Personalized Reading Support for Second-Language Web Documents by Collective Intelligence

IUI 2010, Feb. 2010, Hong Kong, China

The University of Tokyo
Yo Ehara,
Nobuyuki Shimizu, Takashi Ninomiya, and Hiroshi Nakagawa

http://en.newikipedia.org/
Second-language, reading support

Second-language (L2):

any language learned after the first language (cf. Wikipedia)
e.g. English for the speaker (me)

Among ways of *reading support*, we focus on:

- **Glossing Web documents**
  = to annotate words in a document with their meaning.

Web document:

Han Chinese groups paramilitary riots
Los Angeles Times - David

A group of Han Chinese called paramilitaries deploy by the thousands in a bid to
Existing Glossing Systems

Show *glosses* (meaning) when words are clicked.

*pop-jisyo* (Coolest.com, 2001) shows glosses in pop-ups

*popIn* (Cheng, 2008) embeds glosses within a Web document.

---

**Han Chinese groups:** paramilitary

*Los Angeles Times* - *David Liu*

A group of Han Chinese called paramilitaries in China. State police and paramilitaries deploy by the thousands in a bid to...
How they work?

By mediating between **browsers** and **Web servers**

Though they tell which words the users don’t know, click logs have been discarded.
Use click logs to predict "u knows t"? (u: user, t: word) Users implicitly collaborate each other to train predictor
Demo

Access (currently supports only Japanese glosses): 
http://en.newikipedia.org/wiki/item

where item is the item of your interest in Wikipedia.

To avoid cumbersome log-in procedure:
• A browser is regarded as a user.
• Mean English ability is used for the first access.

Note that this system can support any Web pages.

Actually we once created a system that works on most of the English Web pages, http://www.socialdict.com/URL. In some Web pages, however, their JavaScript scripts didn’t co-operate with our JavaScript script and resulted in corrupted display. As this is not our main focus, we prepared a version limiting to Wikipedia and use this version as a demo.
Prediction: personalized

in the sense that prediction differs from user to user

yellow: Predicted to be known to the user

red: Predicted to be unknown to the user

Example from the same document:

low ability

high ability

When the crew asked if there were doctors (博士資格の医師, 博士資格を持つ医師), the captain (船長) said "no". One person was injured, so the plane was turned back to Brussels (ブリュッセル). The captain (船長) and the co-pilot (コパイロット) were prisoners of war (海軍兵). Dr. Julian Struyven, 72, a cardiologist (心臓内科医) and radiologist (放射線科医) from Brussels (ブリュッセル), went to the cockpit and examined (検査) the pilot (パイロット) who was held prisoner in Brussels (ブリュッセル).

"He was not alive," (生きていない) Dr. Struyven told AP. There was "no chance" (全くの希望が無かった) of saving (救う) the pilot (パイロット), he said.

Dr. Struyven said he suspected the pilot (パイロット) had suffered a cardiac arrest (心臓の病気). He said he used a defibrillator (除細動器) to try and revive (蘇生) the pilot (パイロット), but it was too late.

Pilots are subject to rigorous medical checks which increase in frequency with age.
Research Questions

1. To predict, *word difficulty* and *users’ ability* needs to be estimated from the click logs. Can we estimate meaningfully? So that these measures are comparable to those used in *language testing*?
   – Yes. Language testing uses *IRT* model for these measures and we can use it for this task as well.

2. Can the system learn click logs dynamically (every time a user clicks)?
   – Yes. We can use SGD, an on-line algorithm, to train *IRT* model.
I will explain IRT and SGD
Item response theory (IRT)

Probabilistic models used in many testing studies including existing language testing like TOEFL. Testing = estimate difficulty and ability from test results.

Rasch model: simplest version of IRT.

Notations:
User $u \in U$, Words $t \in T$, $y \in \{0,1\}$
$y=1$: $u$ knows $t$, $y=0$: $u$ doesn’t know $t$
Accumulated click logs $(y_n,u_n,t_n)$:
$(y_1, u_1, t_1), (y_2, u_2, t_2), ..., (y_N, u_N, t_N)$
Rasch Model

Input:
\((y_1, u_1, t_1), (y_2, u_2, t_2), \ldots, (y_N, u_N, t_N)\)

Parameters:
\(\theta_u: u's \text{ ability}\)
\(d_t: t's \text{ difficulty}\)

Model:
\[
P(y_n = 1 \mid u_n, t_n) = \sigma(\theta_u - d_t)
\]
\[
\sigma(x) = \frac{1}{1 + \exp(-x)} \quad \text{(sigmoid function)}
\]

Estimation: \(ML\) or MAP (prior on ability and difficulty)
\[
\hat{\theta}, \hat{d} = \arg \max_{\theta, d} \prod_{n=1}^{N} P(y_n \mid u_n, t_n)
\]
Rasch model to Log. Reg.

\[ e_u = (0, \ldots, 1, 0 \ldots, 0) \]
\[ e_t = (0, \ldots, 1, 0 \ldots, 0) \]

\[ w_{rasch} = (\theta \ d)^T, \varphi_{rasch}(u, t) = (e_u \ e_t)^T \]

\[ P(y_n = 1| u_n, t_n) = \sigma(\theta_u - d_t) \]
\[ = \sigma(w_{rasch}^T \varphi_{rasch}(u, t)) \]

inner product form

Logistic regression
a.k.a.:
log-linear model
Maximum entropy model
Answer to 1st Research Question

Can word difficulty & user ability be meaningful? comparable to those used in language testing?

Here they are

**Users’ ability**

\[ w_{rasch} = (\theta \ d)^T \]

\[ \phi_{rasch}(u, t) = (e_u \ e_t)^T \]

**Words’ difficulty**

\[ w_{LR} = (\theta \ d \ w_a)^T \]

\[ \phi_{LR}(u, t) = (e_u \ e_t \ \phi_a)^T \]

By adding extra word features, we can extend IRT with comparability remained
Extra features

IRT:

\[ \mathbf{w}_{rasch} = (\theta \ d)^T \]
\[ \phi_{rasch}(u, t) = (e_u \ e_t)^T \]

LR (extended):

\[ \mathbf{w}_{LR} = (\theta \ d \ \mathbf{w}_a)^T \]
\[ \phi_{LR}(u, t) = (e_u \ e_t \ \phi_a)^T \]

Extra word feature vector \( \phi_a \) includes:
- **Google 1-gram**: word frequencies from a trillion Web documents
- **SVL12000**: manually annotated difficulty measure (1 – 12)
Training IRT to estimate parameters

Training consists of parameter *updates*.

- **Batch learning** [L-BFGS, Nocealdal+, 89], [Trust region Newton method, Lin+, 08]
  Converge to the *global optimum* as for Log. Reg.
  An update involves the *whole* training data.

- **Online learning** [SGD, Stochastic Gradient Descent]
  *Not* converge to the global optimum as for Log. Reg.
  An update involves only the datum *that just has come*. 
Answer to 2\textsuperscript{nd} Research Question

Training consists of parameter \textit{updates}.

- Batch learning \([\text{L-BFGS, Nocealdal+}, 89], \text{[Trust region Newton method, Lin+}, 08]\)
  
  Converge to the \textit{global optimum} as for Log. Reg.
  
  An update involves the \textit{whole} training data.

- \textbf{Online learning} \([\text{SGD, Stochastic Gradient Descent}]\)
  
  \textit{Not} converge to the global optimum as for Log. Reg.
  
  An update involves only the datum \textit{that just has come}.

\textbf{2\textsuperscript{nd} RQ: Can the system learn click logs dynamically?}

\textbf{Answer: Yes if we use SGD.}
Evaluation

Subjects: 16 university/graduate students

# of words answered: 12,000 *per a person*

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>never seen the word before</td>
</tr>
<tr>
<td>2</td>
<td>probably seen the word before</td>
</tr>
<tr>
<td>3</td>
<td>absolutely seen the word before but don’t know its meaning / tried to learn the word before but forgot its meaning</td>
</tr>
<tr>
<td>4</td>
<td>probably know the word’s meaning / able to guess the word’s meaning</td>
</tr>
<tr>
<td>5</td>
<td>absolutely know the word’s meaning</td>
</tr>
</tbody>
</table>

unknown

known
Evaluation Settings

Simulated the case a new user starts using our system from an accumulated log

# of data in accumulated log: \( N_0 \)

# of data in the new user’s log: \( N_1 \)

- Data set:

<table>
<thead>
<tr>
<th>( N_0+N_1 ) words ( (10 \leq N_1 \leq 600) )</th>
<th>Training</th>
</tr>
</thead>
<tbody>
<tr>
<td>1400 words</td>
<td>Development</td>
</tr>
<tr>
<td>9999 words</td>
<td>Test</td>
</tr>
</tbody>
</table>

Accuracy = Ratio of words correctly predicted in Test. Averaged over the 16 subjects

- smart.fm log is used as the accumulated log.
  smart.fm is a system whose log stores millions of \( (y_n, u_n, t_n) \).
Effect by adding extra features

Accuracy (%)

\[ N_1 \] (# of training)
Effect by use of online learning

Accuracy (%)

$N_1$ (# of training)

$2\%$ in 300, 600
Conclusions (Contributions)

• Invented a glossing system with personalized prediction that tells *who* knows *which* word by utilizing click logs having been discarded so far.

• 1\textsuperscript{st} RQ: Among binary classifications (e.g. SVM), use of IRT (Log. reg.) is preferable for this task since its measures (ability & difficulty) are comparable to those used in *language testing*.

• Extended IRT by adding extra word features and marked about 5% higher accuracy

• 2\textsuperscript{nd} RQ: SGD enables on-line learning of IRT and learns click logs dynamically with sacrifice of 2% acc.
Thank you for listening!
Aftermath of this presentation

1.: This work is accepted by ACM Transactions on Intelligent Systems and Technology, Special issue on Tutoring and Coaching System.

In the journal version, I simulated the case the 16 users read the 500 docs in the Brown corpus and showed, by using this system, that:

• the users can read more documents
  – (existing researches showed that, to read a document satisfactorily, a reader should know its 95% of words in occurrence.)

• the number of the users’ clicks decrease compared to the case the users simply click and look up every unfamiliar word.

2.: I collected the logs 3 times larger than mentioned in this study from smart.fm, which is closed to the public now. Modeling and analyzing this logs will be my further research.