カタカナ語から英語への翻字翻訳

鈴木久美
Microsoft Research
One Microsoft Way
Redmond WA 98052, USA
hisamis@microsoft.com

Colin Cherry
National Research Council Canada
1200 Montreal Road
Ottawa Ontario K1A 0R6 Canada
Colin.Cherry@nrc-cnrc.gc.ca

概要

本稿ではカタカナ語を英語（あるいはその他のローマ字使用語）に翻字をもちいて翻訳する手法について述べる。本稿で提案する手法は、生成モデルを素性として識別モデルのフレームワークで使うという点で、近年の統計的機械翻訳手法を翻字のタスクに応用したものとなっている。生成・識別ハイブリッドモデルを使用することにより、このタスクでこれまでの手法を凌ぐ結果を得た。さらに、この翻字エンジンを英日機械翻訳で使用し、人手による評価で翻訳の質が向上することを確認した。

キーワード: 統計的機械翻訳　カタカナ語　翻字
machine translation and transliteration

1　はじめに

翻字(transliteration)とは、使われる文字種の異なる言語間で文字を置き換えることである。翻字は本来、文字と文字の交換(e.g. あ→a, th→θ)を指し、その際文字の音価を保存することは必ずしも重要ではない。しかし、外来語を音に基づいて借用し自国語の文字で表記する、という場合では、語単位で外国語の発音に基づいた翻字を行うことがある。たとえば日本語のカタカナは、英語やその他の外国語由来の語を綴るのに用いられ、その際、元の言語の音に基づいた表記がなされるのが普通である。たとえ外国語での綴りが同じでも、翻字の結果が同じとは限らないし（たとえば、"Paris"は「パリ」でも「パリス」でもありうる）、また同一の語が何通りにも翻字されることも多い（「スパゲティ」「スパゲッティー」など）。このように、単語レベルでの翻字は、文字レベルの単純な翻字に比べて言語処理的に面白い問題であるが、とりわけ、外国語由来の語を元の言語に戻す、いわゆる逆翻字(back-transliteration)と呼ばれるプロセスは、先のspaghettiの翻字に見られるようなあいまい性が存在せず、タスクとして評価が容易なので、これまでにも数多くの研究がなされている。また、このタスクにはさまざまな応用が考えられる。たとえば、カタカナ語を元の言語（たとえば英語）に翻字するのは、音に基づいた翻訳を提供することにほかならず、この技術を用いて、

Avoid using a フリーメール account.

といった、未翻訳の文字が残ってしまっている日英翻訳の結果を、

Avoid using a freemail account.

という、実際に役に立つものに変換することができる。また、日本人が英語で作文する際、カタカナ語の英語での綴りが定かでないことはよく経験するが、そのようなときにも翻字エンジンを使って、カタカナ語の入力から英語の綴りを予測し、作文支援をすることができる。

本稿では、カタカナ語から英語への翻字翻訳[1]（「翻字に基づく翻訳」という意味で、ここではこう呼ぶことにする）を高い精度で行う手法を提案する。提案手法は、文字列の変換手法に関する研究と、統計的機械翻訳(SMT)に関する研究の双方をベースにしている。前者の研究(たとえば[13])では、部分文字列変換ルールを生成モデル枠内で使用しているが、本研究では、このような生成モデルを素性として識別モデル枠内で使用し、教師つき学習する手法をとり、先行研究を凌ぐ結果を得た。また翻字翻訳は、文字を単語、文字列を句とみなすと、句を翻訳単位とするphrase-based SMTとみなすことができ、これまでにこのタスクで提案されてきた手法やツールを使用することができる。ここでは、Koehnら[8]のphrase-based SMTに基づき、このアプローチを翻字に応用し、提案モデルと比較した。

本稿では、まず次節で翻字翻訳の先行研究を紹介し、3節で提案手法について詳述する。4節では、既存の手法と提案手法の比較実験の詳細と結果を報告する。さらに5節では、この翻字エンジンを実際に英日機械翻訳で使用し、人手で評価した結果について述べる。

2　関連研究

翻字翻訳に関する先行研究のタスク設定には、異なる言語間の文字列の類似度を用いて翻訳候補のうち最適なものを選択するアプローチと、このタスクを文字列のトランスダクションとみるアプローチの大きく2種に分けられる。前者のアプローチでは、翻字翻訳の候補の生成と類似度の計算は独立して行われ、候補に入っていない単語には翻訳できない。たとえば、[2]では、同時期に集められた英語と日本語のインターネット検索のクエリログから

[1] 正確には「逆翻字翻訳」だが、わずらわしいのでここでは単に「翻字翻訳」あるいは「翻字」とする。

候補を収集し、そこから文字列間の編集距離をEMアルゴリズムを用いて学習し、その類似度を用いて最適な候補を選択する手法が紹介されている。[1]は、日英対訳コーパスから候補を収集し、テキスト中の分布の類似度と音声的な類似度を用いて最適解を選んでいる。上述の2論文はカタカナ語から英語への翻字翻訳を扱っているが、ほかにもアラビア語やロシア語などから英語への翻字タスクで、対訳コーパスから翻字候補を収集し、文字列の類似度を素性とした分類器で最適な翻字候補を選ぶ手法が提案されている。

これらのアプローチとは対照的に、入力と出力の部分文字列の対応に基づいて文字列を変換する翻字翻訳の方法がある。この手法では、出力はあらかじめ用意された候補から選ぶ必要はなく、新規の文字列を生成できるので、前節で述べたアプローチよりもフレキシブルな設計である。この手法に基づく翻字翻訳は、noisy channelモデルに基づく研究が主流である。翻字前の文字列をs、翻字後の文字列をtとすると、$P(t|s) \propto P(s|t) \cdot P(t)$となり、翻字モデル$P(s|t)$と言語モデル$P(t)$に分割できる。このモデルに基づく初期の研究には[9]があるが、[13]ではこの生成アプローチを、文字単位から部分文字列単位に拡張することによって、先行研究を凌ぐ精度を報告している。本研究では、[13]をベースラインとし、その更なる改良を提案する。

どちらのアプローチにせよ、上述の研究は、翻字翻訳を独立のタスクとみなして評価しているが、これを機械翻訳という応用システム内で使用して評価するという動きも出てきている。[6]がその例で、この研究では、アラビア語から英語への翻字翻訳エンジン自体には重きをおかず、それをどうSMTに組み込むかという点、とりわけある語を翻訳すべきかあるいは翻字すべきかを判定するタスクに焦点が絞られている。これは、具体例を挙げると、الجزيرة(アルジャジーラ)という語を翻訳する際、普通名詞と解釈して「島」と翻訳すべきなのか、固有名詞ととって「アルジャジーラ」と翻字すべきなのかを判定する、という問題であり、翻字を機械翻訳に応用するにあたっては、避けては通れない問題である。ただ、日本語のカタカナ語の翻字翻訳に限って言うと、5節で見るように、我々の使用したコーパスでは、カタカナ語の90%以上が翻字翻訳の対象になっているので、この問題を扱わなくても十分な改善を見ることができた。

3 提案手法

3.1 識別モデルと使用素性、ベースライン

提案手法は、一言で言うと[13]で使用された生成モデルを素性として識別モデル内で使用する、というものである。先に述べたように、[13]はnoisy channelモデルに基づいており、sをtに翻字する確率を

$$P(t|s) \propto P_E(s|t) \cdot \max[P_T(t), P_L(t)]$$

としている。ここでP_Eはemission probability(文字列単位でsがtとして翻字される確率)、$P_T(t)$はtransition probability(生成された文字列のその言語らしさ)、$P_L(t)$はlexicon probability(生成された文字列が語である確率)である。本稿の研究では、翻字確率P_Eは、部分文字列対応つきのデータを用いて推定し、文字n-gram確率$P_T(t)$はそのデータの出力(英語)側のみを用いて推定した。使用したデータについては次節で詳述する。また、後に見るように、翻字タスクでは翻訳タスクと違い、語確率P_Lが非常に有効である。このための辞書作成データについても次節で述べる。

本論文では、これらのモデルを素性として、以下のように線形モデル内で使用した。

$$w_E \log P_E(s|t) + w_T \log P_T(t) + w_L \log P_L(t)$$

各サブモデルの重みwは3.2節で述べるように、パーセプトロンで学習した。このように生成モデルのサブモデルを識別モデルで使用し、その重みを学習する手法は、[11]などSMTではスタンダードな手法であるが、翻字のタスクではこれまでなされていない。また、線形モデルには任意のサブモデルを容易に組み込めるという利点があるので、この点を踏まえてSMTで採用されている3つのサブモデル($P_E(t|s)$、tの文字数、文字列変換ルール数)も加えて使用した。

以上、提案手法は、従来の生成モデルに基づいた素性を識別モデルのフレームワークで使用するというハイブリッドモデルとなっている。これはハイブリッドであるから、ベースラインには、(1)従来の生成モデル[13]と、(2)識別モデルであるphrasal SMTのツール[10]をそのまま使用する手法、の2つを採用し、これらと提案手法を比較した。(2)では、提案手法と同じ学習データを使用して、文字を単語とみなし、phrasal SMTシステムを構築した。その際、言語モデルには文字7-gramを使用し、語順の変更(reordering)はしなかった。出力単語辞書の使用はこの手法ではサポートされていないので、辞書データを用いて文字7-gramモデルを構築し、線形モデルに追加した。

なお、識別モデルでは一般に、素性がオンかオフかのみを示すバイナリの素性関数(indicator feature function)がよく用いられる。文字列変換のタスクでも、各々の変換ルールをバイナリの素性関数として使用でき、その際変換ルールのコンテキスト情報も容易に素性として使用できることから、翻字以外の文字列変換タスクでもこのような素性設計が用いられている[5,7]。この点を踏まえて、比較のため、識別モデル枠内でバイナリの素性関数のみを用いた翻字システムも構築した。このシステムの素性は、文字列変換ルールとその入力側のコンテキスト(前後のC文字)、出力側の文字列n-gram($n = 1 \ldots K$)からなる。CとKの最適な値(C=3, K=5)は4.1節のディベロップメントデータを使って設定した。さらに、翻字タスクでは出力語の辞書が有益であるので、この情報をバイナリ化して辞書素性として追加使用した。出力語の頻度を2,000

未満、200未満、20未満、1、0の5つのグループに分け、各語につき、該当のバイナリ素性がオンになる、という仕組みである。

3.2 識別モデルのパーセプトロン学習

モデルの学習には平均化パーセプトロン学習を用いた[4]。学習の入力と出力は、部分文字列対応付きの、カタカナ語と英語の語対である。入力語sを出力語tに変換する一連の文字列変換ルールを派生(derivation)dとすると、パーセプトロン学習は次の2つのステップをそれぞれの$d_i \in D$について行う。

Decode: $\bar{d} = \text{argmax}_{d \in D(src(d_i))} \vec{w} \cdot \vec{F}(d)$

Update: $\vec{w} = \vec{w} + \vec{F}(d_i) - \vec{F}(\bar{d})$

ただし、$D = d_1 ... d_n$は部分文字列対応済みの学習データ、$D(src(d))$は、dと同様の入力文字列に可能な派生の集合、$\vec{F}(d)$はdの素性関数ベクトル、$\vec{w}$はその重みベクトルである。

学習データの部分文字列レベルの対応付与には、SMT用の既存の語対応ツール[14]を利用した。その際、語順(文字列順)の変更は行わず、文字列はs、t側とも最長3文字・最短1文字に限って使用した。同じ制限はdecode時にも使用した。

4 実験結果と考察

4.1 実験データ

学習用データは日英のウィキペディアを使用して作成した。日本語の記事で、見出し語がカタカナのみであり、英語の記事が存在するもの、という条件で、カタカナと英語の見出し語のペアを約6万件抽出し、このうち2,000件をディベロップメントデータ、別の2,000件をテストデータに使用し、残りの56,000件を学習に使用した。学習データの中には、「コンピュータゲーム」と"computer and video games"、「ハンス・アクセル・フォン・フェルセン」と"axel von fersen the younger"など、厳密には翻字のペアではないものが10%程度含まれており、これらを除くことは有益と思われたので[2]、上述のアラインメントで使用されたルールのうち、2回以下しか使用されなかったルールとそれらを使っている見出し語を除外する、という簡単な方法でこれらのルールと語のペアを除外した。この結果、最終的な学習データのサイズは約4万件になった。このうち約38,000件をP_Eと$P_T(t)$のパラメタ学習に、残りの2,000件をパーセプトロン学習に使用した。また、このデータの英語側は、文字n-gram確率$P_T(t)$の計算にも使用した(n=7)。

ディベロップメントデータとテストデータでは、現実的な応用に使用することを鑑みて、このようなデータクリーニングは行わず、そのまま使用した。ディベロップメントデータは、パーセプトロン学習の繰り返しの回数や、文字n-gramベースの言語モデル$P_T(t)$のnなどのパラメタのチューニングに使用した。

また、提案モデルの有効性がウィキペディアドメインに限られてしまうのではないか、という疑問に答えるため、テストデータにはさらに別ドメインのデータを準備した。このデータは、翻字が翻訳に有効であることを示すため、実際のMTのユーザログから抽出した。具体的には、まずBing Translator (http://www.microsofttranslator.com/)の日英翻訳エンジンに提出された文あるいは句とその翻訳結果をランダムに5,000個抽出する。この中から、翻訳結果の英文に未翻訳のまま残ってしまったカタカナ語(312個あった)を取り出し、これらを含む原文に人手で翻訳を付与し、さらに当該カタカナ語に相当する英語の語(翻字とは限らず翻訳も含まれる)を対応させて翻字の正解とした。ちなみにこのデータのカタカナ語で、ウィキペディアの見出しから抽出されたデータにも含まれていたものは、17(5.5%)にとどまり、ウィキペディアとはかなり異なったドメインであることがわかる。

なお、単語unigram言語モデル$P_L(t)$用の辞書作成にはLDCから入手できる英語Gigawordコーパスと、Bing Translatorの言語モデルトレーニングに使用されたデータの両方を使用した。後者は前者のコーパスの一部を含んでいるので、重複して出現した語に関しては大きいほうの出現数を使用した。最終的な語彙は異なり数で500万語であった。

	システム	正解率	素性数
ベースライン	Phrasal SMT	30.7	8
	[13] (A)	33.5	–
提案手法	[13] 線形モデル	31.7	–
	+パーセプトロン学習	42.4	3
	+SMT素性 (B)	44.1	6
バイナリ素性識別モデル	辞書素性なし	28.6	3.1M
	辞書素性あり (C)	44.2	2.5M

表1: ディベロップメントデータの正解率(%)

4.2 実験結果と考察

ベースラインと提案手法のディベロップメントデータにおける1-best解の正解率を表1に示す。まず、ベースラインの2手法だが、Phrasal SMTの手法よりも、生成モデルで出力単語辞書を使用した[13]の方がよい結果となった。このタスクでの出力単語辞書の重要性を示している。次に、提案手法であるが、「[13]線形モデル」は[13]のサブモデルを線形モデル枠内で単一の重み(w_E=w_T=w_L=1)で使用したときの正解率であり、これは[13]を下回る。しかし、モデルの重みをパーセプトロンで学習したモデル

[2] 本来はこれらのペアを除外せずに、comparableと見て部分文字列レベルで使用するのが理想的であるが、今回の実験では見送った。また、「ウスユキガモ」"marbled duck"など、生物の名前がカタカナ語で表記され、ウィキペディア内ではそれが繰り返し使用されるため、「ガモ」を"duck"と翻字するルールが学習されてしまうという問題も発生している。この問題は現在のところ未解決である。

	Wikipedia		MT Log	
システム	1best	10best	1best	10best
A (ベースライン)	33.5	57.9	38.8	57.0
B (提案手法)	**43.0**	**65.6**	42.9	**58.3**
C(バイナリ素性識別)	42.5	63.5	**43.6**	57.7

表2: テストデータの正解率(%)

は、正解率が42.4%と大きく改善した。3.1節で述べた、SMTでよく使用される3つの素性を追加すると正解率はさらに44.1%に向上した。最後に、比較のため構築したバイナリ素性識別モデルであるが、こちらでも、辞書素性を使用すると正解率が16%近くも向上することから、この素性が非常に有益であることが伺える。バイナリ素性識別モデルは、文字列変換タスク一般において最新の手法であるが、提案手法はこれと比べても遜色のない正解率を実現している。

表2はテストデータにおける正解率である。最適解の正解率に加え、Nベスト解(N=10)の正解率も示した。ウィキペディアのデータを使用して構築した提案手法が、MTのログという別のドメインでも効果的であることがわかる。なお、不正解のなかには、「レチノイド」(正解はretinoids)をretinoidと翻字したり、「シエラレオネ」(sierra leone)をsierraleoneとするなど、正解とは異なるが実際には有効であろうものも含まれている。表2ではMTログデータでの10ベスト正解率は60%に満たないが、次節で見るようにMTに組み込んだときの有効性が80%近くあるのはこのためと思われる。

5 機械翻訳への組み込み

翻字を翻訳システムに組み込む最も単純な方法は、翻訳文に残ってしまったカタカナ語を後処理で翻字する方法であるが、このやり方だと10-best解を活用できない。したがって、ここではカタカナ語と10-best解のペアを、線形モデルに基づくSMTシステム[12]に新たなモデルとして追加するやり方を取った。その際、翻字モデルのスコアは確率にもどして使用した[3]。

この翻字ありSMTシステムと、翻字を組み込む前のシステムとの違いを人手で比較した。比較に使ったデータは、4.1節のMTログのテストデータから抜粋した120個であり、評価は日本語を解さない2人の英語話者が行った。まず翻字ありとなしのシステムの文単位での出力に、1(使えない翻訳)から4(完璧な翻訳)までのスコアをつけてもらったところ、平均で翻字なしでは1.5、翻字ありでは2と、大きな改善がみられた。次に、翻字ありとなしの翻訳結果のペアを各々比較し、どちらがよいかを評価してもらったところ、120のサンプルのうち、95(=80%)で、2人の評価者ともに翻字ありのシステム解の方が適切である、という評価を得た。日英翻訳の結果にカタカナ語がまぎれているよりは、たとえ正解ではなくても翻字の結果(英語)を表示したほうがよいのでは、と思われがちであるが、実際には「アップローダ」(uploader)を、"applaud"と翻字してしまったケースなど、あやまった翻字が混乱をもたらすこともある。このようなケースを考慮すると、80%の文で翻訳の質の向上が見られたことは、本アプローチの有効性を示していると考えられる。

[3] この評価で使用した10-best解は表2のシステムCによる。

6 おわりに

本稿では、カタカナ語から英語への翻字に焦点を絞って説明した。本稿で得られた知見を、その他の言語間の翻字タスクにも応用していくのが今後の課題である。

*本研究は第二著者のMicrosoft Research在籍中に行われた。本研究の成果の詳細は[3]に詳しい。

参考文献

[1] Slaven Bilac and Hozumi Tanaka. 2005. Extracting transliteration pairs from comparable corpora. 言語処理学会第11回年次大会.

[2] Eric Brill, Gary Kacmarcik, and Chris Brockett. 2001. Automatically harvesting katakana-English term pairs from search engine query logs. In *NLPRS*.

[3] Colin Cherry and Hisami Suzuki. 2009. Discriminative substring decoding for transliteration. In *EMNLP*.

[4] Michael Collins. 2002. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *EMNLP*.

[5] Markus Dreyer, Jason Smith and Jason Eisner. 2008. Latent-variable modeling of string transductions with finite-state methods. In *EMNLP*.

[6] Ulf Hermjakob, Kevin Knight and Hal Daumé III. 2008. Name translation in statistical machine translation - learning when to transliterate. In *ACL*.

[7] Sittichai Jiampojamarn, Colin Cherry and Grzegorz Kondrak. 2008. Joint processing and discriminative training for letter-to-phoneme conversion. In *ACL*.

[8] Philipp Koehn, Franz J. Och and Daniel Marcu. 2003. Statistical phrase-based translation. In *HLT-NAACL*.

[9] Kevin Knight and Jonathan Graehl. 1998. Machine transliteration. *Computational Linguistics*, 24(4).

[10] Robert Moore and Chris Quirk. 2007. Faster beam-search decoding for phrasal statistical machine translation. In *MT Summit XI*.

[11] Franz J. Och. 2003. Minimal error rate training for statistical machine translation. In *ACL*.

[12] Chris Quirk, Arul Menezes and Colin Cherry. 2005. Dependency treelet translation. In *ACL*.

[13] Tarek Sherif and Grzegorz Kondrak. 2007. Substring-based transliteration. In *ACL*.

[14] Hao Zhang, Chris Quirk, Robert C. Moore and Daniel Gildea. Bayesian learning of non-compositional phrases with synchronous parsing. In *ACL*.