修 士 学 位 論 文

文字ベース双方向 LSTM を用いた日本語固有表現抽出

平成29年度

茨城大学大学院理工学研究科

情報工学専攻
CAO RUI

平成 29 年度茨城大学大学院理工学研究科情報工学専攻　修士学位論文

文字ベース双方向 LSTM を用いた日本語固有表現抽出

著者：CAO RUI (16NM714Y)
指導教員：新納 浩幸 教授

論文要旨

本論文では日本語のテキストから固有表現を抽出するタスクに対して文字ベースの双方向 LSTM を用いた手法を試みる。

固有表現抽出はテキストの中から人名や組織名といった固有表現を抽出するタスクであり、情報抽出や質問応答システムの基盤技術として重要である。固有表現抽出はテキストを単語列とみなし、各単語に BIO のタグを付与する系列ラベリング問題として定式化できる。そのためラベル付きの訓練データを大量に用意することができれば、条件付き確率場 (Conditional Random Field, CRF) の学習手法を利用して高精度に解決できる。ただし対象が日本語テキストの場合、テキストを単語列に分割する際に単語区切りの基準が訓練データと一致していないことも多く、テキストを単語列に変換する処理に、単語区切りの問題が生じる。また、複数の訓練データを合わせて利用する場合にも、各訓練データの単語区切りの基準が異なるため、訓練データを大規模化する際にも単語区切りの問題が生じる。

本論文ではこの単語区切りの問題に対処するために、テキストを単語列ではなく文字列としてとらえ、固有表現抽出を文字に対して BIO のタグを付与する問題として定式化する。これにより上記で述べた問題が生じない。また従来学習手法として CRF が用いられてきたが、近年、Deep Learning の双方向 LSTM を用いた手法が state of the art となっている。本論文では文字ベース双方向 LSTM を実装し、IREX のデータを利用して、その効果や問題を確認する。

実験の結果、文字ベース双方向 LSTM は CRF と同等以上の性能を示した。また訓練データの大きさが本質的に重要であり、今後は転位学習を利用してこの問題に対処する。

Master ' s Thesis in Scholastic 2017, Major in Computer and Information Sciences,
Graduate School of Science and Engineering, Ibaraki University

Named entity extraction from Japanese text with a text-based bidirectional LSTM

Author : CAO RUI (16NM714Y)
Adviser:Prof. Hiroyuki Shinnou

ABSTRACT

In this paper, we tried slove the problem of named entity extraction from Japanese text by use a character-based bidirectional LSTM.

Named entity extraction is a task to extraction the named entity witch have the meaing with person name, organization.As the base technology it is important for the task of information extraction and question answering system.Named entity extraction can be formulated as a sequence labeling problem that regards text as a word sequence and assigns a BIO tag to each word.This problem can be solved with high accuracy by using CRF(Conditional Random Field) if we can prepare a large amount of labeled training data.However, If we got a japanese text, When we dividing the text into word strings, The criterion for single word separation often does not coincide with the training data.So we will got a problem when attempt to converting texts to word strings.Also when using multiple training data together to attempt increase the size of training data we will got the same problem because the criteria for word segmentation is different.

In this paper, in order to deal with the problem of word segmentation.we consider let the text as not a word string but as a character string. Conversion the problem of named entity extraction as a a problem of assigning BIO tag to characters. By do this the problem described above will not occur.CRF has been used as the general method.Recent years,The bidirectional LSTMo of Deep Learning has been become to the most advanced technology.In this paper. we implement character-base bidirectional LSTM and confirm the effect and problem by using IREX data.

As a result, Character-based bidirectional LSTM showed a better performance compare with CRF.And the size of the training data is enssentially important, we wil try to slove this problem by using dislocation learing in the future.

目 次

第 1 章　序論 **7**
1.1 概要 7
1.2 本論文の構成 7

第 2 章　固有表現抽出 **8**
2.1 定義 8
2.2 目的 8
2.3 具体例 8
2.4 分類 8
2.5 系列ラベリング問題としての定式化 9

第 3 章　BIO タグ付与による解決 **10**

第 4 章　系列ラベリング問題 **11**
4.1 定義 11
4.2 特徴 11
4.2.1 注目している要素以外の情報も使える 11
4.2.2 クラスの数が膨大になりやすいこと 11
4.3 モデル 11
4.3.1 隠れマルコフモデル (Hidden Markov Model) 11
4.3.2 多値分類器の逐次適用 13
4.3.3 条件付き確率場 (Conditional Random Fields,CRF) 13
4.4 各モデルのメリット　デメリット 14
4.4.1 隠れマルコフモデル： 14
4.4.2 多値分類器の逐次適用： 15
4.4.3 条件付き確率場： 15

第 5 章　LSTM による BIO タグの付与 **16**
5.1 LSTM 16
5.1.1 定義 16
5.1.2 ゲート 16
5.1.3 セルの更新： 17
5.1.4 忘却： 17
5.1.5 更新： 17
5.1.6 セルの更新の全体： 17
5.1.7 予測の出力： 17

5.1.8 LSTM の順伝播計算： 18
5.1.9 LSTM の逆伝播計算： 18
5.2 LSTM による BIO タグの付与 18
5.3 ビタビアルゴリズム 19

第 6 章 実験 **20**
6.1 実験設定 20
6.2 実験結果 21

第 7 章 考察 **24**

第 8 章 結論 **25**

表 目 次

5.1 LSTM による BIO タグの付与 19

6.1 ８種類の固有表現 20
6.2 固有表現のラベル 21
6.3 実験結果 22

図 目 次

6.1 epoch 毎のモデルの評価 23

第1章　序論

1.1　概要

本論文では日本語のテキストから固有表現を抽出するタスクに対して文字ベースの双方向 LSTM を用いた手法を試みる。

固有表現抽出はテキストの中から人名や組織名といった固有表現を抽出するタスクであり、情報抽出や質問応答システムの基盤技術として重要である。固有表現抽出はテキストを単語列とみなし、各単語に BIO のタグを付与する系列ラベリング問題として定式化できる。そのためラベル付きの訓練データを大量に用意することができれば、条件付き確率場 (Conditional Random Field, CRF) の学習手法を利用して高精度に解決できる。ただし対象が日本語テキストの場合、テキストを単語列に分割する際に単語区切りの基準が訓練データと一致していないことも多く、テキストを単語列に変換する処理に、単語区切りの問題が生じる。また、複数の訓練データを合わせて利用する場合にも、各訓練データの単語区切りの基準が異なるため、訓練データを大規模化する際にも単語区切りの問題が生じる。

本論文ではこの単語区切りの問題に対処するために、テキストを単語列ではなく文字列としてとらえ、固有表現抽出を文字に対して BIO のタグを付与する問題として定式化する。これにより上記で述べた問題が生じない。また従来学習手法として CRF が用いられてきたが、近年、Deep Learning の双方向 LSTM を用いた手法が state of the art となっている。本論文では文字ベース双方向 LSTM を実装し、IREX のデータを利用して、その効果や問題を確認する。

実験の結果、文字ベース双方向 LSTM は CRF と同等以上の性能を示した。また訓練データの大きさが本質的に重要であり、今後は転位学習を利用してこの問題に対処する。

1.2　本論文の構成

本論文では，はじめ固有表現に関することを紹介しました．第 3 章でデータの前処理として，精度が良いと報告されている「Inside-Outside 法」のバリエーションの 1 つである、 IOB2 と呼ばれるチャンクタグ集合を用いる手法について述べました．第 4 章で系列ラベリング問題についてまとめました．第 5 章では LSTM による BIO タグの付与する方法を提案し，第 6 章では実験で利用したデータとその結果示した，第 7 章では実験結果について考察した，最後に第 8 章ではこの研究の結論を述べた．

第2章　固有表現抽出

2.1　定義

固有表現抽出とは，人名や地名，組織などと言った固有名詞と，日付，場所などに関する表現をテキストから抜き出すことを指しています，自然言語処理に置ける情報抽出の一分野のことである．

2.2　目的

現在世の中に存在する新聞記事，小説などのテキストには大量の固有表現（Name Entity）が含まれています，形態素解析を用いて行う場合，辞書に登録されていない固有表現を存在したと，未知語として扱われ，解析の誤りが生じます，その原因で，大量の固有表現を辞書に登録する必要があります，人手でそれらを登録することは困難である．

この問題を解決する為に計算によって大量のテキストから固有表現を自動的に抽出技術出するが生まれた．

2.3　具体例

太郎は 5 月 18 日の朝 9 時に花子に会いに行った。

という文に含まれる固有表現を抽出すると以下のようになります．

$< PERSON >$ 太郎 $< /PERSON >$ は $< DATE >$ 5 月 18 日 $< /DATE >$ の $< TIME >$ 朝 9 時 $< /TIME >$ に $< PERSON >$ 花子 $< /PERSON >$ に会いに行った。

ここで、¡..¿¡/..¿で囲まれた部分が固有表現であり、¡..¿は表現の分類を示すタグである。

2.4　分類

固有表現には人名や日付表現などいくつかの分類があり，この分類を定義する必要がある．

IREXでは，人名「PERSON」,地名「LOCATION」，組織名「ORGANIZATION」，日付表現「DATE」，時間表現「TIME」，金額表現「MONEY」，割合表現「PERCENT」，固有物名「ARTIFACT」の８種類を定義しました.

2.5　系列ラベリング問題としての定式化

現在用いられているほとんどの固有表現抽出手法では、タスクを系列ラベリング問題としてモデル化し、モデルのパラメータを教師付き学習の枠組みでデータから学習する。

第3章　BIOタグ付与による解決

現在さまざまなな手法を提案されていますが，データの前処理として，SVM（サポートベクトルマシン）を用いた固有表現抽出において精度が良いと報告されている「 Inside-Outside 法 」 のバリエーションの 1 つである、 IOB2 と呼ばれるチャンクタグ集合を用いる手法を利用すると考えています.

Suddenly, the tall German guy talked to me.

O　　B　I　I　I　O O O

ここで，B は人をさす表現の開始地点（beginning）を，I はその内部（inside）を，O はその外部（outside）を表している．こうすると，複数語からなる表現でもうまく表せる．抽出の為に使ったこのようなラベル IOB 2 タグを呼ばれます．I と O だけでは，連続して出現した表現に対応できません.

O の次に O になることが多いですが，O の次に I になることはありません．このように，どんなラベルが来るかは相互に依存しており，これは系列ラベリングの問題と見なすことができます.

第4章　系列ラベリング問題

4.1　定義

系列ラベリング問題とは「ある系列 X の各要素に適切なラベル列 Y を付与する問題」である．日本語固有表現抽出では Xn は処理対象の文中の文字、Yn は固有表現の存在とクラスをエンコードするラベルである．

Nurture passes nature.
[名詞] [動詞] [名詞]

4.2　特徴

4.2.1　注目している要素以外の情報も使える

多値分類では，要素 xi を与えて，その要素 xi がどのグラス yj に属するかを判定するだけである，系列ラベリングでは，要素の系列 x(=x1, x2, x3,....) のように与えています．こうすると,「形容詞の後には名詞が来安い」などの情報を考慮できるようになりました．

4.2.2　クラスの数が膨大になりやすいこと

例（品詞タグ付け）:

This is a pen.
[代名詞][動詞][冠詞][名詞]

品詞の数が n 個ならば、その組み合わせは n の 4 乗になります，クラスも代名詞 代名詞 代名詞 代名詞」「代名詞 代名詞 代名詞 名詞」... のように数が膨大になります，計算することは困難である．

4.3　モデル

4.3.1　隠れマルコフモデル (Hidden Markov Model)

系列： x = (x1, ... xl)

ラベル列： y = (yi, ... yl)

推定方法：

$$同時確率\ P(x,y)\ から\ y^{max} = \arg\max y_2 P(x,y) \tag{4.1}$$

仮定：現在の要素は直前の要素しか依存してません

$$\begin{aligned} P(x,y) &= P(x_l, y_l | x_{l-1}, y_{l-1}, ..., x_l, y_l) P(x_{l-1}, y_{l-1}, ..., x_l, y_l) \\ &= P(x_l, y_l | x_{l-1}, y_{l-1}) P(x_{l-1}, y_{l-1}, ..., x_l, y_l) \end{aligned} \tag{4.2}$$

同時確率：

$$P(x,y) = \prod_{i=1}^{l} P(x_i|y_i) P(y_i|y_{i-1}) \tag{4.3}$$

未知パラメータ：

$$P(x|y), P(y|y^{'}) \tag{4.4}$$

学習データ：

$$D = (x^{(1)}, y^{(1)}).....(x^{(D|)}, y^{(D|)}) \tag{4.5}$$

最尤推定解析解：

$$\begin{aligned} P(x|y) &= \frac{D\ 内で\ x\ に\ y\ がラベルされた数}{D\ 内で\ y\ がラベルとされた数} \\ P(y|y^{'}) &= \frac{D\ 内で\ y^{'}の次に\ y\ が出現した数}{D\ 内の\ y^{'}のうち次の要素が存在する\ y^{'}の数} \end{aligned} \tag{4.6}$$

求めたいもの：

$$\begin{aligned} y_{max} &= \arg max_y P(x,y) = \arg max_y log P(x,y) \\ &= \arg max_y \sum_{i=1} \log P(x_i, y_i | x_{i-1}, y_{i-1}) \end{aligned} \tag{4.7}$$

4.3.2 多値分類器の逐次適用

品詞タグ付ける時，その単語自身 (xi), 直前の単語 (xi-1)，直後の単語 (xi+1) などが素性として使える．ただし，yi-1 や yi+1 はまだわからないので使えない．そこで，y1, y2, y3,... と，前から順に推定していって，yi の推定にはすでに推定した yi-1 を用いることを考える．yi-1 は推定した値なので，ひょっとしたら間違っていることもあるかもしれない．よって，yi を推定する際には，yi-1, xi-1, xi, xi+1 などから素性を抽出することができる．学習段階ではすべてのラベルがわかっているので，正しい yi-1 を用いて素性ベクトルを作ることができる．これが分類器の逐次適用である．それに，分類器の逐次適用では，様々な素性を用いて xi のベクトルを表現することができる．例えば，直前や直後に出現した単語や，xi を構成する部分文字列などを用いても良い．一般的に計算時間は多くなるが，より高精度な分類が可能になる．

4.3.3 条件付き確率場 (Conditional Random Fields,CRF)

系列： x = (x1, ... xl)

ラベル列： y = (yi, ... yl)

推定方法：

$$\text{条件付き確率 } P(y|x) \text{ から } y^{max} = \arg\max y_2 P(x,y) \text{ を推定する} \tag{4.8}$$

仮定：

対数線形モデル $P(y|x) = exp(w \cdot \phi(x,y))/\mathbb{Z}_{x,w}$に以下を仮定する：

$$\phi_k(x,y) = \sum_{i=1} \phi_k(x, y_t, y_(t-1)) \tag{4.9}$$

条件付き確率：

$$\phi_k(x,y) = \frac{1}{Z_(x,w)} \prod_t \exp(w \cdot \phi(x, y_t, y_(t-1))) \tag{4.10}$$

条件付き確率：

$$w \tag{4.11}$$

学習データ：

$$D = (x^(1), y^(1).....(x^(D|), y^(D|))) \tag{4.12}$$

最急勾配法：

$$\begin{aligned} L(w) &= \sum_{(x^i,y^i)\in D} logP(y^{(i)}|x^{(i)}) \quad \text{と置くと} \\ w^{new} &= w^{old} + \varepsilon \bigtriangledown_w Lw^{old} \\ \bigtriangledown_w Lw^{old} &= \sum_{(x^i,y^i)\in D} (\phi(x^{(i)}, y^{(i)}) - \sum_y P(y|x^{(i)}\phi(x^{(i)}, y)) \\ \sum_y P(y|x^{(i)}\phi(x^{(i)}, y) &= \sum_t \sum_{y_t, y_{t-1}} P(y_{t-1}, y_t|x^{(i)}\phi(x^{(i)}, y_t, y_{t-1})) \end{aligned} \tag{4.13}$$

求めたいもの：

$$\begin{aligned} y^{max} &= \arg max P(y|x) \\ &= \arg \max_y \frac{1}{Z_{x,w}} \exp(\sum_t w \cdot \phi(x, y_t, y_{t-1})) \\ &= \arg \max_y \sum_t w \cdot \phi(x, y_t, y_{t-1}) \end{aligned} \tag{4.14}$$

4.4 各モデルのメリット　デメリット

4.4.1 隠れマルコフモデル：

解析解が早く求まる
簡単に試せる
素性に制限
未知の観測値を捉えない

4.4.2　多値分類器の逐次適用：

様々な素性を考えられる
ラベル系列全体の確からしさを考慮できない

4.4.3　条件付き確率場：

様々な素性を考えられる
ラベル系列全体の確からしさを考慮できる
学習が遅い

第5章　LSTMによるBIOタグの付与

5.1　LSTM

5.1.1　定義

LSTM とは Long Short-Term Memory の略です. 時系列問題に対する Deep Learning の手法 Recurrent NN の 1 つである．LSTM は RNN の中間層のユニットを LSTM block と呼ばれるメモリと 3 つのゲートを持つブロックに置き換えることで実現されています。LSTM はゲートを使うことで勾配消失を解決した。

5.1.2　ゲート

LSTM には、3 つのゲートから入力を受け取り、以下のように表される:

$$\begin{aligned} gate_{forget} &= \sigma(W_{fx}X_t + W_{fh}h_{t-1} + b_f) \\ gate_{input} &= \sigma(W_{ix}X_t + W_{ih}h_{t-1} + b_i) \\ gate_{out} &= \sigma(W_{ox}X_t + W_{oh}h_{t-1} + b_o) \end{aligned} \tag{5.1}$$

各入力ゲートでは、活性化関数にシグモイドが使われている。そして各ゲートの役割は以下になる：

Forget Gate

前のセルの状態から，どの部分を保持するかどの部分を忘却するかを制御する．つまり状態の忘却である．

Input Gate

新しく計算した情報のどの部分をセルの状態に加えるのかを制御する．つまり状態の更新である．

Out Gate

セルの状態のどの部分を予測として出力するかを制御する．

5.1.3　セルの更新：

LSTM には，セルの更新と予測の更新があり，以下のように表される．

$$\begin{aligned} \tilde{C} &= tanh(W_{cx}C_t + W_chh_(t-1) + b_c) \\ C_t &= gate_{forget} \cdot C_{t-1} + gate_{input} \cdot \tilde{C} \\ h_t &= gate_{out} \cdot tanh(C_t) \end{aligned} \tag{5.2}$$

5.1.4　忘却：

上の式の Ct の

$$gate_{forget} \cdot tanh(C_t) \tag{5.3}$$

の部分で，前のセルの状態と忘却ゲートの入力をかけて、前のセルのいくつかの部分を忘却する。

5.1.5　更新：

$$\begin{aligned} \tilde{C} &= tanh(W_{cx}C_t + W_chh_(t-1) + b_c) \\ & gate_{input} \cdot \tilde{C} \end{aligned} \tag{5.4}$$

入力ゲートと更新元の情報 (C チルド) をかけて、セルの状態に新たな情報を加える。

5.1.6　セルの更新の全体：

$$C_t = gate_{forget} \cdot C_{t-1} + gate_{input} \cdot \tilde{C} \tag{5.5}$$

5.1.7　予測の出力：

$$h_t = gate_out \cdot tanh(C_t) \tag{5.6}$$

セルの状態に出力ゲートの情報をかけて、予測を出力する。

5.1.8 LSTM の順伝播計算：

$$
\begin{aligned}
\bar{z}^t &= W_z x^t + R_z y^(t-1) + b_z \\
z^t &= g(\bar{z}^t) \\
\bar{i}^t &= W_{in} x^t + R_{in} y^{t-1} + p_{in} \bigodot c^{t-1} + b_{in} \\
i^t &= \sigma(\bar{i}^t) \\
\bar{f}^t &= W_{for} x^t + R_{for} y^{t-1} + p_{for} \bigodot c^{t-1} + b_{for} \\
f^t &= \sigma(\bar{f}^t) \\
c^t &= i^t \bigodot z^t + f^t \bigodot c^{t-1} \\
\bar{o}^t &= W_{out} x^t + R_{out} y^{t-1} + p_{out} \bigodot c^t + b_{out} \\
o^t &= \sigma(\bar{o}^t) \\
y^t &= o^t \bigodot h(c^t)
\end{aligned}
\tag{5.7}
$$

5.1.9 LSTM の逆伝播計算：

$$
\begin{aligned}
\delta y^t &= \Delta + R_z^T \delta z^{t+1} + R_{in}^T \delta i^{t+1} + R_{for}^T \delta f^{t+1} + R_{out}^T \delta o^{t+1} \\
\delta o^t &= \delta y^t \bigodot h(c^t) \bigodot \sigma'(\bar{o}^t) \\
\delta c^t &= \delta y^t \bigodot o^t \bigodot h'(c^t) + p_{out} \bigodot \delta o^t + \\
& p_{in} \bigodot \delta i^{t+1} + p_{for} \bigodot \delta f^{t+1} + \delta c^{t+1} \bigodot f^{t+1} \\
\delta f^t &= \delta c^t \bigodot c^{t-1} \bigodot \sigma'(\bar{f}^t) \\
\delta i^t &= \delta c^t \bigodot z^t \bigodot \sigma'(\bar{i}^t) \\
\delta i^z &= \delta c^t \bigodot i^t \bigodot g'(\bar{z}^t)
\end{aligned}
\tag{5.8}
$$

5.2 LSTM による BIO タグの付与

最初の手順として,LSTM により各タグに対する確率を求める，そして，各文字に対して，選択したタグの確率の和が最大になるようにタグを選択します．選択する際に，あるパースの確率の和が最大としても，このパースが以下のルールを従うすべきである．この部分はビタビアルコリズムで求められます.

$$
\begin{aligned}
O --> B \text{ OK} \\
O --> I \text{ 駄目}
\end{aligned}
$$

表 5.1: LSTM による BIO タグの付与

	私	は	茨	城	に
t_0	p_{00}	p_{01}	p_{02}	p_{03}	p_{03}
t_1	p_{01}	p_{11}	p_{12}	p_{13}	p_{03}
t_2	p_{02}	p_{21}	p_{22}	p_{23}	p_{03}
t_3	p_{03}	p_{31}	p_{32}	p_{33}	p_{03}
t_5	p_{04}	p_{41}	p_{42}	p_{43}	p_{03}
...	...	...	...	...	p_{03}
t_{18}	p_{180}	p_{181}	p_{182}	p_{183}	p_{03}

5.3 ビタビアルコリズム

入力: x

for all y_j

$t(1, y_i) = P(x_1, y_1)$

for j = 2 to $|D|$

for all y_j

$$t(j, y_j) = \max_{y_{j-1}}[logP(x_j, y_j|x_{j-1}, y_{j-1}) + t(j-1, y_{j-1})]$$
$$s(j, y_j) = \underset{y_{j-1}}{argmax}[logP(x_j, y_j|x_{j-1}, y_{j-1}) + t(j-1, y_{j-1})]$$
$$y_{|x|}^{max} = \underset{y}{argmax}t(|x|, y)$$
$$forj = |x| - 1to1$$
$$y_j^{max} = s(j+1, y_{j+1}^{max})$$

第6章　実験

6.1　実験設定

提案手法の有効性や問題を確認するために IREXhttp://nlp.cs.nyu.edu/irex/NE/により配布されている CRL 固有表現データを用いて提案手法による固有表現抽出の実験を行った。このデータは毎日新聞 95 年 1 月 1 日から 10 日までの全記事、約 1 万文に対して、IREX で定義された 8 種類の固有表現をタグ付けしたデータである。本実験で抽出する固有表現も IREX で定義された以下の 8 種類の固有表現である。

表 6.1: 8 種類の固有表現

固有表現	意味	例
ARTIFACT	固有物名	ノーベル賞、Windows10
LOCATION	地名	イギリス、千葉県
ORGANIZATION	組織	自民党、NHK
PERSON	人名	安倍晋三、メルケル
DATE	日付	5 月 29 日、2018/01/31
TIME	時間	午前三時、10:30
MONEY	金額	141 円、8 ドル
PERCENT	割合	15 %、3 割

CRL 固有表現データのうちプログラムで問題なく取り出せた 4,296 文を利用する。そのうち 3,951 文を訓練データ、345 文をテストデータとした。文字で換算すると訓練データは 255,882 文字、テストデータは 26,187 文字となる。各文字に対して表 6.2 のラベルが付与される。ただし表 6.2 の中で `OPTIONAL` は対象にしなかったので、実際に付与するタグは 17 種類である。

以下に例を示す。以下がタグ付けされた元データの 1 文である。

<PERSON> 村山富市 </PERSON> 首相は <DATE> 年頭 </DATE> にあたり <LOCATION> 首相官邸 </LOCATION> で <ORGANIZATION> 内閣記者会 </ORGANIZATION> と <DATE> 二十八日 </DATE> 会見し、<ORGANIZATION> 社会党 </ORGANIZATION> の <ORGANIZATION> 新民主連合 </ORGANIZATION> 所属議員の離党問題について「政権に影響を及ぼすことにはならない。

表 6.2: 固有表現のラベル

ID	固有表現ラベル	意味
0	NONE	固有表現とは無関係
1	ARTIFACT-B	固有物名の開始
2	ARTIFACT-I	固有物名の中
3	DATE-B	日付の開始
4	DATE-I	日付の中
5	LOCATION-B	地名の開始
6	LOCATION-I	地名の中
7	MONEY-B	金額の開始
8	MONEY-I	金額の中
9	OPTIONAL-B	オプションの開始
10	OPTIONAL-I	オプションの中
11	ORGANIZATION-B	組織の開始
12	ORGANIZATION-I	組織の中
13	PERCENT-B	割合の開始
14	PERCENT-I	割合の中
15	PERSON-B	人名の開始
16	PERSON-I	人名の中
17	TIME-B	時間の開始
18	TIME-I	時間の中

上記のデータを本手法が利用できるように文字単位に区切り、各文字に固有表現のラベルを付与したものが次である。各文から 2 行のデータが作成される。 1 行目が文字単位の文、2 行目が 1 行目の各文字に付けたラベルの列である。

村 山 富 市 首 相 は 年 頭 に あ た り 首 相 官 邸 で 内 閣 記 者 会 と 二 十 八 日 会 見 し 、 社 会 党 の 新 民 主 連 合 所 属 議 員 の 離 党 問 題 に つ い て 「 政 権 に 影 響 を 及 ぼ す こ と に は な ら な い 。
15 16 16 16 0 0 0 3 4 0 0 0 0 5 6 6 6 0 11 12 12 12 12 0 3 4 4 4 0 0 0 0 11 12 12 0 11 12 12 12 12 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

6.2 実験結果

評価では再現率と適合率を調べる。学習したモデルが文字に対して NONE 以外のラベルを付与し、その付与したラベルが正解と合致した数を A とする。

またシステムが文字に対して NONE 以外のラベルを付与した総数を S_1、正解データ中の NONE 以外のラベルの総数を S_2 とするとき、再現率（R）と適合率（P）は以

下で算出される。

$$P = \frac{A}{S_1}, R = \frac{A}{S_2}$$

そして学習したモデル全体の評価値は以下の F-値によって算出する。

$$F = \frac{2 \cdot P \cdot R}{P + R}$$

モデルは訓練データを使って学習された双方向 LSTM のモデルである。各 epoch 毎にモデルを保存し、各 epoch 毎のモデルをテストデータを利用して評価した。

結果を表 kekka1 と図 kekka2 に示す。

表 6.3: 実験結果

epoch	適合率	再現率	F-値
1	0.4625	0.4533	0.4579
2	0.5475	0.5406	0.5440
3	0.6084	0.5877	0.5979
4	0.6617	0.6369	0.6491
5	0.7056	0.6866	0.6960
6	0.7300	0.7074	0.7185
7	0.7383	0.7252	0.7316
8	0.7411	0.7257	0.7333
9	0.7696	0.7383	0.7536

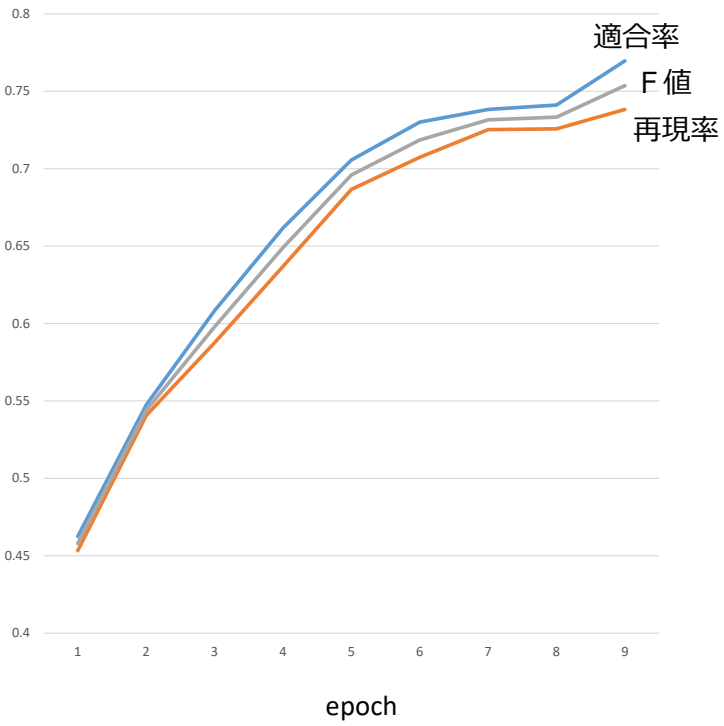

図 6.1: epoch 毎のモデルの評価

第7章　考察

本論文では，テキストを単語列ではなく文字列として捉え，固有表現抽出を文字に対して BIO のタグを付与する問題として定 式化する．単語境界を揃える面倒な処理が必要がないとなっています．Deep Leariing の双方向 LSTM を用いて，現在最も精度の高い手法 CRF より良い精度が期待できると考え，結果に CRF と同等以上の性能を示した．実験の評価が不十分である．今後の実験で充実させます．それに，別のコーパスを用いて，結果を比較して，効果を確認に行くと考えています．

第8章　結論

本論文では文字ベースの日本語固有表現抽出システムを作成し，正解率の低下を押さえて再現率の改善を試した．テキストを単語ではなく文字列としてとらえ，単語区切りの問題が生じない．実験の結果、文字ベース双方向 LSTM は CRF と同等以上の性能を示した。また訓 練データの大きさが本質的に重要であり、今後は転位学習を利用してこの問題に対処する。

謝辞

この研究を修士論文として形にすることが出来たのは，指導教員の新納浩幸教授の厳しくも優しい指導を賜りました．新納先生に深謝致します．また，日常の議論の際，多くの知識やヒントを頂いた佐々木稔講師， 古宮嘉那子講師と新納研究室の皆様に感謝致します．

参考文献

[1] 言語処理のための機械学習入門 高村 大也 (著), 奥村 学 (監修) コロナ社

[2] https://www.weblio.jp/wkpja/content/%E5%9B%BA%E6%9C%89%E8%A1%A8%E7%8F%BE%E6%8A%BD%E5%87%BA_%E5%9B%BA%E6%9C%89%E8%A1%A8%E7%8F%BE%E6%8A%BD%E5%87%BA%E3%81%AE%E6%A6%82%E8%A6%81#.E5.9B.BA.E6.9C.89.E8.A1.A8.E7.8F.BE.E5.88.86.E9.A1.9E

[3] https://sitest.jp/blog/?p=9788

[4] https://www.slideshare.net/Takatymo/ss-64274683

[5] https://qiita.com/t_Signull/items/21b82be280b46f467d1b

[6] https://qiita.com/guitar_char/items/cb4d397cc809b95f1b59

[7] http://www.tkl.iis.u-tokyo.ac.jp/top/modules/newdb/extract/190/data/2008no14fukushima.pdf#search=%27IOB+2%E3%81%A6%E3%82%99%E3%81%AF%E5%9B%BA%E6%9C%89%E8%A1%A8%E7%8F%BE%27

ソースリスト

```
xp = cuda.cupy
wdic = {}
f = open('wdic.pkl','rb')
wdic = pickle.load(f)
f.close()
trainx = {}
trainy = {}
f = open('train.dat','r')
xl = f.readline()
yl = f.readline()
tn = 0
while (xl and yl):
xl = xl.rstrip()
yl = yl.rstrip()
ws = xl.split()
xxl = []
for w in ws:
if w in wdic:
xxl += [ wdic[w] ]
else:
xxl += [ 0 ]
trainx[tn] = xxl
trainy[tn] = yl.split()
tn += 1
xl = f.readline()
yl = f.readline()
f.close

class MyLSTM(chainer.Chain):
def __init__(self, lay, v, k, cl, dout):
super(MyLSTM, self).__init__(
embed = L.EmbedID(v, k),
H = L.NStepBiLSTM(lay, k, k, dout),
W = L.Linear(2 * k, cl),
```

```
)
def __call__(self, hx, cx, xs, t):
accum__loss = None
xembs = [ self.embed(x) for x in xs ]
xss = tuple(xembs)
hy, cy, ys = self.H(None, None, xss)
y = [self.W(item) for item in ys]
for i in range(len(y)):
tx = Variable(xp.array(t[i], dtype=xp.int32))
loss = F.softmax__cross__entropy(y[i], tx)
accum__loss = loss if accum__loss is None else accum__loss + loss
return accum__loss

  demb = 100
vsize = len(wdic)
classsize = 19

  model = MyLSTM(2, vsize, demb, classsize, 0.5)
cuda.get__device(0).use()
model.to__gpu()
optimizer = optimizers.Adam()
optimizer.setup(model)

  bc = 0
xs = []
ts = []
for epoch in range(10):
s = []
for sn in range(len(trainx)):
x = trainx[sn]
y = trainy[sn]
xs += [ xp.array(x, dtype=xp.int32) ]
ts += [ xp.array(y, dtype=xp.int32) ]
bc += 1
if (bc == 10):
model.cleargrads()
hx = chainer.Variable(xp.zeros((2, len(xs), demb), dtype=xp.float32))
cx = chainer.Variable(xp.zeros((2, len(xs), demb), dtype=xp.float32))
loss = model(hx, cx, xs, ts)
loss.backward()
optimizer.update()
xs = []
```

```
ts = []
bc = 0
if (sn % 50 == 0):
print(sn, "/", len(trainx)," finished")
outfile = "nsteplstm-" + str(epoch) + ".model"
serializers.save_npz(outfile, model)
```