

Improving reading comprehension step by step using Online-Boost text readability classification system

Lei La · Nan Wang · Dong-ping Zhou

Received: 28 June 2014 / Accepted: 11 November 2014 / Published online: 19 November 2014
© The Natural Computing Applications Forum 2014

Abstract Online reading exercise becomes the universal tool for a wide variety of second language learning systems. Readability sorting is a key step to display suitable reading materials for the learners. Traditional text readability classification techniques cannot meet the request for online learning perfectly as they do not have real-time classification ability and cannot get the information of learners' language levels. This paper presents a novel framework for online reading exercise which is based on the Online-Boost text readability classification algorithm. We first modified the multinomial Naïve Bayes model to give the reading materials initial readability. We then proposed an Online-Boost algorithm for the text readability update and learners' reading comprehension evaluation according to the learners' answers correct rate of the text. Finally, the system would deliver reading materials with different difficulties to testers with different levels of reading ability in real time. The experimental result reveals that the novel method has ideal ease of use and can significantly improve the performance of second language learners.

Keywords Readability sorting · Text classification · Online learning · Reading comprehension · Boosting · Naïve Bayes

1 Introduction

Reading comprehension is one of the most important part of foreign language learning and is a key criteria for

evaluating learners' language skills. According to the famous *input hypothesis* [1] of Stephen D. Krashen and many other linguists' research, the most efficient way to improve reading comprehension is giving reading materials that have slightly higher level than the reader's reading ability. Reading too many easy texts will become meaningless repetition. On the other hand, second language learners will lose their confidence and interests if the difficulty of texts are too high [2]. Therefore, classifying reading materials by their readability plays a crucial role in foreign language learning.

Readability classification of traditional paper-based reading exercise usually needs manual work of language experts. Obviously, it takes a huge human resource overhead. To overcome the high manual cost drawback, researchers have focus on automatic readability sorting based on computers.

With the development of distributed networks and web service technology, online learning systems and online testing systems such as TOFEL iBT[®] [3] become more and more popular. Internet-based learning and testing systems tend to contain more and more reading materials. Therefore, manual classification becomes increasingly impossible. Furthermore, the online systems should have a big advantage of real time. It will become a great regret if the web-based learning systems could not real-time update texts' readability ranging according to the correct rate of the learner's answers and display reading materials of different readability for users with different reading comprehension.

The most intuitive criteria of readability include word length, number of affix, abstraction level of words, number of polysemy, sentence length [4]. Scholars constitute the formula *Flesch Reading Ease* in [5]. To classify documents according to their readability by machines, the studies in

L. La (✉) · N. Wang · D. Zhou
First Research Institute of Ministry of Public Security, Beijing,
People's Republic of China
e-mail: lalei1984@aliyun.com

[6–8] use variants of Flesch Reading Ease. Such methods are easy to deploy; however, the weights of indicators are experience dependant and the classification results have greater subjectivity.

To enhance the accuracy of classification, Socher et al. [9] introduced statistical model into readability sorting. However, the runtime complexity also rises rapidly along with the performance improvement. Schwarm and Ostendorf [10] applied a support vector machine (SVM) for measuring the perplexity of reading materials. The result is really outstanding if not consider the time and computational consumption.

The work in [11] investigated a lack of training data classification problems and presented a machine learning-based comparator for readability classification when reducing the dependence on training set. However, the large granularity of readability is sometimes disadvantage to finding the suitable difficulty levels for readers.

A measure based on an extension of multinomial Naïve Bayes classification that combines multiple language models to estimate the most likely grade level for a given passage is proposed in [12] which can achieve high precision with a low time and computational overhead.

Unfortunately, above methods can neither real-time update the readability nor respond to the reading comprehension of users.

The ideology of classifier committees in which using committees of classifiers stems from the intuition that a team of experts, by combining their knowledge, was proved to produce better results than a single expert alone [13], especially in AdaBoost algorithm family [14–16]. Moreover, when using language learners as the base classifiers and the reading materials as the training set, Boosting algorithms can get the real-time readability of texts and the reading comprehension of users. The problem is that Boosting-based algorithms are difficult to achieve readability initialization.

In this paper, we investigate the readability classification problem for online foreign language learning. This article contributes the following:

1. A modified method based on the Smoothed Unigram model [12] for better adapt to the online documents readability sorting.
2. A Boosting-based classification algorithm for update texts' readability in real time according to readers' response.
3. A novel weighting mechanism for evaluating users' reading comprehension.
4. A novel online reading system which can automatically make readability classification and select suitable reading materials for learners in different levels adaptively.

In the remaining sections, we start by reviewing the key problems of text classification especially readability classification (Sect. 2). In Sect. 3, the modified Smoothed Unigram model for online documents readability initialization is presented. A Boosting-based classification algorithm is proposed in Sect. 4 for real-time update texts' readability and learners' reading comprehension. Adaptive selection of reading materials according to the language levels of users is achieved in Sect. 5. The overall structure of the novel system is shown in Sect. 6. Finally, Sect. 7 summarizes the paper.

2 Background

2.1 Review of text categorization

Similar to other data classification tasks, text classification also called text categorization (TC) can be defined as the problem of approximation an unknown category assignment function $F: D \times C \rightarrow \{0, 1\}$, where D is the set of all possible documents and C is the set of predefined categories:

$$F(d, c) = \begin{cases} 1, & d \in D \text{ and } d \text{ belong to class } c \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

The approximating function $F: D \times C \rightarrow \{0, 1\}$ is called a classifier. The task is to build a classifier that produces results as close as possible to the true category assignment [17]. The most popular text classification algorithm include Naïve Bayes [18], decision tree [19], SVM, neural networks, Rocchio [20], k -nearest neighbors (k NN) [21] and Boosting-based algorithm such as AdaBoost [22]. Generally speaking, Naïve Bayes has the worst precision and the best efficiency, decision tree and neural networks are middle-level algorithm, k NN, SVM and AdaBoost have the highest classification accuracy [23]. However, the computational consumption of SVM is larger than many other TC methods.

2.2 Readability classification

Text classification is widely used in many different aspects of real world. The most common use of TC is to determine topic type of documents. For example, determine whether a text download from www.time.com belongs to politics, economics or entertainment. Spam filtering, orientation analysis and readability sorting are also major applications of text classification. The schematic of different applications of TC as shown in Fig. 1.

As an important issue in text mining, readability sorting has a wide application in education, publication and search engine. The two major methods to distinguish the difficulty

of the text is readability formula and statistical language model-based method.

Readability formula-based classification has an 80-year history and more than a 100 variants. Currently, the most classic readability formulas are Flesch Reading Ease, Gunning Fog Index [24] and Automated Readability Index [25]. Although the ease of use, formula-based readability sorting gradually become less attractive because the unsatisfactory classification accuracy.

Instead of processing texts as strings, statistical language model-based readability classification can deep into the interior of language [26]. It got considerable development in past decade. Similar to other machine learning-based text classification problems, readability sorting needs a labeled document collection as the training set. The training set should be preprocessed. The preprocessing includes words segmentation, part-of-speech (POS) tagging, exclusion of stop words. Features would be selected or extracted from the pretreated training set. After the computation of

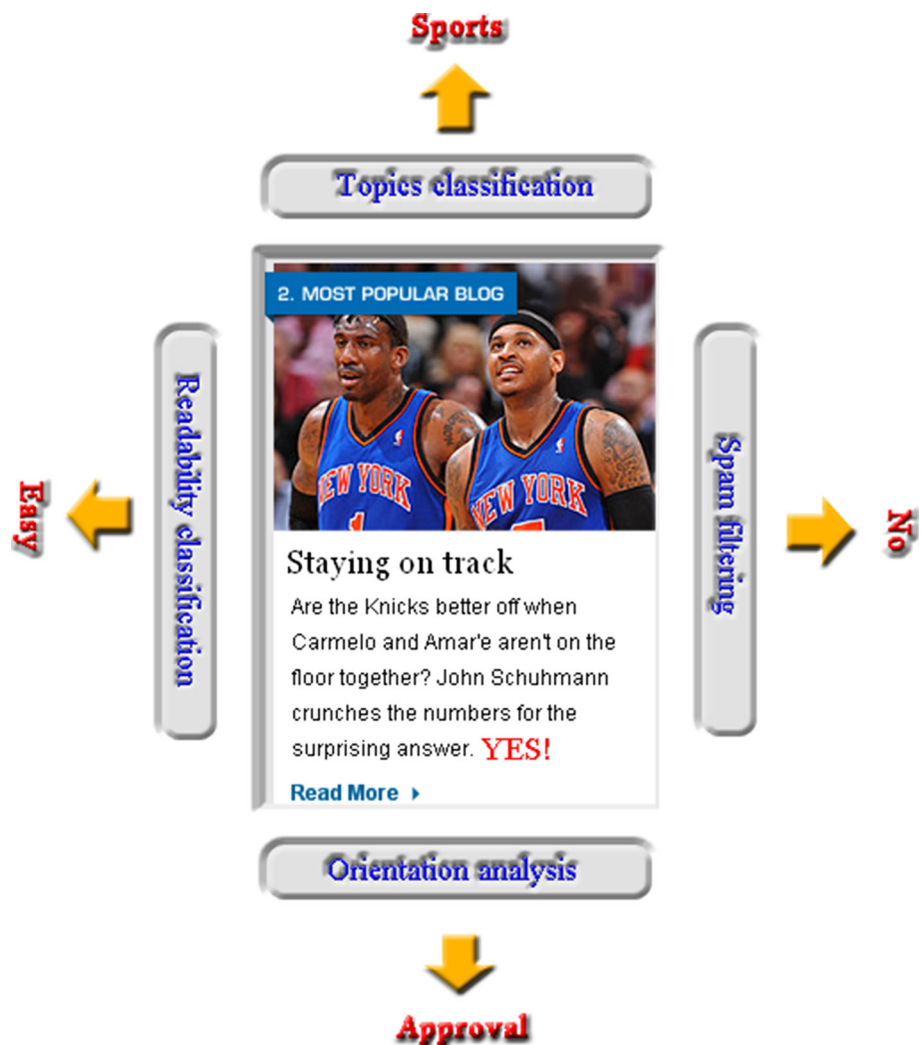
features, readability model is built by referring the manual labels. Then, the testing set will be classified by the built model according to their readability. The flow chart of statistical language model-based readability sorting is shown in Fig. 2.

How to select the most representative features and how to avoid large scales of training set dependence are the two important problems which would deeply affect the performance of readability classification.

3 Initial readability sorting

Although Boosting-based classification algorithm may be very powerful in readability update and learners' reading comprehension evaluation, it cannot be used in the initial readability sorting because it is low efficient to find the suitable features for representing the reading difficulty. Therefore, we need to learn from traditional readability

Fig. 1 Different applications of TC



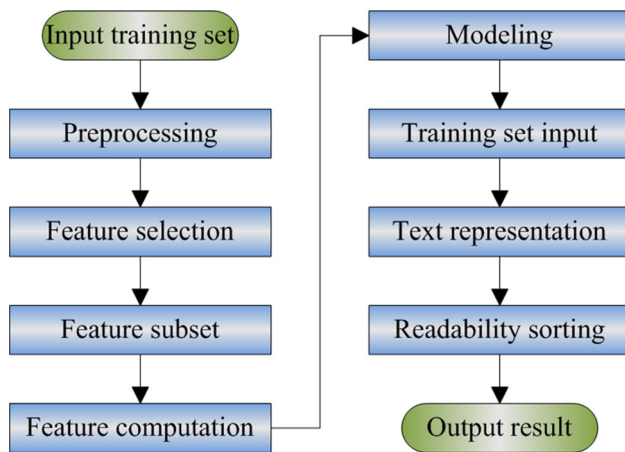


Fig. 2 Flow chart of statistical language model-based readability sorting

classification algorithm to determine the initial readability levels of reading materials.

The language modeling Naïve Bayes-based approach of readability classification [12] algorithm proposed by Kevyn Collins–Thompson and Jamie Callan is an accurate tool for text difficulty prediction. It called Smoothed Unigram model. In addition, Smoothed Unigram model has no domain limitation. They excluded the measurement of sentence length. However, different from native language learners, it is an important indicator about readability for foreign language learners [27]. Therefore, the Smoothed Unigram model should be modified to meet the request for foreign language learning.

3.1 Original Smoothed Unigram model

Smoothed Unigram model is based on a variation of the multinomial Naïve Bayes classifier. In text classification terms, each class is described by a language model corresponding to a predefined level of difficulty. For English online texts, it trained 12 language models corresponding to the 12 English grade levels.

The language models used in [12] are simple: They are based on unigrams and assumed that the probability of a token is independent of the surrounding tokens, given the grade language model. A unigram language model is defined by a list of types (words) and their individual probabilities. Although this is a weak model, it can be trained from less data than more complex models and turns out to give good accuracy for our problem.

In Smoothed Unigram model, a document D is assumed to be generated according to the following steps:

1. Using the prior distribution $P(L_i)$, choose a readability level model L_i from the set of unigram models L . The distribution of L_i in the word space W is multinomial.

2. Using the distribution $P(N|L_i)$, determine the words number N of a passage in tokens.
3. Modeling features of the passage by *bag of words* model, using the Naïve assumption which promised the independence between each token to sampling L_i N times. Build the model L_i according to the N tokens.

Assuming $c(t)$ is the count of type t in D . The probability of D given in model L_i is:

$$P(D|L_i) = P(N|L_i) \cdot N! \prod_{t \in D} \frac{P(t|L_i)^{c(t)}}{c(t)!} \quad (2)$$

Obviously, the runtime complexity of Eq. (2) is huge because of the computation of the high times power and factorial. Therefore, Smoothed Unigram uses Bayes' rule to find the most likely level language model given in the text D . In other words, find the model L_i which maximized $N(L_i|D)$ as:

$$P(L_i|D) = \frac{P(L_i)P(D|L_i)}{P(D)} \quad (3)$$

As the assumptions which the levels have equal initial distribution and the word number N of passage is independent to the levels, former function can be simplified as:

$$N(L_i|D) = \log Z + \sum_{t \in D} c(t) \log P(t|L_i) \quad (4)$$

Logarithm is used in function (4), and $\log Z$ represents combined factors involving passage length and the uniform prior $P(L_i)$. Hitherto, the Smoothed Unigram model is built simply by Eq. (4).

3.2 Modified Smoothed Unigram

The effectiveness of original Smoothed Unigram is proved in [12]. However, two characteristics make it not very ideal for online language learning and second language learning. Firstly, in original Smoothed Unigram, sentence length is not considered as a syntactic component. It is certainly suitable for native language learners. However, for foreign language learners, sentence length is a major reason which affects the readers' understanding [28]. Therefore, the average sentence length and maximum sentence length should be taken into the readability calculation. Secondly, 12-level division of text difficulty will lead to large online time consumption. Moreover, classified reading materials into too many grades will make the readabilities discrimination between two adjacent grades blurred and limit the systems' accuracy.

To solve the problems above, we modified Smoothed Unigram model in two aspects: take the sentence length into consideration and change the readability division to seven levels.

Table 1 Correspondence between letters and readability

Level 1	Level 2	Level 3	Level 4	Level 5	Level 6	Level 7
A	B	C	D	E	F	G
Very easy	Easy	A bit easy	Middle	A bit hard	Hard	Very hard

Assuming the number of sentences in a text is n , the length of sentence i is S_i , and the comprehensive sentence length σ can be used as an additional readability indicator as:

$$\sigma = \frac{\sum_{i=1}^n S_i \cdot \max(S_i)}{n} \tag{5}$$

Above function only considered the number of words in the longest sentence. Its utility is easy but may lead to noisy sentence sensitive. Noisy sentence means the sentence is extremely long but it only appears once in the text and has a very low semantic importance for understanding the full passage. A sentence length threshold δ can be introduced to minimize the interference of noisy sentence. The length of the longest sentence could be replaced by the number of over-long sentences as:

$$\sigma = \frac{\sum_{i=1}^n S_i \cdot \sum_{j=1}^m S'_j}{n} \tag{6}$$

where S' is the set of sentences in which the sentence length is beyond δ . The number of sentences in S' is m , S'_j is the length of j th sentence in S' . According to the statistic in [29], the average length of English sentences is more or <15.4 words in different corpora, and we use 22 as the value of δ in this article.

Using supplementary indicator σ to improve the ability to distinguish between text readability, the final equation of original Smooth Unigram model can be modified as:

$$N'(L_i) = k\sigma \left[\log Z + \sum_{t \in D} c(t) \log P(t|L_i) \right] \tag{7}$$

where $N'(L_i)$ is the final readability, $k \in [0, 1]$ is an empirical coefficient.

In order to increase the dissimilarity between classes, we merged the 12 grades of readability in Smoothed Unigram into seven levels. By the benefits of this modification, the system’s runtime complexity will reduce and user experience of readability will be more obvious. The seven levels of readability can be represented using uppercase letters A–G with the increasing difficulty. Correspondence between letters and readability is shown in Table 1.

Finally, the initial readability sorting based on modified Smooth Unigram model will work following the below steps:

1. *Preprocessing* Make pretreatment for input documents in vocabulary level.
2. *Feature selection* Represent documents by most representative features.

3. *Modeling* Construct Smoothed Unigram model.
4. Calculate the readability of document using the model and sentence length.
5. Output letters which represent the readability of input reading materials.

4 Online-Boost algorithm

Boosting is a voting-based method for improving the performance of classification by using a group of weak classifiers instead the attempt to build single powerful strong classifier [29]. In Boosting, the classifiers are trained sequentially. Before training the i th classifier, the training set is reweighed with greater weight given to the documents that were misclassified by the previous classifiers. The reweight strategy is suitable for updating the readability. What’s more, online learners can be used as a group of base classifiers. Therefore, Boosting is a helpful methodology for online readability update and learners’ reading comprehension evaluation.

The original Boosting algorithm uses three weak classifiers (c_1, c_2, c_3) to form a committee. It divides a large training set into three parts (X_1, X_2, X_3) randomly and uses X_1 to train c_1 firstly. Then, it uses the subset of X_1 which is misclassified by c_1 and X_2 together as the training set of c_2 . The rest can be done in the same manner.

4.1 AdaBoost algorithm family

The AdaBoost algorithm is the best known example of Boosting approach [30]. Because of its high performance, AdaBoost become a big algorithm family. AdaBoost.MH [31], AdaBoost.P [32], AdaBoost.L [33], Multi-class Adaboost [34] and asymmetric AdaBoost [35] are the most important variants of AdaBoost family. Schapire [36] restated the original AdaBoost, some later researchers perfect its details [37–39].

To control the computational cost in a reasonable range, AdaBoost uses a dual-weighted process to choose training sets and classifiers. The detailed steps of AdaBoost are as follows:

1. Given training set $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ where x_i is the training sample and $y_i \in \{1, -1\}$ denotes x_i ’s category label ($1 \leq i \leq n$).
2. Let $f_j(x_i)$ denote i th feature of j th document.

3. Define the initial distribution of documents in the training set $D_t(i) = \frac{1}{N}$
4. Searching weak classifier $c_t(t = 1, 2, \dots, T)$: for j th feature of every sample, a weak classifier c_j can be obtained, and thus get the threshold θ_j and orientation P_j to minimum the error ε_j as below:

$$\varepsilon_j = \sum_{i=1}^n D_t(x_i) |c_j(x_i) \neq y_i| \quad (8)$$

Therefore, the weak classifier c_j is:

$$c_j(x) = \begin{cases} 1, & P_j f_j(x) < P_j \theta_j \\ -1, & \text{otherwise} \end{cases} \quad (9)$$

5. Choose c_j from the whole feature space which has the minimal error ε_j as the weak classifier.
6. Recalculate the feature of samples:

$$D_{t+1}(i) = \frac{D_t(i) e^{-\alpha_i y_i c_t(x_i)}}{Z_t} \quad (10)$$

where Z_t is a normalization factor which makes $\sum_{i=1}^n D_{t+1}(i) = 1$ and α_i is the weight.

7. Repeat the steps above T times and get T optimal weak classifiers with different weights.
8. Combine weak classifiers according to their weight to construct a strong classifier:

$$C_{\text{strong}}(x) = \text{sign} \left(\sum_{t=1}^T \alpha_t c_t(x) \right) \quad (11)$$

Training set utilization can be enhanced using the algorithm above through adjusting the weights of misclassified texts. In addition, the performance of base classifiers is evaluated by the weighting process. These attributions of AdaBoost are useful for online readability and reading comprehension update.

However, the direct use of AdaBoost in online language learning system is impossible. Several significant differences exist between traditional AdaBoost and the request of online readability classification. First of all, online learning systems emphasis on users' experience; therefore, the readability should be judged by users. However, in traditional AdaBoost, this work is undertaken by machine learning algorithms such as SVM, neural networks. Secondly, the classification ability of base classifiers is transparent to the user. In online learning system, the classification ability equals user's reading comprehension when using learners as the weak classifier. Therefore, it is important to make users know their level of language skill. Thirdly, AdaBoost cannot give base classifiers corresponding documents according to their classification ability. This disadvantage should be overcome to make the system automatically select reading materials with suitable

difficulty for users of different reading comprehension. In addition, the online system must have higher robustness because human classifiers have more randomness than machine classifiers. In other words, the online learning system may face more noise.

4.2 Readability updating

The advantage of using online learners as base classifiers in the system will no longer need preprocessing. Obviously, not only words segmentation and POS but also feature selection and document representation are not necessary for human to make classification. More than 60 % time cost will be saved [40] by this way.

The most intuitive criterion of learners-based readability classification is average correct rate. The high average correct rate reveals the low semantic difficulty. Without considering the users' language skills, when all the questions of a text are answered correctly by all the users, it will be regarded as extremely easy text and vice versa. Three assumptions are very important in the initial step of readability update and reading comprehension evaluation:

1. Readability of texts is independent of each other.
2. Language skills of learners are independent.
3. All learners have the same initial reading comprehension.

Assuming the premier readability of document k evaluated online is $R_1(k)$, it can be calculated as:

$$R_1(k) = 1 / \left(\frac{\theta \sum_{i=1}^n C_i}{nQ} \right) \quad (12)$$

where θ is the initial readability determined by modified Smooth Unigram of document i , C_i is the number of right answer given by learner i , n is the number of learners, Q is the sum of questions in text i .

Similar to the first step of readability update according to the average correct rate, the initial reading comprehension of learner j $S_1(j)$ can be computed as:

$$S_1(j) = \frac{\sum_{i=1}^m \theta_i T_i}{\sum_{i=1}^m Q_i} \quad (13)$$

where m is the number of texts, θ_i is the readability of text i given by modified Smoothed Unigram model, T_i is the sum of right answer in text i , Q_i is the question number of text i .

Since system calculated and recorded the data of users' language skills, in the second step of readability update can take more information into account for getting a more precious result. It is easy to imagine that if a reading material has a low correct rate when trained by a high language skill user, it may have very low readability. Similarly, high correct rate text which will be answered by low reading comprehension learners reveals it is less

difficult. In other words, correct rate of low language skill users and error rate of high language skill users are more important and should have greater weights.

To determine high and low reading comprehension, two thresholds α and β are introduced into the system. When the language skill of learner j is lower than α , its contribution for readability will be replaced by a lower bound R_{MIN} , and when its language skill is higher than β , its contribution for readability will be replaced by an upper bound R_{MAX} . Assuming the readability of document k in the second update round is $R_2(k)$, it can be calculated as:

$$R_2(k) = R_2(A) + R_2(B) + R_2(C) \tag{14}$$

where $R_2(A)$, $R_2(B)$ and $R_2(C)$ are defined as:

$$\begin{cases} R_2(A) = R_{MIN}, & S_1(j) \leq \alpha \\ R_2(B) = 1/\prod_i^n S_1(j)C_j, & \alpha \leq S_1(j) \leq \beta \\ R_2(C) = R_{MAX}, & S_1(j) \geq \beta \end{cases} \tag{15}$$

The readability update procedure can keep on running by an iterative way according to the function above:

$$\begin{cases} R_i(A) = R_{MIN}, & S_{i-1}(j) \leq \alpha \\ R_i(B) = 1/\prod_i^n S_{i-1}(j)C_j, & \alpha \leq S_{i-1}(j) \leq \beta \\ R_i(C) = R_{MAX}, & S_{i-1}(j) \geq \beta \end{cases} \tag{16}$$

Using the iterative equation above, system updates the readability of reading materials in real time according to the learners' performance of article understanding. The work steps of readability update is shown in Fig. 3.

Above function only considered the symmetry case which all learners are in the same round. However, the most commonly situation is asymmetric. The randomness of online behavior makes it impossible that all the users practice same number of documents.

Fortunately, formula (16) can meet the request of asymmetric situation when making a small modification. Assuming the practiced texts of learn i is n_i , replacing the former definition of $R_i(B)$ by:

$$R_i(B) = \frac{\sum_{i=1}^n S_i(n_{i-1})}{\sum_{i=1}^n n_{i-1} \cdot S_i(n_{i-1})} \tag{17}$$

Hitherto, the system can update readability of reading materials in real time in different situations. Furthermore, the online readability update algorithm is noise insensitive according to the analysis above.

4.3 Reading comprehension evaluation

In the former subsection, only average correct rates are considered when making reading comprehension evaluation. It is the simplest but not the best way to examine users' language skills. To improve the performance of reading ability evaluation, some statistical characteristics are introduced.

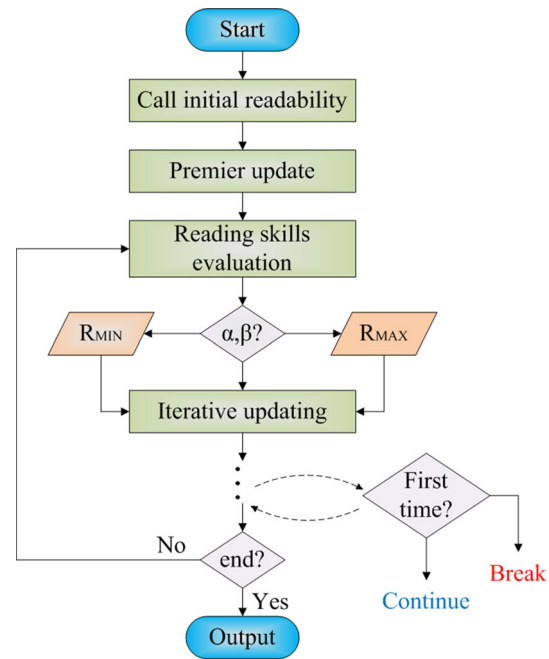


Fig. 3 Work steps of readability update

Stability is an important indicator for performance evaluation. If the correct rate of a learner's answers in different texts fluctuate greatly, his language ability will be considered not so good even he has the same correct rate with other users. To avoid the runtime complexity cause by power computing, following variant of correct rate variance can be used:

$$V_i = \sum_{i=1}^n \left[\frac{T_i}{Q_i} - \sum_{i=1}^n \frac{T_i}{Q_i} / n \right] \tag{18}$$

The above formula also has the role of filtering. Imagine the situation that a weak-skill language learner encounters the article which he had read, the correct rate would be enhanced by this kind of passages. On the other hand, when a high comprehension user just tests the function of the system and gives the random answers, his correct rate will be reduced. Influence of these interfering factors could be limited by function (18).

Correct rates of texts with different readability should play different roles in reading comprehension evaluation. The readability can be used as weight of correct rate:

$$S(j)^A = \frac{1}{m} \left(\sum_{j=1}^m R(j)T_j / C_j \right) \tag{19}$$

Another valuable statistical characteristic is the correct rate of a learner when he is tested by difficult reading materials. When two users have approximately same average correct rate, the one who have higher correct rate in harder test probably has better language skills. The standard of difficulty can be defined as:

$$R(j) > \frac{1}{2} \left(\max_{j \in m} \{R(j)\} + \frac{1}{m} \sum_{j=1}^m R(j) \right) \quad (20)$$

When readability meets the above constraint, the reading comprehension can be calculated as:

$$S(j)^B = \xi \frac{1}{m} \left(\sum_{j=1}^m R(j) T_j / C_j \right) \quad (21)$$

where $\xi > 1$ is an empirical parameter.

Similarly, the concept of easy can be defined as:

$$R(j) < \frac{1}{2} \left(\min_{j \in m} \{R(j)\} + \frac{1}{m} \sum_{j=1}^m R(j) \right) \quad (22)$$

When the readability of document j satisfied above constraint, its dyslexia is very low. Therefore, the correct rates of these reading materials should make less contribution in the comprehension evaluation. An empirical parameter $\zeta \in [0.5, 1]$ can be used as:

$$S(j)^C = \zeta \frac{1}{m} \left(\sum_{j=1}^m R(j) T_j / C_j \right) \quad (23)$$

Algorithm: Online-Boost reading comprehension evaluation

Input: reading materials $D = \{d_1, d_2, \dots, d_m\}$ with questions
Users' answers

Output: reading comprehension of learners

```

1 begin
2   calculate the average correct rate of users
3   call readability calculated by Smoothed Unigram model
4   for( i=1, i <= n, i++)
5     for( j=1, j <= m, j++)
6       initial reading comprehension calculation
7   continue to receive users' answers
8   float S(1) = S(2) = S(3) = 0
9   for( i=1, i <= n, i++)
10    for( j=1, j <= m, j++)
11      call readability of texts from update algorithm
12      if ( R(j) < t1 )
13        S(1) = S(j)A
14      else if ( t1 <= R(j) <= t2 )
15        S(2) = S(j)B
16      else
17        S(3) = S(j)C
18      else if ( wj < β && w'j < β )
19        Wj = wMIN
20      calculate the variance of correct rate
21      S'(j) = S(1) + S(2) + S(3)
22      variance > threshold? S(j) = S'(j); S(j) = ρS'(j)
23      outliers value of weithts? goto step 2: continue
24 end

```

Fig. 4 Pseudo code of reading comprehension evaluation

In this way, the weights of correct rates in easy texts are limited by function (23). Integrating the analysis above, the final reading comprehension $S(j)$ of learner j can be evaluated by following function:

$$S(j) = S(j)^A \cdot S(j)^B \cdot S(j)^C \quad (24)$$

When learners use the system more than once, their reading comprehension can be evaluated and updated by using Eq. (24) iteratively.

However, similar to function (16), above function also just considers the symmetry case in which all the reading materials are in the same round. Function (19) can be modified to increase the scope of application of the system by using the number of documents which learner j has read instead the sum of texts. The detail of reading comprehension evaluation is shown in its pseudo code (Fig. 4):

where t_1 is the lower bound of readability and t_2 is the upper bound of readability. By introducing the parameter ρ , variance of correct rate between different reading materials is considered when making the reading comprehension evaluation.

4.4 Overview of Online-Boost

As the former analysis, update of readability needs to call the computation results of reading comprehension evaluation and the evaluation of language skills needs to call the readability. Therefore, some gradation strategy should be designed to avoid the system fall into an infinite loop.

As we know, the initial readability of different reading materials has given by modified Smoothed Unigram model. The initial readability can be used for first-step reading comprehension evaluation and the evaluated reading comprehension can be used for first-step readability update. Readers can choose the level of first reading text as their language skills are unknown. In this way, a cross-iterative mechanism can be used as the final integration strategy. The algorithm which can package deal with readability update and reading comprehension evaluation is called Online-Boost in this paper. Its operation sequence is shown in Fig. 5.

Following steps above, the online language learning system can update documents' readability and evaluate users' reading comprehension in real time. The comprehensive weighting method guaranteed system's accuracy. Furthermore, efficiency of the system will be enhanced significantly without the procedures of feature selection and feature computation.

5 Preparing for application

As the core of online foreign language learning system, Online-Boost algorithm is constructed in previous section.

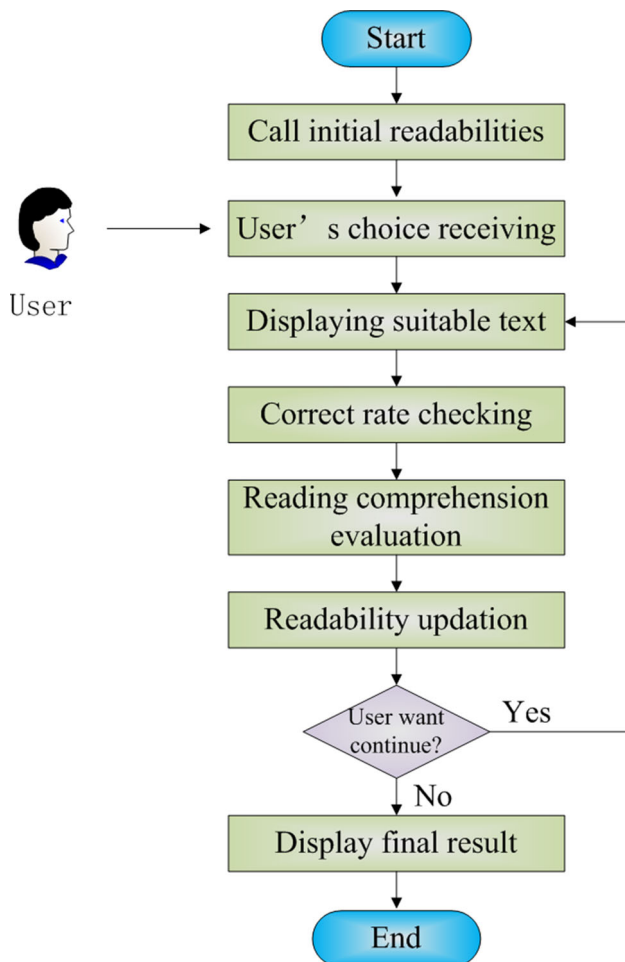


Fig. 5 Operation sequences of Online-Boost

However, the algorithm could not be used directly to build a system because condition missing exists in the system.

5.1 Scalability

Reading materials of the system should be dynamic. It means system managers could increase documents to the system after it became online. Constantly updated reading resources will increase the attractiveness of the system and make the system to meet the needs of more users with different interests. Therefore, the system should have a high scalability for new texts.

To achieve scalability, we create tables in the database which are devoted to store new reading materials. When the new documents are entered, they will be stored in a dedicated database space. Then, the modified Smoothed Unigram model will call these new texts for initial readability computation. After the initial readability sorting, the new reading materials will be considered as known difficulty documents, so they will be stored and updated together with other texts. Thus, the dedicated tables are

cleared to receive new documents. The detailed steps are as follows:

1. The database is divided into two parts: D_1 for storing readability unknown documents and D_2 for storing readability known documents.
2. Input new reading materials set R_1 and store them in D_1 .
3. Call modified Smoothed Unigram model to calculating initial readabilities of R_1 .
4. Dump all the documents in R_1 and their initial readabilities in D_2 and empty D_1 .
5. Use texts in D_2 as the learning materials and D_1 to receive new documents.

5.2 Concurrency control mechanism

As an online system which may face many users at the same time, it must have the ability of concurrency control. In online learning system, the most important concurrency is users submit answers of a document at the same time. Different from the classic online ticketing problem that only one user's submission could be accepted, in the language learning system, all results submitted at the same time by all users should be accepted by the learning system to draw a comprehensive conclusion of the document's readability.

Many mechanisms can be used in different scenarios for concurrency control such as lock-based protocol, time-stamp-based protocol, mechanism based on validity checking and multi-version concurrency control (MVCC) [41]. The characteristics of database transactions in online foreign language learning system are as follows:

1. Higher requirement in response speed of read operations relative to write operations.
2. More read operations than write operations.
3. Write conflict is not frequent because many reading materials in the same difficulty grade can be used.

In a word, it is a higher read concurrency, higher read response and lower write concurrency system. Therefore, MVCC should be used to improve the efficiency of the system.

5.3 Anonymous access

Registration is needed for recording and updating users' reading comprehension and readability updating. However, registration process may make potential users of the system lose patience. Anonymous access may lead to serious interference such as malicious submissions of random answers and repeat test with same documents. The former behaviors will falsely increase difficulty of reading

materials, and the later will lead to over-evaluation of readability.

To solve problems brought by anonymous access above, a treated differently strategy could be used.

6 Experiment

We use the novel system for foreign language (English) education in The Second Junior High School of Pingluo. Fifty-three students participated in the use of the new systems in a semester. We used two methods to evaluate the effect of the novel system. The first method is system satisfaction survey. We give a survey for each student, and the effect of the novel system is divided in 6 levels: very helpful, helpful, a little helpful, hard to say, helpless and totally helpless. As a subjective feelings of students for the language learning system, the experimental results are shown in Table 2.

Changes in academic performance of students can be used as objective criteria for the effect of the novel method and system. We choose six test papers from a standard examination paper library which are divided into several standard difficulty levels, two test papers in each difficulty level. The exams were taken before and after the use of new system, spaced 4 month. The changes of average score are shown in Table 3.

Experimental results show that the systems can significantly improve learners' language skills, and their feeling of the learning system is also quite good.

Moreover, we test the novel methods efficiency. We simulated that 100 users submitted their answers at the same time, in this situation, the time delay of the system is less than 1 s.

Table 2 Results of effect survey

Level	Very helpful	Helpful	A little helpful	Hard to say	helpless	Totally helpless
Sum of votes	9	24	13	4	1	2

Table 3 Changes of average scores before and after the use of new system

Average score	Difficult level		
	Difficulty level 1	Difficulty level 2	Difficulty level 3
Before system be used	83.2	81.7	79.8
After system be used	89.6	86.4	84.7

7 Conclusion

Readability is a crucial indicator for second language learning. In this article, a readability-based language learning system is completely constructed. It uses modified Smoothed Unigram model for reading material's initial readability classification. Correct rate of users' answers will be used for readability update and reading comprehension evaluation after the system online. The updating and evaluation are achieved by Online-Boost algorithm which is the core of the system. Similar to AdaBoost, Online-Boost algorithm uses the methodology of Boosting which forms a committee by a group of experts. Different from AdaBoost algorithm family members, Online-Boost uses language learners as base classifiers. In this way, high time and computational feature selection procedure is avoided and the classification results are more in line with the user's subjective experience. In addition, in traditional Boosting-based categorization algorithm, weak classifier has higher classification ability is not important. Users only focus on the final classification result. However, in Online-Boost, because users are weak classifiers, their classification ability (reading comprehension) will be solved as vice products with no extra cost. Moreover, the system is designed with scalability. Since the system was used in a junior high school for English learning, its effect for foreign language study can be evaluated in a real teaching environment. The experimental results reveal that the method proposed in this article is helpful for improving language learners' reading comprehension.

References

- Krashen SD (1989) The input hypothesis: issues and implications. *Mod Lang J* 73(4):440–464
- Klingner JK, Artiles AJ, Barletta LM (2006) English language learners who struggle with reading: language acquisition or LD? *J Learn Disabil* 39(2):107–128
- <http://www.ets.org/toefl/ibt>
- Mc Laughlin GH (1969) SMOG grading—a new readability formula. *J Read* 20(5):639–646
- Farr JN, Jenkins JJ, Paterson DG (1951) Simplification of Flesch reading ease formula. *J Appl Psychol* 35(5):333–337
- Courtis JK, Hassan S (2002) Reading ease of bilingual annual reports. *J Bus Commun* 39(4):394–413
- Graesser AC, McNamara DS, Louwerse MM, Cai Z (2004) Coh-Matrix: analysis of text on cohesion and language. *Behav Res Methods* 36(2):193–202
- Nagy WE, Anderson RC (1987) Learning word meanings from context during normal reading. *Am Educ Res J* 24(2):237–270
- Socher R, Bauer J, Manning CD, Ng AY (2013) Parsing with compositional vector grammars. In: *The annual meeting of the Association for Computational Linguistics (ACL 2013)*, Sofia, Bulgaria, pp 213–220

10. Schwarm SE, Ostendorf M (2005) Sorting texts by readability. In: Proceedings of the 43rd annual meeting on Association for Computational Linguistics (ACL '05), pp 523–530
11. Tanaka-Ishii K, Tezuka S, Terada H (2010) Narrow-band analyzer. *Comput Linguist* 36(2):503–527
12. Schwenker F, Trentin E (2014) Pattern classification and clustering: a review of partially supervised learning approaches. *Pattern Recognit Lett* 37:4–14
13. Feldman R, Sanger J (2007) *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge University Press, New York, pp 77–78
14. Huanling T, Jun W, Zhengkui L (2010) An enhanced AdaBoost algorithm with Naive Bayesian text categorization based on a novel re-weighting strategy. *Int J Innov Comput Inf Control* 6(11):5299–5310
15. Masnadi-Shirazi H, Vasconcelos N (2011) Cost-sensitive boosting. *IEEE Trans Pattern Anal Mach Intell* 33(2):294–309. doi:10.1109/TPAMI.2010.71
16. Vu TT, Braga-Neto UM (2010) Small-sample error estimation for bagged classification rules. *EURASIP J Adv Signal Process* 2010:1–12
17. Xiaoyong L, Hui F (2012) A hybrid algorithm for text classification problem. *Prz Elektrotech* 88(1B):8–11
18. Ganiz MC, George C, Pottenger WM (2011) Higher order Naive Bayes: a novel non-IID approach to text classification. *IEEE Trans Knowl Data Eng* 23(7):1022–1034. doi:10.1109/TKDE.2010.160
19. Tan S, Li Y, Sun H et al (2014) Interpreting the public sentiment variations on twitter. *IEEE Trans Knowl Data Eng* 26(5):1158–1170
20. Yuanping Z, Mingzhu T, Jia Y (2007) Rocchio text classification based on ontology. In: 7th international conference of Chinese computing (ICCC 2007), China, 2007, pp 266–271
21. Kwon O-W, Lee J-H (2003) Text categorization based on k -nearest neighbor approach for Web site classification. *Inf Process Manag* 39(1):25–44
22. Rätsch G, Onoda T, Müller K-R (2001) Soft margins for AdaBoost. *Mach Learn* 42(3):287–320
23. Javed I, Afzal H, Majeed A et al (2014) Towards creation of linguistic resources for bilingual sentiment analysis of twitter data. In: 19th international conference on applications of natural language to information systems, Montpellier, France, pp 232–236
24. Mikolov T (2012) *Statistical language models based on neural networks*. Ph.D. thesis, Brno University of Technology
25. Crossley SA, Greenfield J, McNamara DS (2008) Assessing text readability using cognitively based indices. *Tesol Q* 42(3):475–493
26. Kanungo T, Orr D (2009) Predicting the readability of short web summaries. In: Proceedings of the second ACM international conference on web search and data mining, NY, USA, pp 202–211
27. Ganiz MC, George C, Pottenger WM (2011) Higher order Naive Bayes: a novel non-IID approach to text classification. *IEEE Trans Knowl Data Eng* 23(7):1022–1034
28. Miranda V, Jaco D, Henk F (2012) Ethnic concentration in the neighbourhood and majority and minority language: a study of first and second-generation immigrants. *Soc Sci Res* 41(3):555–569
29. Abuom TO, Roelien B (2012) Characteristics of Swahili–English bilingual agrammatic spontaneous speech and the consequences for understanding agrammatic aphasia. *J Neurolinguist* 15(5):885–893
30. González-Ortega D, Díaz-Pernas FJ, Martínez-Zarzuela M, Antón-Rodríguez M, Díez-Higuera JF, Boto-Giralda D (2010) Real-time hands, face and facial features detection and tracking: application to cognitive rehabilitation tests monitoring. *J Netw Comput Appl* 33(4):447–466
31. Schapire RE, Singer Y (2000) BoosTexter: a boosting-based system for text categorization. *Mach Learn* 39(2–3):135–168
32. Gambina A, Szczureka E, Dutkowska J, Bakunc M, Dadlez M (2009) Classification of peptide mass fingerprint data by novel no-regret boosting method. *Comput Biol Med* 39(5):460–473
33. Schapire RE (2005) Boosting with prior knowledge for call classification. *IEEE Trans Speech Audio Process* 13(2):174–181. doi:10.1109/TSA.2004.840937
34. Zhu J, Rosset S, Zou H, Hastie T (2006) Multi-class AdaBoost. Stanford Education <http://www.stanford.edu/~hastie/Papers/samme.pdf.2006>
35. Masnadi-Shirazi H, Vasconcelos N (2007) Asymmetric boosting. In: Proceedings of the 24th international conference on machine learning (ICML '07), NY, USA, pp 609–616
36. Hach F, Numanagić I, Alkan C, Sahinalp SC (2012) SCALCE: boosting sequence compression algorithms using locally consistent encoding. *Bioinformatics* 28(23):3051–3057
37. Ting KM, Zheng ZJ (2003) A study of AdaBoost with naive Bayesian classifiers: weakness and improvement. *Comput Intell* 19(2):186–200
38. Yijun S, Sinisa T, Jian L (2006) Reducing the overfitting of AdaBoost by controlling its data distribution skewness. *Int J Pattern Recogn Artif Intell* 20(7):1093–1116
39. Song E, Huang D, Ma G (2011) Semi-supervised multi-class Adaboost by exploiting unlabeled data. *Expert Syst Appl* 38(6):6720–6726
40. Uguz H (2011) A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm. *Knowl Based Syst* 24(7):1024–1032
41. Larson P, Diaconu C, Zwilling MJ, Freedman CS (2011) Optimistic multi-version concurrency control system used for controlling concurrently executing transactions, assigns created version of data records of data store as two timestamps indicating lifetime of version. US Patent US 2011153566-A1 (online). http://www.patentlens.net/patentlens/patent/US_201153566-A1