2F1-04

Webページ中のユーザが知らない語を予測する読解支援システム

A reading support system for Web pages with prediction capability of the words unknown to users

江原　遥[*1]　二宮　崇[*2]　清水　伸幸[*2]　中川　裕志[*2]
Yo Ehara　Takashi Ninomiya　Shimizu Nobyuki　Hiroshi Nakagawa

[*1]東京大学情報理工学系研究科
Graduate School of Information Science and Technology, The University of Tokyo

[*2]東京大学情報基盤センター
Information Technology Center, The University of Tokyo

Novel intelligent interface eases the browsing of Web documents written in the second languages of users. It automatically predicts words unfamiliar to the user and glosses them with their meaning in advance. If the prediction succeeds, the user does not need to consult a dictionary; even if it fails, the user can correct the prediction. The correction data are collected and used to improve the accuracy of further predictions of that user. Every user's language ability is estimated by a state-of-the-art language testing model, which is trained in a practical response time using recently proposed batch learning method. This paper totally omits collective intelligence effect and assumes only a small subset of a user's vocabulary is used to predict that user's whole vocabulary. This assumption limits the number of training data, and thus, enables the use of batch learning method because the number of click logs per a user is virtually limited. Evaluation results for the system in terms of prediction accuracy are encouraging even without collective intelligence effect.

1. はじめに

近年，英文 Web ページを読むニーズが増えているのに伴い，読解支援の重要性が増している．第二言語で書かれた Web ページを読む際には，ユーザが知らない語（ユーザ未知語）が読解を妨げる原因の一つとなる．この問題に対処するためには，Web ページ中にあるユーザ未知語の語義を高速に表示する方法が挙げられる．

Web ページ中にある英単語を素早く辞書で引いて対応する辞書の項目（語義）を閲覧するためのインターフェースとして，Web ページを改変し，ページ中のユーザ未知語をクリックしたりマウスオーバーしたりすることで，辞書の項目を表示するシステムが提案されてきた．このようなシステムを，本研究では，**語義注釈システム**と呼ぶ．

図 1 に挙げる **pop 辞書**[*1]は，日本語母語話者向け語義注釈システムの先鞭をとったシステムで，マウスオーバーすると語義をポップアップで表示する．また，**popIn** では，[*2] 選択したユーザ未知語の語義を Web ページ中に埋め込んでいる．

語義注釈システムでは，ユーザがクリックした単語を記録することにより，ユーザのユーザ未知語のログが蓄積される．このログを，本研究では**単語クリックログ**と呼ぶ．単語クリックログは，読解の障害となるユーザ未知語のリストであるので，読解支援にとって有用な情報であると考えられる．既存の語義注釈システムでは，単語クリックログは活用されてこなかったが，単語クリックログを解析することにより，読解の障害となるユーザ未知語を予測して予め語義を付与することで，ユーザはより高速にユーザ未知語の語義を参照することが可能となる．

筆者らは，ユーザの回答パターンが記録されている単語クリックログから学習することによって，既存の語義注釈システムにユーザの語彙を予測する機能を付加したシステム[*3]を提案している．本システムは，ユーザ未知語を自動的に予測し，その語に語義の注釈を付与する．ユーザが本システムにログインし，本システムを通して Web ページを閲覧した図が図 2 である．赤く着色された部分がユーザ未知と判別された部分であり，語義注釈が付与されている．黄色く着色された部分が既知と判別された部分である．

本システムは，任意の Web ページ全てについて適用可能であり，Gmail アカウントを用いてログインした後に，対象としたい Web ページの URL の前に `http://www.socialdict.com/` をつけてブラウザでアクセスすることで，語義注釈の処理を施すことができる[*4]．

2. システムの構造

本節では，提案する語義注釈システムの構造について説明する．図 3 に提案するシステムの構造を図示する．

(0) ユーザはユーザ識別子 u を本システムに渡し，システムにログインする．このログインは，Gmail などのアカウントを用いて行うことも可能であるが，ユーザビリティの向上のためには，ブラウザをユーザとみなしてログインを不要にした方がよいと考えられる．

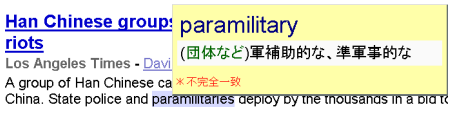

図 1: pop 辞書での注釈の例．

連絡先: 江原　遥，東京大学大学院情報理工学系研究科，東京都文京区本郷 7-3-1 東京大学 総合図書館 4F 情報基盤センター 学術情報研究部門，03-5841-2729，ehara@r.dl.itc.u-tokyo.ac.jp

*1 http://www.popjisyo.com/
*2 http://www.popin.cc/
*3 http://www.socialdict.com/
*4 例：http://www.socialdict.com/http://www.cnn.com/

(1) ユーザは，ブラウザを通じて $l \in URL$ を本システムに渡す．URL は URL の集合とする．ここで，URL を直接指定するのはユーザにとって手間であるため，対象となる Web サイトをあらかじめ定めてしまい URL の入力を不要とした方が，ユーザに取って使いやすいとも考えられる．

(2) システムは，渡された l が指し示す **Web サーバ** (Web Server) にアクセスする．

(3) Web サーバは l を受け取り，l に対応する Web ページ D を探索し（$D = find(l)$），システムに返す．

(4) システムは，D に注釈をつけて返す．この処理については本文を参照．

Web ページは，図 4 (a) に示すように，テキストを葉，葉以外のノードをタグとするような木構造で表現することが可能である．全ての Web ページの集合を Dom_D，全ての木構造の集合を Dom_T と表す．Web ページ $D \in Dom_D$ を受け取り，木構造 $R' \in Dom_T$ を返す処理を $R' = Parse(D)$ と書く．逆に木構造 $R'' \in Dom_T$ を受け取り Web ページ $D' \in Dom_D$ を返す処理を $D' = Unparse(R'')$ と書く．

図 3(4) では，図 4(a) に対して，**トークン化**と**予測機能による注釈**を行い，図 4(b) のように変換する．これらを，それぞれ **Tokenize** と **PredictGloss** という 2 つの，木構造を取り木構造を返す関数で表現する．**Tokenize** は木構造 $R \in Dom_T$ を受け取り，R の葉であるテキストをトークン化した木 $R' \in Dom_T$ を返す．図 4(a) をトークン化したものが，図 4(b) の赤字部分を除いた木構造である．図 4(b) 中でトークン化されたテキストの親タグとなる <**span**> では，クリック時に辞書を引き，語義を受け取る動作（図 3(5),(6) にそれぞれ相当）を実現するプログラムが JavaScript で記述され，埋め込まれる．

PredictGloss は木構造 $R' \in Dom_T$，注釈関数 g，ユーザ識別子 u，判別器の重み $w^{(k)}$ を受け取り，R' の葉に対して，ユーザ u のユーザ未知語 t を $h(u, t, w^{(k)})$ の符号で判断し，t のみに注釈 $g(t)$ をつけて返す．ただし，R' はトークン化されていると仮定する．注釈関数 g は，トークン $t \in T$ を受け取り，t に注釈をつけた文字列 $g(t)$ をつけて返す関数である．

図 3(5), (6) では，**AJAX**（asynchronous JavaScript and XML）を用いてブラウザとシステムが通信を行う[*5]．

(5) D' 中のトークン t が最初にクリックされると，(4) での予測が訂正されたと判断し，単語の既知・ユーザ未知の情報 y をシステムに送出する．(4) で既知と判断されたトークン t がクリックされれば，ユーザ未知（$y = 0$）が送出される．(4) でユーザ未知と判断されたトークン t がクリックされれば，既知（$y = 1$）が送出される．

(6) もし $y = 0$，すなわち，ユーザ未知の場合は，システムは t に対応する注釈 $g(t)$ を返し，$g(t)$ がブラウザで表示される．

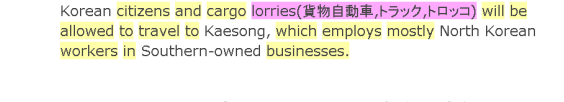

図 2: 本システムでの注釈の例

*5 この通信には，**jQuery** と呼ばれる JavaScript ライブラリを用いている．`http://semooh.jp/jquery/`

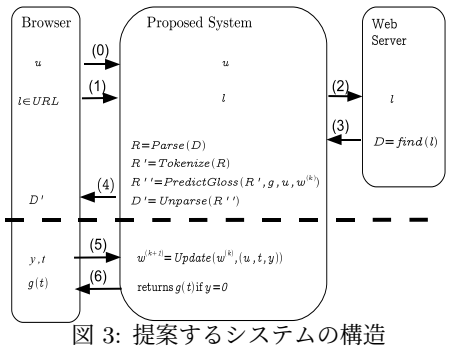

図 3: 提案するシステムの構造

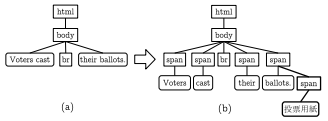

図 4: (a) Web ページの木構造, (b) ブラウザに返却される木構造．太線部分が付与される注釈の例．

(u, t, y) のデータの組は，判別器の重みベクトル $\boldsymbol{w}^{(k)}$ を更新するのに使用される．$Update(\boldsymbol{w}^{(k)}, (u, t, y))$ の詳細については，§3. で述べる．本システムは，Web アプリケーションに特化したクラウド計算機環境である Google App Engine (GAE)[*6]上で動作するように実装した．

3. 予測手法

項目反応理論 [e][*7]（item response theory, IRT）は，人間の能力を測定するテストの設計に使用される確率的なモデルの総称である．TOEFL (Test of English as a Foreign Language) をはじめとする既存の言語テストの設計にも使用されているため，本研究でも項目反応理論を用いることが妥当であると考えられる．

項目反応理論は，テスト結果を入力として受け取る．テストは，$|T|$ 個の項目（設問）から構成されるとし，項目の集合を T と書く．被験者の集合を U とし，被験者数を $|U|$ と書く．被験者 $u \in U$ の項目 $t \in T$ に対する反応を $y \in Y$ とすると，(u, t, y) の組が 1 件のテスト結果となる．ただし，Y は，反応の種類の集合である．以上より，テスト結果は，その件数 N 件とすると $\{(u_n, t_n, y_n) | n \in \{1, \ldots, N\}\}$ と表すことが可能である．これが項目反応理論への入力となる．本研究では，単語クリックログをテスト結果 $\{(u_n, t_n, y_n) | n \in \{1, \ldots, N\}\}$ とみなす．ユーザ $u_n \in U$ の，文書中の個々の単語 $t_n \in T$ に対する反応を $y_n \in Y$ とみなす．Y は，本研究では，$Y = \{0, 1\}$ であるような二値変数とする．$y_n = 1$ のとき，ユーザ u_n は単語 t_n を知っている（既知）とし，$y_n = 0$ のとき，ユーザ u_n は単語 t_n を知らない（ユーザ未知）とする．

本研究では，項目反応理論のうち最も単純な Rasch モデルを，次のように改良して用いた．Rasch モデルでは，$P(y_n = 1 | u_n, t_n) = \sigma(\theta_{u_n} - d_{t_n}) p(\theta_{u_n}, d_{t_n})$ を最尤推定する．ただし，$\sigma(x) = \frac{1}{1+exp(-x)}$ はロジスティックシグモイド関数であり，$p(\theta_{u_n}, d_{t_n})$ は事前分布である．また，θ_{u_n} は被験者 u_n の能力パラメータで θ_{u_n} が高いほど，被験者 u_n の正答率が増加する．また，d_{t_n} は項目 t_n の難易度パラメータで d_{t_n} が高いほど，被験者 u_n の項目 t_n に対する正答率が低下する．

*6 `http://code.google.com/intl/ja/appengine/`
*7 日本語ではその他，**項目応答理論**，**テスト理論**などとも呼ばれる．

図 3 における $PredictGloss$ の内部で使用される判別関数 h は，$h(u_n, t_n, w^{(k)}) = \log P(y_n = 1|u_n, t_n; w^{(k)}) - \log P(y_n = 0|u_n, t_n; w^{(k)})$ と定義され，$h(u_n, t_n, w^{(k)}) \geq 0$ のとき既知，$h(u_n, t_n, w^{(k)}) < 0$ のときユーザ未知と判定される．ここで，予測精度を向上させるために，単語の難しさに関する素性（特徴量）を以下のように導入した．$\boldsymbol{e}_u$ を u 番目の要素のみ 1 で他は 0 のサイズ $|U|$ のユニットベクトル，$\boldsymbol{e}_t$ を t 番目の要素のみ 1 で他は 0 のサイズ $|T|$ のユニットベクトルとする．すると，尤度を，重みベクトル $\boldsymbol{w}_{rasch} = (\boldsymbol{\theta} \ \ \boldsymbol{d})^{\mathrm{T}}$ と素性ベクトル $\boldsymbol{\phi}_{rasch}(u,t) = (\boldsymbol{e}_u \ \ \boldsymbol{e}_t)^{\mathrm{T}}$ を用いて，数式 (1) と表すことができる．ただし，$\boldsymbol{\theta} = (\theta_1, \ldots, \theta_u, \ldots, \theta_{|U|})$，$\mathbf{d} = (-d_1, \ldots, -d_t, \ldots, -d_{|T|})$ である．数式 (1) は，ロジスティック回帰の定義にそのまま従うことから，Rasch モデルはロジスティック回帰の特殊な場合であることがわかる．

$$\begin{aligned} P(y_n = 1|u_n, t_n) &= \sigma\left(\theta_{u_n} - d_{t_n} p(\theta_{u_n}, d_{t_n})\right) \\ &= \sigma(\boldsymbol{w}_{rasch}^{\mathrm{T}} \boldsymbol{\phi}_{rasch}(u_n, t_n)) p(\boldsymbol{w}) \quad (1) \end{aligned}$$

ここで，事前分布は尤度関数の重みパラメータの分布であるので，簡単のために単に $p(\boldsymbol{w})$ と書いた．$\boldsymbol{w}$ を m 次元のベクトルとすると，m 次元の多次元ガウス分布 $\mathcal{N}(\mathbf{0}, \frac{1}{C}\boldsymbol{I})$ がよく用いられるので，本研究でもこれを用いる．

Rasch モデルがロジスティック回帰の中でどのような特徴を持つかを考える．Rasch モデルの本質的な特徴は，ユーザ u の能力パラメータ θ_u が $\theta_u = \boldsymbol{\theta}^{\mathrm{T}} \boldsymbol{e}_u$ と t なしで，項目 t の難易度パラメータ d_t が $d_t = \boldsymbol{d}^{\mathrm{T}} \boldsymbol{e}_t$ と u なしで書き表せ，能力パラメータや項目の難易度パラメータの計算に u と t の両方に依存する素性を用いない点にある．多くの文献では素性ベクトルをユニットベクトルに限定して Rasch モデルが解説されているが，その理由は，Rasch モデルを適用しようとする問題をユーザや項目についての素性を取得できない問題に限定して解説しているからであり，素性ベクトルがユニットベクトルとなるのは，Rasch モデルの本質的な特徴ではない．

従って，ユーザや項目についての素性を取得できる場合は，ユーザと項目の両方に依存する素性ベクトルを用いない限り，これらの素性を素性ベクトルに組み入れることで Rasch モデルを拡張してもよく，推定される重みは，やはり能力パラメータや難易度パラメータとみなすことができる．本研究では，不特定多数のユーザを仮定しているためユーザについては有効な素性が取得できないが，単語については単語頻度などを難易度の素性として用いることができる．素性ベクトルがユニットベクトルだけからなる場合，能力パラメータと難易度パラメータの両方を訓練データ (y_n, u_n, t_n) だけから推定しなくてはならないが，このように単語頻度などの難易度の素性を組み入れると，難易度パラメータの推定が容易になるため，訓練データ (y_n, u_n, t_n) の情報を能力パラメータの推定に集中して使用することができるため，結果として判別精度の向上に貢献すると考えられる．

そこで，本研究では，数式 (1) における Rasch モデルを，ユーザの能力に関する素性ベクトルとそれに対応する重みはそのままに，単語の難易度に関する素性ベクトルとそれに対応する重みだけを次のように置き換えて，拡張することにより判別精度の向上を試みた．重みベクトル $\boldsymbol{w}_{rasch}$ を $\boldsymbol{w}_{ext} = (\boldsymbol{\theta} \ \ \boldsymbol{w}_d)^{\mathrm{T}}$ に，素性ベクトル $\boldsymbol{\phi}_{rasch}$ を $\boldsymbol{\phi}_{ext}(u,t) = (\boldsymbol{e}_u \ \ \boldsymbol{\phi}_t)^{\mathrm{T}}$ に置き換えた．対応する事前分布を $\boldsymbol{w}_{ext}$ とする．

次に，パラメータの推定手法について説明する．パラメータの推定は，図 3 の $Update$ 関数によるパラメータ更新を繰り返して行われる．パラメータを一回更新する際に，データセット全体（この場合は，$\{(u_n, t_n, y_n)|n \in \{1, \ldots, N\}\}$）に対して最適化を行うパラメータ推定手法をバッチ学習法という．一方，パラメータを一回更新する際に，計算を省略し，N とは無関係な定数個（例えば，1 個）のデータだけを見てパラメータを更新する手法をオンライン学習という．Rasch モデルのパラメータを MAP 推定を用いて推定する方法は，バッチ学習法に分類される．

数式 (1) の対数は凸関数であり，これをバッチ学習法で MAP 推定（最尤推定）した場合，最適な $\boldsymbol{w}$ が求まることが知られている．バッチ学習法は，一回の更新で N 個のデータを見るので，少なくとも $O(N)$ 以上の計算量を必要とするため，データ数 N が増加した際には実用的な時間で学習を行うことができなくなる．

しかし，本研究では，ユーザ 1 人あたりクリックできる単語数は，事実上有限である．例えば，ユーザ 1 人が何百万語もクリックすることは考えにくい．また，同じ単語を何度もクリックした場合，最新の結果だけをデータとみなすという考え方もできる．そこで，個々のユーザの予測を，そのユーザのクリックログのみを用いて行うことにすれば，N が増加する場合は考える必要はなく，N を定数として考えることができる．ロジスティック回帰のバッチ学習法については，近年，高速な学習法が提案されており [c]，ユーザ 1 人のクリックログのみを学習データとして用いれば，実用的な時間内に学習を終えることが可能であると考えられる．

4. 実験と考察

本実験では，あるユーザに対する予測を，そのユーザの単語のクリックログのみから行った場合の予測精度を測定した．具体的には，あるユーザについて，異なり語で N_1 個の単語のクリックログが得られたと想定し，そのユーザのテストセットに含まれる単語のうち異なり語で何% について既知/ユーザ未知を当てられたかを 1 人の予測精度の値とした．

精度評価のために，1 人，後述する **SVL12000** という語彙集のうち 11,999 語について，5 段階[*8]の自己申告形式で回答させる方法で，被験者（東京大学を中心とする大学生，大学院生 16 人）の語彙力を測定した．このうち，意味を確実に知っている場合のみを既知の場合とし，残りをユーザ未知の場合とした．

この 11,999 語のうち，後述の **Google 1-gram** の単語頻度の素性が取れる 11,271 語を，271 語のデベロップメントセット，10,400 語のテストセットと，残りの 600 語の訓練データ候補セットに重なりがないようランダムに分割した．この 600 語の訓練データ候補セットのうち，$N_1 \in \{10, 30, 100, 300, 600\}$ 語の単語を訓練データセットとして用い，N_1 に対する判別精度を測定した．§3. で述べた事前分布の C パラメータは，{0.0315, 0.125, 0.5, 1.0, 2.0, 8.0, 32.0, 128.0, 512.0, 2048.0, 8192.0, 32768.0} から，デベロップメントセットに対して最適な値を選択した．

置き換える素性を変化させることにより，3 種類の実験設定を用意し，それぞれ **Rasch**，**EXT** と **IRT+EXT** と名付けた．

IRT §3. で述べた Rasch モデルをそのまま使用し，§3. における単語のユニットベクトル $\boldsymbol{e}_t$ をそのまま素性ベクトルとしている．

*8 単語を知らない度合いの大きい順に，見たこともない，見たことがある気がする，確実に見たことはあるが意味は知らない／覚えたことがあるが意味を忘れている，意味を知っている気がする／意味が推測できる，意味を確実に知っている．

EXT §3. で述べた方法により，Rasch モデルを次のように拡張した．単語に関する素性からユニットベクトルを除き，代わりに **Google 1-gram** と **SVL12000** という 2 種類の素性を組み入れた．**Google 1-gram** は，約 1 兆ページの Web ページ中の英単語の頻度であり，人手を介していない [a]．この頻度を全頻度で正規化して確率値に変換したものを p とすると，$-logp$ を素性に組み入れた．**SVL12000** は，基本的な語彙 12,000 語に対し，英語母語話者を含むチームが人手で 12 段階の難易度をつけた語彙リストである [d]．

IRT+EXT **IRT** の素性ベクトルと **EXT** の素性ベクトルは disjoint（互いに素）であることに注目し，**IRT** の素性ベクトルと **EXT** の素性ベクトルを合わせたものである．

表 1: 素性ベクトルの置き換えによる予測精度の変化 (%)．

	$N_1 = 10$	30	100	300	600
IRT	65.84	66.23	66.38	67.23	66.58
EXT	**74.82**	**79.06**	**79.48**	**79.50**	**79.76**
IRT+EXT	74.63	77.66	79.27	79.23	79.52

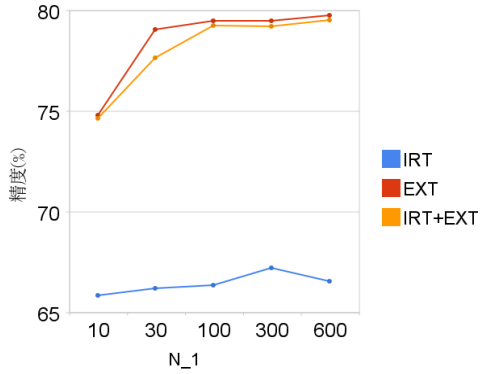

図 5: 素性ベクトルの置き換えによる予測精度の変化．

表 4.，図 5 に実験の結果を示す．まず，ベースラインとなる **IRT** は N_1 に対してほぼ平坦となっており，学習の効果が現れていないことがわかる．この理由は，単語の難易度を表した素性を全く追加していない **IRT** では，テストセットの単語が訓練データセットに現れていない場合，判別器はその単語の難易度を概算することすらできないためである．のように，あるユーザの部分的な語彙情報である単語のクリックログを訓練データとし，そのユーザの残りの語彙情報を予測する場合は，テストセットの単語は訓練データセットに現れることはないので，**IRT** は全く使えない事になる．

一方，単語の難易度を表す素性に置き換えた **EXT** では，この素性に対する重みが学習され，テストセットの単語が訓練データセットに現れていない場合であっても，その単語の素性の値を通じて単語の難易度を概算し，予測することが可能である．**EXT+IRT** は，**EXT** と比較して，若干精度が落ちている．この理由は，前述のように，**IRT** のユニットベクトルの素性が今回の実験設定では意味をなさないにもかかわらず，これを入れたために，この素性に対応する重みの分だけ，有用な **EXT** が軽く重み付けられたためであると考えられる．

学習に要した時間は，精度が最も良かった **EXT** で，訓練データが 600 語の場合，全ユーザ 16 人の全 C パラメータ 12 通り，192 通りの平均が 0.01331 秒，最も時間がかかった訓練でも 0.02281 秒であることから，1 ユーザの訓練には十分実用的な時間であると考えられる．ただし，実験に用いたのは CPUXeon 5160 3.0GHz 2core x 2 の環境であり，今回用いた訓練データはメモリに乗るサイズであった．実験には [c] の実装である [b] を用いた．

5. 結論と今後の課題

筆者らは，既存の語義注釈システムを拡張し，任意の英文 Web ページに対して，ユーザが知らない英単語をそのユーザのクリックログの解析をもとに予測することで，Web ページの読み込み時にそれらの語に対して予め訳を付与する機能を加えることを提案した．

予測に，TOEFL などの言語テストに使用されている項目反応理論の基本的な場合である Rasch モデルを使用することで，判別に利用される重みベクトルの値を英語力や単語の難易度として解釈できる性質を判別器が持てるようにした．さらに，Rasch モデルにおいて素性ベクトルを置き換えても，ユーザと単語の両方に依存する素性を入れなければ，この性質は保たれることを示した．また，ユーザ一人あたりのクリックログに含まれるデータ数は事実上限られているので，そのユーザの残りの単語の既知/ユーザ未知を予測することにし，近年提案された高速な学習手法である [c] を用いれば，バッチ学習でも実用的な時間内で学習を終了させることができることを述べた．

実験の結果，素性ベクトルをより有用な素性に置き換えることで，予測精度が向上することが示された．また，1 ユーザの 1 訓練あたり 0.01331 秒という実用的な時間で訓練が行えることが示された．

以下に，今後の課題を述べる．本稿では，あるユーザ一人のクリックログのみを訓練に用いてバッチ学習し，そのユーザの残りの単語の既知/ユーザ未知を予測したが，多くのユーザのクリックログを組み合わせて学習に用いることが考えられる．この場合，訓練データはユーザ一人のクリックログではなく，全クリックログになるので，もはや N は定数ではなく，オンライン学習を使用するなどの工夫が必要であると考えられる．また，本稿のようにあるユーザ一人のクリックログのみを学習に用いた場合でも，1 ユーザの 1 訓練あたりは実用的な時間であっても，同時に訓練しなければならないユーザ数が増えた場合の対応は今後の課題となる．また，素性に関する課題としては，ユーザの得意分野，単語が属する分野，文書の分野などの，分野（トピック）を素性に入れて考慮した判別を行うことが考えられる．

参考文献

[a] Web 1T 5-gram Version 1, Linguistic Data Consortium, Philadelphia (2006)

[b] LIBLINEAR: A library for large linear classification, year, year, year, year, The Journal of Machine Learning Researchpp. 1871–1874 (2008)

[c] Trust region Newton method for logistic regression, year, year, The Journal of Machine Learning Researchpp. 627–650 (2008)

[d] Standard Vocabulary List 12,000 (1998), Data available at `http://www.alc.co.jp/goi/PW_top_all.htm`

[e] 項目反応理論 [理論編]-テストの数理-, 朝倉書店 (2005)