをクックパッドコーパスと呼ぶ。クックパッドコーパスの統計情報を表 3 に示す。

4.1.3 単語分類器の教師データ

本研究では、単語の分類器の学習と評価に料理オントロジー [3] を使用した。料理オントロジーは、料理ドメインに出現する単語について、属性と上位下位関係、そしてその

*1 <https://cookpad.com/>

*2 <https://dumps.wikimedia.org/jawiki/>

表 4 単語分類に用いる教師データ

Class	Frequency
材料-魚介	452
材料-肉	350
材料-野菜	935
材料-その他	725
調味料	907
調理器具	633
動作	928
その他	896

表 5 単語の分類性能

Class	Precision	Recall	F-Score	Support
材料-魚介	0.49	0.43	0.46	90
材料-肉	0.86	0.70	0.77	70
材料-野菜	0.61	0.68	0.64	187
材料-その他	0.72	0.67	0.69	145
調味料	0.77	0.78	0.77	181
調理器具	0.77	0.72	0.75	127
動作	0.93	0.98	0.96	186
その他	0.70	0.73	0.71	179

同義語に関する情報を整備したデータセットである。我々は、このデータセットの属性データを分類器のラベルとして用いた。各単語について、その同義語も用いた。

また、料理オントロジーでは、1つの単語が複数のクラスに属することがある。本研究ではこのような単語(4単語)は教師データから除外した。複数のクラスに属する単語についても分類を行うためには、

- マルチラベル学習を行う
- 複数クラスの組み合わせを1クラスとみなす

のどちらかのアプローチをとる必要がある。これは、計算コストの増加やクラスのスパースネス問題を引き起こす。本研究では、目標タスクは単語分類ではなく固有表現抽出であり、簡単のためにこのような事例を除外する。

単語分類器には、調理手順中に含まれる単語が入力される。すなわち、料理オントロジー中のどの属性にも含まれない単語が多数存在する。このため、料理オントロジーデータを拡張し、「その他」の属性が必要となる。我々は、レシピ NE コーパスの開発データに含まれるが料理オントロジーには含まれない単語を列挙し、その中から料理オントロジー中のどのクラスにも含まれない単語のリストを作成した。その後、2人のアノテータによって、単語リストのうち、「その他」の属性に含まれると思われる単語を列挙した。2人のアノテーション結果が一致した単語のみを採用した結果、896単語からなる単語リストが得られた。我々は、この単語リスト中の単語に「その他」のクラスを付与し、料理オントロジーに追加した。この結果得られた教師データを、学習データ(3,738件)・開発データ(932件)・テストデータ(1,165件)の3種類のデータに分割した。

4.2 比較手法

本研究では以下の手法を用いて実験と比較を行った。

LR 第2章で説明した点推定による笹田らの手法 [9]

LR+DP LRの出力に対して動的計画法を適用して最適なラベル列を求める笹田らの手法

Lample 第2章で説明したLampleらのBiLSTM-CRFを用いた手法 [5]

Proposed 第3章で提案した文字分散表現に基づく単語分類器を用いた手法

LampleとProposedでは、50次元の文字分散表現、 2×25 次元の文字BiLSTM、100次元の単語分散表現と 2×100 次元の単語BiLSTMを用いる。得られる単語特徴量を全結合層によってラベルの種類次元に変換し、CRFを適用してラベル列を求める。学習は負の対数尤度の最小化によって行われる。学習ではAdaDelta[12]を使用し、ミニバッチサイズを10とする。AdaDeltaのハイパーパラメータは $\rho = 0.95$, $\epsilon = 10^{-6}$ とする。また、勾配の爆発を防ぐために勾配のクリッピングを行う。勾配のクリッピングのしきい値は5.0とする。

本研究では、単語の埋め込み層の重みは、

Uniform $[\frac{-3}{\text{dim}}, \frac{3}{\text{dim}}]$ の範囲で一様サンプリングして初期化する。

+Wikipedia Wikipediaコーパスで学習した分散表現を用いて初期化する。ただし、Wikipediaコーパス中に出現しない単語に対してはUniformを用いて初期化する。

+Cookpad クックパッドコーパスで学習した分散表現を用いて初期化する。ただし、クックパッドコーパス中に出現しない単語に対してはUniformを用いて初期化する。

の3種類の方法を用いて初期化を行い、抽出性能への影響力を比較する。

本研究では、分散表現はSkip-gram with Negative Sampling (SGNS) [13]を用いて学習する。SGNSのパラメータは、それぞれ分散表現の次元を100、文脈窓幅を5、負例の数を5とし、実装にはGensim[14]を用いる。

Proposedでは、50次元の文字分散表現と 2×25 次元のBiLSTMを用いた単語の分類器を用いる。単語の特徴量を文字BiLSTMで獲得し、全結合層に入力して辞書のクラス数次元に変換する。最後にソフトマックス関数を適用することで単語が辞書中の各クラスに属する確率を求める。得られた確率を用いてソフトマックスクロスエントロピーを計算し、これを最小化する。最適化にはAdaDeltaを用い、固有表現抽出器の学習で用いたハイパーパラメータを使用する。同様に、固有表現抽出器の学習と同じように勾配のクリッピングを行う。

表 6 レシピ NE コーパスに対する固有表現の抽出性能

Method	Accuracy	Precision	Recall	F-value
Sasada1 (LR)	0.91	0.79	0.85	0.82
Sasada2 (LR+DP)	0.91	0.81	0.86	0.83
Lample (Uniform)	0.91	0.77	0.86	0.81
Lample (+Wikipedia)	0.93	0.81	0.87	0.84
Lample (+Cookpad)	0.93	0.82	0.89	0.85
Proposed (Uniform)	0.91	0.78	0.85	0.82
Proposed (+Wikipedia)	0.93	0.81	0.87	0.84
Proposed (+Cookpad)	0.93	0.83	0.89	0.86

4.3 実験結果

4.3.1 単語分類の実験結果

我々は、学習データを用いて分類器を学習し、開発データでのロスの値が最小となったモデルを選択してテストデータで評価した。テストデータでの分類結果を表5に示す。各クラスでの分類性能のマクロ平均を計算すると、精度は0.74、再現率は0.74、F1値は0.73となった。学習データの追加やハイパーパラメータのチューニングによってさらなる分類性能の向上が期待できるが、本研究では単語の特徴量を獲得することが目的であるため、追加の最適化は行っていない。

最も分類誤りが多かった「材料-魚介」のクラスの誤りについて、詳細に分析を行った。その結果、「河豚」を「材料-肉」と分類する例や「豆鮓」を「材料-野菜」と分類する例が見られた。これらは、「豚」や「豆」の影響で誤りが発生したと思われる。このような分類誤りがどの程度固有表現抽出の結果に影響を与えているか調査することは今後の課題である。

4.3.2 固有表現抽出の実験結果

学習データを用いて固有表現抽出器を学習し、開発データでのロスの値が最小となるエボックの抽出器を用いてテストデータでの評価を行った。比較手法の固有表現抽出性能の比較を表6に示す。評価には、オープンソースの系列ラベリング評価ツールキットの seqeval^{*3} を用いた^{*4}。Sasadaら[9]の手法、Lampleらの手法[5]および提案手法の固有表現抽出の実験結果を表6に示す。実験の結果、提案手法は料理ドメインの固有表現抽出で最高性能であった笹田ら手法および一般ドメインの固有表現抽出で良い性能を発揮するLampleらの手法の性能を上回った。また、(i)分散表現は固有表現の抽出性能に大きく寄与すること、(ii)同一ドメインで学習した分散表現を用いることが効果的であることが確認された。

^{*3} <https://github.com/chakki-works/seqeval>

^{*4} 我々は、笹田らが公開している PWNER ツールキット (<http://www.ar.media.kyoto-u.ac.jp/tool/PWNER/home.html>) の評価スクリプトにバグを発見した。このバグを修正した結果、性能が著者が論文[9]で報告している値と比較して低くなることわかった。このため、既存研究に対しても再評価を行い、その結果を示した。

実際に Lample らの手法の出力と提案手法の出力を比較した。Lample らの手法では、「焼き色が付いたら」という文の「焼き色」という単語に対して、食材を表す「B-F」というラベルを付与した。一方で、提案手法では食材の状態を表す「B-Sf」というラベルが付与することに成功していた。これは、単語の文字情報を考慮し、モデルが「焼き色」という単語は食べ物らしくないと判断した結果であると考えられる。

5. 結論

本研究では、料理ドメインにおける固有表現抽出のタスクについて、文字単位の分散表現を用いて学習された単語分類器の分類結果を活用した抽出器を提案した。このため、提案手法では、辞書には含まれない単語についても辞書情報に基づいた特徴を抽出できる。実験として、提案手法を料理ドメインの固有表現抽出タスクに対して適用し、既存研究よりも高い性能でレシピテキストから固有表現を抽出できることを示した。

今後の課題として、単語の分類器が固有表現抽出器の性能にどの程度寄与しているかを調査することが挙げられる。寄与の度合いを調査する方法としては、辞書データのサイズを変化させて分類器を学習し、抽出器の性能の変化を観察することが考えられる。単語の分類器が抽出器の性能に大きく寄与しているなら、辞書情報を充実させることによりさらなる性能向上が望める。加えて、単語分類において多義語をどのように扱うかも検討したい。また、より一般的なドメインへの適用として、CoNLL2003 コーパスと WordNet を用いて提案手法の実験を行う予定である。

参考文献

- [1] Sato, M., Shindo, H., Yamada, I. and Matsumoto, Y.: Segment-Level Neural Conditional Random Fields for Named Entity Recognition, *Proceedings of International Joint Conference on Natural Language Proceedings of ssing*, No. 1, pp. 97–102 (2017).
- [2] Mai, K., Pham, T.-H., Nguyen, M. T., Duc, N. T., Bollegala, D., Sasano, R. and Sekine, S.: An Empirical Study on Fine-Grained Named Entity Recognition, *Proceedings of International Conference on Computational Linguistics*, pp. 711–722 (2018).

- [3] Nanba, H., Takezawa, T., Doi, Y., Sumiya, K. and Tsujita, M.: Construction of a cooking ontology from cooking recipes and patents, *Proceedings of ACM International Joint Conference on Pervasive and Ubiquitous Computing Adjunct Publication*, pp. 507–516 (2014).
- [4] Sang, E. F. T. K. and De Meulder, F.: Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition, *Proceedings of The SIGNLL Conference on Computational Natural Language Learning*, pp. 142–147 (2003).
- [5] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K. and Dyer, C.: Neural Architectures for Named Entity Recognition, *Proceedings of Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 260–270 (2016).
- [6] Hochreiter, S. and Schmidhuber, J.: Long short-term memory, *Neural Computation*, Vol. 9, No. 8, pp. 1–32 (1997).
- [7] Lafferty, J., McCallum, A. and Pereira, F. C. N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data, *Proceedings of International Conference on Machine Learning*, pp. 282–289 (2001).
- [8] 笹田鉄朗, 森信介, 山肩洋子, 前田浩邦, 河原達也: レシピ用語の定義とその自動認識のためのタグ付与コーパスの構築, *自然言語処理*, Vol. 22, No. 2, pp. 107–131 (2015).
- [9] Sasada, T., Mori, S., Kawahara, T. and Yamakata, Y.: Named entity recognizer trainable from partially annotated data, *Proceedings of International Conference of the Pacific Association for Computational Linguistics*, Vol. 593, pp. 148–160 (2015).
- [10] Neubig, G., Nakata, Y. and Mori, S.: Pointwise Prediction for Robust, Adaptable Japanese Morphological Analysis, *Proceedings of Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 529–533 (2011).
- [11] Harashima, J., Michiaki, A., Kenta, M. and Masayuki, I.: A Large-scale Recipe and Meal Data Collection as Infrastructure for Food Research, *Proceedings of Language Resources and Evaluation Conference*, pp. 2455–2459 (2016).
- [12] Zeiler, M. D.: ADADELTA: An Adaptive Learning Rate Method, *CoRR* (2012).
- [13] Mikolov, T., Chen, K., Corrado, G. and Dean, J.: Efficient Estimation of Word Representations in Vector Space, *Proceedings of International Conference on Learning Representations* (2013).
- [14] Rehurek, R. and Sojka, P.: Software Framework for Topic Modelling with Large Corpora, *Proceedings of LREC Workshop on New Challenges for NLP Frameworks*, pp. 45–50 (2010).