英語版 Wikipedia を対象とした ユーザが知らない語を予測する読解支援システム

江原 遥
東京大学情報理工学系研究科
{ehara,ninomi,shimizu}@r.dl.itc.u-tokyo.ac.jp

二宮 崇，清水 伸幸，中川 裕志
東京大学情報基盤センター
nakagawa@dl.itc.u-tokyo.ac.jp

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．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．この問題に対応するため，“語義注釈システム”が提案されてきた．語義注釈システムを用いると，ユーザは，知らない語（ユーザ未知語）に遭遇した場合，クリックまたはマウスオーバーなどの語の選択操作により，語義を表示させ，語の意味を知ることができる．図 1 に挙げる pop 辞書[1]では，マウスオーバーしたユーザ未知語の語義をポップアップで表示している．また，ドラッグ操作で選択したユーザ未知語の語義を Web ページ中に埋め込むシステムも提案されている[2]．

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，本稿では “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，予め語義を付与して読解を容易にすることが可能となると考えられる．

本研究では，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を提案する．システムは，`http://socialdict.appspot.com/` にて運用されている．また，英語版 Wikipedia に対しては，最初の *en.* を *enn* に置き換えることでも利用出来るように設計した[3]．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．

ユーザの語彙を予測する際には，単に予測するのではなく，既存の被験者の語彙力を測定する手法と何らかの互換性があるモデルを用いることが望ましい．また，多数のユーザからリアルタイムに収集される単語クリックログは，逐次的に増加するので，予測には逐次アルゴリズムを用いることが望ましい．

[1] `http://www.popjisyo.com/`
[2] `http://www.popin.cc/`
[3] 例：`http://en.wikipedia.org/wiki/Main_page` に対して，`http://ennwikipedia.org/wiki/Main_page`

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 に渡す．既存のシステムでは，ユーザ適応をしないため，この動作は不要であった．

(1) ユーザは，ブラウザ (Browser) を通じて $l \in URL$ を本システム (Proposed System) に渡す．URL は URL の集合とする．

(2) 本システム 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（$D = find(l)$），本システムに返す．

図 1: pop 辞書での注釈の例．

(4) 本システム 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 “” 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[4]．

(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．

The easing of border restrictions, to begin Friday, means more South Korean citizens and cargo lorries(貨物自動車,トラック,トロッコ) will be allowed to travel to Kaesong, which employs mostly North Korean workers in Southern-owned businesses.

図 2: 本システムでの注釈の例

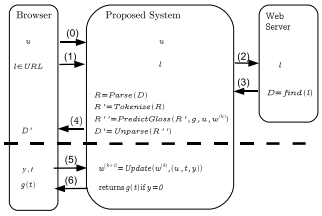

図 3: 提案するシステムの構造

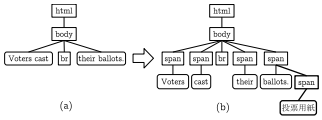

図 4: (a) Web ページの木構造, (b) ブラウザに返却される木構造．太線部分が付与される注釈の例．

(6) もし $y = 0$，すなわち，ユーザ未知の場合は，本システム 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．

(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)[5]上で動作するように実装した．

3 予測手法

項目反応理論 [6][6]（item response theory, IRT）は，テストの設計に使用される確率的なモデルの総称であり，特に人間の能力を測定するためによく用いられている．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

[4] 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/`

[5] `http://code.google.com/intl/ja/appengine/`

[6] 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．

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．本システム の応用では，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，本システム の応用では，$Y = \{0, 1\}$ 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，$P(y_n = 1|u_n, t_n) = \sigma(\theta_{u_n} - d_{t_n})$ を最尤推定する．ただし，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．図 3 における $PredictGloss$ の内部で使用される判別関数 h は，$h(u_n, t_n, w^{(k)}) = \log P(y_n = 1|u_n, t_n) - \log P(y_n = 0|u_n, t_n)$ 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．また，σ はロジスティックシグモイド関数である．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 $\boldsymbol{w}_{rasch} = (\boldsymbol{\theta} \ \ \boldsymbol{d})^{\mathrm{T}}$ と特徴量ベクトル $\boldsymbol{\phi}_{rasch}(u, t) = (\boldsymbol{e}_u \ \ \boldsymbol{e}_t)^{\mathrm{T}}$ 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．

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

数式 (1) において，重みベクトル $\boldsymbol{w}_{rasch}$ を $\boldsymbol{w}_{LR} = (\boldsymbol{\theta} \ \ \boldsymbol{d} \ \ \boldsymbol{w}_a)^{\mathrm{T}}$ に，特徴量ベクトル $\boldsymbol{\phi}_{rasch}$ を $\boldsymbol{\phi}_{LR}(u, t) = (\boldsymbol{e}_u \ \ \boldsymbol{e}_t \ \ \boldsymbol{\phi}_a)^{\mathrm{T}}$ に置き換えることにより，$\boldsymbol{\phi}_a$ に素性を追加することが可能である．追加した素性は，Google 1-gram と SVL12000 である．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る [2]．SVL12000 は，基本的な語彙 12,000 語に対し，人手で 12 段階の難易度をつけた語彙リストである [5].

次に，パラメータの推定手法の改良について説明する．パラメータ更新の際に，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．Rasch モデルのパラメータを最尤推定を用いて推定すると，バッチ学習法となる．$Update$ 関数にバッチ学習法を用いると，ユーザがクリックするたびにデータセット全体を参照し最適化を行う必要が生じるので，§1 で述べたスケーラビリティと即応性が低下する．この問題を解決するために，パラメータを逐次的に推定する逐次学習法が提案されている．本システムでは，逐次学習法の 1 つである Stochastic Gradient Descent (SGD)[1] を用いた．SGD では，最尤推定における n に関する和を省略して，$Update$ 関数を次のように定める．

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \eta_k \nabla E_n(\mathbf{w}^{(k)}) \quad (2)$$

数式 (2) は，$\eta_k = \frac{1}{\lambda(k+k_0)}$ であるときに収束する．ただし，λ, k_0 は非負の定数である．SGD は，逐次学習法であり，逐次的にデータが蓄積される本システムに適すると考えられるため，本研究では，SGD を採用した．

4 実験

実験では，テストセットに対する既知/ユーザ未知の予測精度を測定した．辞書引きログが N_0 個蓄積されたところに，ユーザが 1 人新規にシステムを使い始め，N_1 個の単語の既知/ユーザ未知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何% について既知/ユーザ未知を当てられたかを 1 人の精度とした．

精度評価のために，1 人 12,000 語について，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，の 5 段階の自己申告形式で回答させる方法で，被験者（東京大学修士大学院生 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．12,000 語のうち 11,999 語を $N_1 = 600$ 語までの訓練データセット，1400 語のデベロップメントセット，9999 語のテストセットに分けた．

単語クリックログの代わりに，単語クリックログと同じデータ構造を持つ **smart.fm** [7]というシステムのログで代用した．**smart.fm** は，Web 上で単語を学習するためのシステムであり，学習済みの単語を学習項目から除外するために既知/ユーザ未知をユーザに問う．このデータが，辞書引きログと同じデータ構造を持つ．10,526 人分のデータを **smart.fm** から取得し，継続的にシステム

[7] http://smart.fm/

を利用していると思われる 675 人分のデータを実際に用いた．

§3 において，Rasch モデルを拡張した効果を実験を通じて評価した．まず，素性追加の効果について実験を行った．その結果，“Rasch” が素性を追加していない場合の精度（Accuracy），“LR” が素性を追加した場合の精度である．約 5%精度が向上した．

次に，もう一つの改良である逐次化の効果を測定するために，他手法と比較する実験を行った．表 4 中の “LR” と “SGD” は，それぞれ，今回使用している Rasch モデルのバッチ学習法 [4]，逐次学習法 (SGD) を表す．また，Rasch モデル以外の他の二値分類の手法との比較も行った．表 4 中の “SVM (Linear)” は線形カーネルの SVM (Support Vector Machine)[3]，“SVM (RBF)” は RBF カーネルの SVM である．まず，“LR” が $N_1 = 300, 600$ の時に，SVM よりも良い精度を達成していることが分かる．この事実は，本システムに Rasch モデルを使用することが，言語テストで実際に使用されている点に加え，予測精度の点からも妥当であることを示している．次に，逐次学習法 “SGD” を用いると，バッチ学習法である “LR” に対して 1%〜2% ほどの精度の減少に抑えられることが分かった．

表 1: 各手法の予測精度 (%).

	$N_1 = 5$	30	100	300	600
SVM (Linear)	**74.78**	**78.08**	78.88	79.20	79.27
SVM (RBF)	67.61	77.27	**79.16**	79.55	79.91
SGD	73,84	73.19	78.50	77.93	78.80
LR	73.25	77.89	79.09	**80.03**	**80.01**

5 結論と今後の課題

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）が読解の障害となる．この問題に対処するため，ユーザがクリックなどの操作によりユーザ未知語の語義を表示することのできる語義注釈システムが提案されてきた．語義注釈システムにおいては，クリックされた単語のログをとることにより，単語クリックログが蓄積される．既存の語義注釈システムでは，単語クリックログが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，その語に語義の注釈を付与する予測機能を持った語義注釈システムを提案した．

予測には，TOEFL などの言語テストに使用されていることから，項目反応理論の一種である Rasch モデルを使用した．まず，Rasch モデルを改良し，素性を追加して予測精度を向上させた．次に，高いスケーラビリティと即応性を実現するために，Rasch モデルに対してバッチ学習法ではなく，逐次学習法である SGD を用いた．

実験のために，16 人に 1 人 12,000 語について単語を知っている度合いを尋ねた評価用データを作成した．前述の 2 通りの改良について，それぞれ実験を行った．まず，素性追加により判別精度が約 5%向上した．次に，逐次学習法である SGD を使用することによる判別精度の減少は非逐次のバッチ学習法 (LR) に比べて 1〜2% 程度であることが分かった．また，LR は，SVM などの他の 2 値分類手法と比較してもほぼ同程度の精度を達成することも分かった．

今回，英語版 Wikipedia に対しては，最初の *en.* を *enn* に置き換えることでも利用出来るように設計した．今後の課題としては，英語版 Wikipedia の各項目を特徴量に加えた場合の予測精度を測定することなどが挙げられる．

参考文献

[1] C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[2] T. Brants and A. Franz. *Web 1T 5-gram Version 1*. Linguistic Data Consortium, Philadelphia, 2006.

[3] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[4] R.E. Fan, K.W. Chang, C.J. Hsieh, X.R. Wang, and C.J. Lin. LIBLINEAR: A library for large linear classification. *The Journal of Machine Learning Research*, Vol. 9, pp. 1871–1874, 2008.

[5] SPACE ALC Inc. Standard vocabulary list 12,000, 1998. Data available at `http://www.alc.co.jp/goi/PW_top_all.htm`.

[6] 豊田秀樹. 項目反応理論 [理論編]-テストの数理-. 朝倉書店, 2005.