外国人名のカタカナ表記自動推定システムの作成

安江 祐貴　　佐藤 理史　　松崎 拓也
名古屋大学大学院工学研究科

1　はじめに

2020年に開催される東京オリンピックでは，外国からの参加者(選手および役員)の名前を，現地の言語つまり日本語で表記することが求められている．これは，漢字圏を除くほとんどの国からの参加者の人名を，カタカナで表記することを意味する．このためには，原語をカタカナ表記に翻訳すること，すなわち，トランスリタレーション(transliteration)が必要となる．

コンピュータを用いた自動トランスリタレーションの研究は，Knightらの研究[1]を始めとして，多くの蓄積がある[2]．しかしながら，日本の報道や放送といった，多数の外国人名を翻訳しなければならない現場においては，自動トランスリタレーションの技術は全く使われておらず，いまだに人手の翻訳に頼っているのが現状である．

オリンピックの場合，1万人を超える参加者の日本語訳(大多数は，カタカナ表記)を準備する必要があるが，参加者のアルファベット表記の名簿が公開されるのは開催の数週間前であり，これを全て人手で翻訳するのは時間的に厳しい．また，参加国は200国・地域を超えるため，原言語も多様である．このため，色々な手段を使って，参加予定者の名簿を作成して事前翻訳を準備するが，事前翻訳で完全にカバーできるわけではない．そのため，外国人名の翻訳作業に対して，コンピュータによる支援が求められている．

このような背景により，我々は，本年度よりコンピュータ支援による外国人名カタカナ表記の標準化・統一化に関する研究を開始した．本稿では，この研究における自動トランスリタレーションの実装について報告する．

2　支援システム

2.1　システムの全体像

本研究で作成予定のシステムの全体像を図1に示す．本システムの入力は，外国人名 (アルファベット表記) とその人物に関する情報(国籍，性別) であり，出力は外国人名 (カタカナ表記) である．システムの機能は，大きく次の2つからなる．

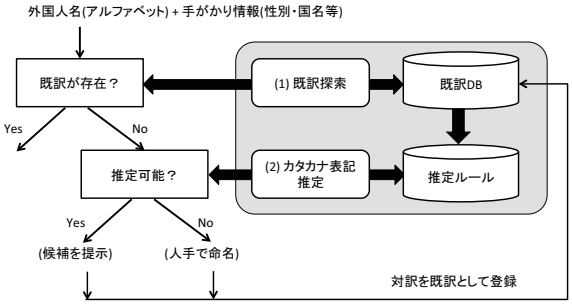


図 1: システムの全体像

1. **既訳検索**：既訳が存在する場合は，それを提示する．
2. **カタカナ表記推定**：既訳が存在しない外国人名に対し，もっともらしいカタカナ表記(複数可)を推定する．

新たに採用されたカタカナ表記は，採用された時点で既訳データベースに登録する．これにより，同じ人物の名前を何度も翻訳することを避けるとともに，同一人物一表記の原則が遵守されるように支援する．

2.2　カタカナ表記推定の実現法

カタカナ表記の推定器(トランスリタレータ)の実装には，Mecab[3]を利用する．Mecabは，Finite State Transducer (FST)の能力を持つため，形態素解析処理だけでなく，汎用的なテキスト変換ツールとして使用可能である．同時に，Conditional Random Field (CRF)を用いた学習機構を備えているため，学習用コーパス(我々の場合は，人名対訳データ)から，トランスリタレータを自動的に作成することができる．

図2に，Mecabを用いたトランスリタレータの実現法の概略を示す．この実現法において検討が必要なのは，主に，学習用コーパスをどのような形で準備するかである．以下では，その詳細について検討する．

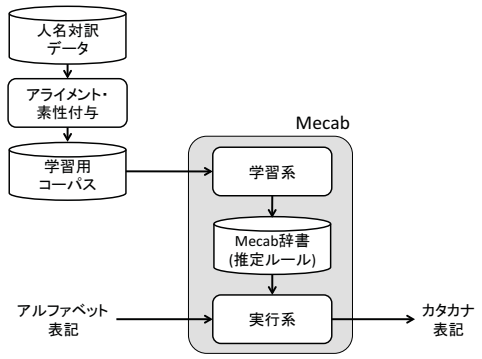


図 2: Mecabによるトランスリタレータの実現

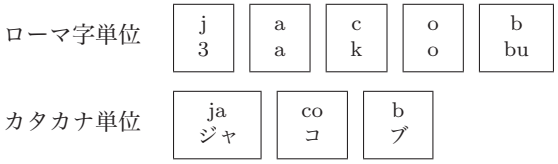


図 3: アライメントの単位

3 検討項目

トランスリタレータの実装において，次の5つの項目を検討した．

3.1 学習データの量

機械学習の適用においては,「学習データの量が十分であるか」ということが問題となる．このことを明らかにするため，学習データの量と精度の関係を調べる．

3.2 アライメントの単位

アルファベット表記とカタカナ表記の間で，どのようにアライメント(部分対応関係)をとるかは自明ではない．トランスリタレーションは,「音」に基づく翻訳であるため，直感的には音を介したアライメントが良いように思われるが，Karimiらによるサーベー[2]によれば，表記に基づくアライメントの方がよい性能を示すと報告されている．この報告に基づき，我々は，ローマ字単位とカタカナ単位の2種類のアライメントを比較する(表3)．

なお，我々は，ローマ字表記として，より音を意識した特別なローマ字表記を採用する．たとえば，タ行の子音は，表1に示すように3種類に区別する．

対訳人名に対するアライメント付与は，まず，カタカナ表記をローマ字表記に変換し，外国人名対訳収集[4]の際に作成したプログラム[1]を用いて，ローマ字単位のアライメントを付与する．カタカナ単位のアライメントは，この結果をカタカナ単位にまとめることで作成する．

[1]ダイナミックプログラミングに基づく方法を採用．可能な部分対応とそのコストは人手で作成．

表 1: タ行のローマ字表記

タ(ta)	ティ(ti)	トゥ(tu)	テ(te)	ト(to)
		テュ(tju)		
チャ(ca)	チ(ci)	チュ(cu)	チェ(ce)	チョ(co)
ツァ(2a)	ツィ(2i)	ツ(2u)	ツェ(2e)	ツォ(2o)

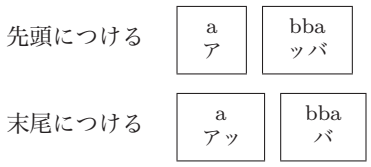


図 4: 促音の位置

3.3 促音の位置

促音「ッ」は，原音との対応が不明瞭であり，アライメントをとる際，どのように扱えば良いか自明ではない．Mecabを利用する場合,「ッ」を部分対応の先頭につける方法と，部分対応の末尾につける方法の2つの選択肢がある．たとえば,「Abba/アッバ」のカタカナ単位のアライメントでは，図4のような選択肢となる．これらの選択肢の優劣を調査する．

3.4 素性の追加

機械学習において，何を素性とするかはもっとも重要な検討項目である．Mecabを利用する場合，トランスリタレーションを実現するための学習用コーパスを構成する1対訳は，次のような形式となる(カタカナ単位を用いた場合)．

```
ja      ジャ,ja
co      コ,co
b       ブ,b
EOS
```

学習用コーパスの1行は，部分対応の1つに対応する．この1行には，カタカナ列(あるいは，ローマ字列)と元のアルファベット列以外にも，素性を追加することが可能である．

今回は，次の4種類を検討する．

1. 追加素性なし．
2. **長音素性**を追加．長音素性は1ビットの素性で，その部分対応の日本語側に長音記号「ー」が含まれるか否かを表す．

3. **促音素性**を追加．促音素性は1ビットの素性で，その部分対応の日本語側に促音「ッ」が含まれるか否かを表す．
4. **長音・促音素性**を追加．長音・促音素性は2ビットの素性で，1ビット目はその部分対応に長音記号が含まれるか否か，2ビット目は促音を含まれるか否かを表す．

Mecabの学習では，こうして定義した素性の列から，CRFの学習に用いる素性の列を作り出す方法を定義する[2]．unigram(単一行の素性の組み合わせ)とbigram(連続する2行の素性の組み合わせ)が可能である．今回の調査では，原則として，異なる情報を表現する組み合わせを，CRF素性として定義する．

3.5 学習パラメータ

機械学習を用いる際,「どの程度学習データにフィットさせるのか」という問題に必ず直面する．Mecabでは，CRFのハイパーパラメータCがこれを制御するパラメータである．このパラメータの精度への影響を調査する．

4 実験と結果

実験には，外国人名対訳辞書の自動編纂[5]のために収集したフルネームの人名対訳データ215,406件を用いた．まず，フルネールの人名対訳データを姓と名に分割し，それらから重複を取り除いて，129,138件のデータを得た．この中からランダムに選んだ9,224件を評価用データとして用い，残りの119,914件を学習データとして用いた．以上の説明から明らかなように，トランスリタレーションを行なう単位は，フルネールを構成する姓または名である．

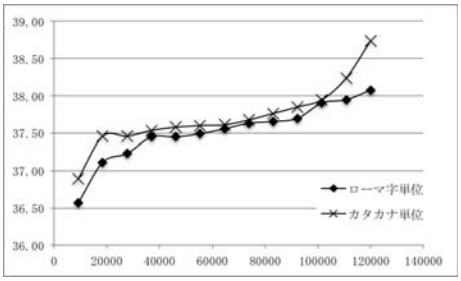
システムの評価指標には，システムの出力の上位1位が正解と一致する割合(Top1)，上位3位以内に正解が含まれる割合(Top3)，上位5位以内に正解が含まれる割合(Top5)，を用いた[3]．ここで「正解と一致する」とは，文字列として完全に同一であることをさす．

4.1 アライメント単位および学習量と精度

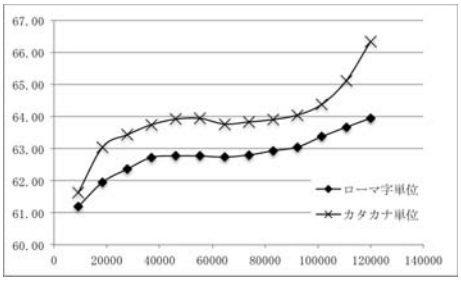
アライメント単位および学習量と精度の関係を図5に示す．この図のグラフの縦軸はシステムの精度(%)，横軸は学習に用いたデータの件数である．なお，この実験では，追加素性なし，促音位置=先頭，CRFのハイパーパラメータ$C = 1.0$ (Mecabのデフォールト)を用いた．

これらのグラフから，以下のことがわかる．

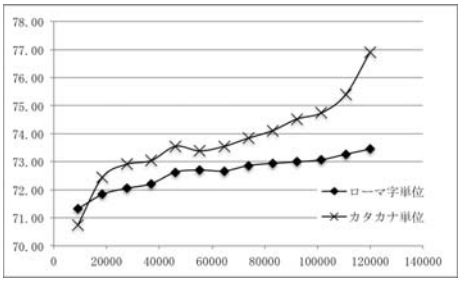
[2]feature.defという名称のファイルで定義する．
[3]Mecabは複数の候補を出力することができる．



(a) Top1



(b) Top3



(c) Top5

図 5: アライメント単位、学習データ量と精度

- アライメント単位に関しては，Top1ではそれほど差が見られないが，Top3とTop5では，明らかにローマ字単位よりカタカナ単位の方が精度が良い．
- 学習に用いるデータ量の増加に従って精度向上が見られる．これは，学習が飽和していないこと(学習データの不足)を意味する．

この結果を受けて，以降の実験では，すべての学習データ(119,914件)を用い，アライメント単位はカタカナ単位を用いる．

4.2 促音位置および追加素性と精度

アライメントにおける促音の位置，および，特殊音素性の追加と，それぞれの場合のトランスリタレーションの精度を表2に示す．なお，この実験でも，CRFのハイパーパラメータは$C = 1.0$を用いた．

この表より，以下のことがわかる．

- 長音素性，長音・促音素性を追加することにより，精度が若干向上する．しかしながら，精度向上の

表 2: 追加素性、促音位置と精度

追加素性	促音位置	Top1	Top3	Top5
なし	先頭	**38.96**	**66.35**	**76.91**
	末尾	38.32	65.82	76.13
促音	先頭	**38.94**	66.35	76.88
	末尾	38.64	**66.59**	**77.18**
長音	先頭	**39.03**	**66.65**	**77.21**
	末尾	38.59	66.01	76.76
長音・促音	先頭	**39.19**	66.59	77.20
	末尾	39.03	**66.85**	**77.32**

表 3: 学習パラメータと精度

C	Top1	Top3	Top5
0.25	**40.18**	67.51	77.61
0.50	39.82	**67.57**	**77.79**
1.00	39.03	66.85	77.32
1.50	38.24	65.82	76.81

効果は限定的である．促音素性は，単体では効果はない．

- アライメントにおける促音位置も，精度にほとんど影響しない．

以上の結果から，これらの検討項目には，それほど注意を払う必要はないと考える．我々は，一番精度が高かった，長音・促音素性の追加，および，促音位置=末尾を採用する．

4.3 学習パラメータと精度

学習パラメータと精度の関係を表3に示す．この表からわかるように，パラメータCの値も，精度にそれほど大きな影響を与えなかった．我々は，$C = 0.5$を採用する．

4.4 長音・促音の有無と精度

本実験で最終的に得られた精度は，上位5位までに正解が含まれる割合(Top5) が77.79%である．この精度は，それほど満足できるものではない．

正解が得られない名前はどのような名前であるかを調査したところ，名前に長音を含まれるか否かが大きく関係していることが判明した．表4に，長音と促音の有無と精度との関係を示す．この表から明らかなように，長音を含む名前のTop5は，含まない名前のTop5より約10%低い．長音を含む名前は，評価データ9,224件中3,747件(40.6%)あり，その精度の低さが全体の精度を押し下げている．

5 おわりに

今回の調査において，精度に最も大きく影響したものは，アライメント単位である．より小さなローマ字

表 4: 長音・促音の有無と精度

	件数	Top1	Top3	Top5
長音と促音を含まない	4553	46.76	72.46	82.03
促音のみを含む	924	40.15	70.45	81.17
長音と促音の両方を含む	324	33.95	60.80	72.22
長音のみを含む	3423	28.92	58.98	71.63
合計	9224	39.82	67.57	77.79

単位を用いた場合，対応の種類が少なくなるという長所がある．一方，より大きなカタカナ単位を用いた場合は，アライメント単位のbigramしか考慮対象にできないMecabにおいては，より広い範囲(文脈)を考慮できるという長所がある．今回の調査では，カタカナ単位がよいという結果が出たが，どのような名前で差が出ているのか，より詳細な調査が必要であろう．

本稿の冒頭で述べたように，オリンピックの参加者の名簿の翻訳に当たっては，多数の原言語を対象としなければならない．この点が，これまでの(対象言語対を限定した)トランスリタレーション研究と大きく異なる点である．この問題に対する我々の方針は，次のとおりである．

1. 言語または国籍が不明の大量の人名対訳データから，ベースとなるトランスリタレータを構成する．(本報告)
2. 言語または国籍が既知の少量の人名対訳データを追加学習データとして用い，トランスリタレータを再学習する．

本研究でMecabを利用する大きな理由は，この追加学習が可能であるからである．追加学習の効果については，稿を改めて報告する予定である．

謝辞 本研究では，平成26年度の放送文化基金の助成(テーマ「コンピュータ支援による外国人名カタカナ表記の標準化・統一化」)を受けている．

参考文献

[1] Kevin Knight and Jonathan Graehl. Machine transliteration. *Computational Linguistics*, Vol. 24, No. 4, pp. 599–612, 1998.

[2] Sarvnaz Karimi, Falk Scholer, and Andrew Turpin. Machine transliteration survey. *ACM Computing Surveys*, Vol. 43, No. 3, 2011.

[3] 工藤拓. MeCab. `http://taku910.github.io/mecab/`. [Online; accessed 10-Jan-2016].

[4] Satoshi Sato. Crawling English-Japanese person-name transliterations from the Web. In *Proceedings of the 18th international conference on World Wide Web*, pp. 1151–1152, 2009.

[5] 佐藤理史. 辞書の見出し語集合と代表性. 言語処理学会第18回年次大会論文集, pp. 8915–918, 2012.