拡張固有表現の分類とテンプレートによる案内文の一考察

石塚大貴 [1]　　中沢実 [1,a)]

近年 AI への関心が高まり，同時に文章生成方法も注目を浴びている．その方法のほとんどが時系列データに基づく統計的な分析であり，文章全体を対象に生成を行うため，特定の対象の案内文を生成しようとすると，適切な文章が得られないことが多い．そのため，私は拡張固有表現を用いて生成対象の文章をうまく表現できないか考えた．本論文では，まず Wikipedia の記事から各拡張固有表現を識別する方法について述べる．その後，拡張固有表現を用いた観光地の案内文の生成方法の検討について記述する．

1. はじめに

近年，AI への関心が高まり，同時に文章生成方法も注目を浴びている．その方法のほとんどが時系列データに基づく統計的な分析であり，文章全体を対象に生成を行うため，特定の対象の案内文を生成しようとすると，適切な文章を得られないことが多い．そのため，本研究は拡張固有表現を用いて生成対象の文章をうまく表現できないかを考察した．本論文では，まず，Wikipedia から拡張固有表現を分類するための方法について述べる．その後，拡張固有表現を用いた観光地の案内文の方法の検討について記述する．

2. 固有表現・拡張固有表現について

固有表現とは，文章中に存在する単語を重要な要素としてカテゴリに分類した単語である．アメリカ合衆国の DARPA が組織した評価プロジェクトである MUC (Message Understanding Conferences) [1] では，組織名 (ORGANIZATION)，人名 (PERSON)，地名 (LOCATION)，日付表現 (DATE)，時間表現 (TIME)，金額表現 (MONEY)，割合表現 (PERCENT) の 7 種類の固有表現が定義されている．また，日本国内の評価ワークショップである IREX (Information Retrieval and Extraction Exercise) [2] では MUC が定義した 7 種類の固有表現と固有物名 (Artifact) の 8 種類を定義されている．

また，関根により固有表現を拡張した拡張固有表現 [3] が提唱されている．固有表現と比べたときの，拡張固有表現の利点は 2 つある．1 つ目は，固有表現の種類の定義が MUC や IREX よりも多いことである．固有表現として定義された種類数は，MUC で 7 種類，IREX で 8 種類しかないのに対して，拡張固有表現では 200 種類以上の固有表現が定義されている．また，各固有表現の間には親子関係がある．例えば，「製品名」の子にあたる固有表現には「美術作品名」が含まれており，さらにその子に当たる固有表現には「映画名」や「番組名」などが含まれている．この 2 つの利点により，拡張固有表現を用いると，単語を細かく分類できるため，文章の分析をするのに役立つ．

しかし，日本語拡張固有表現をカテゴリ別に分類したデータセットとして公開されているものは存在しない．そこで，本研究では Wikipedia [4] を用いた拡張固有表現の分類について実施．

3. 拡張固有表現獲得について

拡張固有表現を分類するための準備ついて説明する。

Wikipedia は，インターネット百科事典で誰でも加筆・編集することができる．そのため，最新の情報が掲載されている場合が多い．また，記事には Fig.1 のようにカテゴリや記事同士のリンクが存在したり，編集を効率化や記事の情報整理をしたりするためにテンプレートが使用されている．

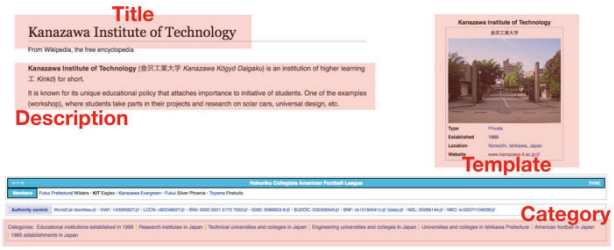

Fig. 1 Feature of Wikipedia Article

これらの特徴から拡張固有表現を分類するためのデータとして Wikipedia は適していると考えた．そのため，本研究では Wikipedia の記事に対して，拡張固有表現のタグ付けを人手で行い，残りの記事についてはニューラルネットワークを用いてタグ付けを行う．

記事一つひとつに対して人手でタグ付けするのは，時間や労力がかかり，タグ付けする複数の人の間での方法を統一することが難しい．これらの問題を考慮し，Fig.2 に示すタグ付けツールを作成した．

Fig. 2 The Management Tool of Tagging with Named Entities

このツールは Wikipedia の記事に対して拡張固有表現をタグ付けするためのものである．タグ付け期間中は学内のネットワークを介してアクセスすることができるようにし，20 名の研究室メンバーの協力を得た．その結果，37,290 記事に対して 144 種類の拡張固有表現をタグ付けることができた．

4. 各拡張固有表現における記事の統計

ニューラルネットワークによる拡張固有表現の分類を行う前に，精度の高い分類を行うために，各拡張固有表現の

1 金沢工業大学
KIT, Nonoichi, Ishikawa, 921-8501, Japan.
a) minoru.nakazawa.jp@ieee.org

記事の特徴を分類した．その結果，記事中の特徴として拡張固有表現の分類に対して有効的だと感じた「タイトル」，「カテゴリ名」，「リンク先の記事のタイトル」，「使用しているテンプレート」を特徴にすることにした．次に各特徴の統計データの例を示す．

4.1 タイトルについての統計

Table1 に各拡張固有表現の記事タイトルに含まれる文字の頻出が多いものを集めたデータを示す．例として，拡張固有表現の中で比較的分かりやすい 5 種類をピックアップしている．以下，他の特徴についても例として同じ 5 種類についてみていく．Table 1 を見ると各固有表現の特徴がよく表れているのがわかる．

Table. 1 Frequent Characters Ranking Making up Titles

	Person	God	Spa	Province	Park
1	・	神	温	州	園
2	―	天	泉	の	公
3	の	命	湯	省	会
4	田	の	川	―	―
5	子	大	県	県	総

4.2 カテゴリ名についての統計

Table2 に各拡張固有表現の記事のカテゴリ名に含まれる文字の頻出が多いものを集めたデータを示す．タイトル以外の特徴は，ひらがなやカタカナがあまりにも多いため一文字として扱っている．タイトル同様に拡張固有表現の特徴がよく表れている．

Table. 2　Frequent Character Ranking Making up Category

	Person	God	Spa	Province	Park
1	Hiragana	神	Hiragana	州	Hiragana
2	人	Hiragana	温	Hiragana	園
3	物	日	泉	Katakana	公
4	―	本	地	国	市
5	学	津	県	省	地

4.3 リンク先の記事タイトル

Table3 に各拡張固有表現の記事の「リンク先の記事タイトル」に含まれる文字の頻度が多いものを集めたデータを示す．リンク先の記事タイトルでは数字も多いため一文字として扱っている．「リンク先の記事タイトル」は上記 2 つの特徴よりも特徴が出ていないが，これはリンクしている記事が複数存在し，年月のページにリンクさしている場合が多いためである．しかし，比率を見ると十分に特徴と言えると考えたため，今回の研究では採用することにした．

Table. 3 Frequent Charactores Ranking Making Up Titles of Linked Articles

	Person	God	Spa	Province	Park
1	―	神	泉	―	Number
2	Katakana	社	温	Katakana	年
3	Number	―	Number	Number	Katakana
4	・	日	年	・	市

4.4 使用しているテンプレートについての統計

Table4 に各拡張固有表現の記事の「使用しているテンプレート」の上位 5 つを示す．他の特徴に比べてかなり，特徴があるのがわかる．

Table. 4 Template Usage Frequency Ranking

	Person	God	Spa	Province	Park
1	Reflist	Shinto-stub	日本の温泉地	Refilist	公園
2	Normdaten	Reflist	温泉	仮リンク	Reflist
3	生年月日と年齢	神道	脚注ヘルプ	Lang	Commonscat
4	脚注ヘルプ	脚注ヘルプ	Commonscat	Commonscat	Pref-stab
5	Cite web	神道 横	ウィキポータルリンク	Cite Web	Cite web

4.5 統計についてのまとめ

4 つの特徴について詳しく考察すると，拡張固有表現を分類するための特徴としては十分であることがわかった．

5. 分類タスクの定義と検証・評価方法

本章では，ニューラルネットワークの構造について記述する前に前提としての分類タスクの定義と検証・評価方法について触れる．

5.1 分類タスクの定義

分類タスクでは，インプットとして各記事の特徴をニューラルネットワークに入力し（特徴の変形については次章で説明する），アウトプットとして拡張固有表現か否かを拡張固有表現ごとに出力する．注意しなければならない点は，１つの記事に複数の拡張固有表現が割り当てられる場合があることである．例えば，「トマト」は「植物名」と「食べ物名」の 2 つの拡張固有表現に分類される．つまり，本研究で扱う分類タスクは，多クラス分類である．

5.2 検証方法

本研究では検証方法として，Stratified K-Fold Cross Validation (SCV) [5] を使用する．この手法ではデータセットの標本を K 個に分割し，そのうち 1 つをテスト事例として，残りの K-1 個をトレーニングデータとして学習させる手法である．また，分割の方法として，各クラスができるだけ均等になるように分割するのも特徴である．本研究では，K = 5 とする．

5.3 評価方法

本研究では分類結果の評価方法として Area Under the Curve (AUC) [6] を使用する．AUC は ROC 曲線を用いて算出される方法で，多クラス分類の際に用いられる評価方法である．

6. ニューラルネットワーク構造

本章では，拡張固有表現を分類するためのニューラルネットワークについて説明する．

6.1 特徴の変形

4 種類の特徴の変形について説明する．また特徴を変形後のベクトルはニューラルネットワークのインプットとして扱う．

6.1.1 タイトルの変形

記事のタイトルは拡張固有表現そのものを表している重要な部分なため，並び順を考慮したいと考え，特徴の変形には工夫を凝らした．変形手法は，Convolution Neural Network (CNN) [7] のフィルタ処理から着想を得た．まず，タイトルを 32x32 の行列に変形する．変形手順は以下の通りである．

1) 横方向には,タイトルの各文字を UTF-8 に変換する．各文字が 4 バイトに満たない場合は，左側に全体で 32bit になるように 0 で埋める．
   例えば,「大」を UTF-8 の 16 進数で表すと 0xE5A4A7 となるが，これを 0x00E5A4A7 と置く．
2) 縦方向には，変形した文字列を縦に並べる．文字数が 32 文字の場合はぴったりと収まる．タイトルの文字列が 32 文字より短い場合は，全体 32 行になるまで上側を行単位で 0 埋める．タイトルの文字列が 32 文字よりも長い場合は，タイトルの文字列を後ろから 32 文字分だけ使う．

次に 32x32 の行列を CNN のフィルタ処理の要領でフィルタにかける．Fig.3 にフィルタ処理のイメージを示す．使用するフィルタは，3x32 のフィルタを 32 種類，4x32 のフィルタを 32 種類，5x32 のフィルタを 32 種類，計 96 種類のフィルタを使用する．各フィルタは正規分布に従い初期化を行なっている．フィルタ処理のストライドは 1 とし，フィルタ適用後は max pooling にかけて 96 次元に変換する．

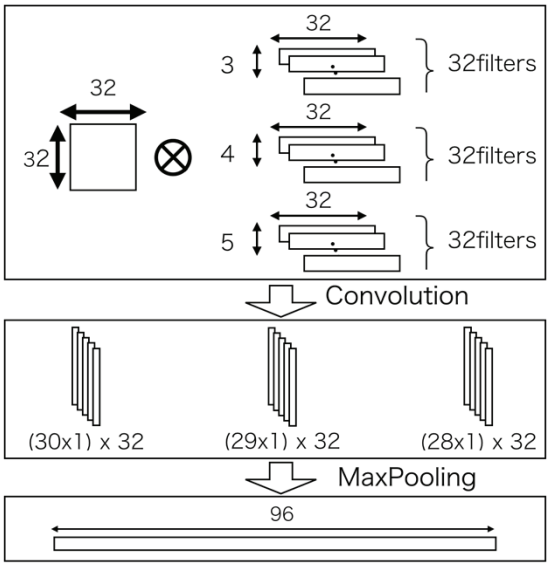

Fig. 3 The Image of Filtering

6.1.2 カテゴリ名の変形

Wikipedia では，1 つの記事が複数のカテゴリに属することが多い．変換方法は，まず各カテゴリ名の文字列を文字に分解し，次にカテゴリ毎にその文字が含まれているかいないかで確率に変換する．例えば,「りんご」の記事を見たとき,「果物」と「食物」のカテゴリに属していたとする．ただし，果: 1/2 = 0.5，物: 2/2 = 1.0，食: 1/2 = 0.5 といった具合である．対象の文字はカテゴリ名で頻出の高い上位 192 文字とする．このことから，この変換によって各記事の特徴は 192 次元に変換される．

6.1.3 リンク先の記事タイトル

Wikipedia では，1 つの記事から複数の記事へリンクが貼っていることが多い．変換方法は，特徴として「リンク先の記事タイトル」を使用すること以外は「カテゴリ名」の特徴の変換方法と同じである．そのため説明は省略する．

6.1.4 使用しているテンプレート

Wikipedia では記事を作成するとき，複数のテンプレートを使用している場合が多い．変換方法は，テンプレート名の文字を扱うのではなく，そのテンプレートが使用しているか否である．本研究で，対象のテンプレートは，全記事で使用率の高い 192 種類のみである．つまり，192 次元の特徴へと変換される．

6.1.5 各特徴を結合

変形後の特徴のまとめとして,変形後の各特徴数を Table 5 に示す．各特徴を変形した後はニューラルネットワークのインプットとして入力を行う．

Table. 5 The number of Dimension of Each Feature After Conversion

The Name of Feature	Dimension
Title	96
Category Name	192
Title of Linked Article	192
Used Template	192
Total	**672**

6.2 ニューラルネットワークの構造

拡張固有表現の認識のためのニューラルネットワークの構造 Fig.4 に示す．このニューラルネットワークは，インプットとして結合した 672 次元特徴を受け取り，アウトプットとして 144 次元のベクトルで返す．アウトプットでは，入力された記事の特徴が各拡張固有表現に当てはまるかを推測された値を 0〜1 範囲で返す．活性化関数として Sigmoid 関数 [8] を，最適化手法として Adaptive Moment Estimation (Adam) [9] を採用した．

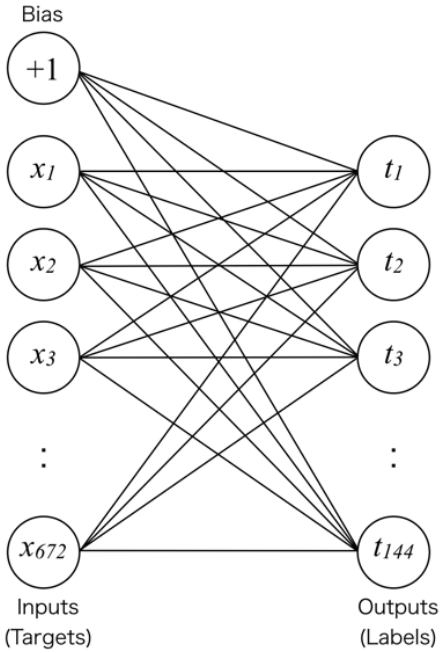

Fig. 4 The Structure of Neural Network for Classification

7. 拡張固有表現の分類結果

本章では，ニューラルネットワークでの学習した結果を示す．

7.1 AUC(Area under ROC curve)

Table6 に学習結果の SCV における各 Fold における

AUC の値を示す．小数点第 5 桁以下は四捨五入で丸めてある．AUC は 0.9 以上だと非常に良い精度であると言われているため，良い結果が得られたといえる．

Table. 6 The Results of Classification

Fold #	1	2	3	4	5	Avg.
AUC	0.9689	0.9688	0.9687	0.9689	0.9689	**0.9687**

7.2 結果例

Table7 に分類がうまくいった拡張固有表現の単語例を，Table 8 に分類がうまくいかなかった拡張固有表現単語例を示す．

分類がうまくいった例を見ると，以下のような点が考えられる．

1) 学習のためのデータセットとして，タグ付けされた単語例が比較的十分である．具体的には 80 記事以上タグ付けされた拡張固有表現に関しては，比較的分類精度が高い．
2) Wikipedia の中で特徴的な「カテゴリ」を使用している場合が多い．例えば，人名では「○○学者」や「○○の人物」「○○人」「○○家」，温泉名では，「○○県の温泉」，都道府県州名では，「○○の州」や「○○の県」，会社名では「○○メーカー」や「○○の企業」などである．
3) Wikipedia の中で特徴的な「テンプレート」を使用している場合が多い．例えば，人名では職業に関するテンプレート，温泉名では「日本の温泉地」，都道府県州名では，「○○の州」や「Infobox settlement」，会社名では「基礎情報 会社」などである．
4) 記事タイトルに注目的な文字列が含まれているものが多い．具体的には，人名では「・」，温泉名では「温泉」，都道府県州名では「州」や「県」，会社名ではカタカナや英語などである．

一方で，分類がうまくいかなかった例では，以下のような点が考えられる．

1) 対象の拡張固有表現に当てはまる Wikipedia の記事がそもそも少ない．
2) Wikipedia の「設定されているカテゴリが広義を表す意味である場合」や「割り当てられているカテゴリ数が少ない記事」が多い．例えば，神名では「○○の神社」や「○○の歴史」「○○の建造物」，公園名では「○○市」や「○○の施設」，「○○法」や「○○社会」などである．
3) テンプレートを使用していない記事が多い．
4) タイトルに特徴的な文字が含まれていない記事が多い．

また，拡張固有表現に関わらず記事によっては分類精度が優れないものがある．その特徴的な記事を以下にまとめる．

1) タイトルが異常に長い記事．
2) 情報量が少なすぎる記事．
3) カテゴリ設定が存在しない記事．
4) 別の記事に転送されている記事．

これらの考察から，本研究で考案した手法では Wikipedia の記事や拡張固有表現によって，分類精度にかなりの差が出ることがわかった．

また，どの拡張固有表現にも属さない記事についてのデータセットを作成していないため，拡張固有表現を割り当てるべきでない記事が，複数の拡張固有表現に割り当てられてしまう事象が起こっている．今後はこのようなことが起こらないように，他クラス分類ではどのクラスにも属さないといったラベルも必要であるということを認識した．

Table. 7 Examples of Words when Classification Succeed

Named Entity Name	Example Words
Person	フリッシュ，ドミニク・ピノン，宇井あきら，エリックディッカーソン，高塚清一 etc.
Spa	武芸川温泉，平内海中温泉，三船温泉，鉛温泉，奥津温泉 etc.
Province	ダッカ県，アルコロン州，ナジャフ県，カラブリア州，ルクワ州 etc.
Company	ガイアノーツ，タバック，ユニフード，JUNKEI-GLOVE，KINGMAX etc.

Table. 8 Examples of Words when Classification Failed

Named Entity Name	Example Words
God	穴師坐兵主神社，神田みか，マチャミ，マミーソース，中社古墳 etc.
Park	和泉本町，神田須田町，ミヤコーバス塩釜営業所，名名古屋市山田図書館，広田専用軌道 etc.
Offence	特定情報提供役務，障害者基本計画，経営学検定，おかしな二人 etc.
Flora Part	生物科学，計画的陳腐化，水酸化鉱物，榎木洋子，ジェノヴァのレ・ストラーデ・ヌオーヴェとパラッツィ・デイ・ロッリ制度 etc.

8. テンプレートによる案内文の生成方法

本章では，分類した拡張固有表現を使用する応用例としてテンプレートによる案内文の生成方法について述べる．

8.1 GeoApp の開発

今回は実際の利用状況を想定できるようにするため，スマートフォンで使えるアプリ「GeoApp」を作成した．Fig.5, Fig.6, Fig.7 に GeoApp のスクリーンショットを示す．

Fig.5 はアプリを立ち上げたときはじめに表示される画面である．この画面では，実際にユーザがどこにいるのかを設定することができる．研究では，実際に現地に行くのが面倒なため，対象の場所（現在地）を自由に設定できるようにしている．このアプリでは“場所”のことを“Geo”と呼んでいる点に注意しなければならない．また，このアプリは，API サーバを必要とし実際に外部に API サーバを設置するとなるとコストがかかるため，localhost に設置している．補足として，API サーバは研究のため我々が開発したものである．

Fig.6 は現在地を「兼六園」に設定したときの周辺のスポットをリスト形式で表示している画面である．周辺のスポットの取得は API サーバから行っている．API サーバでは，Twitter [10] から位置情報付きのツイートの中で「I'm

at」で始まるものだけを抽出し，観光地リストに置き換えている．実際には Swarm [11] や Foursquare [12] といった位置情報共有アプリを使用したユーザが Twitter と連携しているために Twitter にもツイートとして流れてくる．Swarm や Foursquare の API は制限が厳しく多くのデータを取得することが困難なため，一度に 100 ツイートまで取得できる Twitter を使用している．

Fig. 5 Screenshot #1 of GeoApp

Fig. 6 Screenshot #2 of GeoApp

Fig.7 は周辺スポット一覧で「兼六園」をタップしたときに表示される画面である．この画面では、スポットの名前や緯度経度，スポットをマップで表示するためのボタン，生成された案内文が表示される．「マップで開く」ボタンを押すと，スマートフォンで特に設定していない限り iOS では「マップ」，Android では「Google Maps」でスポットが表示される．案内文は API サーバから取得している．

8.2 テンプレートによる案内文の生成

テンプレートによる案内文の生成方法について記述する．案内文の生成方法は，Web 上から案内文の対象となるスポットの情報を見つけてきてそれをテンプレートに当てはめる形である．

まず，テンプレートの形式について説明する．テンプレートの形式は，拡張固有表現の名称を {} (中括弧) で囲った形で記述する形である．また，複数の種類の拡張固有表現が入る可能性がある場合には，| (縦棒) で各拡張固有表現を区切るように記述する．例えば，「{City} の {Dish} はとても美味しい」というテンプレートがあり，Web 上から市区町村名として金沢市，料理名としてカレーライスが取得できたとする．この場合は案内文として「金沢市のカレーライスはとても美味しい」が生成される．また，複数の種類の拡張固有表現が入る可能性のあるテンプレートについても説明する．例えば，「{Character|Dish} はとても人気があります」というテンプレートがある場合，中括弧には，ひゃくまんさん（キャラクター名）やカレーライス（料理名）の両方を入れることができる．

Fig. 7 ScreenShot #3 of GeoApp

Web からの情報は 2018 年 12 月 31 日の Wikipedia の本文を対象としている．最初は Google 検索 [13] のトップ 5 の Web ページやツイートからの取得も考えていたが，コンテンツのみの抽出を行う必要があり，構造化のされ方が千差万別なため今回は断念した．

次に学習した拡張固有表現と Web から取得したコンテンツを照らし合わせる方法について述べる．取得したコンテンツは，形態素解析ツール MeCab [14] によって単語分割を行う．MeCab の辞書として Wikipedia 中の単語の形態素解析にも対応した辞書 mecab-ipadic-NEologd [15] を使用する．この方法により単語分割を行い，学習した拡張固有表現と分割された単語がマッチするのかどうかを検証する．マッチした場合には，スポットごとにテンプレートに当てはまる拡張固有表現として登録を行っている．

8.3 案内文生成の例

本研究では，実際に兼六園，東京タワー，東京ディズニーランド（以下ディズニーランドと略す）の 3 つのスポットに対して同様のテンプレートを使って生成文を作成して結果を考察した．Table 9 にテンプレートによる生成例を示す．表よりテンプレートに拡張固有表現を当てはめると簡単な文であれば，それらしい案内文を生成することができることがわかる．しかし，案内文として役に立つテンプレートを考えることは非常に難しく，どのスポットにも当て

はまるものでなくてはならない．また，あるスポットに対して同じ種類の拡張固有表現の単語が複数存在する場合，今回の研究では頻出度の一番高い単語を採用する．しかし，この方法だと文脈にあったものが選ばれるとは限らない．例えば，利用料金がそれぞれ正しくない．そのため，テンプレートによる案内文の生成では具体的な案内を作成するのは非常に難しい．この問題を解決するためには，Web から取得したコンテンツの文脈関係をコンピュータに理解させなければならない．テンプレートのルールももっと複雑にしなければ対応できないと考えられる．

Table. 9 Examples of Generated Guidance Sentences

Template	Spot	Generated Sentence
{Province} の {City}に位置しています	兼六園	石川県の金沢市に位置しています
	東京タワー	東京都の千代田区に位置しています
	ディズニーランド	千葉県の浦安市に位置しています
利用料金は {Money}です	兼六園	利用料金は 310 円です
	東京タワー	利用料金は 500 円です
	ディズニーランド	利用料金は 500 円です
{Railroad|Road}で行けます	兼六園	北陸新幹線で行けます
	東京タワー	大江戸線で行けます
	ディズニーランド	国道 357 号線で行けます
近くには {Station|Airport}などがあります	兼六園	近くには金沢駅があります
	東京タワー	近くには浜松町駅があります
	ディズニーランド	近くには成田国際空港があります

9. まとめ

本論文では，まず「Wikipedia の記事から拡張固有表現を分類方法」について述べ，次に応用例として分類した拡張固有表現を用いて「テンプレートによる案内文を生成方法」について述べた．

Wikipedia の記事から拡張固有表現を分類する方法では，まずを作成したツールを用いて分類のためのデータセットの作成を行い，拡張固有表現を割り当てた記事に対して統計的に分析を行った．その後，記事の特徴の行列変換に工夫を凝らし，そのデータをインプットして，対象の記事がどの拡張固有表現に当てはまるかを分類するためのニューラルネットワークを構築した．その結果，AUC による評価では，かなりの精度を得ることができた．しかし，拡張固有表現や記事によって分類精度にかなりの差があり，うまく分類できていない記事も多い．改善点として「拡張固有表現にタグ付けすべきでない記事もデータセットに含めること」や「ニューラルネットワークのインプット部分で，“カテゴリ名”や“リンク先の記事タイトル”の特徴変形を文字の頻出度で行うのではなく，文字の並びを考慮した変形処理を行ったり，文字単位ではなく単語単位での変形処理を行ったりすること」などが挙げられる．

テンプレートによる案内文の生成方法では，分類した拡張固有表現を用いてテンプレートによる案内文の生成を行った．GeoApp を作成することで拡張固有表現の利用状況を想定することができた．テンプレートにより案内文の生成は簡単なものであればできることを確認した．しかしながら，どのスポットにも当てはまるようなテンプレートを考えることは難しく，同じ種類の拡張固有表現の単語が複数あった場合には，どの単語適切かを見極める必要がある．実際に使えるアプリにするためには更なる改善が必要だと思われる．

本研究では総合的に様々な技術を用いて拡張固有表現について考察した．研究を行う過程で，拡張固有表現の抽出は自然言語処理の分野で非常に役に立つ項目であると認識することができた．また，解決すべき課題や作成するアプリケーションに対応していくために，今後は固有表現をより使いやすいものにアレンジしていく必要があると考える．また，迅速に固有表現を使った問題解決プロダクトを開発する場合，人間が全ての固有表現を定義するのではなく，ある程度自動的に分類・識別・抽出できる技術が必要である．本研究はこのような状況に対して解決できるような参考を与え，自然言語処理による問題解決の幅が少しでも広がることを今後の課題としている．

参考文献

1) Ralph Grishman and Beth Sundheim: “Message understanding conferences-6: A brief history,” 16th Conference on Computation linguistics, 1996, pp.466-471.
2) Satoshi Sekine and Hitoshi Isahara: “Japanese named entity evaluation: analysis of results,” COLING ’00 Proceedings of the 18th conference on Computational linguistics – Volume 2, 2000, pp1106-1110.
3) 関根の拡張固有表現層:「関根の拡張固有表現階層 -7.1.1- 」 , (https://sites.google.com/site/extendednamedentity711/, 2019 年 7 月 10 日アクセス)
4) Wikipedia:「Wikipedia」(https://ja.wikipedia.org/，2019 年 7 月 10 日アクセス)
5) scikit-learn 0.20.2 documentation: 「sklearn.model_selection.StratifiedKFold」, (https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html, 2019 年 7 月 10 日アクセス)
6) Anthony K Akobeng: “Understanding diagnostic tests 3: receiver operating characteristic curves,” Acta Paediatr, 2007, pp.644-647.
7) Ian Goodfellow, Yoshua Bengio and Aaron Courville: *Deep Learning*, 2016, pp.326-339.
8) Jun Han and Claudio Moraga: “The Influence of the Sigmoid Function Parameters on the Speed of Backpropagation Learning,” International Workshop on Artificial Neural Network: From Natural to Artificial Neural,1995, pp.195-201.
9) Diederik Kingma and Jimmy Ba: “Adam: A method for stochastic optimization,” arXiv 1412.6980, 2014, pp.1-15.
10) Twitter:「Twitter」, (https://twitter.com/, 2019 年 7 月 10 日アクセス)