仮名漢字変換ログを用いた単語分割・読み推定の精度向上

高橋 文彦[1,a)]　森 信介[2,b)]

概要：単語分割・読み推定の課題として，未知語の多いテキストを頑健に解析できないという問題がある．本研究ではこのような問題に対処するために，文章を作成するときに用いる仮名漢字変換のログを参照する方法を提案する．仮名漢字変換ログとは，インプットメソッドで文章を作成するときの履歴であり，単語境界や入力記号列の情報を含んでいるため，アノテーションデータと見なすことができる．一方で変換ログは，誤った確定結果などを含むためノイズありのアノテーションデータだといえる．本論文では，ノイズを含んだアノテーションデータを学習データに利用する３つの方法を提案する．実験では，Twitterを題材として提案手法を評価し，単語分割・読み推定ともに精度が向上することを確認し，提案手法の有効性を示した．

1. はじめに

音声認識 [1] や機械翻訳 [2]，仮名漢字変換 [3] では，単語分割・読み付与された日本語テキストを機械的に処理することで実行される．このため日本語テキストを単語分割・読み付与する処理が最も基本的で精度を大きく左右する [4]．現在テキスト解析は，大半の単語が既知語で構成されるテキストであれば，高い精度で解析できることが知られている [5]．しかし一方で，未知語の多いテキストは頑健に解析することができない．例えばウェブテキストには固有名詞や新語などの未知語が頻繁に出現するため，高い精度で解析を行うことが難しい．本研究では，このようなテキスト解析時における未知語の問題に対処するために，仮名漢字変換ログを用いる方法を提案する．仮名漢字変換ログとは，インプットメソッドで文章を作成するときの変換の履歴であり，単語境界や入力記号列の情報を含んでいるため，アノテーションデータと見なすことができる．未知語候補も変換候補に挙げる仮名漢字変換システム [6] を用いて変換ログを収集することで，ユーザーが未知語を変換結果として選択すると未知語を含むアノテーションデータが獲得できる．このアノテーションデータを言語資源として用いることで，仮名漢字変換精度が向上することがすでに知られている [7]．しかし変換ログは，後述するようにノイズありのアノテーションデータだといえる．本研究では，

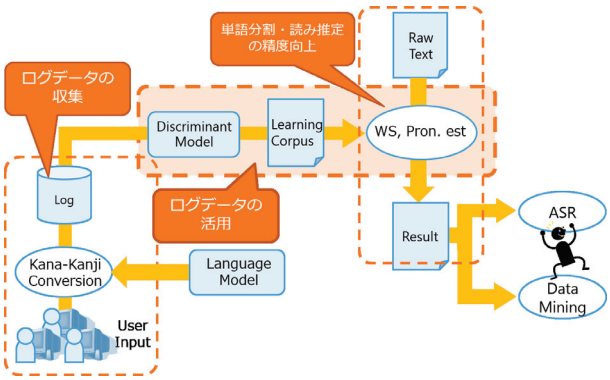

図 1　システムの概要

変換ログを単語分割器・読み推定器の学習データとして使えるように加工して単語分割・読み推定の学習に用いる．本研究の仮名漢字変換ログを単語分割・読み推定に利用する方法は，仮名漢字変換という人が自然に行う言語処理を通じて半自動的にアノテーションデータが得られる．本研究の概要を図 1 に示す．

本論文では，未知語が多く含まれたテキストとしてツイートを対象として提案手法を評価する．ツイートとは Twitter 社により提供されるサービス Twitter[*1]における，短文の投稿である．ツイートは災害時の情報解析 [8]，抑鬱傾向の推定 [9]，音声対話 [10] などの応用研究で利用されており，これらの応用の前処理として単語分割や読み推定が利用されている．しかし一方で，後述の実験で明らかにするが，ツイートの単語分割・読み推定精度は不十分である．従って，ツイートの単語分割・読み推定の精度向上は要求に迫られた課題といえる．

[1] 京都大学情報学研究科
京都府京都市左京区吉田本町
[2] 京都大学学術情報メディアセンター
京都府京都市左京区吉田本町
[a)] takahasi@ar.media.kyoto-u.ac.jp
[b)] forest@i.kyoto-u.ac.jp

[*1] https://twitter.com

2. 関連研究

本論文が提案するのは，人の自然な行動から単語分割や読み推定に有用な情報を獲得する枠組みである．ここでは、単語分割や読み推定の関連研究について概観し、様々な情報を用いてそれらの精度向上を図る研究について述べる．

単語分割は、日本語や中国語などの単語境界を明示しない言語に対する最初の処理であり、多くの研究がある．研究の初期は、人手で作成した規則に基づく方法 [11] が主流であったが、90 年代の中ごろからコーパスに基づく方法が主流となっている．統計的手法としては、まず、単語や品詞あるいは自動推定したクラスの n-gram モデルによる方法が提案された [12][13][14]．次に条件付き確率場に基づく方法により精度が向上することが示された [15]．精度向上には、機械学習手法の改善と同様に言語資源の追加が重要であることが改めて認識されるに至って、言語資源を有効活用 [16] するために、一部の単語にのみアノテーションがなされた部分的単語分割コーパスから条件付き確率場学習可能とする拡張が行われた [17]．さらに、能動学習を可能とするために学習時間を短縮することなどを意図して、点予測による方法が提案されている [18]．本論文では、基礎となる単語分割の手法として、この点予測による方法を採用している．

一方、読み推定の研究は、音声合成のフロントエンドとして、音声言語処理の分野でなされてきた．読みに加えて、アクセントを同時に推定する統計的手法が提案されている [19]．この手法は、単語と読みとアクセントの組の n-gram モデルに基づいている．読み推定についても、柔軟な言語資源の参照を意図して、点予測による方法が提案されている [20]．本論文では、基礎となる読み推定の手法として、この点予測による方法を採用している．

単語分割については、人が言語処理での利用を意図して作成した学習データ (コーパスや辞書) 以外を用いて精度向上を実現する方法が近年研究されている．例えば人のために執筆された辞書の見出し語の利用が挙げられる [21]．人のための辞書の見出し語は、自然言語処理の単語分割基準に照らし合わせると複合語になっていることが多く、その利用方法は自明ではない．この論文では、見出し語の両端に単語境界があるという情報を自動単語分割に活用している．ほかに、Wikipedia などの HTML タグのある文章に対して、そのタグを単語境界とみなして、単語分割器を学習する方法が提案されている [22] [17] [23] ．本論文で利用する仮名漢字変換ログも、人が意図して作成した言語資源ではないという点で、これらの研究と類似している．しかしながら、後述するように、仮名漢字変換ログは多くのノイズを含んでおり、利用がより困難であると考えられる．また、読みの情報を取得することも可能であり、読み推定の精度向上にも貢献する．

3. 未知語を提示する仮名漢字変換

本研究で用いる仮名漢字変換システムでは，ユーザーに未知語候補を提示し，その未知語が選択されることで，未知語を文脈と共に獲得することができる．このため，既知語のみを提示する通常の仮名漢字変換システムでなく，既知語に加えて未知語候補も変換候補に挙げる仮名漢字変換システムとして，単語と入力記号列の組を単位とする確率的モデルによる仮名漢字変換 [24] を用いた．

3.1 擬似確率的コーパス

本研究では，インプットメソッド利用者が未知語候補を変換候補から選択して変換結果を確定することで，未知語が変換ログに残り，未知語の獲得が可能となる．したがって，仮名漢字変換の変換候補に既知語のみならず，未知語候補を提示する必要がある．この方法として本研究では擬似的確率コーパスから仮名漢字変換の語彙を決定する．

本研究では擬似確率的コーパスを作成するために，アノテーション情報のないテキストから文献 [25] の方法を用いて単語境界を付与し, 文献 [24] の方法を用いて読み情報を付与する．擬似確率的単語分割読み付与コーパスは，確率的単語分割読み付与コーパスの高コストな計算量を軽減する方法として，単語分割読み付与済みコーパスで確率的単語分割読み付与コーパスを近似する方法を用いている．具体的には，まず確率的単語分割コーパスに対して以下の処理を最初の文字から最後の文字まで $(1 \leq i \leq n_r)$ 行なう．

(1) 文字 x_i を出力する．
(2) 0 以上 1 未満の乱数 r_i を発生させ P_i と比較する．$r_i < P_i$ の場合には単語境界記号を出力し，そうでない場合には何も出力しない．

これにより，確率的単語分割コーパスに近い単語分割済みコーパスを得ることができる．これを擬似確率的単語分割コーパスと呼ぶ．同様にして，擬似確率的単語分割コーパスの各単語に対して，最初の単語から最後の単語までその都度発生させた乱数と読みの確率の比較結果から該当単語の読みを決定する．これにより，確率的読み付与コーパスに近い読み付与済みコーパスを得ることができる．これを擬似確率的単語分割読み付与コーパスまたは，単に擬似確率的コーパスと呼ぶ．単語境界確率と読み確率は，5.2 項の点予測を用いて，単語分割読み付与済みコーパスから推定したロジスティック回帰に基づくモデルで計算する．

下記の例では，1 行目の文では「艦これ」を「艦」と「これ」に分割しているが，2 行目の文では「艦これ」を 1 単語としてコーパスにアノテーションされている．これは「艦」と「こ」の間が確率的に分割され，単語境界有無の揺れが生じた結果である．この例では,「艦」と「艦これ」という未知語候補がコーパスにアノテーションされるが，インプットメソッド利用者が「艦これ」を変換候補から選択

することで，ログにこの情報が残り,「艦これ」という未知語が獲得される．

擬似確率的単語分割コーパスの例

昨日 | 艦 | これ | や |っ| て | た
艦これ |っ| て | 面白 | い |？

3.2 表記と読みの組を単位とする言語モデル

仮名漢字変換システムの言語モデルとして，文献 [24] の単語と読みの組を単位とする言語モデルを用いる。確率的モデルによる仮名漢字変換 [6] は，キーボードから直接入力可能な入力記号 $\mathcal{Y}$ の正閉包 $y \in \mathcal{Y}$ を入力として，日本語の文字 $\mathcal{X}$ の正閉包を変換結果として出力する．この際，以下の式が示すように，単語 w を入力記号列 $\boldsymbol{y}$ の組 $u = \langle w, \boldsymbol{y} \rangle$ を単位とする言語モデルによる生成確率を評価基準とする．

$$\begin{aligned}\underset{w}{\mathbf{argmax}}\, P(\boldsymbol{w}|\boldsymbol{y}) &= \underset{w}{\mathbf{argmax}} \frac{P(\boldsymbol{w},\boldsymbol{y})}{P(\boldsymbol{y})} \\ &= \underset{w}{\mathbf{argmax}} P(\boldsymbol{u})\end{aligned}$$

ここで単語列 $\boldsymbol{w}$ は表記文字であることに注意されたい．$P(\boldsymbol{u})$ は，u を単位とする n-gram モデルを用いて，以下のようにモデル化される．

$$P(\boldsymbol{u}) = \prod_{i=1}^{h} P(u_i|\boldsymbol{u}_{i-n+1}^{i-1})$$

$$P(u_i|\boldsymbol{u}_{i-n+1}^{i-1}) = \begin{cases} P(u_i|\boldsymbol{u}_{i-n+1}^{i-1}) & \text{if } u_i \in \mathcal{U} \\ P(\texttt{UU}|\boldsymbol{u}_{i-n+1}^{i-1}) M_{u,n}(u_i) & \text{if } u_i \notin \mathcal{U} \end{cases} \quad (1)$$

ここで $\mathcal{U}$ は言語モデルの語彙 (単語と入力記号列の組の集合) を表す。この式の中の u_i $(i \leq 0)$ と u_{h+1} は、単語を単位とする場合と同様に、文頭と文末に対応する記号 BT である。また UU は未知の組を表す記号である。

式 (1) の $M_{u,n}(u) = M_{u,n}(\langle w, \boldsymbol{y} \rangle)$ は未知語モデルである。従来手法と同様に、大きな学習コーパスを用いれば実際の使用における未知語率は極めて低く、また未知語に対する正確な仮名漢字変換は困難であると考えて、アルファベット $\mathcal{U}$ 上の未知語モデルの代わりにアルファベット $\mathcal{Y}$ 上の未知語モデル $M_{y,n}(\boldsymbol{y})$ を用いることとする。以上から、仮名漢字変換は、以下の式のようになる。

$$P(u_i|\boldsymbol{u}_{i-n+1}^{i-1}) = \begin{cases} P(u_i|\boldsymbol{u}_{i-n+1}^{i-1}) & \text{if } u_i \in \mathcal{U} \\ P(\texttt{UU}|\boldsymbol{u}_{i-n+1}^{i-1}) M_{y,n}(\boldsymbol{y}_i) & \text{if } u_i \notin \mathcal{U} \end{cases} \quad (2)$$

ここで $\boldsymbol{y}_i = y(u_i)$ は $u_i = \langle w_i, \boldsymbol{y}_i \rangle$ の入力記号列である。なお、$M_{u,n}(u)$ の代わりに $M_{y,n}(\boldsymbol{y})$ を用いることは以下の式で与えられる近似であり、$\mathcal{Y} \subsetneq \mathcal{X}$ であるので、入力記号列のみからなる文字列を未知語として出力することになる。

$$M_{u,n}(u) = M_{u,n}(\langle w, \boldsymbol{y} \rangle) \approx \begin{cases} M_{y,n}(\boldsymbol{y}) & \text{if } w \in \mathcal{Y}^+ \\ 0 & \text{if } w \notin \mathcal{Y}^+ \end{cases}$$

この式の $M_{y,n}(\boldsymbol{y})$ のパラメータは、学習コーパスにおける語彙 $\mathcal{U}$ に含まれない表記と入力記号列の組の入力記号列から推定する。これは、学習コーパスにおける未知の組の単語を入力記号列に置き換えた結果から $M_{u,n}(u)$ を推定しているのと同じである。

式 (2) の $P(u_i|\boldsymbol{u}_{i-n+1}^{i-1})$ と $P(\texttt{UU}|\boldsymbol{u}_{i-n+1}^{i-1})$ は、語彙に BT と UU を加えた $\mathcal{U} \cup \{\texttt{BT}, \texttt{UU}\}$ 上の n-gram モデルである。パラメータは、単語に分割されかつ入力記号列が付与されたコーパスから以下の式を用いて最尤推定する。

$$P(u_i|\boldsymbol{u}_{i-n+1}^{i-1}) = \frac{N(\boldsymbol{u}_{i-n+1}^{i})}{N(\boldsymbol{u}_{i-n+1}^{i-1})}$$

ここで，$N(\boldsymbol{u})$ はコーパス中の表記と読みの組列 $\boldsymbol{u}$ の出現回数を表す．

3.3 連語クラス n-gram モデル

本論文では，文献 [26] の連語クラス言語モデルの単位を表記と読みの組に拡張して用いた．連語クラス言語モデルは連語言語モデルとクラス言語モデルを複合した言語モデルである．連語言語モデル [27] は変換精度を向上させ，クラス言語モデル [28] はモデルを小さくすることが知られている．

3.3.1 連語 n-gram モデル

連語言語モデルは，複数の表記と読みの組の連接を連語にまとめ上げ，連語を単位とする n-gram 言語モデルである．連語言語モデルでは単語列 $\boldsymbol{u} = u_1 u_2 \cdots u_m$ は連語列 $\boldsymbol{\gamma} = \gamma_1 \gamma_2 \cdots \gamma_{m'}$ に変換され，

$$p(\boldsymbol{u}) \stackrel{def}{=} p(\boldsymbol{\gamma})$$

と定義される．ただし，連語 γ_i は表記と読みの組列を表す．従って，$p(\boldsymbol{\gamma})$ は単語 n-gram モデルと同様に，

$$\begin{aligned} p(\boldsymbol{\gamma}_1^{\boldsymbol{m'}}) &= \prod_{i=1}^{m'} p(\gamma_i|\boldsymbol{\gamma}_1^{\boldsymbol{i-1}}) \\ p(\gamma_i|\boldsymbol{\gamma}_1^{\boldsymbol{i-1}}) &\approx p(\gamma_i|\boldsymbol{\gamma}_{\boldsymbol{i-k}}^{\boldsymbol{i-1}}) \end{aligned}$$

と計算される．ただし，$k = n - 1$ である．

$$p(\gamma_i|\boldsymbol{\gamma}_{\boldsymbol{i-k}}^{\boldsymbol{i-1}}) \stackrel{def}{=} \frac{N(\boldsymbol{\gamma}_{\boldsymbol{i-k}}^{\boldsymbol{i}})}{N(\boldsymbol{\gamma}_{\boldsymbol{i-k}}^{\boldsymbol{i-1}})}$$

ここで，$N(\boldsymbol{\gamma})$ はコーパス中の連語列 $\boldsymbol{\gamma}$ の出現回数を表す．連語の決定は，コーパスを n 分割し クロスエントロピーが低下するように採用する．

3.3.2 クラス n-gram モデル

クラス言語モデルは，類似した単語をグループにまとめ上げ，クラスを単位とする n-gram 言語モデルである．表記と読みの組 u をクラス c に写像するクラスマップを f とすると，

$$p(u_i|\boldsymbol{u}_{\boldsymbol{i-k}}^{\boldsymbol{i-i}}) \stackrel{def}{=} p(c_i|\boldsymbol{c}_{\boldsymbol{i-k}}^{\boldsymbol{i-i}})p(u_i|c_i)$$

と定義できる．ただし，$c_* = f(u_*)$ である．$p(c_i|\boldsymbol{c}_{\boldsymbol{i-k}}^{\boldsymbol{i-i}})$ と，$p(u_i|c_i)$ は，学習コーパスから次のように最尤推定で求める．

$$p(u_i|\boldsymbol{c}_{\boldsymbol{i-k}}^{\boldsymbol{i-1}}) \stackrel{def}{=} \frac{N(\boldsymbol{c}_{\boldsymbol{i-k}}^{\boldsymbol{i}})}{N(\boldsymbol{c}_{\boldsymbol{i-k}}^{\boldsymbol{i-1}})}$$

$$p(u_i|c_i) \stackrel{def}{=} \frac{N(u_i)}{N(c_i)}$$

ここで，$N(\boldsymbol{c})$ はコーパス中のクラス列 $\boldsymbol{c}$ の出現回数を表す．クラスマップ f は，コーパスを n 分割しクロスエントロピーを基準とする方法 [29] で推定する．

3.3.3 連語クラス言語モデル

コーパスに対して連語化した後にクラス化をして，連語クラス言語モデルを構築する．連語クラスタリング言語モデル構築が出来るツールとして本論文では kasuga[*2][26] を用いる．

3.4 確率的仮名漢字モデル

確率的仮名漢字モデルは，日本語文を単語列 $\boldsymbol{w}$ とみなし，単語と入力記号列との対応関係がそれぞれ独立であると仮定することで以下の式で表される．

$$M_{PM}(\boldsymbol{y}|\boldsymbol{w}) = \prod_{i=1}^{h} P(\boldsymbol{y}_i|w_i)$$

ここで，部分入力記号列 $\boldsymbol{y}_i$ は単語 w_i に対応する入力記号列であり，$\boldsymbol{y} = \boldsymbol{y}_1\boldsymbol{y}_2\cdots\boldsymbol{y}_h$ を満たす．確率 $P(\boldsymbol{y}_i|w_i)$ の値は，単語ごとに入力記号列が付与されたコーパスから最尤推定する．

4. 仮名漢字変換ログの収集

本研究では，仮名漢字変換ログを収集し，これを言語資源として利用することで単語分割・読み推定の精度を向上させる．ここでは，仮名漢字変換ログを収集するインプットメソッド，収集した変換ログの特性について，変換ログを利用する際に問題となる点について説明する．

4.1 変換ログを収集するインプット メソッド

仮名漢字変換ログを収集するために，サーバーサイドで仮名漢字変換を行うインプットメソッド KAGAMI[*3] を作成した．クライアントとサーバーの動作の様子を図 2 に示す．

*2 https://plata.ar.media.kyoto-u.ac.jp/koji/kasuga/
*3 https://plata.ar.media.kyoto-u.ac.jp/takahasi/kagami/

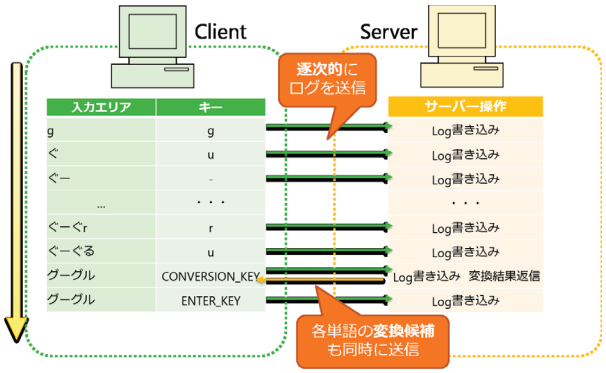

図 2 変換ログを収集するインプット メソッド

インプット メソッドを使う過程は入力過程，変換過程，確定過程の 3 つに分けられる．入力過程はキーボード操作により入力文字列が入力される過程である．この過程における入力文字列が文の読み情報となる．変換過程は Space Key などで入力文字列が表記文字列へ変換される過程であり，変換結果から他の変換候補を選択する過程を含む．この変換過程で文に単語境界情報が付与される．確定過程は Enter Key などで表記文字列を決定する過程である．入力過程，変換過程，確定過程の順に進み入力が完了する．ただし，表記文字列が平仮名のみで構成される場合に多いが，変換過程はスキップできる．

KGAMI は，各過程のログと共にその時間と IP アドレスを逐次的にサーバーに送信する．各過程のログは以下である．

入力過程のログ
入力文字列を入力する際の過程のキーボード操作であり，入力文字の他に文字削除やカーソルの移動を含む．

変換過程のログ
入力文字列を Space Key などで変換する過程のキーボード操作であり，変換結果や変換候補の他に変換後に分割位置を指定して変換する過程のキーボード操作を含む．

確定過程のログ
変換結果を Enter Key などで確定する過程のキーボード操作であり，確定結果などが含まれる．

サーバーは，3 節で述べた仮名漢字変換システムによる変換と，クライアントから受け取った変換ログをログファイルへ書き出しを行う．仮名漢字変換システムでは，次のコマンドと結果を返す．

- `CONVERT` : 入力文字列を仮名漢字変換し，その結果を返す．
- `CONVERT_WITH_1ST_BOUNDARY` : 入力文字列を，指定された分割位置で分割するという制約の下仮名漢字変換し，その結果を返す．
- `LIST_CANDIDATE` : 入力文字列の読みを持つ辞書中の単語と，片仮名化，半角文字化した文字列を返す．

表 1 ‘それに比べると安めかと’というツイートの仮名漢字変換ログ

時間	確定結果	変換過程有無	備考
18:37:11.219621	そ _れ _に	false	変換していない確定結果
18:37:12.603286	くらっ/くらっ| ベル/べる	true	誤って確定した結果
18:37:14.945918	比べ/くらべ | る/る	true	修正の入力
18:37:15.328247	と	false	
18:37:19.828312	も _の _の	false	完成したツイートには残らなかった確定結果
18:37:22.427933	安め/やすめ | か/か | と/と	true	

4.2 仮名漢字変換ログ

一つの仮名漢字変換ログは，確定結果一つに対する入力過程のログ，変換過程のログ，確定過程のログで構成さる．変換ログを収集するインプット メソッド によって得られた変換ログの一部を，確定した時間（確定時間）と確定結果，変換過程の有無と共に表 1 に示す．

変換ログの主要な情報は確定過程における確定結果である．多くの場合確定結果の単位は完全な文ではなく文断片である．また，変換過程がない変換ログの確定結果は単語境界情報が含まれない．さらに，誤まって確定した結果や，2 文字の人名などを他の単語を用いて 1 文字ずつ入力した場合などを含むため読み情報が確かだと限らない．したがって，変換ログはノイズありの単語分割済みかつ読み付与済みの文断片からなるコーパスと見なすことができる．

変換ログをコーパスとして利用するに当たっての問題は大きく分けて 2 つある．表 2 に例を示す． 1 つ目の問題として，ノイズを含む点である．この問題はさらに誤って確定した場合（確定誤り），2 文字の人名などを他の単語を用いて 1 文字ずつ入力した場合（分割入力），表示文字は正しいが分割位置が誤っている場合（分割位置誤り）の 3 つに分類できる． 2 つ目の問題として，情報量が少ない点である．入力の単位が文断片であり前後の文脈が無いため，n-gram 言語モデルにおいての情報が少ないという問題（細分化）である．

本研究では，このような変換ログをコーパスとして使えるように加工して，自動単語分割器や読み推定器から参照する．

5. 仮名漢字変換ログを用いた単語分割・読み推定

本研究では収集した仮名漢字変換ログを，単語分割・読み推定の学習データとして利用できるよう仮名漢字変換ログを加工する必要がある．ここでは，本論文で提案する仮名漢字変換ログの利用方法と，その学習データを利用するために部分的アノテーションから学習できる推定器について説明する．

5.1 仮名漢字変換ログの利用

ノイズありの単語分割済みかつ読み付与済みの文断片からなるコーパスである変換ログを学習データに利用するために，本論文では 3 つの方法を提案する．

5.1.1 確定結果の部分的アノテーションコーパス

確定結果は単語境界，読み情報が付与された部分的アノテーションコーパスと見なすことが出来る．このため，確定結果をそのままコーパスとして利用する．この方法による部分的アノテーションコーパスを本論文では，AS-IS-log と呼ぶ．

表 1 の例をこの方法でコーパスにすると次のようになる．

AS-IS-log の例

```
そ _れ _に
くらっ/くらっ| ベル/べる
比べ/くらべ | る/る
と
も _の _の
安め/やすめ | か/か | と/と
```

AS-IS-log は誤った確定結果を含み， 1 つ 1 つの文断片が短い．

5.1.2 チャンキングした確定結果

細分化の問題を回避するために，確定結果の時間を参照して連結する方法を提案する．

変換ログの確定時間と次の変換ログの入力過程のログの開始時間の差が s 以下の場合，この確定結果を連結する．本論文では，$s = 0.5$[s] とした．この方法による部分的アノテーションコーパスを本論文では，CHUNK-log と呼ぶ．

表 1 の例をこの方法でコーパスにすると次のようになる．

CHUNK-log の例

```
そ _れ _に | くらっ/くらっ| ベル/べる
比べ/くらべ | る/る | と | も _の _の
安め/やすめ | か/か | と/と
```

CHUNK-log は確定誤りのログを含むが， 1 つ 1 つの文断片が AS-IS-log に比べて長い．

5.1.3 ツイートに対する自動アノテーション

作成されたツイートに変換ログをアライメントし，単語分割位置と読みの情報を付与する方法を提案する．この方法により，確定誤りと細分化の問題を回避できると考えられる．確定誤りの変換ログはアライメントされないため学

表 2　仮名漢字変換ログを利用するに当たっての課題

		例
ノイズを含む	確定誤り	あの/あの | 手/て | ー/ー | ション/しょん
	分割入力	有村/ありむら | 架/か | 純/じゅん
	分割位置誤り	に/に | ログ/ろぐ | イン/いん | し/し
情報量が少ない	細分化	[今日/きょう | は/は] [晴れ/はれ] [で/で | す/す]

習データから除外され，文として完成しているツイートにアライメントするため細分化の問題を回避出来る．

変換ログを収集するインプットメソッドでは，完成したツイートを取得をしていないので，まずツイートとそのツイートを作成した際の変換ログを対応づける必要がある．このために KAGAMI 利用者の利用期間のツイートをすべて収集し，以下の条件の $a \wedge (b \vee c)$ を満たす変換ログをツイートに対応づけた．

a. ツイートした時間から１０分以内の確定結果を含む
b. 確定結果の文字列がツイートと３文字以上一致する
c. 直前の変換ログと IP アドレスが一致する

部分文字列一致のみでなく，IP アドレスの一致を条件に含めたのは，誤って確定した結果や推敲の過程で除外された結果を対応づけるためである．また，部分文字列の一致する文字数を３文字以上としたのは，日本語文に 1 文字または２文字の助詞が頻出するためである．

次に，ツイートに対応付けした変換ログの確定結果をアライメントする．変換ログを時系列順で並べ，過去のデータから順にツイートに単語分割位置と読みの情報を付与する．この方法による部分的アノテーションコーパスを本論文では，ALIGN-log と呼ぶ．

表 1 の例をこの方法でコーパスにすると次のようになる．

ALIGN-log の例

そ _れ _に | 比べ/くらべ | る/る | と | 安め/やすめ | か/か | と/と

ALIGN-log は，確定誤りのログや完成したツイートに残らなかった確定結果を含まず，CHUNK-log よりも１つ１つの文断片が長い．

5.2　点予測による単語分割・読み推定

確定結果は文の断片であるので，自動単語分割器や読み推定器は，部分的にアノテーションされたコーパスからの学習が可能である必要がある．そのため，本研究では点予測による単語分割・読み推定を採用した．点予測とは，分類器の素性として，周囲の単語境界や読みなどの推定値を利用せずに，周囲の文字列の情報のみを利用する方法である．

点予測による単語分割の入力は文字列 $\boldsymbol{x} = x_1 x_2 \cdots x_n$ であり，各文字間に単語境界の有無を示す単語境界タグ $\boldsymbol{t} = t_1 t_2 \cdots t_{n-1}$ を出力する．単語境界タグ t_i が取り得る値は，文字 x_i と x_{i+1} の間に単語境界が「存在する」か「存在しない」の２種類である．したがって，単語境界タグの推定は，２値分類問題として定式化される．点予測による単語分割では，文字 n-gram，文字種 n-gram，単語辞書素性の３種類の素性を参照する線形サポートベクトルマシン [30] による分類を行っている．

点予測による読み推定の入力は単語列であるが，読み推定対象の単語以外の単語境界情報を参照しない．この設計により，一部の単語にのみ単語境界や読み情報が付与された部分的アノテーションコーパスが利用可能となる．

(1) 学習コーパスに出現し，複数の読みが付与されている単語は，単語毎の分類器で読みを推定する．
(2) 学習コーパスに出現し，唯一の読みが付与されている単語には，その読みを付与する．
(3) 学習コーパスに出現せず，辞書に出現する単語には，辞書の読みを付与する．
(4) 未知語の場合は，サブワード辞書で学習した未知語モデルによって推定される．

分類器で読みを推定する (1) の場合は，点予測を用いる．点予測による読み推定は，読みを推定する単語 w とその直前の文字列 $\boldsymbol{x}_-$ と直後の文字列 $\boldsymbol{x}_+$ を入力とし，これらのみを参照して単語 w の読みを推定する多値分類問題として定式化される．参照する文字列の窓幅を m' とすると，入力において参照される文脈情報は $\boldsymbol{x}_-, w, \boldsymbol{x}_+ = x_{-m'} \cdots x_{-2} x_{-1}, w, x_1 x_2 \cdots x_{m'}$ となる．すなわち，この文字列と w の前後に単語境界があり，内部には単語境界がないという情報のみから w の読みを推定する．読み推定の分類器には，$\boldsymbol{x}_-\boldsymbol{x}_+$ に含まれる文字 n-gram，$\boldsymbol{x}_-\boldsymbol{x}_+$ に含まれる文字種 n-gram を利用する．単語境界とは異なり，読み推定は多値分類である．したがって，各単語の読み候補毎の分類器をつくる．つまり，ある単語に読み候補が３つ存在すれば分類器はその単語に対して３つ作り，推定には 1 対多方式を用いて多値分類を行う．

この機能があるテキスト解析器として本論文では KyTea[31] を用いる．また，単語分割器・読み推定器ともに窓幅 $m' = 3$ とした．

6.　評価実験

変換ログを用いた学習データを用意し，実際のツイートの単語分割を行う．これを人手によるアノテーションと比較し評価する．

表 3 確率的単語分割・読み推定のための学習コーパス

コーパス			
分野	文数	単語数	文字数
BCCWJ	56,753	1,324,951	1,911,660
新聞記事	8,164	240,097	361,843
英語辞書	11,700	147,809	197,941
辞書			
分野	単語数		
UniDic			234,652
単漢辞書			37,040

6.1 仮名漢字変換システムと仮名漢字変換ログ

アノテーション情報のないテキストから，3.1 項で説明した擬似確率的コーパスを作成する．この未知語候補を含んだコーパスの語彙が，インプットメソッドの変換候補として提示され利用者に選択されることで，本研究は未知語を獲得できる．アノテーション情報のないテキストとして，ツイートと BCCWJ[32] の NonCore データを用いた．ツイートは，13,467,927 件のツイートを収集し，メンション（宛先），ハッシュタグ（検索用のインデックス），URL，ティッカーシンボル（企業情報検索用のインデックス）を除いた本文部分を抽出した．また本文に改行を含むツイートは改行文字前後で文を分割した．つまり，改行文字を 1 文字含むツイートは 2 文に分割される．この結果，786,331 文を得た．BCCWJ の NonCore データは 358,078 文を用いる．これらの 2 つのテキストを合わせた 1,207,182 文から擬似確率的コーパスを作成する．

単語境界確率と読み確率を計算するために，KyTea[31] を用いる．表 3 の学習データを用いて，ロジスティック回帰 [30] を用いたモデルを学習した．この単語分割・読み推定器を用いて，ツイートと BCCWJ の NonCore データの単語境界確率，読み確率を計算し，3.1 項の方法で擬似確率的コーパスを作成した．このコーパスを用いて 3 節の未知語を提示する仮名漢字変換システムを作成した．

この仮名漢字変換システムを 2014/04/13-2014/10/21 の間に 5 人に利用してもらい，22,569 件の変換ログを集めた．この変換ログを実験に使用する．

6.2 テストデータ

2014/05/19-2014/05/22，2014/06/02-2014/06/04 に収集した 2,659,168 件のツイートからランダムにシャッフルし 1,592 件のツイートを選択した．このツイートに対して人手でアノテーションを行った．アノテーション基準は BCCWJ の短単位に準拠し，これに加えて活用語尾を分割する．

これらのツイートから，6.1 項と同様に，メンション，ハッシュタグ，URL，ティッカーシンボルを除いた本文部分を抽出した．これらのツイッター特有のシンボルを除いた理由としては，正規表現で抽出が可能なので応用研究 [8][9] において単語分割の対象にならないと判断したためである．次に，本文に改行を含むツイートは改行文字前後で文を分割した．これらの処理によって，1,592 件のツイートから 2,976 文を得，これをテストデータとした．

表 4 実験で用いるコーパス

学習データ			
記号	文数	単語数	文字数
BCCWJ-train	56,753	1,324,951	1,911,660
AS-IS-log	22,523	-	65,250
CHUNK-log	6,572	-	65,250
ALIGN-log	1,850	-	52,387
テストデータ			
記号	文数	単語数	文字数
TWI-test	2,976	37,010	58,316
BCCWJ-test	6,025	148,929	212,261

表 5 ツイートの単語分割精度

	再現率	適合率	F 値
BCCWJ-train	89.80	94.17	91.93
BCCWJ-train + AS-IS-log	90.17	94.02	92.05
BCCWJ-train + CHUNK-log	90.61	94.34	92.44
BCCWJ-train + ALIGN-log	90.12	94.23	92.13

表 6 一般分野テキストの単語分割精度

	再現率	適合率	F 値
BCCWJ-train	99.01	98.97	98.99
BCCWJ-train + AS-IS-log	98.96	98.89	98.93
BCCWJ-train + CHUNK-log	99.05	98.88	98.97
BCCWJ-train + ALIGN-log	98.99	98.93	98.96

6.3 実験の設定

実験で用いるコーパスを表 4 に示す．変換ログ由来のコーパスは部分的アノテーションコーパスなので，単語数を明記していない．

BCCWJ-train は現代日本語書き言葉均衡コーパスの学習セット，AS-IS-log は変換ログの確定結果（5.1.1 参照），CHUNK-log は確定結果を時間差で連結したもの（5.1.2 参照），ALIGN-log は確定結果をツイートにアライメントしたもの（5.1.3 参照），TWI-test は人手でアノテーションしたツイートの本文，BCCWJ-test は現代日本語書き言葉均衡コーパスのテストセットである．

BCCWJ-train のみを学習データ，AS-IS-log,CHUNK-log,ALIGN-log をそれぞれ BCCWJ-train に追加した学習データ，とする 4 つの学習データでそれぞれ表 3 の辞書と共に単語分割器・読み推定器を学習し，TWI-test，BCCWJ-test に対して単語分割・読み推定を行う．

TWI-test は未知語を多く含んだテキストとして，BCCWJ-test は一般分野のテキストとして実験で用いる．

表 7　未知語への変換ログによる改善

分類	例	改善
表記揺れ	素晴しい (素晴らしい)	○
連濁	(掘り) ごたつ	○
長音化	おいしーい	○
小文字化	ぁなた	×
記号化	あや ι い	×
口語的表現・方言	やっぱ	○
オノマトペ	ぺっちゃり	○
感動詞	いやっほー	○
顔文字・アスキーアート	(^o^)	×
新語	ググる	○
固有名詞	パズドラ	○

6.4　単語分割の評価

単語分割は，単語単位でアライメントを取り，再現率，適合率，その調和平均 (F 値) で評価した．TWI-test の単語分割精度を表 5 に，BCCWJ-test の単語分割精度を表 6 に示す．BCCWJ-test と TWI-test の単語分割精度を BCCWJ-train で比較すると TWI-test の単語分割精度の方が 7%程低く，やはりツイートの単語分割が困難な問題であることがわかる．また，ツイートに対する BCCWJ-train の結果を見ると，適合率が再現率に比べて高いため，過分割が起きていることがわかる．これは，未知語の一部を既知語だと誤認し，未知語の内で分割されていることが原因である．例として,「艦これ」などが挙げられる．これを解決するためにやはり未知語を含んだ文からの学習が必要である．

ツイートの単語分割において，変換ログを用いた学習データを用いると精度が向上し．特に CHUNK-log を学習データに追加すると有意 ($p = 0.05$) に精度が向上した．未知語の分割に関して改善が見られたので，文献 [33] を参考に表 7 にまとめる．小文字化，記号化，顔文字・アスキーアートは本論文の仮名漢字変換システムの変換ログでは改善できない．小文字化，記号化の改善方法としては，置き換わる文字に規則があるので，すべての可能な未知語候補を元の単語の変換候補として提示すれば変換ログとして獲得が可能である．顔文字・アスキーアートは，文献 [34] の方法を用いることで抽出可能だが，構成される文字の読みとインプットメソッド利用者の考える入力に違いがあるため変換ログとしての獲得が困難である．

ALIGN-log を追加する方法は CHUNK-log を追加する方法に比べて精度が低かった．この方法の問題点は，推敲の過程やツイートを作成後につぶやかなかった時の変換ログが利用されないという点にある．実際に ALIGN-log に用いられた変換ログは，すべての変換ログの 51%(11398/22569) 程度だった．

次にログの量とその時の単語分割精度のグラフを図 3 に示す．0.51×xALIGN-log は，ALIGN-log の横軸を 0.51 倍したグラフであり，ALIGN-log に実際に利用されているログの量をシュミレートしたグラフである．いずれのグラフもログの量に応じて単調増加でないことからノイズが含まれていることがわかる．AS-IS-log は，精度が上下するが，やはりノイズが多く含まれるため，ログの量を増やし続ければ精度が上がり続けるわけではない．CHUNK-log は，精度が向上し続けているため，さらにログの量を増やすことでさらなる精度向上が期待される．ALIGN-log は，8,000 件までは精度が向上するものの，8,000 件以降は精度が微増するにとどまっている．0.51×xALIGN-log は，CHUNK-log と同等のログの量で比較しても精度が低かった．これは，アライメントがうまくいってないことにより，ノイズがのったコーパスになっていると考えられる．

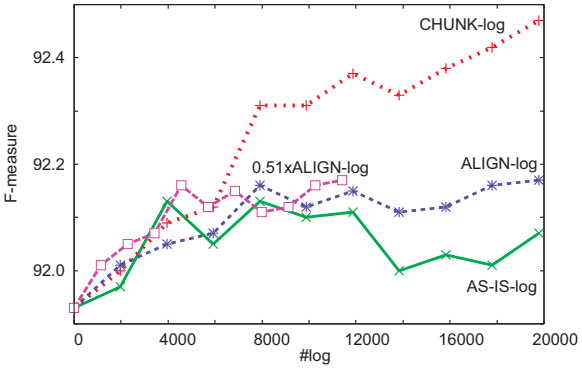

図 3　ログの量に応じた単語分割精度

表 8　ツイートの読み推定精度

	再現率	適合率	F 値
BCCWJ-train	95.14	93.94	94.53
BCCWJ-train + AS-IS-log	95.14	93.86	94.50
BCCWJ-train + CHUNK-log	95.20	93.96	94.58
BCCWJ-train + ALIGN-log	95.17	93.96	94.56

表 9　一般分野テキストの読み推定精度

	再現率	適合率	F 値
BCCWJ-train	99.37	99.35	99.36
BCCWJ-train + AS-IS-log	99.36	99.34	99.35
BCCWJ-train + CHUNK-log	99.37	99.35	99.36
BCCWJ-train + ALIGN-log	99.38	99.36	99.37

また，一般分野テキストの単語分割において，変換ログを用いた学習データを用いても大きく精度が低下することはなかった．これは，変換ログを用いたモデルが一般分野でも十分な解析精度を保つことを意味する．

6.5　読み推定の評価

読み推定は，文献 [20] を参考に，読み情報列を文字単位でアライメントを取り，再現率，適合率，その調和平均 (F 値) で評価した．TWI-test の読み推定精度を表 8 に，BCCWJ-test の読み推定精度を表 9 に示す．BCCWJ-test と TWI-test の読み推定精度を BCCWJ-train で比較すると TWI-test の読み推定精度の方が 5%程低く，単語分割と

同様に，やはりツイートの読み推定が困難な問題であることがわかる．

ツイートの読み推定において，CHUNK-log と ALIGN-log を学習データに用いるとわずかに精度が向上した．一方で AS-IS-log を学習データに用いるとわずかに精度が低下した．ログ由来の学習データの読み情報を確認したところ，誤った読み情報が散見された．これが読み推定の精度向上の障害となっていると考えられる．

7. おわりに

本論文では，未知語が多く含まれるテキストの単語分割・読み推定の精度向上を目的とし，仮名漢字変換のログを利用する方法を提案し，実験的に評価した．仮名漢字変換ログを学習データとして利用することが効果的であり，特に確定結果を連結するとより効果的であることが示された．

謝辞

KAGAMI を利用して仮名漢字変換のログを御提供いただいた皆さんに感謝します．

参考文献

[1] Jelinek, F.: Self-Organized Language Modeling for Speech Recognition, Technical report, IBM T. J. Watson Research Center (1985).
[2] Koehn, P.: *Statistical Machine Translation*, Cambridge University Press (2010).
[3] Chen, Z. and Lee, K.-F.: A New Statistical Approach To Chinese Pinyin Input, *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pp. 241–247 (2000).
[4] 須藤克仁，永田昌明，森　信介：日英特許翻訳における日本語単語分割の分野適応の検討，言語処理学会第 18 回年次大会発表論文集 (2012).
[5] Kudo, T., Yamamoto, K. and Matsumoto, Y.: Applying Conditional Random Fields to Japanese Morphological Analysis, *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 230–237 (2004).
[6] 森　信介：無限語彙の仮名漢字変換，情報処理学会論文誌，Vol. 48, pp. 3532–3540 (2007).
[7] 山口洋平，　森信介，河原達也：変換ログを用いた仮名漢字変換精度の向上，言語処理学会第 17 回年次大会発表論文集 (2011).
[8] Takeshi, S., Makoto, O. and Yutaka, M.: Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors, *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pp. 851–860 (2010).
[9] Tsugawa, S., Mogi, Y., Kikuchi, Y., Kishino, F., Fujita, K., Itoh, Y. and Ohsaki, H.: On estimating depressive tendencies of Twitter users utilizing their tweet data, *VR'13*, pp. 1–4 (2013).
[10] Higashinaka, R., Kawamae, N., Sadamitsu, K., Minami, Y., Meguro, T., Dohsaka, K. and Inagaki, H.: Building a Conversational Model from Two-Tweets, *IEEE Transactions on ASRU*, pp. 330–335 (2011).
[11] Kurohashi, S., Nakamura, T., Matsumoto, Y. and Nagao, M.: Improvements of Japanese Morphological Analyzer JUMAN, *Proceedings of the International Workshop on Sharable Natural Language Resources*, pp. 22–28 (1994).
[12] 丸山　宏，荻野紫穂，渡辺日出雄：確率的形態素解析，日本ソフトウェア科学会第 8 回大会論文集，pp. 177–180 (1991).
[13] Nagata, M.: A Stochastic Japanese Morphological Analyzer Using a Forward-DP Backward-A* N-Best Search Algorithm, *Proceedings of the 15th International Conference on Computational Linguistics*, pp. 201–207 (1994).
[14] 森　信介，長尾　眞：形態素クラスタリングによる形態素解析精度の向上，自然言語処理，Vol. 5, No. 2, pp. 75–103 (1998).
[15] 工藤　拓，山本　薫，松本裕治：Conditional Random Fields を用いた日本語形態素解析，情報処理学会研究報告，Vol. NL161 (2004).
[16] Mori, S. and Neubig, G.: Language Resource Addition: Dictionary or Corpus?, *Proceedings of the Nineth International Conference on Language Resources and Evaluation*, pp. 1631–1636 (2014).
[17] Tsuboi, Y., Kashima, H., Mori, S., Oda, H. and Matsumoto, Y.: Training Conditional Random Fields Using Incomplete Annotations, *Proceedings of the 22nd International Conference on Computational Linguistics* (2008).
[18] Neubig, G., Nakata, Y. and Mori, S.: Pointwise Prediction for Robust, Adaptable Japanese Morphological Analysis, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pp. 529–533 (2011).
[19] 長野　徹，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及びアクセントの同時推定，情報処理学会論文誌，Vol. 47, No. 6, pp. 1793–1801 (2006).
[20] Mori, S. and Neubig, G.: A Pointwise Approach to Pronunciation Estimation for a TTS Front-end, *Proceedings of the InterSpeech2011*, Florence, Italy, pp. 2181–2184 (2011).
[21] Mori, S. and Oda, H.: Automatic Word Segmentation using Three Types of Dictionaries, *Proceedings of the Eighth International Conference Pacific Association for Computational Linguistics* (2009).
[22] Yang, F. and Vozila, P.: Semi-Supervised Chinese Word Segmentation Using Partial-Label Learning With Conditional Random Fields, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 90–98 (2014).
[23] Jiang, W., Sun, M., Lu, Y., Yang, Y. and Liu, Q.: Discriminative Learning with Natural Annotations: Word Segmentation as a Case Study, *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pp. 761–769 (2013).
[24] 森　信介，笹田鉄郎，Graham, N.：確率的タグ付与コーパスからの言語モデル構築，自然言語処理，Vol. 18, No. 2 (2011).
[25] 森　信介，小田裕樹：擬似確率的単語分割コーパスによる言語モデルの改良，自然言語処理，Vol. 16, No. 5, pp. 7–21 (2009).
[26] Maeta, H. and Mori, S.: Statistical Input Method based on a Phrase Class n-gram Model, *Workshop on Advances in Text Input Methods* (2012).
[27] Deligne, S. and Bimbot, F.: Language modeling by Variable Length Sequences: Theoretical Formulation and Evaluation of Multigrams, *Proceedings of the Interna-*

tional Conference on Acoustics, Speech, and Signal Processing, pp. 169–172 (1995).
[28] Brown, P. F., Pietra, V. J. D., deSouza, P. V., Lai, J. C. and Mercer, R. L.: Class-Based *n*-gram Models of Natural Language, *Computational Linguistics*, Vol. 18, No. 4, pp. 467–479 (1992).
[29] 森　信介，土屋雅稔，山地　治，長尾　真：確率的モデルによる仮名漢字変換，情報処理学会論文誌，Vol. 40, No. 7, pp. 2946–2953 (1999).
[30] Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R. and Lin, C.-J.: LIBLINEAR: A Library for Large Linear Classification, *Journal of Machine Learning Research*, Vol. 9, pp. 1871–1874 (2008).
[31] Flannery, D., Miyao, Y., Neubig, G. and Mori, S.: A Pointwise Approach to Training Dependency Parsers from Partially Annotated Corpora, *Journal of Natural Language Processing*, Vol. 19, No. 3 (2012).
[32] Maekawa, K.: Balanced Corpus of Contemporary Written Japanese, *Proceedings of the 6th Workshop on Asian Language Resources*, pp. 101–102 (2008).
[33] 勝木健太，笹野遼平，河原大輔，黒橋禎夫：Web 上の多彩な言語表現バリエーションに対応した頑健な形態素解析，言語処理学会第 17 回年次大会発表論文集，pp. 1003–1006 (2011).
[34] 渡邉謙一，高橋寛幸，但馬康宏，菊井玄一郎：系列ラベリングによる顔文字の自動抽出と顔文字辞書の構築 (2013).