

ユーザが知らない語を予測する読解支援システム SocialDict とそのリーダビリティ測定への拡張

江原 遥^{†1} 二宮 崇^{†2}
清水 伸幸^{†2} 中川 裕志^{†2}

英文 Web ページ中のユーザが知らない語を単語クリックログから予測する読解支援システム SocialDict を拡張し、リーダビリティも測定可能にする手法を提案する。

A reading support tool with prediction capability and its extension to readability measurement

YO EHARA,^{†1} TAKASHI NINOMIYA,^{†2}
NOBUYUKI SHIMIZU^{†2} and HIROSHI NAKAGAWA^{†2}

We have been proposing a system to support reading English Web pages for ESL (English as a second language) users by annotate the words they may not know by personalized prediction of those words using the word click logs. In this paper, we propose its extension to readability measurement. As this paper is not refereed, the content of this paper is not counted as a conference paper.

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（非既知語）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 ページ中にある非既

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、ページ中の非既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、本稿では、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} 選択した非既知語の語義を 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、ユーザの非既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、読解の障害となる非既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、読解の障害となる非既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非既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 SocialDict^{*3}を提案している⁴⁾。本システムは、非既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非既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^{*4}。



図 1 pop 辞書での注釈の例。

^{†1} 東京大学大学院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

^{*1} <http://www.popjisyo.com/>

^{*2} <http://www.popin.cc/>

^{*3} <http://www.socialdict.com/>

^{*4} 本稿では説明のために既知と判断された語も着色しているが、筆者らの過去の発表の質疑応答において、この場合は着色しない方がよいという意見もあった。

SocialDict 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 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。^{*1}

本稿では、この SocialDict が用いているユーザが分からない単語を予測する機能を利用して、文書のリーダビリティを測定する手法を提案する。本研究をリーダビリティ測定に応用するためには、個々の単語の既知/非既知の予測と、文書のリーダビリティを結びつける必要がある。この結びつけは、ESL (English as a Second Language) におけるリーディング (読解) の研究において著名な I.S.P. Nation によって、サーベイ論文⁸⁾ で論じられている。

⁸⁾ において、I.S.P. Nation は、text coverage という量を定義し、文書を十分な理解するためには、98% の text coverage が必要であるとしている。text coverage は、文書の延べ語数に対する知っている単語の延べ語数の比率であり、自然言語処理の用語を用いれば、その文書における、知っている単語の 1-gram 確率の総和と表現できる。

⁸⁾ では、この 98% という数値は、I.S.P. Nation のグループが⁷⁾ において、non-native speakers (NNS) に対してテキスト中の出現頻度の低い語を意味のない語に置き換えることによって確実にその語が分からなくなるようにして読ませた後、多肢選択式の理解度テストを行い、text coverage と理解度テストの結果を回帰した結果得られたと報告している。具体的には、text coverage が 80% では十分な理解度に達した者は一人もおらず、90% では小数であるが十分な理解度に達した者がおり、95% では 90% より十分な理解度に達した者が増えたもののやはり全体から見れば小数であったのに対し、100% では殆どが十分な理解度に達した、としている。

また、興味深いことに⁸⁾ では、この text coverage 98% という数値が⁷⁾ とは独立に、native speakers (NS) に対して行われた³⁾ と符合していることを報告している。

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 SocialDict での注釈の例

*1 例: <http://www.socialdict.com/http://www.cnn.com/>

以上より、本稿では、テキスト中の単語のうち 98% が非既知であれば、そのテキストを読むことができないと仮定する。

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。

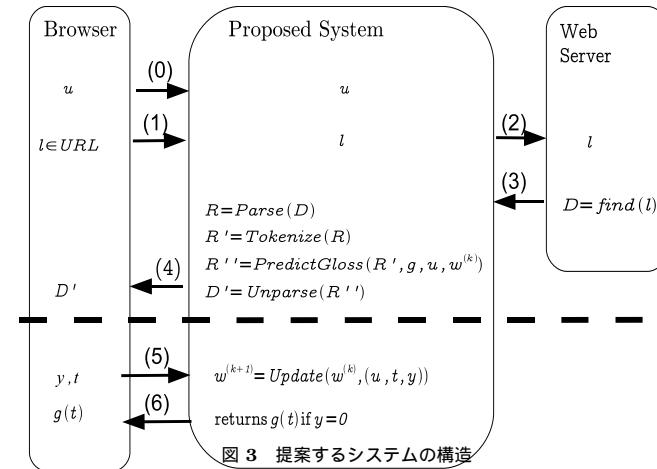


図 3 提案するシステムの構造

- (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。
- (1) ユーザは、ブラウザを通じて $l \in URL$ を SocialDict 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決めてしまい URL の入力を不要とした方が、ユーザにと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。

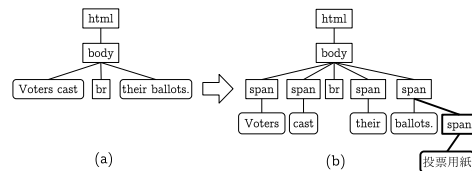


図 4 (a) Web ページの木構造, (b) ブラウザに返却される木構造。太線部分が付与される注釈の例。

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^{*1}。

(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。

(u, t, y) のデータの組は、判別器の重みベクトル $w^{(k)}$ を更新するのに使用される。 $Update(w^{(k)}, (u, t, y))$ の詳細については、§3 で述べる。SocialDict 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)^{*2}上で動作するように実装した。

3. 予測手法

項目反応理論^{10)*3} (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 $\{(u_n, t_n, y_n) | n \in \{1, \dots, N\}\}$ 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 $\{(u_n, t_n, y_n) | n \in \{1, \dots, N\}\}$ 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 (非既知) 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

*1 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://semoooh.jp/jquery/

*2 http://code.google.com/intl/ja/appengine/

*3 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。

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。予測に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、 $h(u_n, t_n, w^{(k)}) < 0$ のとき非既知と判定される。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。 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、 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 $w_{rasch} = (\theta \ d)^T$ と素性ベクトル $\phi_{rasch}(u, t) = (e_u \ e_t)^T$ を用いて、数式 (1) と表すことができる。ただし、 $\theta = (\theta_1, \dots, \theta_u, \dots, \theta_{|U|})$ 、 $d = (-d_1, \dots, -d_t, \dots, -d_{|T|})$ 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。

$$\begin{aligned} P(y_n = 1 | u_n, t_n) &= \sigma(\theta_{u_n} - d_{t_n} p(\theta_{u_n}, d_{t_n})) \\ &= \sigma(w_{rasch}^T \phi_{rasch}(u_n, t_n)) p(w) \end{aligned} \quad (1)$$

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 $p(w)$ と書いた。 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 $\mathcal{N}(0, \frac{1}{\sigma} I)$ 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 $\theta_u = \theta^T e_u$ と t なしで、項目 t の難易度パラメータ d_t が $d_t = d^T e_t$ 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 w_{rasch} を $w_{ext} = (\theta \ w_d)^T$ に、素性ベクトル ϕ_{rasch} を $\phi_{ext}(u, t) = (e_u \ \phi_t)^T$ に置き換えた。対応する事前分布を 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、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、 $\{(u_n, t_n, y_n) | n \in \{1, \dots, N\}\}$ ）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 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。

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。つまり、蓄積された単語クリックログのデータ数 N に対して、一回のクリックごとのパラメータ更新に $O(N)$ 以上の計算が必要となる。従って、単語クリックログのデータ数 N が増加すると、いずれ実用的な時間で学習が行えなくなるため、実運用環境ではバッチ学習は適さないことがわかる。この問題を解決するために、精度を犠牲にする代わりにパラメータを逐次的に推定し、一回のクリックごとのパラメータ更新が $O(1)$ で終了する逐次学習法が提案されている。Rasch モデルに対しては、逐次学習法の 1 つとして Stochastic Gradient Descent (SGD)¹⁾ が利用できる。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。

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

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、実運用している SocialDict , NEWikipedia では、共にこれを用いている。

4. 単純精度実験

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/非既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^{*1}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、残りを非既知の場合とした。

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 e_t をそのまま素性ベクトルとしている。

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²⁾。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、 $-\log p$ 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

*1 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。

ある⁹⁾。

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

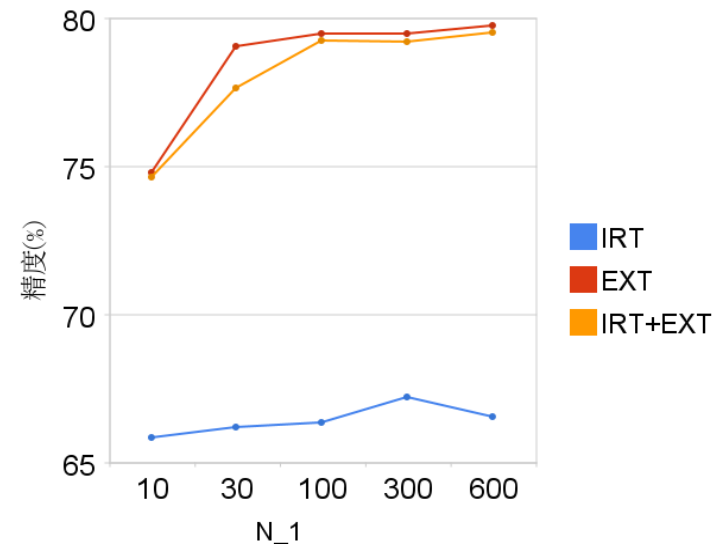


図 5 素性ベクトルの置き換えによる予測精度の変化。

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 CPU Xeon 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⁶⁾ の実装である⁵⁾ を用いた。

5. リーダビリティ実験

前節の結果を踏まえ、最も高精度を記録した EXT の結果を用いて、リーダビリティ実験を行った。リーダビリティ実験では、Brown corpus のリーダビリティを測定した。

Brown corpus はいくつかのカテゴリに分かれて文書がまとめられているが、このうち、44 文書の news、17 文書の reviews、29 文書の fiction の 3 カテゴリについて実験を行った。まず、前節で収集した 16 人についての 11,271 語を用いて、各文書の延べ語数のうち 98% 以上を知っているか否かで、文書が読める/読めないの正解をつけた。“studying” に対して “study” を求めるなど、文書中の語を原型に戻す作業には、nlk の WordNet Lemmatizer を用いた。このとき、文書中に収集した単語以外の単語が現れた場合は、単純に無視した。

次に、前節の予測結果を用いて、読める/読めないを測定した。最高でどの程度正確に予測可能であるかを調べるため、前節の訓練データセットとデベロップメントセットに入っている語については、収集した正解を用いて単語の既知/非既知をつけた。前節のテストセットの単語についてのみ、前節の $N_1 = 600$ 語の場合の予測の結果をそのまま利用して、リーダビリティを測定した。

表 5 に、実験の結果を示す。表の中の値は正解した文書数/全文書数の形式を取ってい

る。正解した文書数とは、読める/読めないが予測と一致した文書数である。ユーザ 10 については、結果が求められなかったので省いた。また、参考までに、各カテゴリにおける Flesch-Kincaid 値 (0 ~ 100) の平均を F-K の欄に書いた。

表 2 リーダビリティ実験結果

user 1	44 / 44	17 / 17	29 / 29
user 2	44 / 44	17 / 17	28 / 29
user 3	26 / 44	12 / 17	13 / 29
user 4	42 / 44	17 / 17	28 / 29
user 5	41 / 44	17 / 17	25 / 29
user 6	34 / 44	14 / 17	19 / 29
user 7	44 / 44	17 / 17	29 / 29
user 8	42 / 44	17 / 17	27 / 29
user 9	4 / 44	7 / 17	10 / 29
user 11	4 / 44	7 / 17	11 / 29
user 12	44 / 44	17 / 17	29 / 29
user 13	44 / 44	17 / 17	29 / 29
user 14	44 / 44	17 / 17	29 / 29
user 15	43 / 44	14 / 17	21 / 29
user 16	44 / 44	17 / 17	29 / 29
F-K	65.35	61.6647058824	82.0689655172

6. 結論と今後の課題

筆者らは、任意の英文 Web ページに対して、クリックログの解析を基にユーザ知らない英単語をユーザの英語力に応じて予測することで、Web ページの読み込み時にそれらの語に対して訳を予め付与する、ユーザ参加型の読解支援システム SocialDict を提案している。

本稿では、この SocialDict が用いているユーザが分からない単語を予測する機能を利用して、文書のリーダビリティを測定する手法を提案した。具体的には⁸⁾ を参考に、98% の text coverage があれば、文書を十分に理解できると仮定した。

§5 において、リーダビリティ実験を行った。このように、単語ごとの予測を用いてリーダビリティを測定した点に本研究の貢献がある。

まず、多くの文書について全正解していることから、提案手法は期待できる結果を示していると解釈できる。一方、ユーザによっては、精度が極端に悪いユーザも存在する。この理由の解明が、今後の課題である。また、fiction については、Flesch-Kincaid 値が他の二つのカテゴリよりも高い (難易度が低い) ためか、他の二つのカテゴリよりも予測精度が悪く

なっている。Fiction と non-fiction で、リーダビリティの基準となる text coverage が異なる可能性を示唆する研究があることが、⁸⁾ で述べられており、この研究との関連も、今後の課題である。

参 考 文 献

- 1) Bishop, C.M.: *Pattern recognition and machine learning*, Springer (2006).
- 2) Brants, T. and Franz, A.: *Web 1T 5-gram Version 1*, Linguistic Data Consortium, Philadelphia (2006).
- 3) Carver, R.: Percentage of unknown vocabulary words in text as a function of the relative difficulty of the text: Implications for instruction., *Journal of Reading Behavior*, Vol.26, No.4, pp.413-437 (1994).
- 4) Ehara, Y., Shimizu, N., Ninomiya, T. and Nakagawa, H.: Personalized reading support for second-language web documents by collective intelligence, *2010 International Conference on Intelligent User Interfaces (IUI 2010)*, pp.51-60 (2010).
- 5) Fan, R., Chang, K., Hsieh, C., Wang, X. and Lin, C.: LIBLINEAR: A library for large linear classification, *The Journal of Machine Learning Research*, Vol.9, pp. 1871-1874 (2008).
- 6) Lin, C., Weng, R. and Keerthi, S.: Trust region Newton method for logistic regression, *The Journal of Machine Learning Research*, Vol.9, pp.627-650 (2008).
- 7) M.Hu, I.N.: Vocabulary density and reading comprehension, *Reading in a Foreign Language*, Vol.13, No.1, pp.403-430 (2000).
- 8) Nation, I.: How large a vocabulary is needed for reading and listening?, *Canadian Modern Language Review*, Vol.63, No.1, pp.59-82 (2006).
- 9) SPACE ALC Inc.: Standard Vocabulary List 12,000 (1998). Data available at http://www.alc.co.jp/goi/PW_top_all.htm.
- 10) 豊田秀樹：項目反応理論 [理論編]-テストの数理-, 朝倉書店 (2005).