Vol.2021-CH-125 No.4
2021/2/13

候文における文字単位の単語分散表現モデルに基づく固有表現抽出手法

吉賀 夏子[1,a)]　堀 良彰[2,b)]　永崎 研宣[3,c)]

概要：江戸期において，各藩で日々の業務を記した「日記」と呼ばれる記録は，現在も全国各地に膨大に残されている．これまで著者らは多様な背景を持つ市民に広く日記内容の読み解きを容易にするため，候文で記述された記事文の低コストな Linked Data 化を試みてきた．本研究では，一連の研究で得た知見から，Linked Data 化システムで必須の固有表現抽出における未知語の判定をより高精度に行うため，深層学習での固有表現抽出に採用される手法のひとつである，文字単位の単語分散表現モデルを用いた手法を採用した．本手法では，従来深層学習するには少ない教師データと現代日本語の大規模コーパスで作られた分散モデルを組み合わせて学習モデルを構築できる．提案手法で構築したモデルを固有表現抽出した結果，形態素解析ツールのユーザ辞書に未登録の語彙に対し，本モデルによる学習で高精度に判定可能であることが示唆された．

キーワード：固有表現抽出，候文，文字単位の単語分散表現，Flair, 古文書

Named Entity Extraction Based on Character Level Embeddings for Understanding Japanese Historical Business Records

Abstract: In the Edo period (1603-1868), Nikki and Nikki Mokuroku, hand-written business records and those catalogs, were kept by clans in various parts of Japan to record their daily activities. In order to facilitate the reading and understanding of the contents of Nikki by citizens with diverse backgrounds, the authors have performed a named entity recognition to create low-cost Linked Data from the translated Mokuroku titles written in Sorobun, an epistolary style mainly used for documenting official papers. In our previous studies, we are aware that it is difficult to recognize unknown vocabraries by using only common morphological analysis tools. In this study we attempted to create a word-by-word embedding model, which is one of the deep learning methods for extracting named entities in order to extract and specify a classname of unknown Japanese words more accurately. We constructed a combined model, recognizing semantic meanings of a word in a sentence. The model consists of a distributed model made from a large corpus of modern Japanese and a small number of labeled Souroubun data. As a result of named entity recognition by the proposed method, it is suggested that the method can accurately recognize the vocabularies and the named entity classnames that is not registered in and user dictionary of a morphological analysis tool.

Keywords: Named Entity Extraction, Japanese Epistolary Style (Sorobun), Character Level Word Embeddings, Flair, Historical Documents

1. はじめに

本研究は，江戸期に「候文（そうろうぶん）」[1] と呼ばれる文語体で記述された文章の翻刻記事に対し，従来より高精度な固有表現抽出 (以下，NEE と呼ぶ．Named Entity Extraction の略．) を行うため，事前学習された現代日本語の単語分散表現に候文からの教師データを追学習したモデルを構築して NEE を行う手法を提案するものである．

筆者らは，これまでに形態素解析ツール MeCab[2] の辞書データを介して，候文で書かれた目録記事文から NEE

[1] 佐賀大学地域学歴史文化研究センター
[2] 佐賀大学全学教育機構
[3] 一般財団法人人文情報学研究所
[a)] natsukoy@cc.saga-u.ac.jp
[b)] horiyo@cc.saga-u.ac.jp
[c)] nagasaki@dhii.jp

© 2021 Information Processing Society of Japan

を行う仕組みを開発した [3]．さらに，未知語や地域色の強い固有表現については，郷土資料の読み解きに明るい地元市民によるクラウドソーシングで固有表現を収集することで，少数による手作業で多くの候文に対し比較的短い期間での NEE を実現した [4]．

しかし，上記の形態素解析用の辞書データを充実させる手法には次に挙げる課題がある．

- 人による確認が済んだ記事に全く出てこない未知語は形態素解析ツールで対応できない．例えば，文中で読点などの区切り文字なしに人名が連続する場合，当時の日本語名が一般的にどのようなものであるかを「ある程度」知っている人間なら，姓名を合わせた人名あるいは複数の人名を抽出できるが，形態素解析ツールのみではうまく抽出できない．
- 一般に多くの単語は多義であり，文脈に応じて NEE 判定を行う必要がある．例えば「肥前」という単語は文脈によって地名にも人名にもなりうる．
- クラウドソーシングの参加者でそれぞれ NEE 判定に多少ずれがある，もしくは判定基準にあいまいさが生じる場合がある．その結果，ユーザ辞書登録に二重の確認が必要となり手間がかかる．

以上の課題を解決するためには，形態素解析ツールの辞書データを応用した従来の NEE 手法とは異なる手法で未知語を予測し，適切な固有表現クラスをラベル付けする必要がある．それを実現する方法の一つとして，大量の候文を形作る文字および単語を「分散表現」と呼ばれる手法で数値化して学習モデルを構築し，そのモデルで候文中の未知語を認識し，適切に固有表現クラスを予測する深層学習による方法が挙げられる．

しかし，実用的な学習モデルの構築に必要な大量の候文サンプルの収集は，現状では困難である．その代替案として，本研究では，Web からダウンロード可能な現代日本語コーパスを分散表現に変換したものを利用する．なぜなら，候文の文法は現代日本語のそれと比べて大きく異なり，文体特有の用語があるものの，現代日本語で用いられている文字および単語自体の意味および用法については現代と大差ないと考えられるためである．よって，現代語コーパスから作られた分散表現に候文の教師データを組み合わせて，NEE 可能な学習モデルを作成可能であると予想される．

本研究では，分散表現を用いた NEE を比較的容易に導入可能な自然言語処理ライブラリ Flair[5] を用いて，Wikipedia 日本語ダンプファイルを基に作られた Flair Embeddings と呼ばれる文字レベル分散表現と目録候文の教師データを実際に組み合わせて学習モデルを構築し，テスト用候文（以降，IOB2 データセットあるいは単にデータセットと呼ぶ.）に対して NEE を実行した．その際に，データセット数に応じた固有表現クラス（表 1）別の F 値 [6] を算出して，NEE 判定精度を評価した．

表 1 固有表現クラス名およびその説明.
Table 1 Named entity classes and the instructions.

固有表現クラス名		説明
Person/JINMEI	人名	人名，呼称
DATE	日時	日時を表す語
PLACE	場所	座標で指定可能な地名
EVENT	出来事	検索キーワードとなり得る語
ROLE	役職、役割	役職，家族関係
TERMS	候文用語	接続詞，定型句
QUANTITY	数量	数および単位を表す語

2. 関連研究

2.1 小城藩日記データベースの候文に対する NEE 関連

2018 年 4 月に公開された「小城藩日記データベース」[7] は，日記目録の全 73,984 記事文をくずし字からテキスト文に翻刻し，データベースにて検索可能にしたウェブサイトである．

江戸期において，我が国には領地と支配権の範囲を総称する「藩」と呼ばれる制度が存在した．全国で 250 程度の大名が治める藩があり，さらにそれぞれの藩には家臣にあたる支藩が連なっていた．支藩をふくめ，多くの藩では行政や出征，治安，冠婚葬祭行事など多様な日々の事柄を記した業務記録すなわち日記が編纂された．佐賀県にも，佐賀藩の支藩にあたる小城鍋島藩で作成された日記 85 年分が現在残存している．さらに，当藩ではその日記を基に，内容を要約し箇条書きにした目録類が用途別に作成された．数種類ある目録のなかで，「日記目録」と呼ばれる，全ての日記内容を時系列に目録化したものは，現在 122 年分と日記よりも多く残存している．この日記目録の存在により，江戸期から現代にわたり検索者は膨大で広範にわたる日誌内容から効率的に目当ての記事を探し出すことができる．

本サイトは，主に歴史研究者が小城藩の動向を史料から精査する目的で構築された．他方，小城市民にとっては郷土史データベースの側面を持つため，市民もまた積極的に目録記事文とその内容を閲覧し，共有する場ともなっている．

ただし，先に述べた通り，各目録記事文中の固有表現自体は現代日本語に共通するものが多いものの，候文特有の文法があり，特に人名および地名には多くの地域固有の単語が含まれている．本データベースを通じて記事内容の理解を促進するには，記事文に対し NEE を行うことで，人による理解はもとより機械的な分析を可能にする必要がある．

筆者らは先の研究で，江戸期の古典籍書誌から固有表現を得る目的で，NEE 技術を応用した Linked Data 自動変換システム [3] を構築した．本システムでは，形態素解析ツール MeCab を一部応用することで，単語を記事文から

© 2021 Information Processing Society of Japan

自動抽出し，あらかじめ設定した人名，地名，出来事，候文用語などの固有表現クラスに仕分ける．

ただし，地域色の濃い当時の語彙を Web 上で収集するのは一般に容易ではない．そこで，著者らは郷土資料研究者に加え，地域市民の協力によるクラウドソーシングで固有表現を収集し，MeCab ユーザ辞書に登録した．その結果，これまで困難であった地域の固有表現収集が可能となり，多くの記事文において NEE が実現した．

2.2 単語分散表現を用いた NEE 手法関連

単語分散表現は，「同じ文脈で出てくる言葉は似たような意味を持つ傾向がある」と考える分布仮説の考え方に基づいて考案された [8]．近年は，word2vec[9] を使った単語の意味の加減算例をはじめ，単語分散表現と呼ばれる数値化手法を用いた研究が盛んである．

さらに，周辺情報を得るだけではなく，単語が文中でどのように使用されているか，すなわち文脈情報を得る必要がある．そのためには，ある単語が複数の意味をもつことを考慮し，それぞれの意味が使用される確率を含めて分散表現を作らなくてはならない．

文脈が重視される NEE では，2018 年に，ELMo[10]，BERT[11]，Flair と呼ばれる，より実用的な手法が公開され，NEE への実用的な深層学習適用に革新をもたらした．これらは，Web などから得られる大量のテキストデータを用いて単語分散表現モデルを構築する事前学習と，少量の固有表現ラベルを付加した教師データで，どの単語に適切な固有表現を適用するかを学習する Fine-tuning を行う．特に，BERT と Flair は日本語を含む 54 言語，89 コーパスの分散表現を用いた NEE 精度比較で高い評価を受けている [12]．

そのうち，Flair は，単語のみでなく文字レベルからの単語分散表現データ構築を行う点が特徴である．このことは，1 節で挙げた課題として挙げた，形態素解析ツールでは未知語を一塊の単語として捉えられない場合に対処する問題を解決するために必要な要素である．加えて，Flair 公式 Github[*1]では，深層学習による NEE を少ないプログラミング工数で手軽に行える Python ライブラリ，プログラム例，および日本語の事前学習データである Flair Embeddings も公開しているため，導入が容易である．

3. Flair での候文 NEE 導入方法

候文に Flair で NEE を行うには，以下の流れで行う．

(1) 固有表現クラスのラベル付き教師データの収集
(2) 文字レベル IOB2 データセットの作成
(3) Flair ライブラリで学習モデル作成用プログラム構築および実行，F 値評価結果を取得

[*1] https://github.com/flairNLP/flair

城	B-JINMEI
州	I-JINMEI
様	O
本	B-PLACE
行	I-PLACE
寺	I-PLACE
へ	O
御	O
葬	B-EVENT
礼	I-EVENT
之	B-TERMS
事	I-TERMS

図 1 IOB2 形式データの例.
Fig. 1 An example of IOB2 format.

なお，作成したモデルには，任意の候文を適用させて NEE 結果を取得することができる．

まず，(1) の候文教師データは，全 73,984 目録記事文中前方 40,000 文までを対象とし，学習およびテスト用候文を合わせたデータセットを 40,000 文からランダムに選択して作成した．また，クラウドソーシングと管理者による MeCab ユーザ辞書用単語の整理および確認の後，あらためて NEE して得られた最新の結果を教師データとした．

次に，この教師データから以下に示す文字レベルの IOB2 形式データに変換する．IOB2 形式データでは，図 1 の通り，NEE 抽出した単語を 1 文字ずつに分割し，単語の先頭文字を B，それ以外を I，NEE として抽出しなかった文字を O とし，固有表現クラス（表 1）を付加してラベル付けを表現する．

最後に，Flair でほぼ定型化された学習モデル作成用プログラムを構築する．具体例は Google Colaboratory 上のサイト[*2]で示す．(2) で作成した候文の IOB2 データは Fine-tuning 用，(3) のプログラム中でダウンロードする Flair Embeddings[*3]は事前学習された分散表現データである．なお，本プログラム中で，IOB2 データセットは初期値でモデル生成用データとテスト用データに 9 対 1 の割合に自動分割される．学習モデル生成後は，プログラムに組み込まれている固有表現クラス別 F 値が出力される．

本研究では，学習モデルの規模と F 値の関係を調査するため，IOB2 データセット 40,000 文から，5,000 文刻みでランダムに選択したデータセットに対し，3 回ずつ学習モデルを構築して，総文字数，総文字種数および F 値の測定を行なった．

[*2] https://colab.research.google.com/drive/1vrxCIx2o-4GiP8zewZO18UngEAQ6Lfnv?usp=sharing
[*3] 本事前学習データは，Flair 公式 Github ページ (https://github.com/flairNLP/flair/blob/master/resources/docs/embeddings/FLAIR_EMBEDDINGS.md) にて取得可能．Wikipedia 日本語版ダンプファイル (2018/12/20) から Flair embeddings に構築された．

© 2021 Information Processing Society of Japan

図 2 各データセット数に対応する総文字数および総文字種数の関係．なお，それぞれの値は 3 回データセットを作成して得た平均．

Fig. 2 Relation between the total number of characters and the total number of character types corresponding to each number of datasets. Each value is the average of the three datasets.

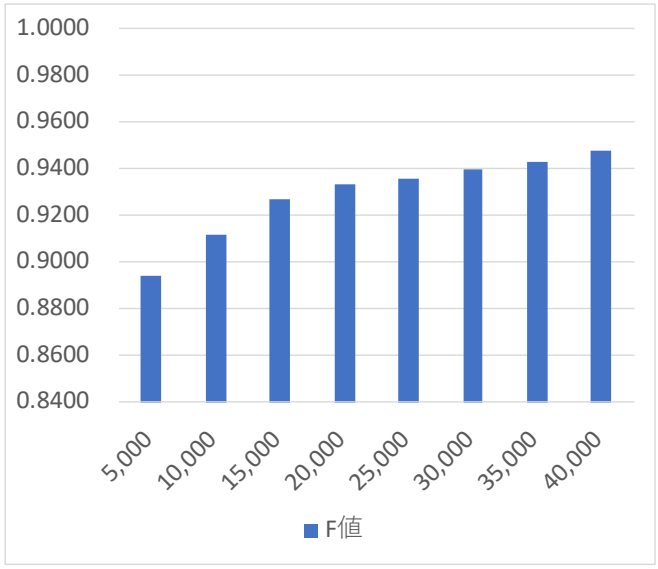

図 3 各データセット数に対応する F 値．なお，それぞれの値は 3 回データセットを作成して得た平均．

Fig. 3 F-score corresponding to each number of datasets. Each value is the average of the three datasets.

4. 結果

前節の手順で得た総文字数と総文字種数の関係を図 2 に示す．その結果，データセット量に比例して総文字数は増加する一方，総文字種数は 2,500 以下に留まった．

次に，各データセットにおける F 値の平均を図 3 に示す．この図から，データセットが 5,000 の時，固有表現クラス全体を通じた F 値は 0.8939 であり，40,000 文では 0.9474 まで精度が上がった．なお，データセット数を増やしても直線的に NEE 精度は上昇しなかった．

さらに，固有表現クラス別に F 値を精査した．その際に，図 4，図 5 および図 6 で示す通り，データセット別 3 回の測定ごとに求め，F 値の推移を可視化した．

その結果，候文用語（TERMS）が全てのデータセットの F 値平均約 0.9801 と最も高く，次いで人名（PERSON / JINMEI）が約 0.9343 であり，データセット数の増減にも大きく影響されなかった（図 4）．続いて地名（PLACE），出来事名（EVENT），ロール名（ROLE）の順に F 値平均がそれぞれ 0.9085，0.9007，0.8947 となり（図 5），データセットを 20,000 以上に増やすことで F 値は概ね 0.9 以上となった．

日時および数量の F 値平均（図 6）は，それぞれ 0.8633，0.8600 と，他のクラスと比較して最も低い結果を示した．データセットを 30,000 文以上にした時点で 0.9 に到達する結果も現れたが，同じデータセットの中でも試行の度にばらつきが見られた．

最後に，図 7 で，形態素解析では抽出困難な候文に対し，構築した学習モデルで NEE を実行する例[*4]を示す．実行例は，30,000 文の学習モデルを使用した．

5. 議論

Flair を介した提案手法により，事前学習済みの現代日本語の単語分散表現を利用することで，日記目録の候文に対する NEE を高精度かつ容易に実行することができた．Flair では BERT とは異なり，文字レベルから分散表現を構築する．そのため，候文を事前に単語分割することなく，単語抽出と NEE を同時に実行可能であり，今回の課題解決に大きな役割を果たしている．

また，今回の結果から，固有表現クラス別の F 値を測定した結果，クラスによって NEE の判定精度に大きな違いがあることが明らかになった．その際に，候文用語および人名の NEE については，10,000 以上のデータセットがあれば実用に値する精度で実行可能であることが示唆された．その一方で，日付および数量に関する NEE は他のクラスに比べて不安定な結果を示した．その理由は，人による判定の難易度そのものがクラスによって大きく異なり，概してそれが今回の F 値測定結果に表れたためと考える．

他方，GPU に高負荷がかかる深層学習では，データセット数が多いと相応の計算資源を必要とする．例えば，データセットが 30,000 以上の場合，無料で利用可能な Google Colaboratory 上[*5]では NEE に失敗した．そのため，より

[*4] Google Colaboratory のソースコードは `https://colab.research.google.com/drive/1_30rwEPSp6P5EOLzFn_glxN-wEX-Dw4I?usp=sharing` を参照のこと．
[*5] 最大 11GB の GPU を使用可能．

© 2021 Information Processing Society of Japan

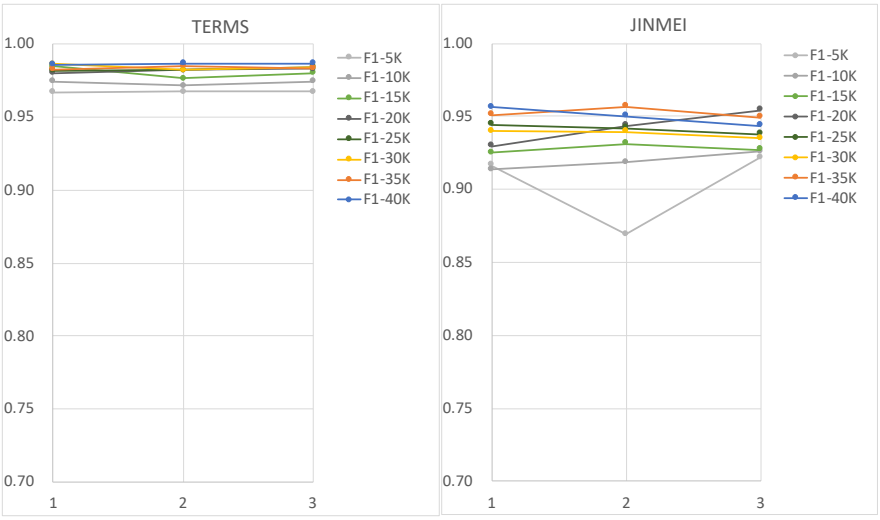

図 4 候文用語（TERMS）および人名（PERSON/JINMEI）の F 値推移．横軸は測定の順番，縦軸は F 値．例えば，F1-5K は，5,000 文での F 値で 1-3 回目の測定結果を表す．

Fig. 4 F-score transitions of TERMS and PERSON. ”F1-5K” represents a F-score based on 5,000 randomly chosen souroubun sentences. Each x-axis shows a trial.

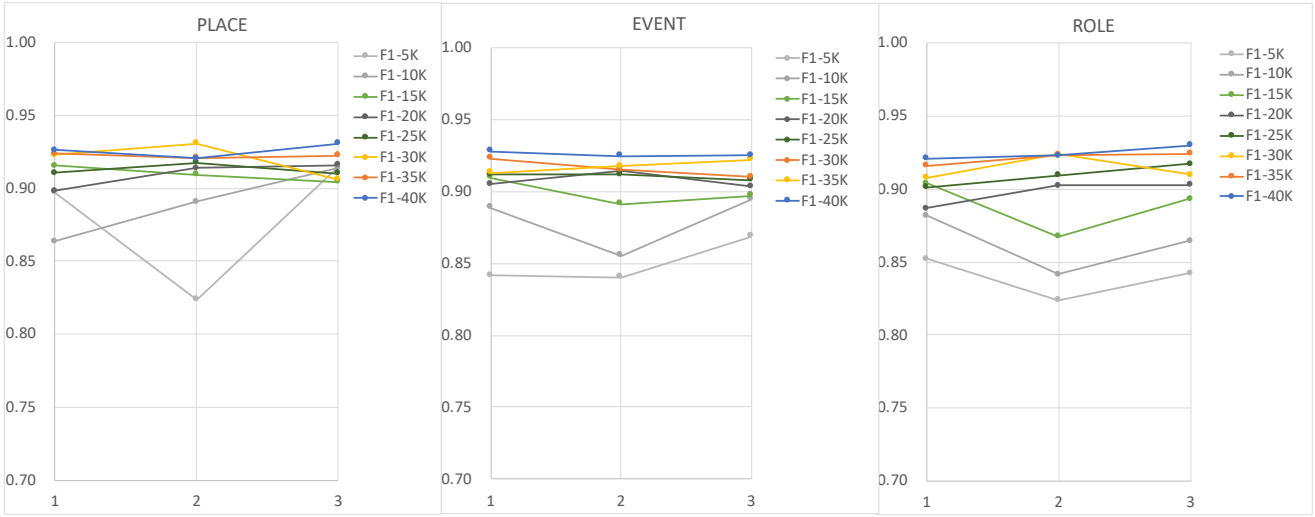

図 5 地名（PLACE），出来事（EVENT）ロール名（ROLE）および の F 値推移．

Fig. 5 F-score transitions of PLACE, EVENT and ROLE.

多くの GPU メモリを積んだ機器もしくはクラウドサービスの調達が必要となり，導入コストがかかる．また，Flair 学習モデルによる NEE は，形態素解析ツールの解析速度に比べて若干低速であることも課題である．

6. まとめ

我が国には，江戸時代に約 260 の藩が存在し，関連する武家には小城藩日記のような候文で書かれた古文書が数多く残存している．現状，それら古文書の翻刻が，クラウドソーシングや画像解析など IT 技術の補助で進められることについてはこれまでに一定の評価を得られている．新しい翻刻の様式が普及することで，今後は翻刻後のテキストの有効活用についても注目されると考える．

本研究では，翻刻後テキスト化された候文の読み解きを促進し，機械可読化して有効活用するため NEE を行うにあたり，形態素解析ツールでは困難な未知語の抽出と文脈を考慮した NEE を，深層学習の単語分散表現を用いる手法で実行した．

その際，導入が容易な単語分散表現ライブラリ Flair を用いて，候文中の単語抽出と NEE を同時に試みた．Flair は，BERT と同様に，通常膨大な計算量が必要な事前学習済みの単語分散表現を Web からダウンロードして学習モデル構築に利用可能であることに加え，独自に文字レベル分散表現である Flair Embeddings を利用することができる．その上，Flair ライブラリの利用で，学習モデルや検証に必要なプログラミングにかかる導入コストを低く抑えら

© 2021 Information Processing Society of Japan

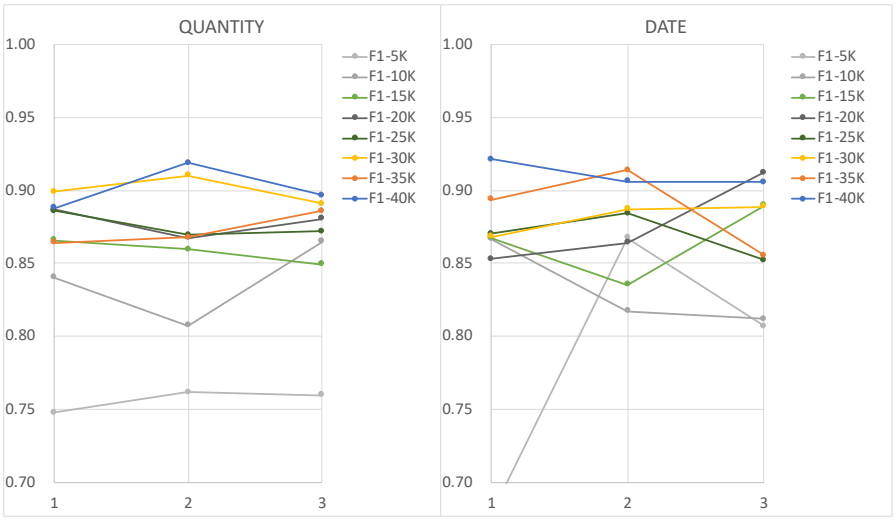

図 6 数量 (Quantity) および日時 (DATE) の F 値推移.
Fig. 6 F-score transitions of QUANTITY and DATE.

例）轟木久右衛門村岡太吉香田利三郎居附之事

[MeCab]

```
轟木　名詞,固有名詞,地名,一般,*,*,,,轟木,,轟木,,固,*,*,*,*,OGI_PLACE
久右衛門　　名詞,固有名詞,人名,一般,*,*,*,*,久右衛門,*,久右衛門,*,固,*,*,*,*,OGI_JINMEI
村岡　名詞,固有名詞,人名,一般,*,*,*,*,村岡,*,村岡,*,固,*,*,*,*,OGI_JINMEI
太吉　名詞,固有名詞,人名,一般,*,*,*,*,太吉,*,太吉,*,固,*,*,*,*,OGI_JINMEI
香田　名詞,固有名詞,人名,一般,*,*,*,*,香田,*,香田,*,固,*,*,*,*,OGI_JINMEI
利　　名詞,普通名詞,一般,*,*,*,リ,利,利,リ,利,リ,漢,*,*,*,*,リ,リ,リ,リ,*,*,"1,0",C3,*
三郎　名詞,固有名詞,人名,一般,*,*,*,*,三郎,*,三郎,*,固,*,*,*,*,OGI_JINMEI
居　　名詞,普通名詞,一般,*,*,*,キョ,居,居,キョ,居,キョ,漢,*,*,*,*,キョ,キョ,キョ,キョ,*,*,1,C3,*
附　　名詞,普通名詞,一般,*,*,*,,,附,,附,,和,*,*,*,*,OGI_ROLE
之事　接尾辞,名詞的,一般,*,*,*,,之事,之事,,之事,,和,*,*,*,*,OGI_TERMS
```

[Flair]

```
轟 <B-JINMEI> 木 <I-JINMEI> 久 <I-JINMEI> 右 <I-JINMEI> 衛 <I-JINMEI> 門 <I-JINMEI>
村 <B-JINMEI> 岡 <I-JINMEI> 太 <I-JINMEI> 吉 <I-JINMEI>
香 <B-JINMEI> 田 <I-JINMEI> 利 <I-JINMEI> 三 <I-JINMEI> 郎 <I-JINMEI>
居 <B-ROLE> 附 <I-ROLE>
之 <B-TERMS> 事 <I-TERMS>
```

図 7 テスト用候文（小城藩日記データベース登録番号：72701）に対する MeCab および Flair での NEE 実行例.
Fig. 7 Example of NEE execution with MeCab and Flair for a test sentence.

れる．今回，日本語版 Flair Embeddings と比較的文例の少ない候文の教師データを組み合わせることで，NEE を F 値 0.9 以上の高精度で実行可能であることを示した．

特に，古文書の読み解きに最も重要な固有表現クラスである人名と，人にとっても通常一定の学習による慣れが必要な候文用語の抽出は，少ない教師データで高精度に実行できることが示された．

ただし，提案手法の深層学習のみで NEE を実行するには，候文サンプルの収集，解析実行時間の短縮などに課題がある．

今後は，日記目録以外の古文書についても広く提案手法の導入を試み，予備知識がなくても古文書の NEE で文の

© 2021 Information Processing Society of Japan

内容を人はもとより機械的にも把握可能な仕組みを創りたいと考えている．

謝辞 本研究は，JSPS 科研費 JP19K20630 の助成を受けたものである．

参考文献

[1] 峰岸 明：国史大辞典（候文体），Vol. 8, 吉川弘文館 (1987).
[2] Kudo, T.: MeCab: Yet Another Part-of-Speech and Morphological Analyzer (ver. 0.996), , available from ⟨http://taku910.github.io/mecab/⟩ (accessed 2021-01-17).
[3] 吉賀夏子，只木進一：古典籍書誌データ構造に対応した Linked Data への半自動変換，情報処理学会論文誌，Vol. 59, No. 2, pp. 257–266 (2018).
[4] 吉賀夏子，只木進一：低コストな Linked Data 化を目指したクラウドソーシングによる固有表現収集の試み，じんもんこん 2019 論文集，Vol. 2019, pp. 239–244 (2019).
[5] Akbik, A., Blythe, D. and Vollgraf, R.: Contextual string embeddings for sequence labeling, *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 1638–1649 (2018).
[6] Wikipedia contributors: F-score — Wikipedia, The Free Encyclopedia, Wikipedia foundation (online), available from ⟨https://en.wikipedia.org/w/index.php?title=F-score&oldid=997534712⟩ (accessed 2021-01-17).
[7] 佐賀大学地域学歴史文化研究センター：小城藩日記データベース，, 入手先 ⟨https://crch.dl.saga-u.ac.jp/nikki/⟩（参照 2021-01-17）.
[8] Harris, Z. S.: Distributional structure, *Word*, Vol. 10, No. 2-3, pp. 146–162 (1954).
[9] Mikolov, T., Chen, K., Corrado, G. and Dean, J.: Efficient Estimation of Word Representations in Vector Space (2013).
[10] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. and Zettlemoyer, L.: Deep contextualized word representations, *Proc. of NAACL* (2018).
[11] Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding, *arXiv preprint arXiv:1810.04805* (2018).
[12] Straka, M., Straková, J. and Hajič, J.: Evaluating Contextualized Embeddings on 54 Languages in POS Tagging, Lemmatization and Dependency Parsing (2019).

© 2021 Information Processing Society of Japan