機械翻訳手法に基づいた日本語の読み推定

羽鳥 潤
東京大学大学院 情報理工学系研究科
コンピュータ科学専攻
東京都文京区本郷 7-3-1
hatori{at}is.s.u-tokyo.ac.jp

鈴木 久美
Microsoft Research
One Microsoft Way
Redmond, WA 98052, USA
hisamis{at}microsoft.com

1　緒論

本稿では[1]、日本語の漢字仮名交じり文における読み付けの問題を取り上げる。ここで、読み付けとは「与えられた文・語句に対してその振り仮名の列を出力するタスク」であると定義する。例えば、入力文「東京都美術館に行った」に対して「とうきょうとびじゅつかんにいった」が正しい出力となる。このタスクは、テキスト音声合成の為の重要な基礎となる他、仮名漢字変換の訓練データを作成する目的にも重要である。

日本語における読み推定では、単漢字の読み候補が多く、非合成的な読みも多く見られる事から、単語分割後に辞書語単位での読み推定を行う方法が主流だった (長野 06; 森 10a)。この方法では、読み推定を辞書に基づいた読み判別問題として解く事で、豊かな文脈情報と高性能な識別的機械学習手法（例えばSVM）を利用する事ができる。

一方で、辞書ベースの手法を用いる事の出来ない未知語に対する読み推定の問題には、これまであまり重きが置かれてこなかった。例えば、森 10a) では、既知語に対して SVM を用いた読み推定を行っているのに対して、未知語に対しては単純な「雑音のある通信路」モデルを適用するに留まっている。そこで、本稿ではまず、ウェブから教師なし学習によって獲得した単語・読みペアを学習データとして用いて、未知語に対する読み推定モデルを構築する。その後、この手法を辞書を用いて拡張する事で、既知語と未知語の両方を含む文の読み推定を行う事ができるモデルを構築する。提案手法は、句ベースの機械翻訳手法に基づいているため、単漢字／部分文字列レベルの翻訳操作と辞書語レベルの翻訳操作の両方を同時に利用する事ができ、既知語と未知語の読み推定を継ぎ目なく行う事が可能になる。また、手法における工夫として、結合 n-グラム (JCK10) や翻訳操作の合成 (CS09) を用いる事で大きな精度向上が見込める事を示す。

2　関連研究

SS06) は既知語の読み判別のためにウェブを利用する手法を考案したが、その対象は既知の固有名詞に限られていた。また、KMIN07)・笹田 09) は、音声データを用いて、新語の読み曖昧性を解消する手法を提案している。結合 n-グラム推定の応用例は、長野 06)・森 10b) 等の研究がある。また、現在の日本語読み推定タスクにおける最高水準にある KyTea(森 10a) は、SVM を用いた二段階アプローチに基づいたシステムであり、まず、点推定による単語分割が行われた後に、各セグメント毎に読み推定が行われる。読み推定の段階では、対象語が既知語であった場合には、文字と字種 n-グラムを素性として SVM による読み判別が行われ、未知語であった場合には、平仮名から漢字への変換確率と出力仮名バイグラム確率に基づく「雑音のある通信路」モデルが用いられる。

3　読み推定モデル

この節では、日本語読み推定タスクに対する我々の提案手法を紹介する。この手法は、統計的機械翻訳の手法に基づいているが、アライメントは単調（交差しない）で、挿入・削除は起きないと仮定する[2]（図 1 参照）。

3.1　デコーダ

デコーダには、統計的機械翻訳において一般的に用いられている、句ベースのデコーダ (ZN04) を利用する。スコアリングには線型モデルが用いられ、ある出力列 t のスコアは、任意の入力列 s と出力列 t のペアに対して定義された、重みベクトル　と実数値素性関数 $\mathbf{f}(s,t)$ の内積として表される。素性としては、CS09) と同様複数の生成確率素性が用いられ、(1) 双方向翻訳確率 $P(t|s), P(s|t)$、(2) 文字 n-グラム確率 $P(t)$、(3) 結合 n-グラム確率 $P(s,t)$、出力文字数、句（翻訳操作）の数を利用している。ここで、結合 n-グラム確率は、入力単語と出力単語のペアの列（「⟨床屋, とこや⟩⟨に, に⟩⟨行く, いく⟩」）に対する n-グラム言語モデル確率である。また、後述する部分文字列モデルでは、これに加えて辞書語素性（辞書語にマッチした句の長さの合計）も用いる。線型モデルの重みベクトルの訓練には平均化パーセプトロンを用いた。

3.2　翻訳操作

図 2 に見られるように、訓練に使用するコーパスは漢字仮名交じり文とその読み仮名のペアからなるため、モデルの訓練を行う為には、まず、単語間のアライメントを取らなければならない。我々は句アライナを用いる方法と、辞書ベースの句デコーダを用いる 2 通りの方法を試した。それぞれの方法に応じて、部分文字列

[1]本研究は、第一著者の Microsoft Research におけるインターンシップ中に行われた。

[2]実際には、「不忍池（しのばずいけ）」のように厳密には単調でない例、「一関（いちのせき）」のような挿入が起こる事もある。

モデル・辞書語モデルの 2 モデルを構築し、実験に用いる。前者が教師無し学習によって構築された未知語モデル、後者がこれを辞書を用いて拡張したものに対応している。以下にその詳細を記す。

3.2.1 部分文字列モデル

部分文字列モデルでは、句ベースのアライナ (ZQMG08) を用いて、教師なし学習によってアライメントを取る。まず、1 つの漢字が長さ 1 以上の読み列に対応しているとしてアライメントをとり、後に翻訳操作の合成（第 3.2.3 節参照）を行う。これによって、必要なメモリ量を抑えながらより大きな単位のアライメントを取れるようにしている。

3.2.2 辞書語モデル

辞書語モデルでは、辞書語を翻訳単位とした句デコーダを用いてアライメントを取る。このデコーダでは、まず辞書語に基づいた翻訳表を作成し、辞書語単位の翻訳操作を用いてアライメントを取る。用いられるデコーダは、第 3.1 節で紹介されたデコーダと本質的に同じものであるが、素性として、順方向翻訳確率と句の数のみが用いられる。この際、用いられる翻訳操作は辞書語単位のものとなるため、辞書に存在しない翻訳操作が含まれているインスタンスは自動的に切り捨てられる。但し、実際には多くの読み辞書は単漢字の読みも含んでいる事から、辞書に存在しない非一般的な単漢字読みが用いられていない限り、未知語の読み推定も単漢字読みにバックオフして行う事ができる。

3.2.3 翻訳操作の合成

CS09) と同様に翻訳操作の合成を用いる。図 1 に見られるように（例：「東京+都」「に+行った」）、隣接する翻訳操作をいくつか合成する事により、読み推定に必要なコンテキストや複合語の読みを考慮に入れる事ができる。例えば、「行った」には「いった／おこなった」という読みの曖昧性があるが、助詞「に」がその前に来ている場合には、ほぼ「いった」であると判別する事が出来る[3]。単漢字モデルにおいては、アライメントは始めに単漢字単位で取られているため、この過程によって非合成的な語句の読みを復元する事ができる。また、結合 n-グラム確率を計算する際には、合成前の操作を記憶しておき[4]、どのような合成が行われても同じアトミックな操作の列に基づいて計算されるようにした。これにより、一貫性のない翻訳操作単位の混合によって言語モデルの質が低下する事を避けている。

4 実験

4.1 辞書

辞書語モデルでは、アライメントを取る際の基準となる辞書が必要となる。本稿の実験では、UniDic 1.3.12 (63 万語)、岩波書店の辞書（33 万語）、社内で利用可能な辞書（23 万語）の 3 種類の辞書を合わせて、重複を除いた後、最終的に 78 万語の辞書として用いた。この辞書は、単漢字モデルにおける辞書素性にも用いられる。

[3]無論、「運動会を体育の日に行った（おこなった）」などの例外はあるが、非常に少数である。

[4]複数の合成が考えられる場合（「東／山口」「東山／口」など）は、訓練データ中で最初に出現したものだけ記憶した。

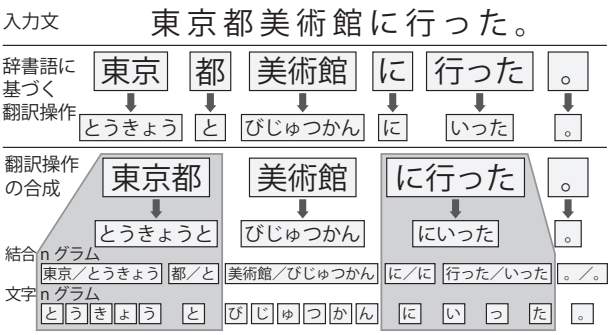

図 1: モデルの概要

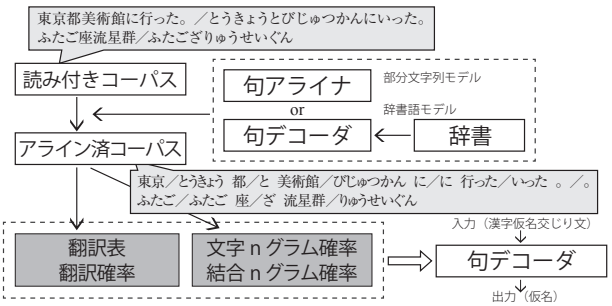

図 2: モデル構築の詳細

4.2 データセット

モデルを訓練するための訓練データとして、まず、ウェブからの自動読み獲得 (HS11) を試みた。この手法では、Wikipedia 本文中の “漢字仮名列（平仮名列）” という表現に着目し、(1) 漢字仮名列の長さは平仮名列の長さを超えない、(2) 漢字仮名列は句読点を含まない、(3) 漢字仮名列の含む仮名は平仮名列にも含まれる、等の制約を元に単語・発音ペアを抽出する。複数の漢字仮名列候補が考えられる時は、それら全てを展開して利用した。これによって、Wikipedia の本文（2010 年 1 月 24 日版）から、46 万対の単語・読みペア（以下 Wiki-Train）を獲得する事ができた。しかし、これらのデータには名詞以外の読みがほとんど含まれておらず、また、相当量のノイズ（約 10%）が含まれている事に注意されたい。また、Wikipedia から自動獲得された単語・読みペア以外に、140 万文からなる新聞記事データ（以下 News-Train）を用いた。単漢字モデルでは、更に前述の辞書が合わせて訓練データとして用いられた。また、提案手法とベースラインシステムの比較用には、現代日本語書き言葉均衡コーパスのモニタ公開データ（2009 年度版）(前川 08, 以下 BCCWJ) を用いた。

次に、評価セットは様々なドメインからなる 6 つのデータセットで構成されている。

News-1(N1)・**News-2**(N2): Microsoft Research IME コーパス (SG05)（新聞記事データ、それぞれ 867・739

モデル	N1	N2	Q1	Q2	PN	WP
KyTea	83.6	85.9	92.9	85.6	52.9	62.9
提案手法 (教師無し)	18.1	11.5	87.9	77.6	71.1	70.9
結合 n-gram(単漢字)	35.6	29.0	91.9	83.0	90.1	72.4
提案手法 (単漢字)	37.6	31.8	93.3	82.7	90.5	**72.9**
結合 n-gram(辞書語)	86.4	84.7	93.0	85.7	91.1	67.2
提案手法 (辞書語)	**89.7**	**88.6**	**95.5**	**87.8**	**92.9**	70.2

表 1: 提案手法・ベースライン手法の精度比較。KyTea・提案手法 (教師無し) 以外のモデルは Wiki-Train・News-Train と全辞書を用いて構築されている。

文[5]、平均文字数は 51.8・44.9）

Query-1(Q1)・**Query-2**(Q2): 社内のクエリログからのデータ（1,049・3,078 インスタンス、平均文字数は 3.8・5.7）。一般名詞から新語までをバランスよく含む。

Name(PN): 人名や地名の難読語から構成された 9,170 インスタンス（平均文字数 3.0）。

Wiki(WP): Wikipedia から自動獲得されたペアから抽出した 2,000 インスタンス（人手でチェック済み）。訓練データとの重複はない。平均文字数 4.1。

線型モデルの重みのテューニングには、各ドメインにおいて予め確保されていた 200 インスタンスをそれぞれ用いた。

4.3 実験設定

アライナには、社内で用いられている EM アルゴリズムに基づいた単調アライナを、単漢字の最大読み長が 4 であるとして用いた。ヌル・シンボルは考慮しなかった。スタック・デコーダのビーム幅は 20 とした。頻度による翻訳操作の足切りは行わなかった。翻訳操作の合成は、部分文字列モデルでは最大 4 操作まで、辞書語モデルでは 3 操作まで考慮した。部分文字列モデルでは文字 5-グラム・結合 4-グラムを、辞書語モデルでは文字 4-グラム・結合 3-グラムを、Kneser-Ney スムージングと共に用いた。これら全てのパラメータ・実験設定は予備実験において性能とメモリ効率を考慮して決定された。また、評価指標としては、インスタンス・レベルでの完全一致による精度を用いた。

4.4 ベースライン手法

ベースライン手法として、書記素・音素変換タスク等で最高水準の精度を記録している結合 n-グラムモデルを用いた[6]。このモデルでは、既存手法に準じる部分文字列モデルと、本稿の提案である辞書語モデルの両方を構築して評価を行った。また、日本語読み推定タスクにおける最高水準のシステムである KyTea 0.13 も比較用に用いた。

5 結果と議論

表 1 は、提案手法とベースライン手法の全体的な性能比較を示している。“KyTea” は、作者のウェブページ[7]でダウンロード可能な「高性能 SVM モデル」を用いている。“提案手法 (教師無し)” は、完全な教師なし学習[8]によって訓練された部分文字列モデルである。しかし、Wiki-Train 中の単語・発音ペアにはほとんど動詞のインスタンスが含まれていない為、このモデルは News-1/2 に含まれている完全な文にほとんど対応する事ができず、それぞれ文レベル精度で 18.1%・11.5%に留まった。一方で、下の 4 モデルは Wiki-Train・News-Train の全インスタンスと 3 つの辞書全てを利用して構築されている。“提案手法 (辞書語)” は、6 データセット中 5 セットで最も良い性能を記録し、手法の有効性とロバスト性を示した。Wiki においては、辞書語モデルは単漢字モデルよりも悪い結果となったが、これは辞書語モデルが、Wikipedia に出現する固有名詞の読みに必要な非一般的な読みを切り捨ててしまう為だと考えられる。

モデル	N1	N2	Q1	Q2	PN	WP
KyTea(ノイズ有)	68.5	65.3	88.0	79.5	**67.9**	**65.8**
KyTea(ノイズ無)	**75.3**	**75.5**	91.5	83.4	61.7	64.1
提案手法 (辞書語)	73.8	75.4	**92.8**†	**84.9**†	62.8	64.3

表 2: BCCWJ・Wiki-Train・UniDic で構築されたモデルの精度比較。“KyTea(ノイズ無)” と “提案手法 (辞書語)” の差については、McNemar テストの結果統計的有意だと判定されたものに “†” が付いている。

表 2 は、KyTea と “提案手法 (辞書語)” を、双方同じデータセット（BCCWJ・Wiki-Train・UniDic）を用いて構築し、評価したものである[9]。ここで、“KyTea(ノイズ有)” は訓練コーパス中の全インスタンスを訓練に用いたが、“KyTea(ノイズ無)” は提案手法の辞書語操作を用いてフィルタされたインスタンスのみを訓練に用いた。表 2 から見て取れるように、このフィルタリングは大きな精度向上に寄与している（Name と Wiki において精度が低下しているが、これは本質的な問題ではない[10]）。フィルタリング後の同一データセットを用いた実験では、提案手法は “KyTea(ノイズ無)” を 4 データセットで上回った。News-1/2 では少し悪い結果となったが、これは、新聞記事データには未知語が少なく、与えられた文脈中で既知語の読みを判別するという最も KyTea に適した実験設定になっている為だと考えられる。また、我々のモデルは拡張性を優先するために頻度の正規化を行っていないため、この訓練データのサイズではその有用性が十分に発揮されているとは言い難い。したがって、提案手法は新聞記事などの標準的なデータで KyTea に準じた性能を持ち、他の未知語を多く含むドメインでは有意に優れていると考えられる。

誤り分析の結果、提案手法の出力には結合／文字 n-

[5] アラビア数字・漢数字等を含むインスタンスはシステム間の出力の整合性を取る事が難しいため取り除いてある。

[6] メモリ使用量の制約から、提案手法と同じ $n = 3$ とした。

[7] http://www.phontron.com/kytea/model.html。作者によると、このモデルは BCCWJ・UniDic を含むいくつかの言語資源を用いて構築されている。表 1 の実験では、メモリの制限のために KyTea を提案手法と同じ資源で訓練する事ができなかったが、表 2 の実験では同じ資源を用いて訓練を行っている。

[8] 訓練とテューニングは共に Wikipedia から自動獲得されたインスタンス (Wiki-Train) を用いており、辞書や開発セットは利用していない。

[9] KyTea の訓練は以下のようにして行われた。まず、BCCWJ と UniDic を用いて KyTea の単語分割モデルを学習し、次に、この単語分割モデルを用いて Wiki-Train のインスタンスを再アラインし、一貫した単語分割を持つコーパスを作成する。このようにして作られたコーパスと BCCWJ・UniDic を用いて最終的なモデルを構築した。

[10] 誤り分析の結果、これらの結果は、UniDic には人名に頻繁に用いられる単漢字読み、例えば「美（み）」「人（と）」のような読みが含まれていない為である事が分かった。そのため、人名を多く含んでいる Name・Wiki での結果が悪くなってしまっているが、この問題は被覆率の大きな辞書を用いる事で解決できる。

モデル	N1	N2	Q1	Q2	PN	WP
提案手法 (辞書語)	89.7	88.6	95.5	87.8	92.9	70.2
- 結合 n-グラム	-5.5	-3.3	-1.5	-3.8	-4.4	-4.2
- 翻訳操作の合成	-3.9	-4.0	-2.6	-1.2	-1.8	-2.9

表 3: Wiki-Train・News-Train・全辞書を用いて訓練された辞書語モデルに対する、素性削減実験の結果。全ての精度低下は統計的に有意であった ($p < 0.01$)。

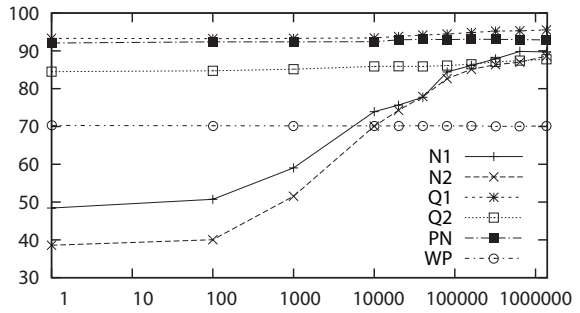

図 3: Wiki-Train に News-Train を徐々に加えた時の、各ドメインにおける提案手法 (辞書語) の性能の推移。

グラム確率の寄与による特徴が多く見られた。例えば、「契約切れ」(KyTea:けいやくきれ) 等の例で、提案手法は正しい解析 (けいやくぎれ) をしており、これは、他の例から「く」の後の「切れ」は「ぎれ」となる、というような性質を学習しているためと考えられる。一方で、我々の手法は字種素性等の一般化された文脈素性を簡単に組み込む事ができない為、「ブランド米」(出力：べい) のような例の解析に失敗したが、KyTea は字種素性により「片仮名列+米 (まい)」というような性質を学習して、解析に成功している。

表 3 は、辞書語モデルにおける素性削減実験の結果を表している。翻訳操作の合成と結合 n-グラム素性の両方が、全てのドメインにおいて有意に精度向上に寄与している事が見て取れる。結合 n-グラムの寄与は、既知語の読み曖昧性解消の為のスムーズな文脈を提供する事 (News-1/2 で特に有用) と、未知語の読み推定の為に単漢字の読み同士の依存関係を取り入れる事 (特に Wiki・Name で有用) の 2 点にあると考えられるが、これらの結果は、結合 n-グラムによってこれら両方の寄与が確認できた事を示している。一方で、翻訳操作の合成は、特に新聞記事において大きな精度向上に貢献している事から、未知語の読み推定よりも、既知語の読み判別に強く貢献していると考えられる。

図 3 は、訓練に用いた News-Train の文数とモデルの性能の関係を表す。まず最初に、モデルは Wiki-Train だけを用いて訓練され、その後、News-Train の文を徐々に増やしながら実験を行っている。この過程は、未知語の読み推定モデルを、既知語を含む文レベルの読み推定が可能なモデルに適応させる過程と考えられる。グラフから見て取れるように、新聞記事からの訓練データを加える事で、新聞記事における性能は大幅に向上している一方で、新聞記事以外の評価セットでは性能がほとんど変化しなかった。従って、Wikipedia から自動獲得されたデータと新聞記事両者の訓練データを混合する事で、特定の分野における性能を低下させることなく、全体の精度を向上させる事が可能である事が示された。

6 結論

本稿では、日本語の読み付け問題に対する統合的な手法を提案した。提案手法は、句ベースの統計的機械翻訳手法に基づき、既知語に対する読み判別問題と未知語に対する読み推定問題を統合的に扱う事を可能にした。提案手法の主要な要素は教師無し学習によって学習され、ノイズに対しても頑健である事から、新たな分野に簡単に適応する事が出来る。また、様々な分野の評価セットにおいて実験を行った結果、提案手法は既存手法よりも優れており、ほぼ全分野において 90%近い精度を持つ事が分かった。更に、提案した翻訳操作の合成と結合 n-グラムの利用は、有意にモデルの精度を向上させる事が分かった。最後に、誤り分析の結果、SVM による既存手法は提案手法とは異なった性質持つ事が分かり、今後、そのような手法が利用している文字種素性や、将来的には音訓読みの依存関係等を利用する事で、更なる精度向上が期待できると考えられる。

謝辞

本稿の実験を行うに当たって KyTea に関する詳しい情報を提供して下さった京都大学の Graham Neubig 氏に、この場を借りてお礼申し上げます。

References

Colin Cherry and Hisami Suzuki. Discriminative substring decoding for transliteration. In *EMNLP-2009*, 2009.

Jun Hatori and Hisami Suzuki. Predicting pronunciation in Japanese. In *CiCLing-2011*, 2011.

Sittichai Jiampojamarn, Colin Cherry, and Grzegorz Kondrak. Integrating joint n-gram features into a discriminative training framework. In *NAACL-2010*, 2010.

Gakuto Kurata, Shinsuke Mori, Nobuyasu Itoh, and Masafumi Nishimura. Unsupervised lexicon acquisition from speech and text. In *ICASSP-2007*, 2007.

Hisami Suzuki and Jianfeng Gao. Microsoft Research IME Corpus. Technical Report MSR-TR-2005-168, Microsoft Research, 2005.

Eiichiro Sumita and Fumiaki Sugaya. Word pronunciation disambiguation using the web. In *NAACL-2006*, 2006.

Richard Zens and Hermann Ney. Improvements in phrase-based statistical machine translation. In *HLT-NAACL 2004*, 2004.

Hao Zhang, Chris Quirk, Robert C. Moore, and Daniel Gildea. In *ACL-2008*, 2008.

笹田鉄郎, 森信介, 河原達也. 未知語を含む文脈情報の自動獲得による統計的仮名漢字変換システムの分野適応. 言語処理学会第 15 回年次大会, 2009.

森信介, Graham Neubig. 仮名漢字変換ログの活用による言語処理精度の自動向上. 言語処理学会第 16 回年次大会, 2010.

森信介, 笹田鉄郎, Graham Neubig. 確率的タグ付与コーパスからの言語モデル構築, 2010.

前川喜久雄. KOTONOHA『現代日本語書き言葉均衡コーパス』の開発. 日本語の研究, Vol. 4, pp. 82–95, 2008.

長野透, 森信介, 西村雅史. N-gram モデルを用いた音声合成のための読みおよびアクセントの同時推定. 情報処理学会論文誌, Vol. 47, pp. 1793–1801, 2006.