Vol.2011-NL-203 No.4
2011/9/16

日本語教育のための誤り訂正ローマ字かな変換

笠原誠司[†1]　小町守[†1]
永田昌明[†2]　松本裕治[†1]

本稿では日本語学習 SNS において，ローマ字で書かれた学習者の文を仮名に変換することで，添削者が訂正を容易に行えるよう支援する方法について述べる．我々のシステムは外国語の単語を検出し，日本語の単語のみを変換する．また単語のスペルに誤りが含まれていても変換することができる．学習者の作文に対し実験を行い既存の日本語入力システムよりも 10%高い単語変換精度を達成した．誤り解析を行うことにより，母音同士を混同しやすい，母語の発音の影響を受けた書き方をしてしまう，といった学習者の誤りの傾向を明らかにした．

Error Correcting Romaji-kana Conversion for Japanese Language Education

Seiji Kasahara,[†1] Mamoru Komachi,[†1]
Masaaki Nagata[†2] and Yuuji Matsumoto[†1]

We present an approach to help Japanese editors on language learning SNS correct learners' sentences written in roman characters by converting them into kana. Our system detects foreign words and converts only Japanese words even if it contains spelling errors. Experimental results show that our system achieves about 10 points higher conversion accuracy than one of traditional input methods. Error analysis reveals tendency of errors made by learners. For example, learners tend to be confused by vowels and make errors caused by nature of their native language.

†1 奈良先端科学技術大学院大学
Nara Institute of Science and Technology
†2 NTT コミュニケーション科学基礎研究所
NTT Communication Science Laboratories

1. はじめに

独立行政法人国際交流基金によると，2009 年現在，133 の国や地域で 365 万人が日本語を学習している．[⋆1]日本語学習者はそれぞれ異なった文化や，母語の統語・音韻論的な影響など様々な背景がある．日本語は普通，中国語から取り入れた表意文字である漢字と，音節を表す文字である仮名を用いて書かれる．仮名には約 50 個の異なる文字がある．これらをすべて覚えるのはラテン言語圏の日本語学習者には負荷が高いため，多くの学習者はローマ字を用いて学習を始める．しかし，日本語母語話者にとっては，仮名で書かれている方がローマ字で書かれているよりも添削しやすい．そのため，ローマ字を仮名に変換することにより，日本語添削者が楽に添削できるようになるが，学習者の文はスペル誤りを含んでいるため，単純なローマ字仮名変換機能はうまく動作しない．一般に用いられてきた日本語入力システムはローマ字を仮名に変換する機能を持っているが，これらは日本語を母語とする者を対象にしているため，学習者の誤りを含む入力を正しく扱うことができない．

本稿ではローマ字で書かれた誤りを含むテキストを，誤り訂正された仮名のテキストに変換するタスクを提案する．これにより，日本語添削者が学習者の文を添削する際，文を容易に読むことができるようになる．我々の手法は 3 つのステップから構成される．すなわち，言語の特定，スペル訂正，そしてローマ字仮名変換である．以下ではそれぞれについて詳しく説明する．

言語の特定　日本語学習者は日本語には存在しない，自分の母語の単語をそのまま書くことがある．日本語のローマ字化の方法は，単純にスペルを反映したものではないため，これらの単語はうまく仮名に変換できない．そのため，これらの単語は仮名に変換せずに残しておいたほうが日本語添削者にとっては読みやすい．本稿では英語の辞書を用意し，マッチする単語は仮名に変換せずに残しておく．

スペル訂正　誤りを含む単語は正しく変換できないので，スペル訂正が有効である．本稿ではコサイン類似度と編集距離を組み合わせて訂正を行う．

ローマ字仮名変換　前述の方法で訂正された文字列に対し，単純な方法を用いて，ローマ字を仮名に変換する．

実験は言語学習 SNS から収集したデータを使用し提案手法で変換した結果と，標準的な日本語入力システムの結果を比較し，正しく変換できなかった場合の誤り傾向を分析し，さ

⋆1 http://www.jpf.go.jp/j/about/press/dl/0542.pdf

らに学習者の誤りの特徴を紹介する．

本稿の構成を述べる．まず 2 節で誤りを含む入力の訂正や変換，言語特定に関する研究を概観する．3 節では，日本語のローマ字化について簡単に触れ，4 節で言語学習 SNS のひとつである Lang-8 とそこから得られる文の特徴を述べる．5 節で我々が実験に用いた誤り訂正かな漢字変換システムの仕組みを順に従って説明する．6 節では実験内容とその結果について記述する．7 節でシステムが訂正に失敗した事例を示し今後の方向性を議論し，8 節で締めくくる．

2. 関 連 研 究

我々の主な関心は誤りを含む入力をどう扱うかというところにある．文献[7] では仮名で書かれた文における誤り検出と訂正を仮名の文字 n-gram 情報を用いて行った．我々の手法もこの方法に似ているが，ローマ字入力を対象としており，言語特定などの異なった難しさを含んでいる．文献[?] では誤りも許容する中国語の入力システムが紹介された．ローマ字仮名変換は，ピンイン漢字変換と似ているが，我々の目的は日本語教師を支援しているところに違いがある．文献[2] では高精度な言語特定手法が紹介された．日本語学習者の誤り訂正は文献[3] でも試みられている．彼らは仮名漢字混じりの文の訂正を試みたが，我々の対象はローマ字で書かれた文である．

3. 日本語のローマ字化

日本語のローマ字化についてはいくつか異なる種類の方式がある．そのうち多く使われているものとして，ヘボン式，訓令式，そして日本式ローマ字がある．中でもヘボン式が最も広く使われている．ほとんどの日本語学習者もヘボン式を使用しているため，提案手法では変換の方式としてこれを使用することにした．ヘボン式は一般的に英語の音韻論に基づいている．これは日本語の単語の発音をローマ字によって表現するので，英語話者には習得が容易な方法である．ローマ字化にこれら以外にもさまざまな亜種が存在するが，ほとんどの亜種は長音を示すための記号の有無によるものである．

4. Lang-8 から得た学習者のローマ字日本語コーパス

ローマ字で書かれた日本語学習のコーパスは筆者らが知る限りでは存在しない．そのため，ローマ字の学習者コーパスを言語学習 SNS の Lang-8[⋆1]から収集した．このサービス上では約 75,000 人のアクティブユーザが様々なトピックについて記事を書き込んでいる．また各ユーザは自身が学習したい言語をプロフィールに登録しているため，この情報をもとに日本語学習者の文だけを集めた．日本語学習者が書いた文が 925,588 文蓄積されており，そのうち 93.4%に当たる 763,971 文が人手での添削を受けている．さらに約 10,000 文がローマ字で書かれたものである．

表 1 に Lang-8 上で見られる文の例を示す．Lang-8 から得られたデータにはそのインターフェースや文化からくる特徴がある．例えば，1 や 2 のように添削後の文の横に，添削者が補足をするためやコミュニケーションを取るためにコメントをつけている文がある．また 3 や 4 の文のように，ローマ字の文とかな漢字混じりの文を併記している場合もある．5 のようにハイフンを複合語の接続に用いている学習者もいる．6 の文では，この文だけではわからない主語を，添削者が前の文脈から推測して補完している．7 の例では学習者の文が正しく書けているということを示すため,「OK desu」とだけ書かれている．8，9，10 からは学習者の誤りの傾向が見て取れる．8 では hanasemasu を hanashimasu と書いてしまい，活用の誤りが見て取れる．また mada を made としており，母音を混同していると推測できる．9 では ha を no と書いており，助詞の選択に困難さがあるようである．10 では amerikajin を americagen と書いており，amerika と america の母語のスペリングからくる誤りと，jin と gen の母語の発音からくる誤りが組み合わさっている．

Lang-8 から収集したローマ字で書かれたほとんどの学習者の文には，単語と単語の間に区切りが置かれているという特徴がある．ただし動詞とその活用語尾は接続されている．本稿ではこの特徴を利用し，本来なら困難である誤りを含む日本語文の単語切れ目を判断する．他の特徴として，助詞の曖昧性がある．例えば,「は」はローマ字では ha に割り当てられるが，wa と発音されるため，そう書かれていることが多い.「を」を書くときの wo と o,「へ」を書くときの he と e にも同じ曖昧性がある．

5. 誤り訂正ローマ字仮名変換

システムは 3 つの機能からなる．言語の特定，類似語探索による誤り訂正，そしてローマ字仮名変換である．この章では各機能がどのように実現されているかについて詳しく述べる

⋆1 http://lang-8.com/

番号	学習者の文	添削された文
1	Onaka ga itai desu !	Onaka ga itai desu ！　だいじょうぶ？
2	suki ni narimasu.	suki ni narimasu.Perfect!好きになります.
3	Isogashikatta.	Isogashikatta. 忙しかった.
4	gakko wa omoshiroi desu.	gakko wa omoshiroi desu. 学校は面白いです.
5	Tokyo ni irutoki, Meiji-jingu mo ni ikimashita.	Tokyo ni irutoki, Meiji-jingu ni mo ikimashita.
6	Noh ni mimashita.	Nihonjin no tomodachi ga Noh wo misetekuremashita.
7	Konnichiwa!	OK desu
8	nihongo ga sukoshi hanashimasu demo made jouzu ja arimasen.	nihongo ga sukoshi hanasemasu demo mada jouzu ja arimasen.
9	Chichi no atama ga ii desu.	Chichi ha atama ga ii desu.
10	watashi wa americagen desu.	watashi wa amerikajin desu.

表 1　Lang-8 に見られる作文の例

5.1　言語の特定

言語特定はローマ字で書かれた入力文と，英語の辞書及びローマ字化された日本語辞書との照合によって行われる．[⋆1]学習者は翻字によってローマ字化せず，自分の母語の単語をそのまま書き込むことがある．我々の目的は，完全な音訳を行うことではないので，ローマ字化された日本語だけを仮名に変換したい．これを実現するため，私たちは英語の単語辞書を用いた．英語の辞書を用いた理由は，学習者の文中で見つかる日本語以外の単語は英単語がほとんどだからある．各言語の辞書を用意することで容易に他の言語にも拡張可能である．この辞書にマッチした単語は仮名に変換されないようにタグを付けておく．辞書には 155,287 個の単語を持つ WordNet 2.1[⋆2]を用いた．誤り訂正しなくて良い正しい単語を除外しておくため，日本語の辞書も用いた．これには IPADic 2.7.0 を使用した．さらに動詞活用の辞書も使用した．なぜなら，学習者の書く文では動詞とその活用語尾は結合されているが，我々の日本語辞書内では分割して登録されているためである．この辞書は 1991 年度の毎日新聞に登場するすべての動詞の活用形から作成した．日本語の係り受け解析器である CaboCha 0.53[⋆3] を用いて動詞をひとつ以上含む文節を抽出した．これにより抽出された活用形は 243,663 個である．

5.2　誤 り 訂 正

英語の辞書にも日本語の辞書にもマッチしなかった単語を以下の手法によって誤り訂正する．誤り訂正は 2 種類の異なる尺度を用いた類似語探索で行われる．すなわち，文字 uni-gram のコサイン類似度と編集距離である．類似語の探索用辞書には IPADic のみを用いた．

5.2.1　類似語探索を用いた訂正候補の生成

我々は編集距離が最も小さいものを訂正候補として選択したい．編集距離とはある文字列を他の文字列に変形されるために必要な編集操作の最小の操作回数である．ここで，編集操作とは挿入，削除，そして置換のことである．しかし，語彙数が大きい時，編集距離の計算コストが高くなり，システムを動かす上で問題となりうる．[⋆4]そこで，編集距離の計算をする前にコサイン類似度による類似語探索で候補数を絞り込むことで計算コストを削減する．コサイン類似度は文字 n-gram の素性を用いて計算されるが，ここで n には 1 を用いた．これは，辞書内の適切な候補をほぼすべてカバーすることができ，かつ不要な候補数を大幅に削減することが出来るからである．例えば，学習者の書いた単語 packu に対する辞書内の類似語をコサイン類似度で探すと，候補の数は 163 個にまで減らすことが出来る．探索された単語の例には kau， pakku， chikau などが含まれる．コサイン類似度による探索は大規模な単語辞書に対しても非常に効率的に行うことが出来る．[4)⋆5]

5.2.2　最尤候補の選択

文字長によって正規化された文字 n-gram のコストによって最尤候補を選択する．この計算はローマ字の文字 5-gram の言語モデルを用いて行う．入力文字列を先頭から 5 文字切り出し，そのコストを言語モデルから求め，末尾までのコストの総和を求めた後，文字長で割ることで入力文字列のコストを求める．言語モデルは 1991 年度の毎日新聞を kakasi を用

⋆1 ローマ字化には kakasi 2.3.4. を用いた． http://kakasi.namazu.org/
⋆2 http://wordnet.princeton.edu/
⋆3 http://chasen.org/~taku/software/cabocha/
⋆4 実験では入力と訂正候補との編集距離が 1 までの文字列のみ使用した．なぜならそうしたときが予備実験で一番良い結果を出したからである．
⋆5 http://www.chokkan.org/software/simstring/

学習者の文	正しい文	仮名に変換された文
yorushiku onegia shimasu.	yoroshiku onegai shimasu.	よろしく おねがい します.
Muscle musical wo mietai.	Muscle musical wo mitai.	Muscle musical を みたい.
Gorofu ga daisuki desu	gorufu ga daisuki desu	ごるふ が だいすき です

表 2 Lang-8 の作文から作成したテストデータの例（下線は訂正箇所を表し，波線は日本語以外の単語を表す）

いてローマ字化したのち，SRILM 1.5.12[⋆1]を用いて作成した．なお言語モデルを作成する際に Witten-Bell スムージングによって平滑化している．

5.3 ローマ字仮名変換

ローマ字列を貪欲法で最長一致する仮名文字列へと変換していく．もし単語が長音記号を含む場合，長音を意味する 2 つの母音へと展開し変換する．ヘボン式で使われていない文字は，英語で似た発音を持つ他の文字として解釈する．つまり，ca, ci, cu, ce, co はそれぞれ ka, shi, ku, se, ko として扱い，母音を伴わない m は n として扱う．この様にした理由は，母語が英語の学習者が多いため，英語の発音を近づけたからである．ほとんどのローマ字のペアは曖昧性なく仮名へと変換されるが，いくつかのペアは複数の可能性を含んでいる．例えば，kinyuu という文字列は「金融」とも「記入」とも読むことが出来る．これは n の音が母音を伴わなくても単独で音節を構成できることに起因する．これらの問題は，本稿での主題からは外れており，また実際には日本語母語話者が添削した文をローマ字に戻すときには曖昧性なくローマ字に変換することができるため，本研究では対象としない．

6. 実　　験

本手法を用いて，ローマ字文字列を誤り訂正した後，仮名に変換するときの評価を行った．また，既存の日本語入力システムと結果を比較することで有用性を確かめた．

6.1 評 価 方 法

誤り訂正の精度について評価を行った．また，適合率と再現率についても評価を行った．適合率と再現率は以下の式で求めた．

$$Recall = \frac{N_t}{N_w}, Precision = \frac{N_t}{N_e}$$

ここで Nt，Nw，Ne はそれぞれ，システムによって誤った単語から正しい単語へ訂正されたものの数，誤りを含む単語の数，システムによって編集された単語の数を表す．

手法	精度	適合率	再現率
Anthy（ベースライン）	74.5	66.7	69.7
提案手法（類似語探索なし）	84.5	76.6	77.3
提案手法（類似語探索あり）	85.0	78.1	78.6

表 3 各システムの誤り訂正の成績

6.2 実 験 設 定

結果を比較するため，Anthy 7900[⋆2]をベースラインとして用いた．これはデファクトスタンダートとなっている日本語入力システムのうちの一つであり，言語特定と類似語探索のどちらも行わない．ここで Anthy の名誉のために，これは誤り訂正のために特別に作られたものではないことを記しておく．さらに我々の手法に関しても，類似語探索を用いたものと，用いていないものについてそれぞれ実験を行ない，結果について比較した．

6.3 使用データ

3 節で述べたように添削者によって編集の仕方にばらつきがあるため，Lang-8 上の学習者の文と，添削者の文を単純に対応付けただけでは正しい添削になっている保証がない．そこで，我々は正解データを自分たちで添削し直し作成した．Lang-8 からローマ字文を 500 文集めてきて使用した．いくつかの文にはすでに添削者によって修正が加えられていたが，質の均一性を保つため自分たちで添削を付け直した．このとき，学習者の文には様々な誤りが含まれていたが，スペル誤りだけ訂正されたデータを作成した．なぜなら，本論文の主題はスペル誤り訂正を施すことにあるからである．表 2 に作成したデータセットの例を示す．

6.4 実 験 結 果

表 3 にスペル訂正の精度を示す．提案手法の単語精度は 85.0%で，Anthy の 74.5%よりも 10 ポイント高かった．我々のシステムで類似語探索を用いなかった場合の精度は 84.5%であった．このことより，言語特定は本手法において非常に重要であることがわかる．正しく訂正された単語の例を表 4 に示す．類似語探索を行わなかった場合の再現率は 77.3%であ

⋆1 http://www-speech.sri.com/projects/srilm/

⋆2 http://anthy.sourceforge.jp/

番号	学習者の入力	仮名	正しい文
1	domou	どもう	doumo
2	Yorushiko onegai shimasu	よるしこ おねがい します	yoroshiku onegai shimasu
3	Merrii kurisamasu, mina-san	めっりい くりさます， みなさん	merii kurisumasu minasan
4	domo arigato guzaimasu	ども ありがと ぐざいます	doumo arigatou gozaimasu
5	nihongo ga scoshi wakarimasu	にほんご が s こし わかります	nihongo ga sukoshi wakarimasu
6	hajimimashtei	はじみま sh てい	hajimemashite
7	donna eigaosaiking mimashitaka	どんな えいがおさいきん みましたか	donna eiga wo saikin mimashitaka
8	Horandajin desu	ほらんだじん です	orandajin desu
9	Nihon go wa totemo musugashi desu	にほん ご わ とても むすがし です	nihon go wa totemo muzukashii desu

表 5 特徴的な学習者の誤り方の例（下線は訂正箇所を示す）

番号	誤りを含む文	仮名	正しい文
1	Soshite, kurama wo durivu wo shimasu	そして, くらま を どらいヴ を します	Soshite, kuruma wo doraibu wo shimasu
2	boku wa nagai ichi-nichi no renshou o shimasu	ぼく わ ながい いちにち の れんしょう お します	boku wa nagai ichi nichi no renshuu o shimasu
3	Terebi gamu wo asobitai desu	てれび げーむ を あそびたい です	terebi geemu wo asobitai desu

表 6 異なる単語にマッチした例（下線は訂正箇所を示す）

誤りを含む単語	仮名	正しい単語	仮名
shuutmatsu	しゅう t まつ	shuumatsu	しゅうまつ
do-yoobi	どよおび	doyoubi	どようび
packu	ぱ c く	pakku	ぱっく

表 4 システムにより正しく訂正された単語の例（下線は訂正箇所を示す）

誤りを含む単語	仮名	正しい単語	編集距離
durivu	づりヴ	doraibu	3
prutugarogo	p るつがろご	porutogarugo	3
musugashi	むすがし	muzukashii	3

表 7 編集距離が遠すぎる単語の例（下線は訂正箇所を示す）

り，類似語探索で誤り訂正することにより 78.6%に改善された．正しく訂正できた単語の例を表 4 に示す．ここで，下線が引かれている単語は誤りを含むもの，波線が引かれているものは日本語ではない単語を表す．適合率も 76.6%から 78.1%へ，類似語探索を用いることで向上した．これらの結果から，類似語探索による誤り訂正は再現率を下げることなく適合率を改善することができ，誤り訂正に有効であると言える．しかし，ベースラインでさえ正解率が低く，このタスクは困難であることを示している．

7. 議　　論

表 3 から読み取れるとおり，類似語探索を用いた誤り訂正はローマ字仮名変換の性能を向上させたが，依然この手法には改善の余地が残されている．ここでは誤りの種類を調査し，解決するための案を紹介する．システムが正しく校正できなかった単語のうち，最も多かった 3 つの誤りのタイプをパターンごとに以下に示す．それは違う単語とのマッチング，入力単語と正しい単語との遠すぎる編集距離，そして複合語である．

7.1 異なる単語とのマッチング

入力された文字列が辞書内に存在していたとき，本来学習者が意図していた単語と異なっていてもマッチしてしまう．その事例を表**??**に示す．例えば，学習者が renhuu という単語を書きたかったが，誤って renshou と入力してしまった場合，この単語を修正することはできない．なぜなら，renshou という文字列は日本語の単語辞書内に存在するからである．この種の問題は，文脈の情報を用いることで改善が期待できるため，単語 n-gram モデルを用いてみる価値はある．

7.2 遠すぎる編集距離

入力からの編集距離がある閾値よりも低い単語は訂正候補として選択されない．その事例を表 7 に示す．たとえば学習者が muzukashii を musugashi と書いてしまったとすると，この 2 つの文字列間の編集距離は 3 である．これは我々の設定した閾値よりも高い値である

学習者の入力	仮名	望む入力
denwabangou	でんわばんごう	denwa bangou
Meiji-jingu	めいじじんぐう	meiji jinguu
nouryokushiken	のうりょくしけん	nouryoku shiken

表 8 訂正できなかった複合語の例

ため訂正されない．閾値の値をより大きくすることもできるが，大きくすればするほどあまり似ていない単語が候補に増えてきてしまう．この問題は，表 5 に見られるような学習者の誤り方の傾向を反映した重み付けを編集距離に与えることで改善されると考えられる．例えば，1，2，3，4 番の文では doumo を domou と書いてしまう，というように学習者が母音を混同している例が学習者の文中で多く見られた．そこで母音の置換や入れ替えには低いコストを設定することで，このような音韻的な混同に基づく誤りを訂正できると考えられる．また，母語の発音の影響と考えられるが，5，6 番の文で見られるように su や shi を書くとき母音を抜かして書いているものが多い．7 番の文では n の発音を ng と書いている．8，9 番の文でも母語の発音の影響とみられる誤り方をしている．これらの場合も同様に学習者の誤り方の傾向を反映した編集距離の重み付けを施すことで精度が改善されると期待できる．例えば，子音だけで日本語を入力する手法が文献[5] で提案されており，母音の編集距離を低くしても意図した入力を推測することができる．

7.3 複 合 語

我々の手法は，辞書の区切りの粒度と学習者の文の区切り粒度が等しい時にのみ有効である．区切りの粒度が異なるためにうまく訂正できなかった例を表 8 に示す．例えば，nouryokushiken という単語は nouryoku と shiken という 2 つの単語としても扱うことができ，実際 IPADic では 2 つの単語として扱われており，nouryokushiken という単一の単語は登録されていない．そのため，マッチングの際に探し当てることができない．この問題を解決するためには，単語分割の技術を各入力文字列に適用することが有効であると考えている．

8. おわりに

本稿ではローマ字で書かれた学習者の文を仮名に変換する手法を紹介した．我々のシステムの目標は言語学習 SNS の添削者が，楽に学習者の文を読めるようにすることである．言語特定と類似語探索を用いたスペル訂正システムを紹介した．このシステムは学習者のローマ字で書かれた文を仮名に変換するタスクにおいて，既存の日本語入力システムよりも 10 ポイント高い精度を達成した．実験結果に対し誤り分析を行なうことで，異なる辞書にマッチングしてしまう場合や，編集距離が遠すぎる場合，複合語などにうまく対応できないことがわかった．また学習者は母音を混同しやすいという傾向や，母語の影響を受けた誤り方をする傾向があるということがわかった．今度は文脈の情報を用いる，文字列を単語分割の手法で解析，学習者の誤り傾向を反映する，などして誤り訂正ローマ字仮名変換の精度を高めたい．

謝　　辞

今回実験でデータを使用させていただいた Lang-8 を運営されておられる喜洋々さんに感謝いたします．

参 考 文 献

1) Zheng Chen and Kai-Fu Lee. A New Statistical Approach to Chinese Pinyin Input. In *Proceedings of ACL*, pp. 241–247, 2000.
2) Yo Ehara and Kumiko Tanaka-Ishii. Multilingual Text Entry using Automatic Language Detection. In *Proceedings of IJCNLP*, pp. 441–448, 2008.
3) Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. Mining Revision Log of Language Learning SNS for Automated Japanese Error Correction of Second Language Learners. In *Proceedings of IJCNLP*, 2011.
4) Naoaki Okazaki and Jun'ichi Tsujii. Simple and Efficient Algorithm for Approximate Dictionary Matching. In *Proceedings of COLING*, pp. 851–859, 2010.
5) Kumiko Tanaka-Ishii, Yusuke Inutsuka, and Masato Takeichi. Japanese input system with digits –Can Japanese be input only with consonants? In *Proceedings of HLT*, pp. 211–218, 2001.
6) Yabin Zheng, Chen Li, and Maosong Sun. CHIME: An Efficient Error-Tolerant Chinese Pinyin Input Method. In *Proceedings of IJCAI*, pp. 2551–2556, 2011.
7) 新納浩幸. 平仮名 N-gram による平仮名列の誤り検出とその修正. 情報処理学会論文誌, Vol.40, No.6, pp. 2690–2698, 1999.