文字分散表現に基づく辞書情報を活用した固有表現抽出器の学習に関する研究

筑波大学
図書館情報メディア研究科

2019年03月

平松　淳

目 次

第 1 章　序章 **1**

第 2 章　関連研究 **6**
2.1　固有表現抽出 6
2.2　ニューラルネットワークを用いた固有表現抽出 7
2.3　レシピテキスト 8

第 3 章　ベースライン **10**
3.1　LSTM 10
3.2　CRF 11
3.3　LSTM-CRF 12
3.4　Lample らの LSTM-CRF 12

第 4 章　提案手法 **14**
4.1　単語の分類器 14
4.2　固有表現抽出器 14
4.3　転移学習 15

第 5 章　実験データ **19**
5.1　ラベル付きコーパス 19
5.2　ラベルなしコーパス 19
5.2.1　Wikipedia コーパス 20
5.2.2　クックパッドコーパス 21
5.3　単語分類器の教師データ 21

第 6 章　実験 **23**
6.1　比較手法 23
6.2　重みの固定 24
6.3　評価指標 25
6.4　実験結果 25
6.4.1　単語分類の実験結果 25
6.4.2　固有表現抽出の実験結果 25

第 7 章　結論 **31**

謝辞 **32**

参考文献 **33**

図 目 次

1.1 アノテーションフォーマットの違い 2
1.2 固有表現抽出の例 3
1.3 部分アノテーションの例 4
1.4 辞書の例 5

3.1 LSTM のセルの概略. 10
3.2 CRF のグラフィカルモデル. 11
3.3 LSTM-CRF のネットワーク図. 12
3.4 Lample ら [1] の単語特徴抽出モジュールの概略図. 13

4.1 本研究で用いた単語分類器の概略図. 15
4.2 分類器の出力を Bi-LSTM の入力に用いたネットワーク 17
4.3 分類器の出力を Bi-LSTM の出力に結合したネットワーク 17
4.4 転移学習の概略 18

6.1 単語分類の性能評価の例. 28
6.2 固有表現抽出の性能評価の例. 28
6.3 教師データの数と分類性能の関係 29

表 目 次

5.1 r-NE コーパスに含まれる固有表現 19
5.2 r-NE コーパスの統計量 20
5.3 クックパッドコーパスの統計量 20
5.4 Wikipedia コーパスの統計量 20
5.5 単語分類に用いる教師データ 21

6.1 ニューラルネットワークのハイパーパラメータの一覧 27
6.2 単語の分類性能 27
6.3 固有表現抽出器の抽出性能の比較 30
6.4 提案手法のラベルごとの予測性能 30

第1章　序章

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 [2, 3]．固有表現抽出では，テキストの中に含まれる人名，組織名，場所名などを固有表現と呼び，これらの自動的な抽出を行う．現在行われている固有表現抽出に関する研究の多くは新聞記事のテキストを用語の抽出対象としている．しかし，現実世界に存在している様々なドメインにはそれぞれ固有の用語が存在し，それらの自動抽出は重要なタスクである [4, 5, 6]．用語の認識は基礎的な言語解析であり，様々な応用技術の基盤となる．例えば，本研究で固有表現抽出の対象とするレシピテキストでは，レシピ中に出現する食材名や調理器具名，また調理操作名などを認識し，認識結果を用いてレシピを機械可読なグラフ構造に変換する研究が存在する [7]．

固有表現抽出は，一般的には教師ありの系列ラベリング問題として定式化されることが多い．系列ラベリング問題とは，入力列を $\mathbf{X} = (x_1, x_2, \ldots, x_n)$ としたときに，ラベル系列 $\mathbf{Y} = (y_1, y_2, \ldots, y_n)$ を予測するタスクである．このとき，y_k は人名や地名，組織名などの固有表現もしくは固有表現の一部である．固有表現は複数の単語からなる可能性があるため，固有表現抽出を系列ラベリング問題として解く際には固有表現は固有表現タグと呼ばれるフォーマットのタグに変換される．固有表現タグの枠組みでは，フレーズを表現するために固有表現名に特殊な接頭辞を付与したタグを用意する．接頭辞のフォーマットは複数存在し，IOB フォーマット，BIO フォーマット，IEBES フォーマットが有名である．
図 1.1 に IOB フォーマット，BIO フォーマット，BIOES フォーマットの例を示す．IOB フォーマットでは，固有表現の開始単語の固有表現タグに I の接頭辞を付与する．ただし，同じカテゴリの固有表現が出現した場合には 2 つ目の固有表現の先頭の固有表現タグに B の接頭辞を付与する．BIO フォーマットは固有表現の開始単語には B の接頭辞，以降の単語には I の接頭辞を付与する．BIOES フォーマットでは，1 単語からなる固有表現には，固有表現タグに S の接頭辞を付与する．また，複数単語からなる固有表現に対しては，開始単語には B，中間単語には I，終了単語には E の接頭辞をそれぞれ付与する．固有表現抽出の研究では BIO フォーマットもしくは BIOES フォーマットが採用されることが多いが，本研究では笹田ら [6] に従い BIO フォーマットを採用する．

例として，「関西国際空港からフランクフルト空港へ向かう」という文を考える．この文は，単語分割の結果，

$$\mathbf{X} = (\text{関西}, \text{国際}, \text{空港}, \text{から}, \text{フランクフルト}, \text{空港}, \text{へ}, \text{向かう}) \tag{1.1}$$

という単語列となる．この単語列に対しては図 1.2 のように固有表現タグが付与され，「関西国際空港」「フランクフルト空港」の 2 つの固有表現が得られる．このときのタグ系列 $\mathbf{Y}$ は以下のようになる．

$$\mathbf{Y} = (\text{B-LOC}, \text{I-LOC}, \text{I-LOC}, \text{O}, \text{B-LOC}, \text{I-LOC}, \text{O}, \text{O}) \tag{1.2}$$

系列ラベリング問題の教師データは，図 1.2 のように文に含まれるすべての単語に対して固有表現タグが付与されていることが一般的である．しかし，この教師データを作成するアノテーションコストは文の極性分析や文書分類のタスクの教師データを作成するアノ

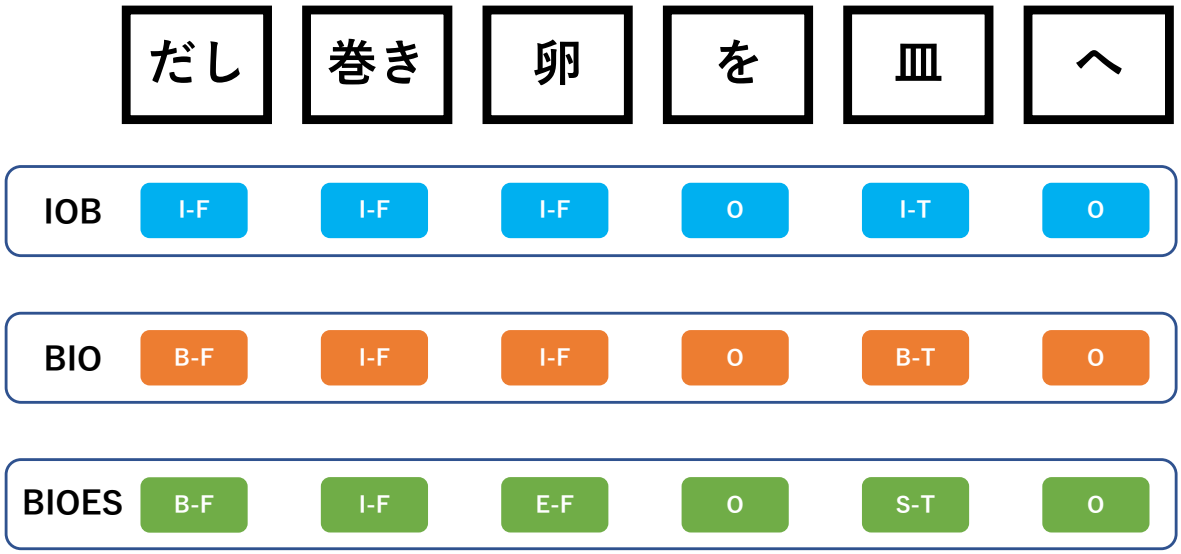

図 1.1: アノテーションフォーマットの違い

テーションコストと比較して高い．なぜなら，文の極性分析や文書分類は一文に対して 1 つのラベルを付与する一方で，系列ラベリングの教師データは入力単語列のすべてにラベルを付与する必要があるためである．また，固有表現抽出のアノテーションガイドラインは文書分類のガイドラインと比較して複雑になる．例えば，図 1.2 の例のとき，2 箇所の「空港」という単語に場所のタグを付与するか,「関西国際空港」および「フランクフルト空港」という単語列に場所のタグを付与するかを決定するのは自明ではない．このような例が与えられたとき，アノテーションガイドラインが存在しないと，アノテーション作業を行うアノテータごとに異なるアノテーションを付与する可能性がある．このため，アノテーションの一貫性を保つためにガイドラインが必要となる．アノテータは，定められたアノテーションガイドラインに準拠してアノテーション作業を行うことを要求される．しかし，これは時として専門的な知識が必要になることがある．さらに，固有表現はそのドメインに特化した用語であることが多い．ドメイン知識を持たない場合にはそのような用語に気づけず，アノテーションの付与に失敗してしまう可能性がある．このため，固有表現抽出の対象となるテキストによってはドメイン知識を持ったアノテータに作業を依頼する必要がある．このような専門知識を持ったアノテータは限られており，アノテーション作業のコストも高い．

このようなアノテーション作業の困難に対して，文中の単語の一部に対してのみ固有表現タグを付与する部分アノテーションと呼ばれるアノテーション方式が存在する [8, 9]．文全体に対してアノテーションをすることは困難であったとしても，文中の単語の中にはアノテーションが容易な単語も存在する．例として，レシピドメインのテキストに対して，「食材」を表すタグを付与するタスクを考える.「水溶き片栗粉と砂糖を鍋に加える」という文について考えると，この文は

$$\mathbf{X} = (\text{水溶き}, \text{片栗粉}, \text{と}, \text{砂糖}, \text{を}, \text{鍋}, \text{に}, \text{加える}) \tag{1.3}$$

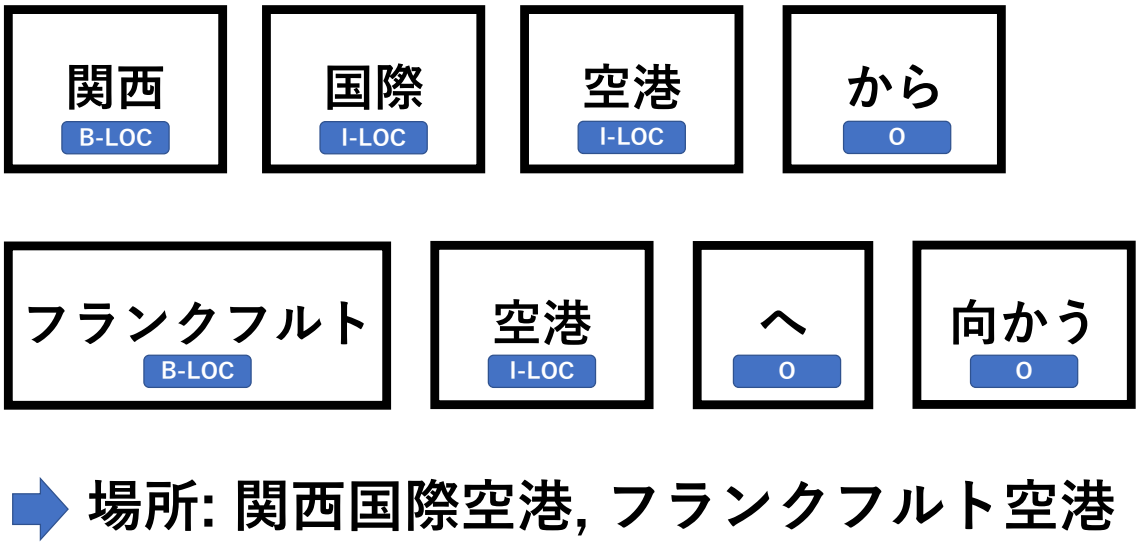

図 1.2: 固有表現抽出の例

という単語列に分割できる．このとき,「水溶き片栗粉」に対して食材のタグを付与するか,単に「片栗粉」に対して食材のタグを付与するかを判断する必要がある．この作業はアノテーションガイドラインを注意深く読むことで可能となる．しかし，多くの場合アノテーションの専門家ではないワーカがアノテーション作業を行うため，このような判断を適切に行えないことがある．一方で,「砂糖」に対して食材の固有表現タグを,「鍋」に対して調理器具の固有表現タグを付与すればよいことは容易にわかる．このようなとき，部分アノテーションの考え方では，図 1.3 のようにアノテーション作業が難しい箇所についてはアノテーションをせず，確信が持てる単語にのみアノテーションを付与する操作が許されている．

部分アノテーションはアノテーション作業のコストを削減し，作業をより容易なものにした点で重要である．しかし，依然として部分アノテーションにおいても系列データに対するアノテーション作業が必要がある．系列全体を考慮したアノテーション作業は単語をいくつかのカテゴリに分類させるような作業と比較してコストが高い．一方で，ある固有表現のカテゴリを与えて，ワーカにこのカテゴリに属する単語を列挙させるタスクは系列全体を考慮する必要がないという意味において固有表現のアノテーション作業と比較してコストが低いタスクである．例えば,「調理器具」のカテゴリに属する固有表現を列挙するタスクでは，文を読まずに作業が可能である．また，列挙が難しい場合，あらかじめ用意した単語をワーカに提示し，その単語がどのカテゴリに属するかを回答してもらう簡単なタスクでも構築可能である．本研究では，単語とその単語のカテゴリのペアで構成されたエントリを収集したデータを辞書と呼ぶ．例として，レシピドメインの辞書を図 1.4 に示す．料理レシピで用いられる用語が収集されていることがわかる．辞書を整備するコストは固有表現アノテーションが付与されたコーパスを構築するコストと比較して低く，さらに辞書は様々なドメインにおいてすでに整備されていることが多い．このため，このような辞書資源を活用することは固有表現タグが付与されたコーパスが十分な規模で存在しな

図 1.3: 部分アノテーションの例

いドメインで固有表現抽出を行うために重要である．このような辞書を活用する手法 [10, 11] は存在するが，これらの研究では辞書に含まれない単語に対する辞書特徴の構築法について改善できる点がある．

本研究では，レシピオントロジーデータ [12] を加工して辞書を構築する．次に，得られた辞書を教師データとした単語分類器を学習する．単語分類器の予測結果を固有表現抽出器の特徴量に組み込み，辞書情報を考慮した予測が可能な手法を提案する．提案手法で用いる単語分類器は単語を文字単位に分割し，文字情報を特徴量としてカテゴリを予測する．このため，辞書中には含まれない単語に対して辞書中で得られた部分文字列の情報を活用した予測が可能である．レシピ固有表現抽出 [6] のタスクにおける先行研究との比較実験を行い，提案手法の有用性を示す．

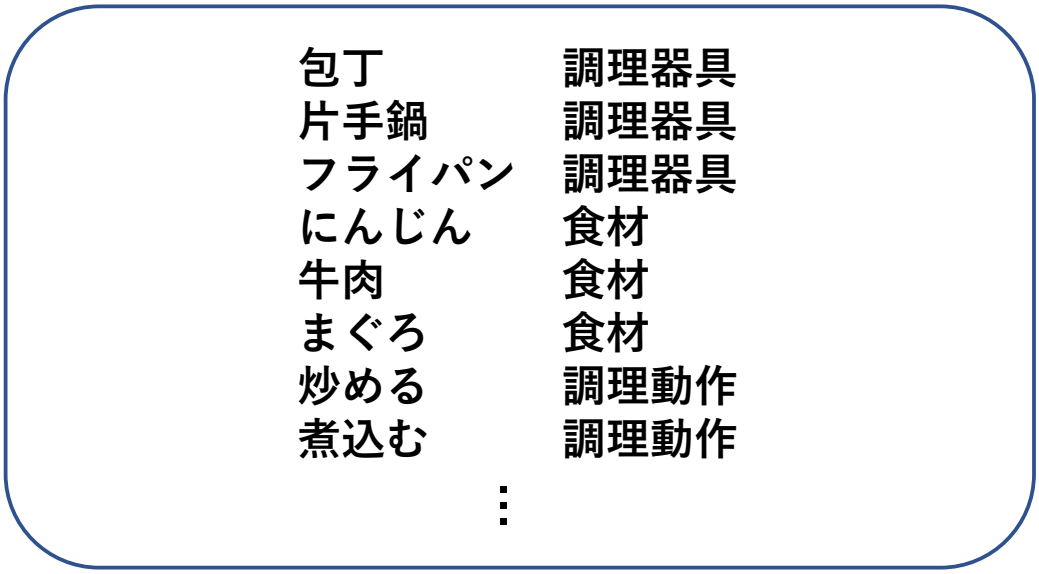
包丁 調理器具
片手鍋 調理器具
フライパン 調理器具
にんじん 食材
牛肉 食材
まぐろ 食材
炒める 調理動作
煮込む 調理動作

図 1.4: 辞書の例

第2章　関連研究

2.1　固有表現抽出

固有表現抽出は MUC (Message Understanding Conference) [13, 14] ではじめに提起されたタスクである．当初は情報抽出のためのタスクであり，MUC は新聞記事からの情報抽出を目的としていた．新聞記事からの情報抽出は現在においても重要なタスクであり，英語の固有表現抽出の研究でもっとも用いられるデータの 1 つである CoNLL 2003 データセット [15] も新聞記事に対して固有表現タグを付与している．しかし，固有表現抽出はニュースドメインだけでなく様々なドメインにおいて重要なタスクである．例えば，Ontonotes データセット [16] や ACE 2005 データセット [17] はニュース記事だけでなく，ウェブ上に存在する様々なテキストに対して固有表現タグを付与している．日本語では，IREX の CRL 固有表現データが存在する [18]．CRL 固有表現データは，新聞 (毎日新聞) の記事に対して人名や組織名などの固有表現タグを付与したコーパスである．

新聞記事やウェブ上の文書の他にも，バイオインフォマティクスの分野においても固有表現抽出の研究が活発に行われている [19, 4, 16, 5, 20, 21]．固有表現タグは，固有表現の抽出対象となるテキストのドメインによって異なり，バイオインフォマティクスのドメインでは病名，タンパク質名，DNA や病名などを固有表現として固有表現タグが付与されている．この他にも，レシピテキスト [6] や将棋の解説文 [22] のような現実世界の様々なドメインで固有表現の抽出が行われている．

固有表現抽出手法は，大きく分けると

1. 辞書情報と統語パターンを用いたルールベースの手法
2. 教師なし学習による手法
3. 教師あり学習による手法

の 3 種類に分類できる．ルールを用いた手法 [23] では，ドメイン固有の辞書情報と統語パターンを用いて固有表現を抽出する．統語パターンを活用したルールベースの手法は，精度の高い抽出が可能である一方再現率が辞書の網羅率に依存し，再現率が低くなることが多い．しかし，すべての固有表現を網羅する辞書を構築することは現実的ではない．この課題に対し，統計的な手法を用いて辞書や統語パターンを学習するアプローチが現れた．

教師なし学習による手法では，統語パターンなどのルールに加え，大規模コーパスを用いて計算した統計量を用いて固有表現を抽出する．Etzioni ら [24] は Web 上の文書を用いて計算した自己相互情報量 (Pointwise Mutual Information; PMI) を用いて固有表現抽出の再現率が向上したことを報告している．

ルールベースの手法と教師なし学習による手法では，統語パターンなどの専門家が定めたルールを活用する．教師あり学習による手法では，このようなルールも教師データから

自動的に学習することを目指す．教師あり固有表現抽出は系列ラベリング問題として定式化されることが多い．機械学習ベースの固有表現抽出器は，ルールベースの手法や教師なし学習による手法と比較して，同程度の精度を保ちつつ高い再現率で固有表現を抽出できる．さまざまなモデルを用いた手法が提案されており，例えば隠れマルコフモデル (Hidden Markov Model; HMM) を用いた手法 [25]，サポートベクトルマシン (Support Vector Machine; SVM) を用いた手法 [26]，ロジスティック回帰 (Logistic Regression; LR) を用いた手法 [8] や条件付き確率場 (Conditional Random Fields; CRF) を用いた手法 [27] などが例として挙げられる．HMM および CRF は系列全体の最適なラベル系列を推定するモデルであるが，SVM と LR は系列中のある単語の最適ラベルを独立に推定する．このように各単語について独立に最適なラベルを推定する方法は点推定と呼ばれる．

2.2 ニューラルネットワークを用いた固有表現抽出

また，近年はニューラルネットワークに基づく固有表現抽出が数多く提案されており，CoNLL 2003 データセットをはじめとした様々なデータセットにおいて高い性能を発揮している [28, 29, 1, 30]．ニューラルネットワークによる手法では，人手で設計された特徴量の代わりに単語の分散表現と呼ばれる特徴量が用いられる．単語の分散表現はニューラルネットワークのパラメータの一部である．このため，固有表現抽出器を学習する際に自動的に分散表現も最適化される．この結果，タスクに合った分散表現が得られるため，人手による特徴量の設計が不要である．Collobert ら [28] は既存の教師あり学習手法と同等の性能を発揮するニューラルネットワークを提案し，以降のニューラルネットワークに基づく固有表現抽出手法の先駆けとなっている．Collobert らは畳み込みニューラルネットワーク (Convolution Neural Networks; CNN) と CRF を組み合わせた手法を用いている．さらに，大規模なコーパスで学習した単語の分散表現をニューラルネットワークの単語の分散表現の初期値として利用することで固有表現の抽出性能の向上を実現している．Huang ら [29] は双方向 LSTM (Bidirectional LSTM; Bi-LSTM) と CRF を組み合わせた LSTM-CRF と呼ばれるニューラルネットワークを提案している．Huang らは人手で設計された特徴量と単語の分散表現の両方を特徴量として利用しており，これらの特徴量は同時に利用できることを示している．また， Huang らが提案した LSTM-CRF のアーキテクチャは近年の固有表現抽出器のスタンダードとなっている．Lample ら [1] や Ma ら [30] は単語レベルの特徴量だけでなく，文字レベルの情報も特徴量に加えた LSTM-CRF ベースのニューラルネットワークを提案している．Lample ら，Ma らは単語を更に分割して文字列を作成し，それら文字に対して分散表現を割り当てている．割り当てた分散表現をニューラルネットワークに入力し，得られた出力ベクトルを文字ベースの単語特徴量としている．この文字ベースの単語特徴量と単語の分散表現を結合し，LSTM-CRF のニューラルネットワークに入力することで固有表現抽出の性能を向上させている．Lample らと Ma らの手法の違いは文字分散表現を入力するニューラルネットワークの種類であり，Lample らの手法は Bi-LSTM を，Ma らは CNN を用いて文字特徴を抽出している．

文字分散表現の他にも，辞書情報が得られる場合にはその情報を活用することで固有表現抽出の性能を向上させることができる．辞書情報を活用するアプローチは古くから多く存在する [25, 31] が，ニューラルネットワークに基づく手法でも辞書情報を活用するアプローチが研究されている [29, 10, 11]．Huang ら [29] は単語が辞書に含まれるかどうかを表すベクトルを特徴量に追加している．また，Pham ら [11] は単語が辞書中の各カテゴリに含まれる確率を要素に持つベクトルを特徴量として用いている．Huang らや Pham らは単語レベルで辞書情報を活用する手法を提案している．これに対し，Sato ら [10] は複合語に

対しても辞書情報を活用する手法を提案している．Sato らは単語列を Ma らの LSTM-CNN-CRF に入力し，得られた結果を用いてラティスを構築する．このとき，ラティス上では複数の単語が 1 つのノードとなりうる．Sato らはこのノードが辞書中に含まれているかどうかを表す特徴量を割り当てている．これにより，辞書中の複合語の情報を考慮することに成功している．

既存手法では，複合語に辞書特徴量を割り当てることは可能である一方，辞書の単語の分割単位が教師データ中の単語の分割単位が一致しない場合に辞書特徴量を活用できない問題が発生する．日本語は単語間に明示的な境界が存在しない言語である．文字列をどのように単語列に分割するかは形態素の定義により異なる．このため，ある形態素解析器を用いて解析を行ったとき 1 形態素となる文字列が別の形態素解析器を用いた場合複数の形態素に分割される可能性がある．例として,「出刃包丁」という文字列について考える．この文字列は「出刃包丁」という 1 つの形態素とみなされる場合と「出刃 包丁」という 2 つの形態素とみなされる場合がある．前者のような解析を行う解析器を解析器 A，後者を解析器 B とする．さらに，解析器 B の出力に基づいて「包丁」という文字列が辞書に登録されると仮定する．このとき，固有表現抽出の教師データが解析器 A によって単語分割されている場合，辞書中の「包丁」というエントリの情報は活用できない．

この問題に対して，本研究では文字分散表現を Bi-LSTM に入力し，出力ベクトルを 1 層のパーセプトロンに入力してカテゴリ数次元のベクトルを得る．次に，得られたベクトルにソフトマックス関数を適用し，辞書中のカテゴリを予測する分類器を構築する．このとき，分類器は単語特徴量を文字特徴量から抽出しているため,「出刃包丁」という単語のカテゴリを予測する際に辞書中の「包丁」というエントリの情報を活用できる．得られたベクトルは単語の辞書中で定義されたどのカテゴリにどの程度の確率で帰属するかを表す確率ベクトルとなっている．得られた確率ベクトルを固有表現抽出器の特徴量に追加することで，分類情報を固有表現抽出で活用できる．

2.3 レシピテキスト

本研究では，提案手法の学習および評価のためにレシピドメインのテキストを利用する．近年，人々は自身が作成した料理のレシピ情報をインターネット上にアップロードしており，多くの研究者がアップロードされたレシピ情報を用いた研究を行っている．前田ら [7] はレシピテキストに対して依存構造解析を行い，計算機可読なグラフ構造に変換する手法を提案している．Salvador ら [32] は Recipe 1M データセット [33] と呼ばれるデータを用いて料理画像の画像特徴量とレシピテキストの言語特徴量を同一の空間へと写像し，マルチモーダルな分散表現を学習している．Salvador らは，この得られた分散表現を用いて，料理画像からレシピを検索する手法を提案している．佐藤ら [34] は日本語のレシピテキストを英語に翻訳するタスクに取り組み，エラー分析を通してレシピに特有な用語に関する分析やフレーズベースの機械翻訳とニューラルネットワークを用いた機械翻訳の比較を行っている．牛久ら [35] は料理動画からレシピテキストを自動生成する手法を提案している．レシピテキストの生成の際には固有表現抽出の技術が活用されている．また，レシピ質問応答 [36] のようなより応用的なトピックの研究も存在する．

レシピテキストを扱う，より基礎的な言語解析に関する研究としてレシピ固有表現抽出が存在する．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 [6]．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た抽出手法を提案している．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，ラベル系列の最適化を行っている．笹田らの手法は，ラベル系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．レシピ固有表現抽出のタスクはレシピの機械可読な形式への変換 [7] や料理動画からのレシピテキスト自動生成 [35] などに応用されている．

第3章　ベースライン

3.1　LSTM

LSTM [37] とは，式 3.1 から式 3.5 で定義されるニューラルネットワークである．

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \mathbf{h}_{t-1} + \mathbf{U}_f \mathbf{x_t} + \mathbf{b}_f), \tag{3.1}$$
$$\mathbf{i}_t = \sigma(\mathbf{W}_i \mathbf{h}_{t-1} + \mathbf{U}_i \mathbf{x_t} + \mathbf{b}_i), \tag{3.2}$$
$$\tilde{\mathbf{c}}_t = \tanh(\mathbf{W}_c \mathbf{h}_{t-1}) + \mathbf{U}_c \mathbf{x}_t + \mathbf{b}_c, \tag{3.3}$$
$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t, \tag{3.4}$$
$$\mathbf{o}_t = \sigma(\mathbf{W}_o \mathbf{h}_{t-1} + \mathbf{U}_o \mathbf{x}_t + \mathbf{b}_o), \tag{3.5}$$
$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t). \tag{3.6}$$

図 3.1 に LSTM のセルを示す．入力ゲート，忘却ゲート，出力ゲートと呼ばれる 3 つのゲートを持ち，これによって過去の情報をどの程度未来へ伝えるかを制御できる．

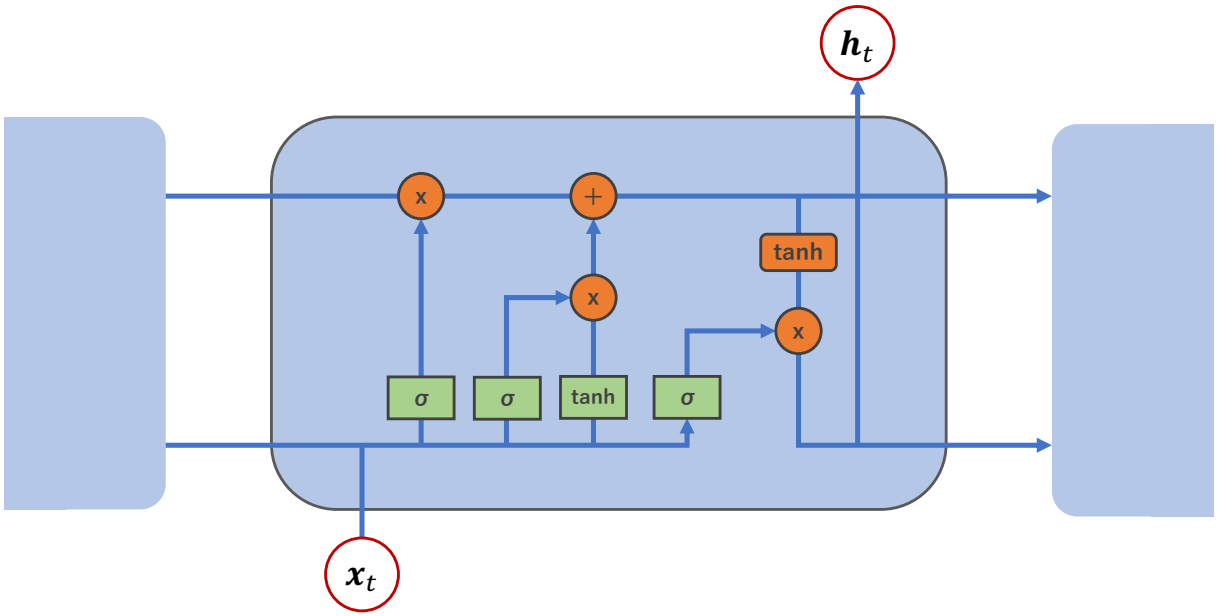

図 3.1: LSTM のセルの概略．

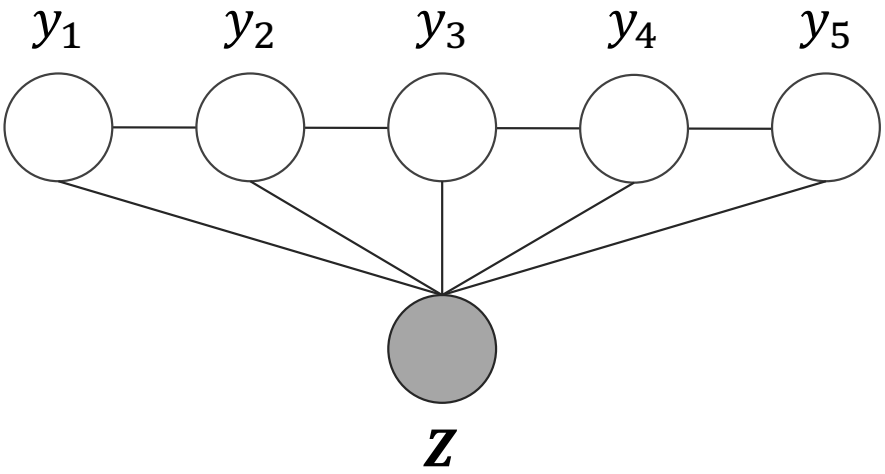

図 3.2: CRF のグラフィカルモデル．

3.2 CRF

CRF [38] は系列ラベリングを行う研究で広く用いられている確率モデルである．CRF は系列が入力されたときに系列の潜在変数を推定する．CRF では，出力系列となる確率変数の間に依存関係があり，これを無向グラフで表現している．変数間の依存関係をグラフィカルモデルで表現したものを図 3.2 に示す．入力系列の素性ベクトルを

$$\mathbf{Z} = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n), \tag{3.7}$$

として，ラベル系列 $\mathbf{y}$ の確率を，

$$P(\mathbf{y} \mid \mathbf{Z}; \mathbf{W}, \mathbf{b}) = \frac{\prod_{i=1}^{n} \psi_i(y_{i-1}, y_i, \mathbf{Z})}{\sum_{y' \in \mathcal{Y}(\mathbf{Z})} \prod_{i=1}^{n} \psi_i(y'_{i-1}, y'_i, \mathbf{Z})}, \tag{3.8}$$

のように定義する．このとき，ψ_i は以下のように定義される．

$$\psi_i(y', y, \mathbf{Z}) = \exp(W^T_{y',y}\mathbf{z}_i + \mathbf{b}_{y',y}). \tag{3.9}$$

$\mathbf{W}$ と $\mathbf{b}$ はラベルの遷移のパラメータであり，ラベル y' からラベル y にどの程度遷移するかを制御する．最適なラベル系列 $\tilde{\mathbf{y}}$ はこの確率 $P(\mathbf{y} \mid \mathbf{Z}; \mathbf{W}, \mathbf{b})$ を最大化するような $\mathbf{y}$ であり，

$$\tilde{\mathbf{y}} = \underset{\mathbf{y} \in \mathcal{Y}(\mathbf{Z})}{\operatorname{argmax}} P(\mathbf{y} \mid \mathbf{Z}; \mathbf{W}, \mathbf{b}), \tag{3.10}$$

と表せる．このようなラベル系列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．

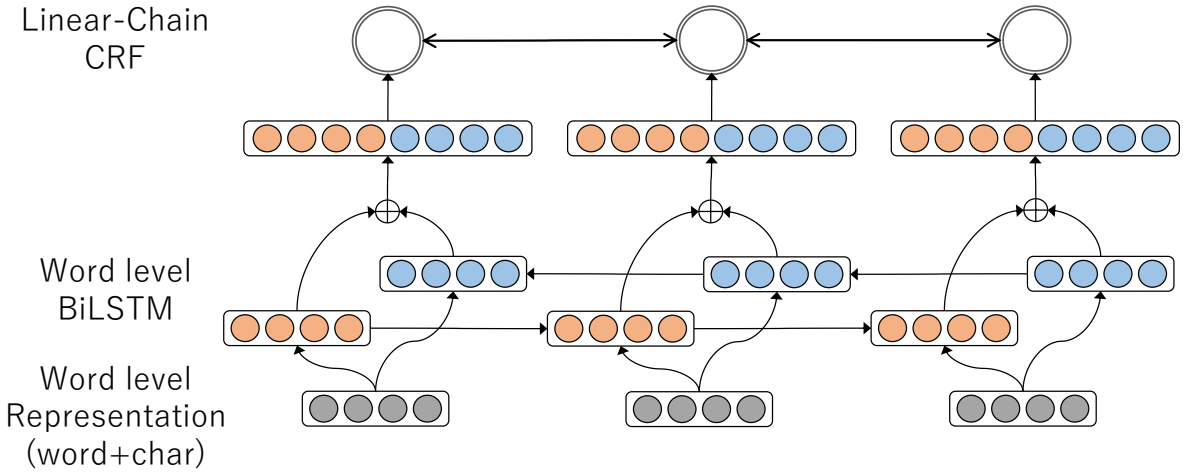

図 3.3: LSTM-CRF のネットワーク図.

3.3 LSTM-CRF

提案手法は Lample ら [1] の手法の拡張となっている．このため，本章では Lample らの手法について詳細に解説する．Lample らの手法は，単語の特徴量を LSTM-CRF と呼ばれるニューラルネットワークに入力する．LSTM-CRF は Huang ら [29] によって提案されたニューラルネットワークである．LSTM-CRF のアーキテクチャを図 3.3 に示す．
LSTM-CRF では，まず特徴量を LSTM に入力し出力ベクトルを得る．次に，得られた出力ベクトルを CRF に入力し，最適なラベル系列を予測する．Huang らは実際には Bi-LSTM と呼ばれる，通常の LSTM に加えて系列を逆順に入力する LSTM を組み合わせた Bi-LSTM と呼ばれるニューラルネットワークを用いている．本稿では，簡単のために入力系列が得られたときに隠れ層の重み $\mathbf{h}$ を出力する関数 Bi-LSTM を定義する．

LSTM-CRF では，LSTM の出力ベクトルの次元とラベルの種類数が異なることがある．このため，LSTM の出力ベクトル $\mathbf{h}_t$ を，

$$\mathbf{z}_t = \mathbf{W}\mathbf{h}_t + \mathbf{b}, \tag{3.11}$$

のように変換する必要がある．ここで，$\mathbf{W}$ は (ラベルの種類数) × (隠れ層の次元数) の重み行列であり，$\mathbf{h}_t$ をラベル数次元のベクトル $\mathbf{z}_t$ に変換する．

3.4 Lample らの LSTM-CRF

Huang ら [29] は LSTM に入力する特徴量として，ニューラルネットワークで一般的に用いられる単語の分散表現に加え，従来の固有表現抽出器で用いられていた単語の表層的な特徴量などを用いた．これに対し，Lample らの手法はルールに基づく素性をまったく使わないアプローチをとっている．Lample らの手法では，単語の分散表現だけでなく，文字分

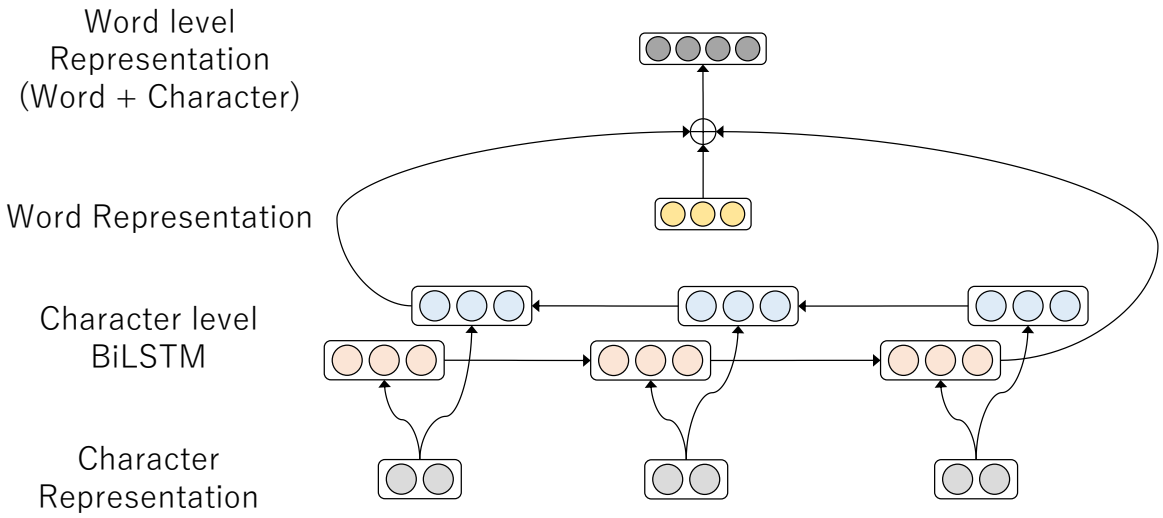

図 3.4: Lample ら [1] の単語特徴抽出モジュールの概略図.

散表現から構成する単語の特徴量も単語レベルの特徴量として採用している．Lample らの文字ベース特徴量は図 3.4 に示すニューラルネットワークによって構成される．単語列を $\mathbf{X} = (x_1, x_2, \ldots, x_n)$，単語列の t 番目の単語に含まれる文字の分散表現を $\mathbf{C}_t = (\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_m)$，ラベル系列を $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ とする．x_t および y_t は離散的な記号であり，$\mathbf{c}_t$ は分散表現であることに注意されたい．このとき，文字分散表現から抽出した単語レベルの特徴量は

$$\mathbf{w}_t^{(char)} = \text{Bi-LSTM}^{(char)}(\mathbf{C}_t), \tag{3.12}$$

と表せる．さらに，得られた文字ベースの単語レベル特徴量と単語の分散表現を

$$\mathbf{x}_t = [\mathbf{w}_t; \mathbf{w}_t^{(char)}], \tag{3.13}$$

のように結合し， LSTM-CRF に入力する単語特徴量を得る．ここで，$\mathbf{w_t}$ は x_t の単語の分散表現である．この操作を文中のすべての単語に対して行い，文全体の特徴量

$$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n), \tag{3.14}$$

を構築する．最後に，得られた単語特徴量を Bi-LSTM に入力し，以下の式のように隠れ状態を得る．

$$\mathbf{h}_t = \text{Bi-LSTM}(\mathbf{X})_t. \tag{3.15}$$

隠れ状態が得られれば，Huang らの LSTM-CRF と同様に Viterbi アルゴリズムを用いて最適なラベル系列を推定できる．

第4章　提案手法

本研究では，Lample ら [1] の手法を拡張し，単語の辞書的な情報を文字ベースで活用する手法を提案する．本章では，提案手法について

1. 文字ベースのニューラルネットワークによる単語分類器
2. 単語分類結果を活用したニューラルネットワークによる固有表現抽出器

の２つに分割して解説を行う．

4.1　単語の分類器

提案手法で用いる単語の分類器を図 4.1 に示す．単語分類器は，単語を文字レベルに分割し，分割した文字の分散表現を Bi-LSTM に入力する．この操作により，単語に対して文字レベルで構成した 1 つのベクトルが得られる．単語の分類器で用いている Bi-LSTM は Stacked Bi-LSTM と呼ばれるニューラルネットワークである．Stacked Bi-LSTM では，特徴量は異なる k 種類の Bi-LSTM に連続して入力される．これにより，中間層のパラメータに自由度が生まれ，モデルの表現力が向上する．得られたベクトルを全結合層に入力し，辞書のカテゴリ数の次元のベクトルに変換する．カテゴリ数次元のベクトルに対してソフトマックス関数を適用することで，単語が辞書の中のどのカテゴリに属するかを表すベクトルが得られる．入力文の t 番目の単語の辞書ベクトルを $\mathbf{a}_t$ とすると，$\mathbf{a}_t$ は

$$\mathbf{h}^{(\mathbf{classifier})}{}_t = \text{Stacked Bi-LSTM}(\mathbf{C}_t), \tag{4.1}$$

$$\mathbf{z}_t = \mathbf{W}\mathbf{h}^{(classifier)}{}_t + \mathbf{b}, \tag{4.2}$$

$$\mathbf{a}_t = \text{Softmax}(\mathbf{z}_t), \tag{4.3}$$

と定義される．ここで，$\mathbf{C}_t$ は図 3 で定義した入力文の t 番目の単語の文字の分散表現のリストである．また，ここでの Bi-LSTM は文字列全体に対する特徴量を抽出する必要があるため，図 4.1 に示すように順方向 LSTM の最終時刻の出力と逆方向 LSTM の最終時刻の出力を結合したベクトルが出力となり，図 3.3 に示す Bi-LSTM とは異なる．

4.2　固有表現抽出器

本研究では，第 4.1 節で定義した辞書ベクトルを固有表現抽出器の特徴量として用いる．本研究で用いる辞書とは，単語とその単語が属するカテゴリからなるエントリを含むデータであり，分類器は単語が入力された際にその単語が帰属するカテゴリを予測する．分類器は単語の各カテゴリに対する帰属度を確率ベクトルの形で出力するため，Huang ら [29]

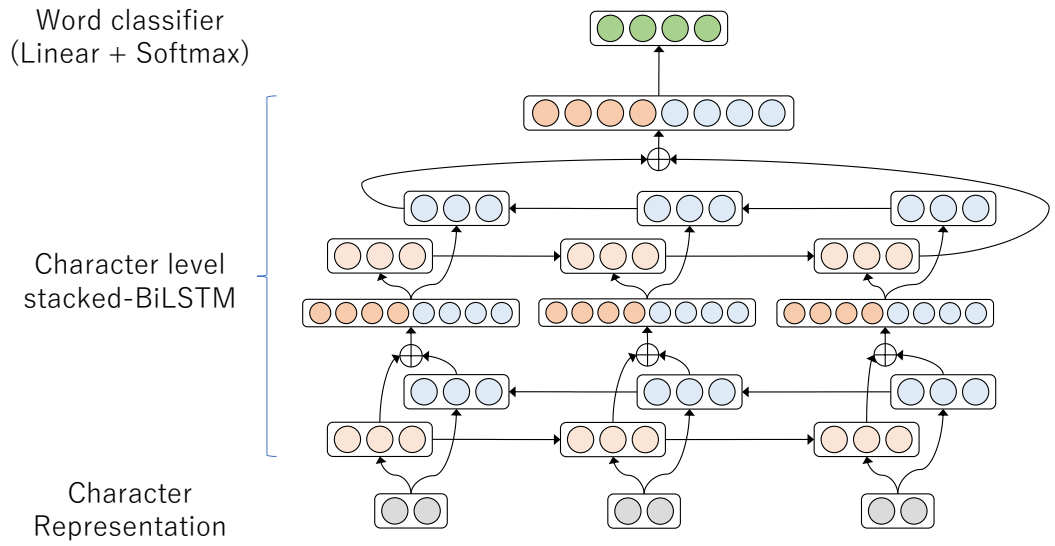

図 4.1: 本研究で用いた単語分類器の概略図.

や Sato ら [10] の手法と比較したときにカテゴリの曖昧性を考慮した出力が可能である．また，辞書ベクトルは文字レベルの分散表現を用いて構築される．このため，Pham ら [11] の手法では辞書中に出現しない単語に対する辞書ベクトルはゼロベクトルになってしまう一方，提案手法は辞書中の部分文字列の情報を考慮したベクトルを出力できる．さらに，提案手法は文字レベルで単語を解釈するため，辞書中の単語と教師データ中の単語の分割単位が異なる場合にも辞書を考慮した出力ができる．辞書中に含まれるカテゴリ情報は単語の分散表現と比較すると解釈性が高く，モデルのエラーの分析が容易になるなどの効果も期待できる．提案手法が利用する辞書を構築するコストは固有表現のアノテーションつきコーパスを構築するコストと比較して小さく，様々なドメインで適用が可能である．

本研究では，辞書ベクトルを 2 種類のアプローチで固有表現抽出器に取り入れる．図 4.2 は辞書ベクトルを Bi-LSTM に入力する特徴量に追加するアプローチを採用した手法である．この手法では，辞書情報は Bi-LSTM によって単語列全体に伝播する．図 4.3 は辞書ベクトルを CRF の入力に追加するアプローチを採用した手法である．この手法では，辞書ベクトルの情報は Bi-LSTM には入力されない．このため，提案手法 1 が辞書情報を文脈情報として活用する一方，提案手法 2 は辞書情報を単語レベルで活用していると解釈できる．

4.3 転移学習

提案手法では，まず単語分類タスクを解き，得られた分類器を固有表現抽出タスクに流用する．このようなアプローチは転移学習と呼ばれ，画像解析の分野で広く行われているが，自然言語処理においてもその有用性が報告されている [39, 40, 41]．転移学習の概略を図 4.4 に示す．転移学習では，学習済みのモデルの重みは一部，もしくはすべてを固定する．どの重みを固定すればよいかはタスク依存であり，実験による検討が必要である．本研究では，

1. 分散表現の重みのみを固定する
2. 分散表現，Bi-LSTM の重みを固定する
3. 分散表現，Bi-LSTM，全結合層 (出力層) の重みを固定する

の３種類の方法による転移学習を行い，性能を比較する．

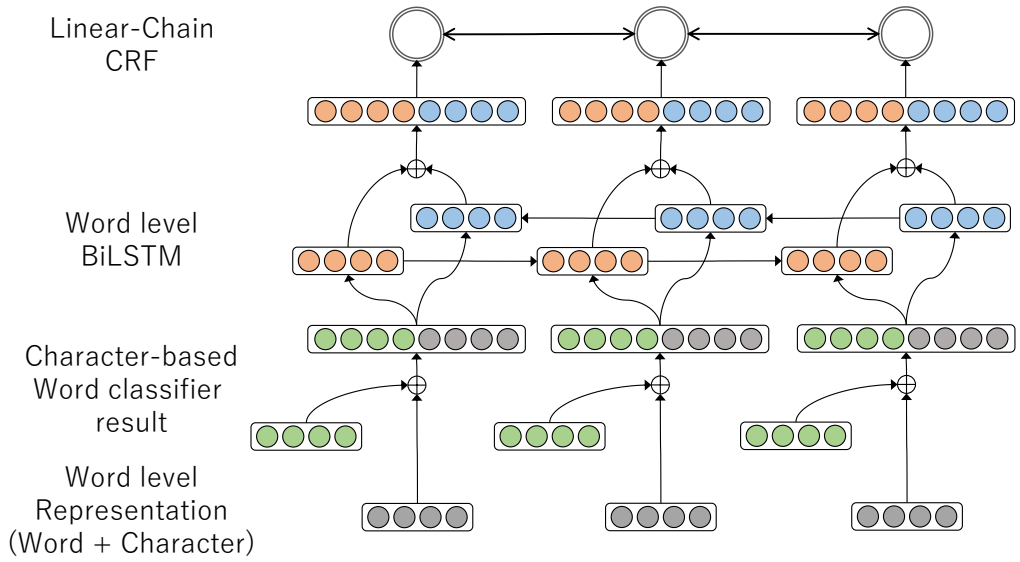

図 4.2: 分類器の出力を Bi-LSTM の入力に用いたネットワーク

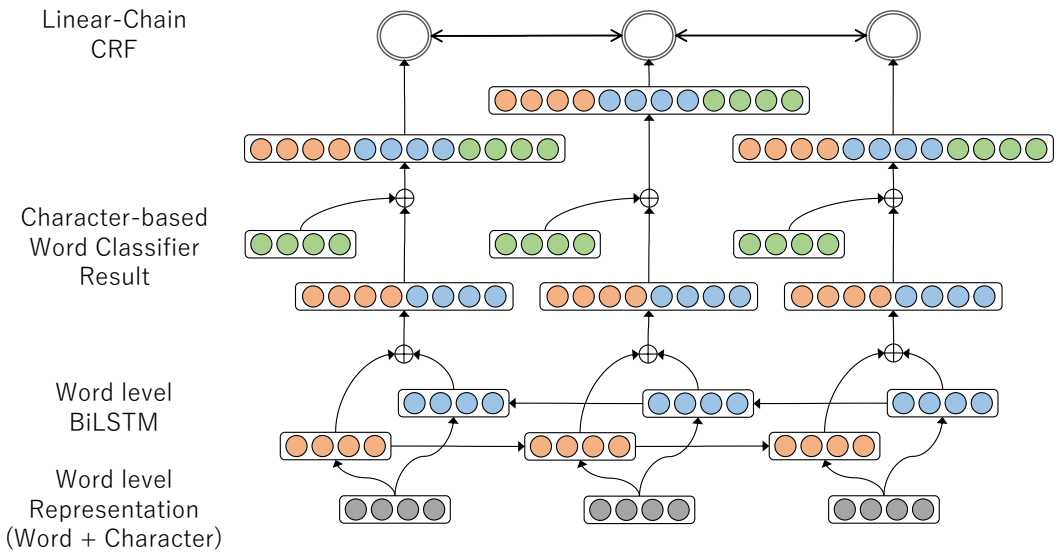

図 4.3: 分類器の出力を Bi-LSTM の出力に結合したネットワーク

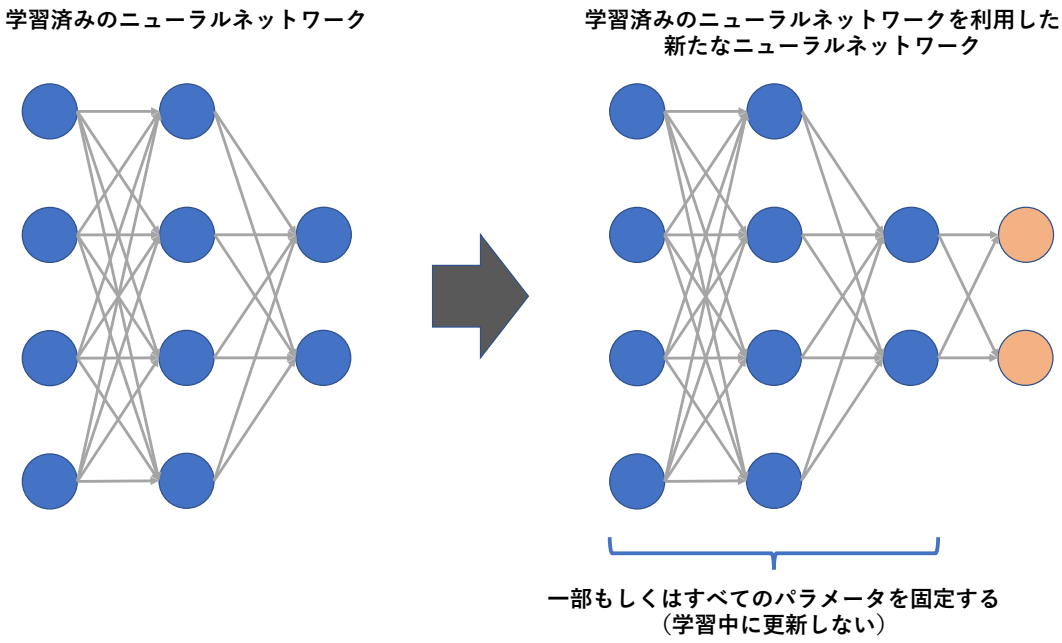
学習済みのニューラルネットワーク
学習済みのニューラルネットワークを利用した
新たなニューラルネットワーク
一部もしくはすべてのパラメータを固定する
（学習中に更新しない）

図 4.4: 転移学習の概略

第5章　実験データ

本節では，本研究で利用したコーパスについて解説する．本研究では，r-NE コーパス，クックパッドコーパス，Wikipedia コーパスの 3 種類のコーパスを利用した．これらのコーパスは，ラベル付きコーパスとラベルなしコーパスの 2 種類に区別できる．ラベル付きコーパス，ラベルなしコーパスそれぞれについて解説を行う．

5.1　ラベル付きコーパス

固有表現抽出器の学習には一般的にラベル付きコーパスが必要となる．本研究では，固有表現抽出器の学習と評価のために r-NE コーパス [6] を利用した．r-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 [8] を用いて単語単位に分割されている．r-NE コーパスに含まれる固有表現の種類および出現頻度を表 5.1 に示す．食材（「さつまいも」「じゃがいも」など）と調理者の動作（「切る」「炒める」など）が最も多く含まれることがわかる．r-NE コーパスの統計情報を表 5.2 に示す．第 5.2 項に示すラベルなしコーパスと比較すると小規模なコーパスであることがわかる．

5.2　ラベルなしコーパス

ニューラルネットワークを用いて固有表現抽出器を構築する際，学習済みの単語分散表現を抽出器の単語分散表現の初期値とすることで性能が向上することが知られている [1]．分散表現は教師ラベルが付与されていないコーパスが得られれば学習可能である．しかし，一般的に分散表現の品質を確保するためには大規模なコーパスが必要である．本研究では，

表 5.1: r-NE コーパスに含まれる固有表現

固有表現	説明	出現頻度
F	食材	6,282
T	道具	1,956
D	継続時間	409
Q	分量	404
Ac	調理者の動作	6,963
Af	食材の動作	1,251
Sf	食材の状態	1,758
St	道具の状態	216

[1] https://cookpad.com/

表 5.2: r-NE コーパスの統計量

文書数	436
文数	3,317
単語数	60,542
異なり単語数	3,390
文字数	91,560
異なり文字数	1,130

表 5.3: クックパッドコーパスの統計量

文書数	1,715,589
文数	12,659,170
単語数	216,248,517
異なり単語数	221,161

表 5.4: Wikipedia コーパスの統計量

文書数	1,114,896
文数	18,375,840
単語数	600,890,895
異なり単語数	2,306,396

大規模なラベルなしコーパスとして，２種類のコーパスを用意した．それぞれを Wikipedia コーパスとクックパッドコーパスと呼ぶ．固有表現抽出器の学習では文字特徴量を用いる一方，単語の分散表現の学習には文字特徴量は必要ない．このため，ラベル付きコーパスについては文字数および異なり文字数の情報を示し，ラベルなしコーパスについては文字数および異なり文字数の情報を示していない．

5.2.1 Wikipedia コーパス

オンライン百科事典 Wikipedia は，Wikipedia で使われているデータベースのダンプファイルを公開している[2]．ダンプファイルのうち，本文記事を含むデータ (2018 年 08 月 01 日のもの) をダウンロードした．このデータを解凍し，必要な前処理を行いコーパスを構築した．Wikipedia のダンプファイルに含まれる記事データは XML によってマークアップが行われている．まず，得られた XML データを Wikiextractor [3] を用いて JSON 形式に変換する．Wikipedia の記事本文データは Media Wiki 記法と呼ばれる特殊な記法でマークアップが行われているが，Wikiextractor によってこれらのマークアップを除去し，プレーンテキストに変換する．Wikiextractor は XML データを複数の JSON ファイルに分割し，分割されたそれぞれのファイルには複数の記事のデータが含まれる．次に， JSON ファイルから記事の本文フィールドを抜き出し，形態素解析を行い単語に分割する．形態素解析器には KyTea [8] を用いる．以上の前処理の結果得られたコーパスを Wikipedia コーパスと呼ぶ．Wikipedia コーパスの統計量を表 5.4 に示す．

[2] https://dumps.wikimedia.org/jawiki/
[3] https://github.com/attardi/wikiextractor

表 5.5: 単語分類に用いる教師データ

カテゴリ	データ数
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

5.2.2 クックパッドコーパス

クックパッドデータセット [42] 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，レシピサービスクックパッド [4]に投稿された献立やレシピのデータが含まれている．我々は，クックパッドデータセットからレシピの調理手順に関するデータを抽出した．このデータに対して KyTea を用いて形態素解析を行い，単語に分割した．得られたコーパス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量を表 5.3 に示す．

5.3 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 [12] 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 つの単語が複数のカテゴリに属することがある．本研究ではこのような単語は教師データから除外することとした．該当する単語は 4 単語であった．複数のカテゴリ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
- 複数カテゴリの組み合わせを 1 カテゴリとみなす

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カテゴリ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，r-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カテゴリ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」のカテゴリを付与し，料理オントロ

[4] https://cookpad.com/

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 種類に分割した．

第6章　実験

6.1　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 部分アノテーションコーパスを用いた学習が可能な笹田ら [9] の手法．
点推定を行っており，点推定のモデルはロジスティック回帰を採用した．
モデルの学習には PWNER ツールキット [1]を利用した．
特徴量として，文字 n-gram，文字種 n-gram，単語 n-gram などの
表層的な情報が用いられている．学習には liblinear [43] が採用されており，
パラメータの最適化にはニュートン法が採用されている．

LR+DP LR の出力に対して動的計画法を適用してラベル系列の最適化を行う手法．
点推定のモデルはロジスティック回帰を採用した．

Lample 第 3 章で解説した Lample ら [1] の LSTM-CRF を用いた手法．
単語レベルの特徴量を単語の分散表現と文字の分散表現を用いて抽出している．

Dictionary Lample らの手法に辞書特徴量をナイーブに組み合わせた手法．
入力系列の各単語について，単語が辞書に含まれる場合はそのカテゴリ情報を
1-of-k 符号化したベクトルを単語特徴量に追加する．
単語が辞書に含まれない場合，ゼロベクトルを単語特徴量に追加する．

Proposed1 第 4 章で解説した単語分類情報を BiLSTM に入力する手法．
図 4.2 の LSTM-CRF のニューラルネットワークである．

Proposed2 第 4 章で解説した単語分類情報を CRF に入力する手法．
図 4.3 の LSTM-CRF のニューラルネットワークである．

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 を適用してラベル系列を求める．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 Adam [43] を使用し，ミニバッチサイズを 10 とする．Adam のハイパーパラメータは $\alpha = 0.001$，$\beta_1 = 0.9$，$\beta_2 = 0.9$ 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．ニューラルネットワークの実装には Chainer [44] を用いる．

本研究では，単語の分散表現は，

Uniform $[\frac{-3}{\mathrm{dim}}, \frac{3}{\mathrm{dim}}]$ 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．
学習された分散表現の語彙に含まれない単語の分散表現は
$[\frac{-3}{\mathrm{dim}}, \frac{3}{\mathrm{dim}}]$ の範囲で一様サンプリングして初期化する．

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．
学習された分散表現の語彙に含まれない単語の分散表現は
$[\frac{-3}{\mathrm{dim}}, \frac{3}{\mathrm{dim}}]$ の範囲で一様サンプリングして初期化する．

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．分散表現は Skip-gram with Negative Sampling (SGNS) [45] 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[46] 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カテゴリ数次元に変換する．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カテゴリ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．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 Adam [43] を使用し，ミニバッチサイズを 10 とする．Adam のハイパーパラメータは $\alpha = 0.001$，$\beta_1 = 0.9$，$\beta_2 = 0.9$ 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．

固有表現抽出器のハイパーパラメータおよび単語の分類器のハイパーパラメータをまとめて表 6.1 に示す．

6.2 重みの固定

提案手法は単語分類タスクで学習したニューラルネットワークを固有表現抽出タスクで流用する，転移学習のアプローチを取っている．転移学習では，すでに学習を行ったニューラルネットワークの重みは学習中に更新しない，ファインチューニングと呼ばれる操作が行われることがある．本研究では，以下の 2 つの設定でファインチューニングを行い，性能を比較する．

LSTM 文字分散表現，Bi-LSTM の重みを学習中に更新しない．
単語分類器の重みについては全結合層の重みだけが更新される．

Linear 文字分散表現，Bi-LSTM，全結合層の重みを学習中に固定する．
単語分類器の重みは学習中に更新されない．

6.3 評価指標

評価指標としては単語分類，固有表現抽出ともに精度 (Precision), 再現率 (Recall), F 値 (Fscore) を用いる．精度と再現率，単語分類の場合は

$$\text{精度} = \frac{\text{\#正しく分類できた単語}}{\text{\#対象のカテゴリに属する単語}}, \tag{6.1}$$

$$\text{再現率} = \frac{\text{\#正しく分類できた単語}}{\text{\#分類器がカテゴリに属すると判定した単語}}, \tag{6.2}$$

と定義される．単語分類の評価の例を図 6.1 に示す．

固有表現抽出の場合は

$$\text{精度} = \frac{\text{\#正しく抽出できた固有表現}}{\text{\#コーパス中に含まれる固有表現}}, \tag{6.3}$$

$$\text{再現率} = \frac{\text{\#正しく抽出できた固有表現}}{\text{\#抽出器が抽出した固有表現}}, \tag{6.4}$$

と定義される．固有表現抽出の評価の例を図 6.2 に示す．F 値は単語分類，固有表現抽出ともに

$$\text{Fscore} = \frac{\text{精度} * \text{再現率}}{2 * (\text{精度} + \text{再現率})}, \tag{6.5}$$

のように計算できる．

6.4 実験結果

6.4.1 単語分類の実験結果

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 6.2 に示す．各カテゴリ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．

最も誤分類が多かった「材料-魚介」のカテゴリの誤りについて，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「豆鯵」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．このような誤分類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．

分類器の学習に用いた単語の数と分類性能の関係を図 6.3 に示す．学習データに用いる単語の数を増加させることで，さらなる分類性能の向上が期待できる．

6.4.2 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ポックの抽出器を用いてテストデータでの評価を行った．評価には，conlleval [2] を用いた．

[2] https://www.clips.uantwerpen.be/conll2000/chunking/output.html

[3] 比較手法の固有表現抽出性能の比較を表 6.3 に示す．提案手法 1 は再現率，提案手法 2 は精度と F 値において比較手法の中で最も高い値となった．固有表現抽出において，最も重要とされる評価指標は F 値であるため，以降では F 値で最高性能となった提案手法 2（Wikipedia コーパスで学習した分散表現で単語の分散表現を初期化，ファインチューニングなし）を提案手法と呼び議論を行う．

提案手法のラベルごとの予測性能を表 6.4 に示す．調理者の動作 (Ac) や調理時間 (D) のタグは高い精度で予測に成功している一方，食材などの分量 (Q) や調理道具の状態 (St) のタグの予測性能は低いことが確認できる．これは，教師データ中に含まれる Q タグと St のタグの数が少ないことや「少々」「おこのみで」「すきなだけ」など，分量を表す表現にはバリエーションがあることなどが原因であると思われる．

次に，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」というラベルを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．

[3]我々は，笹田ら [9] 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) に付属している評価スクリプトに誤りを発見した．このバグを修正した結果，評価値が著者らが論文で報告しているよりも低くなることが判明した．このため，本研究では，conlleval (https://www.clips.uantwerpen.be/conll2000/chunking/output.html) を用いてすべての手法について再度評価を行い，得られた結果を示した．

表 6.1: ニューラルネットワークのハイパーパラメータの一覧

	Hyper-parameter	NER	Classifier
Character Embedding	dimensionality	50	50
Character Bi-LSTM	state size	100	100
	initial state	0	0
	peepholes	no	no
Word Embedding	dimensionality	100	–
Word Bi-LSTM	state size	200	–
	initial state	0	–
	peepholes	no	–
Dropout	dropout rate	0.5	0.5
Adam	α	0.001	0.001
	β_1	0.9	0.9
	β_2	0.9	0.9
	gradient clipping	5.0	5.0
	batch size	10	10

表 6.2: 単語の分類性能

カテゴリ	精度	再現率	F 値
動作	94.01	98.74	96.32
材料-その他	74.58	70.97	72.73
材料-肉	87.72	83.33	85.47
材料-野菜	75.00	78.75	76.83
材料-魚介	60.00	61.54	60.76
調味料	81.37	83.97	82.65
調理器具	78.64	74.31	76.42
負例	69.86	66.23	68.00

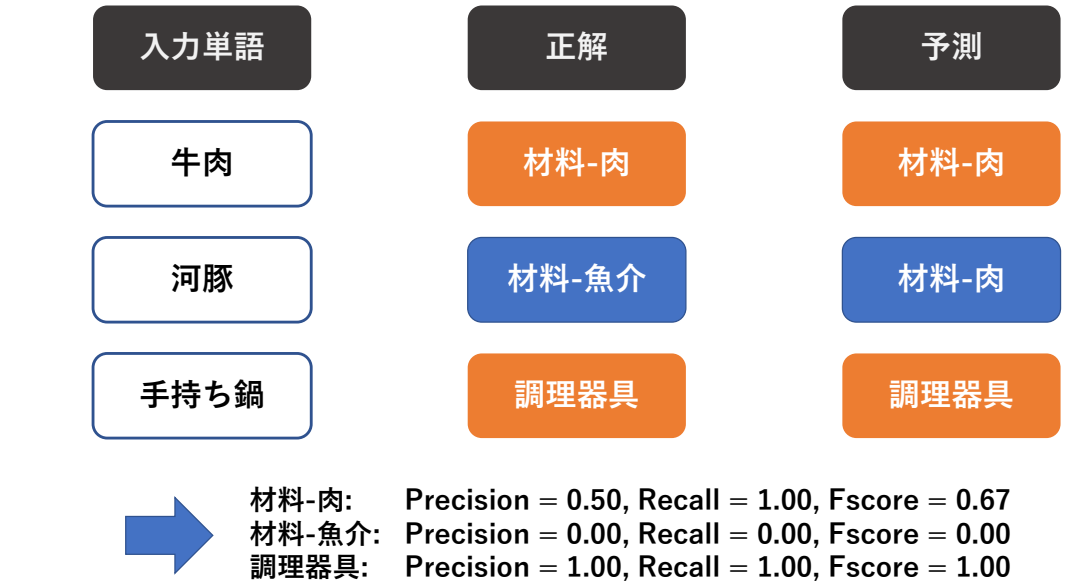

図 6.1: 単語分類の性能評価の例.

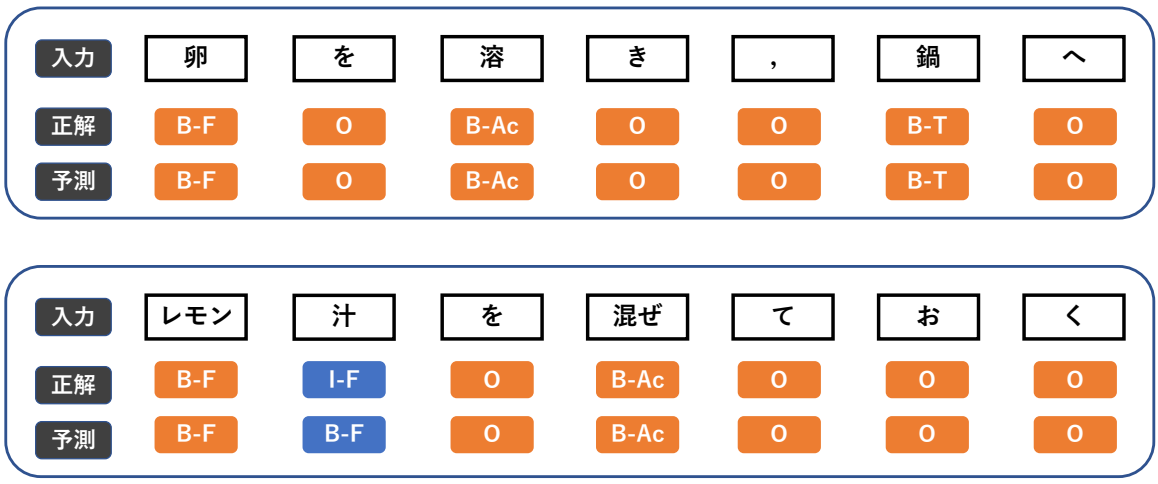

図 6.2: 固有表現抽出の性能評価の例.

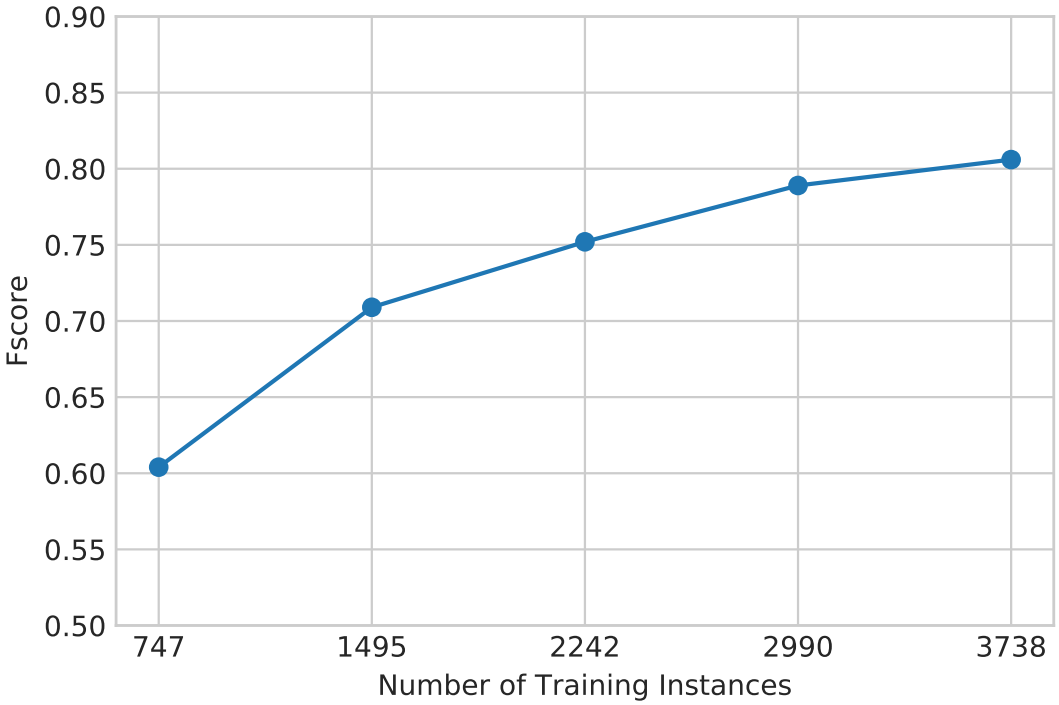

図 6.3: 教師データの数と分類性能の関係

表 6.3: 固有表現抽出器の抽出性能の比較

手法	学習済み分散表現	Fine-tuning	精度	再現率	F 値
笹田ら [9] (LR)	–	–	82.34	80.18	81.2
笹田ら [9] (LR+DP)	–	–	82.94	82.82	82.8
Lample ら [1]	Cookpad	–	84.54 (± 1.22)	88.47 (± 0.69)	86.40 (± 0.89)
Lample ら [1]	Uniform	–	82.59 (± 0.94)	88.19 (± 0.25)	85.24 (± 0.46)
Lample ら [1]	Wikipedia	–	85.31 (± 0.67)	88.22 (± 0.65)	86.68 (± 0.47)
Dictionary	Cookpad	–	83.91 (± 1.21)	88.60 (± 0.41)	86.16 (± 0.72)
Dictionary	Uniform	–	82.36 (± 1.25)	88.28 (± 0.25)	85.18 (± 0.71)
Dictionary	Wikipedia	–	85.44 (± 1.04)	87.67 (± 0.25)	86.50 (± 0.56)
提案手法 1	Uniform	–	82.15 (± 0.95)	88.60 (± 0.41)	85.22 (± 0.49)
提案手法 1	Uniform	Linear	82.74 (± 0.76)	88.44 (± 0.29)	85.44 (± 0.39)
提案手法 1	Uniform	LSTM	82.84 (± 1.03)	88.43 (± 0.46)	85.50 (± 0.55)
提案手法 1	Cookpad	–	84.63 (± 0.24)	88.59 (± 0.28)	86.52 (± 0.24)
提案手法 1	Cookpad	Linear	84.87 (± 0.47)	88.52 (± 0.47)	86.60 (± 0.32)
提案手法 1	Cookpad	LSTM	84.36 (± 0.77)	**89.06 (± 0.40)**	86.60 (± 0.44)
提案手法 1	Wikipedia	–	85.61 (± 0.86)	88.10 (± 0.54)	86.80 (± 0.37)
提案手法 1	Wikipedia	Linear	85.74 (± 1.32)	88.14 (± 0.55)	86.86 (± 0.64)
提案手法 1	Wikipedia	LSTM	84.84 (± 1.01)	88.34 (± 0.53)	86.52 (± 0.60)
提案手法 2	Uniform	–	82.81 (± 0.88)	88.40 (± 0.41)	85.46 (± 0.58)
提案手法 2	Uniform	Linear	83.07 (± 0.61)	88.24 (± 0.43)	85.54 (± 0.42)
提案手法 2	Uniform	LSTM	83.10 (± 0.57)	88.18 (± 0.30)	85.52 (± 0.41)
提案手法 2	Cookpad	–	85.08 (± 1.30)	88.46 (± 0.18)	86.68 (± 0.71)
提案手法 2	Cookpad	Linear	84.15 (± 1.00)	88.20 (± 0.26)	86.08 (± 0.62)
提案手法 2	Cookpad	LSTM	84.78 (± 1.10)	88.22 (± 0.37)	86.44 (± 0.72)
提案手法 2	Wikipedia	–	85.63 (± 0.52)	88.87 (± 0.37)	**87.18 (± 0.34)**
提案手法 2	Wikipedia	Linear	85.55 (± 0.88)	88.34 (± 0.67)	86.86 (± 0.37)
提案手法 2	Wikipedia	LSTM	**85.83 (± 0.53)**	88.25 (± 0.48)	86.98 (± 0.26)

表 6.4: 提案手法のラベルごとの予測性能

固有表現	精度	再現率	F 値
Ac	91.77 (± 1.02)	95.23 (± 0.42)	93.46 (± 0.33)
Af	78.87 (± 3.68)	78.12 (± 1.19)	78.46 (± 2.22)
D	96.63 (± 1.71)	93.88 (± 2.88)	95.23 (± 2.16)
F	85.84 (± 0.94)	89.01 (± 0.65)	87.39 (± 0.59)
Q	58.70 (± 3.81)	70.00 (± 3.19)	63.69 (± 1.82)
Sf	75.12 (± 4.40)	78.17 (± 1.95)	76.52 (± 2.04)
St	66.03 (± 5.64)	52.63 (± 4.70)	58.46 (± 4.52)
T	82.53 (± 2.30)	89.09 (± 1.26)	85.66 (± 1.21)

第7章　結論

本研究では，辞書情報を活用したニューラルネットワークによる固有表現抽出手法を提案した．提案手法は，料理オントロジーデータを辞書として分類器を学習し，得られた分類器の出力を固有表現抽出器の特徴量として活用することで固有表現抽出の性能を向上させた．既存手法では辞書情報は単語レベルでモデルに入力されていたが，提案手法では辞書情報を文字レベル特徴量から構成している．このため，提案手法は辞書中に含まれない単語に対して辞書情報を考慮した特徴量を獲得できる．提案手法の有用性を，レシピドメインにおける固有表現抽出タスク [6] において確認した．レシピ固有表現のタスクで最高性能であった笹田ら [9] の手法および近年一般ドメインの固有表現抽出で用いられることが多い Lample ら [1] の手法と比較した．比較の結果，提案手法は F 値で既存手法を上回る性能を発揮することが明らかになった．これは，提案手法が文字ベースで辞書情報を活用するため，未知語に対しても有用な素性を抽出できていることを示唆している．

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．また，単語分類において多義語をどのように扱うかを検討する必要があると考えられる．本研究では分類器の教師データから多義語を除外しているが，これは多義語を分類するような分類器は学習にかかる計算コストが高く，入力単語について１つのカテゴリにのみ属する仮定をおいたためである．多義語の分類をより現実的な計算コストで行える分類器を構成し，多義語を考慮することでさらなる抽出性能の向上が期待できる．さらに，提案手法の他ドメインでの性能を調査することも今後の課題である．本研究ではレシピテキストを対象に固有表現抽出を行い，その際に辞書資源として難波ら [12] の料理オントロジーデータセットを利用した．辞書資源を構築すれば，新聞記事テキストや医療テキストに対しても提案手法を適用できる．CoNLL 2003 データセットなどの新聞記事テキストや NCBI disease コーパスなどの医療テキストに対してドメインごとの辞書して提案手法を適用し，提案手法の様々なドメインでの抽出性能を調査する必要があると考えられる．

謝辞

本研究の遂行において，学類生時代からご指導，ご鞭撻くださった若林啓助教授に心から感謝します．また，手塚太郎准教授には研究に関する議論で大変お世話になりました．クックパッド株式会社原島純さんには研究に関する助言を数多くいただきました．研究室メンバのみなさん，特に野沢健人さん，山田純也さん，福田拓也さん，福山怜史さん，菊池祥平さんとの日頃の議論によって研究生活が有意義なものとなりました．本当にありがとうございました．

参考文献

[1] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural Architectures for Named Entity Recognition. In *Proceedings of Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, 2016.

[2] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, 2009.

[3] Vikas Yadav and Steven Bethard. A Survey on Recent Advances in Named Entity Recognition from Deep Learning models. In *Proceedings of International Conference on Computational Linguistics*, pages 2145–2158, 2018.

[4] Lorraine Tanabe, Natalie Xie, Lynne H. Thom, Wayne Matten, and W. John Wilbur. GENETAG: A tagged corpus for gene/protein named entity recognition. *BMC Bioinformatics*, 6(SUPPL.1):1–7, 2005.

[5] Rezarta Islamaj Doğan, Robert Leaman, Zhiyong Lu, and National Institutes. NCBI disease corpus: A resource for disease name recognition and concept normalization. *Journal of Biomedical Informatics*, 47:1–10, 2014.

[6] 笹田鉄朗, 森信介, 山肩洋子, 前田浩邦, and 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, 22(2):107–131, 2015.

[7] Hirokuni Maeta, Tetsuro Sasada, and Shinsuke Mori. A Framework for Procedural Text Understanding. In *Proceedings of International Conference on Parsing Technologies*, pages 50–60, 2015.

[8] Graham Neubig, Yosuke Nakata, and Shinsuke Mori. Pointwise Prediction for Robust , Adaptable Japanese Morphological Analysis. In *Proceedings of Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 529–533, 2011.

[9] Tetsuro Sasada, Shinsuke Mori, Tatsuya Kawahara, and Yoko Yamakata. Named entity recognizer trainable from partially annotated data. In *Proceedings of International Conference of the Pacific Association for Computational Linguistics*, pages 148–160, 2015.

[10] Motoki Sato, Hiroyuki Shindo, Ikuya Yamada, and Yuji Matsumoto. Segment-Level Neural Conditional Random Fields for Named Entity Recognition. In *Proceedings of International Joint Conference on Natural Language Proceedings ofssing*, number 1, pages 97–102, 2017.

[11] Khai Mai, Thai-Hoang Pham, Minh Trung Nguyen, Nguyen Tuan Duc, Danushka Bollegala, Ryohei Sasano, and Satoshi Sekine. An Empirical Study on Fine-Grained

Named Entity Recognition. In *Proceedings of International Conference on Computational Linguistics*, pages 711–722, 2018.

[12] Hidetsugu Nanba, Toshiyuki Takezawa, Yoko Doi, Kazutoshi Sumiya, and Miho Tsujita. Construction of a cooking ontology from cooking recipes and patents. In *Proceedings of ACM International Joint Conference on Pervasive and Ubiquitous Computing Adjunct Publication*, pages 507–516, 2014.

[13] Ralph Grishman and Beth Sundheim. Message Understanding Conference-6: A Brief History. In *Proceedings of International Conference on Computational Linguistics*, pages 466–471, 1996.

[14] Nancy Chinchor and Patty Robinson. MUC-7 Named Entity Task Definition. In *Proceedings of Message Understanding Conference*, page 21, 1998.

[15] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of The SIGNLL Conference on Computational Natural Language Learning*, pages 142–147, 2003.

[16] Sameer Pradhan, Alessandro Moschitti, Nianwen XueD, Hwee Tou Ng, Anders Bjorkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. Towards Robust Linguistic Analysis Using OntoNotes. In *Proceedings of Conference on Computational Natural Language Learning*, pages 143–152, 2013.

[17] Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. ACE 2005 multilingual training corpus, 2006.

[18] Satoshi Sekine and Hitoshi Isahara. IREX : IR and IE Evaluation project in Japanese. In *Proceedings of International Conference on Language Resources and Evaluation*, 2000.

[19] J. D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. GENIA corpus - A semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(SUPPL. 1):180–182, 2003.

[20] Amber Stubbs, Christopher Kotfila, and Özlem Uzuner. Automated systems for the de-identification of longitudinal clinical narratives: Overview of 2014 i2b2/UTHealth shared task Track 1. *Journal of Biomedical Informatics*, 58:S11–S19, 2015.

[21] Jiao Li, Yueping Sun, Robin J. Johnson, Daniela Sciaky, Chih Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Thomas C. Wiegers, and Zhiyong Lu. BioCreative V CDR task corpus: a resource for chemical disease relation extraction. *Database : the journal of biological databases and curation*, 2016:1–10, 2016.

[22] Shinsuke Mori, John Richardson, Atsushi Ushiku, Tetsuro Sasada, Hirotaka Kameko, and Yoshimasa Tsuruoka. A Japanese Chess Commentary Corpus. In *Proceedings of Language Resources and Evaluation Conference*, pages 1415–1420, 2015.

[23] Francis Wolinski, Frantz Vichot, and Bruno Dillet. Automatic processing of proper names in texts. In *Proceedings of Annual Conference of the European Chapter of the Association for Computational Linguistics*, pages 23–30, 1995.

[24] Oren Etzioni, Michael Cafarella, Doug Downey, Ana-maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. Unsupervised Named-Entity Extraction from the Web : An Experimental Study. *Artificial intelligence*, 165(1):1–42, 2005.

[25] Guodong Zhou and Su Jian. Named Entity Recognition using an HMM-based Chunk Tagger. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, pages 473–480, 2002.

[26] 山田寛康, 工藤拓, and 松本裕治. Support Vector Machine を用いた日本語固有表現抽出. 情報処理学会論文誌, 43(1):44–53, 2002.

[27] Andrew Mccallum and Wei Li. Early Results for Named Entity Recognition with Conditional Random Fields , Feature Induction and Web-Enhanced Lexicons. In *Proceedings of Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 188–191, 2001.

[28] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural Language Proceedings ofssing (Almost) from Scratch. *Journal of Machine Learning Research*, 12:2493–2537, 2011.

[29] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional LSTM-CRF Models for Sequence Tagging, 2015, 1508.01991.

[30] Xuezhe Ma and Eduard Hovy. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, 2016.

[31] Xavier Carreras, Lluis Marquez, and Lluís Padró. Named entity extraction using adaboost. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, pages 1–4, 2002.

[32] Amaia Salvador, Nicholas Hynes, Nicholas Hynes, Javier Marin, Ferda Ofli, Ingmar Weber, and Antonio Torralba. Learning Cross-modal Embeddings for Cooking Recipes and Food Images. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[33] Javier Marin, Aritro Biswas, Ferda Ofli, Nicholas Hynes, Amaia Salvador, Yusuf Aytar, Ingmar Weber, and Antonio Torralba. Recipe1M: A Dataset for Learning Cross-Modal Embeddings for Cooking Recipes and Food Images, 2018, 1810.06553.

[34] Takayuki Sato, Jun Harashima, and Mamoru Komachi. Japanese-English Machine Translation of Recipe Texts. In *Proceedings of Workshop on Asian Translation*, pages 58–67, 2016.

[35] Atsushi Ushiku, Hayato Hashimoto, Atsushi Hashimoto, and Shinsuke Mori. Procedural Text Generation from an Execution Video. In *Proceedings of International Joint Conference on Natural Language Processing*, pages 326–335, 2017.

[36] Semih Yagcioglu, Aykut Erdem, Erkut Erdem, and Nazli Ikizler-Cinbis. RecipeQA: A Challenge Dataset for Multimodal Comprehension of Cooking Recipes. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, pages 1358–1368, 2018.

[37] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1–32, 1997.

[38] John Lafferty, Andrew McCallum, and Fernando C N Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of International Conference on Machine Learning*, pages 282–289, 2001.

[39] Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, pages 1756–1765, 2017.

[40] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2227–2237, 2018.

[41] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2018, 1810.04805.

[42] Jun Harashima, Ariga Michiaki, Murata Kenta, and Ioki Masayuki. A Large-scale Recipe and Meal Data Collection as Infrastructure for Food Research. In *Proceedings of International Conference on Language Resources and Evaluation*, pages 2455–2459, 2016.

[43] Diederik P Kingma and Jimmy Lei Ba. Adam: A method for stochastic gradient descent. In *Proceedings of International Conference on Learning Representations*, 2015.

[44] Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. Chainer: a next-generation open source framework for deep learning. In *Proceedings of Workshop on Machine Learning Systems*, 2015.

[45] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of International Conference on Learning Representations*, 2013.

[46] Radim Rehurek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of LREC Workshop on New Challenges for NLP Frameworks*, pages 45–50, 2010.