Predicting the Difficulty of Multiple-Choice Cloze Questions for Computer-Adaptive Testing

Ayako Hoshino[1]*
Hiroshi Nakagawa[2]
[1]NEC Common Platform Software Research Laboratories
[2]Unviersity of Tokyo
a-hoshino@cj.jp.nec.com, n3@dl.itcu-tokyo.ac.jp

Abstract

Multiple-choice, fill-in-the-blank questions are widely used to assess language learners' knowledge of grammar and vocabulary. The questions are often used in CAT (Computer-Adaptive Testing) systems, which are commonly based on IRT (Item Response Theory). The drawback of a simple application of IRT is that it requires training data which are not available in many real world situations. In this work, we explore a machine learning approach to predict the difficulty of a question from automatically extractable features. With the use of the SVM (Support Vector Machine) learning algorithm and 27 features, we achieve over 70% accuracy in a two-way classification task. The trained classifier is applied to a CAT system with a group of postgraduate ESL (English as a Second Language) students. The results show that the predicted values are more indicative of the testees performance than a baseline index (sentence length) alone.

1 Introduction

Multiple-Choice (MC) questions are commonly used in many standardized tests, due to its ease of collecting and marking the answers. Multiple-Choice Fill-In-the-Blank[1] (MC-FIB) questions, especially, have proven their effectiveness in testing language learners' knowledge and usage of a certain grammar rule or a vocabulary item. The TOEIC[2] test, for example, has questions in this format. Below is an example of a multiple-choice cloze question:

There is a [ ] on the tree.
1) bird 2) birds 3) orange 4) oranges

The sentence is called the *stem*, in which the blank is shown as the brackets. The alternatives consist of one right answer and the distractors.

*This work has been done as a part of Ph.D. studies at University of Tokyo.
[1]For brevity, we use *cloze* instead of fill-in-the-blank.
[2]Test of English for International Communication, run by ETS (Educational Testing Services), U.S.

CAT is a technology that are motivated to offer a better assessment by being adaptive to the testee, where the computer administers subsequent questions depending on the precedent performance of the testee. IRT provides the theoretical background on most of the current CAT systems, in which a testee's ability (or *latent trait*) and the difficulty of a question are described in values in a common unit called *logit*. In IRT, the difference in the ability and difficulty is projected to the probability of the users' getting the right answer, using a sigmoid function. For a fuller introduction to IRT, the readers are guided to Baker et al. [1].

IRT is a well-researched area, where many positive results have been reported. For example, Urry reports that an IRT-based CAT system achieves sufficient precision with 20 questions, whereas pen-and-paper testing requires 100 questions [2]. However, less attention has been focused on the cost of adapting the model, which is, as commonly practiced, the cost of conducting the pre-test results on the comparable group of testees. In cases where pre-testing is not possible, all questions are assumed to be of equal difficulty at the onset, then the difficulty of the questions is updated as the users' responses accumulate [1].

This research is motivated to combine CAT with recently emerging AQG technology (Automatic Question Generation, explained in the following section), which will render possible a novel assessment system that adaptively administers questions from the automatically generated question set. However, one obstacle is that the above-mentioned problem of adapting an IRT model to the testees' level, whose cost will be higher as the available number of questions increases. Existing methods of IRT model adaptation will be inpractical when unlimited number of newly-generated questions are added to the question pool.

In this situation, it is vital to have a means of automatically calculating a rough prediction of difficulty, or inferring the difficulty of a question from the performance of the targeted group on similar questions. The use of supervised machine learning would be worth exploring, which will also be an attempt to have the computer gain a general notion of the difficulty of MC-FIB questions.

The rest of the paper is organized as follows: We review relevant work on difficulty prediction in Section 2. In Section 3, the proposed method is presented with evaluation results on a closed data set. In section 4, the trained classifier is tested in a subject group experiment. Section 5 concludes the paper.

2 Related Work

There is only limited literature in the field of computational linguistics on MC-FIB questions for language proficiency testing. Among the attempts in generating MC questions from an input text [3] [4] [5] [6] [7]. One of the AQG studies provide a method to compute the complexity of reading comprehension questions [8][3] to be used in a CAT system. Their measure is defined as a weighted sum of complexity values on the stem sentence, the paragraph, the answer sentence and so forth.

[3]In their study, the format of questions was neither MC nor FIB, thus the answer is composed by the testee.

In fact, the complexity of a sentence alone has been studied for decades in computational linguistics, and many indices have been proposed. Segler, in his work on extracting example sentences for language learners, compares traditionally proposed indices [9]. The indices include sentence length (number of words in the sentence), parse-tree depth (maximum number of levels from the root to a word), and the combination of such factors. Segler's comparison reveals that it's very hard to beat the sentence length, which is the simplest measure.

The complexity of the stem sentence affects the difficulty of the above-presented question. But other factors, such as similarities between the right answer and the distractors would surely influence the difficulty of MC-FIB, and thus should be taken into account.

The Flesch-Kincaid Index is a readability measure for a passage, which is widely used among educators. The index is defined as follows:

$$R_{FRE} = 206.8 - 1.05X - 84.6Y$$

where X is the average number of syllables in a word and Y is the average number of words in a sentence. The index can be applied to a sentence with Y fixed to one. This version of Flesch/Kincaid score is composed so the value is interpreted as the grade in an American elementary school.

Some improvements of the readability measures have been proposed [10]. Miyazaki et al. proposed an individualized measure for reading text extraction [11].

Evaluation of the existing indices are done often manually with add-hock parameters. In this study, a supervised machine learning technique is used to tune the parameters in combining feature values.

3 Difficulty Prediction

In this study, we use the technique of supervised machine learning for the task of difficulty prediction. We first explore the learning algorithms, and train the best performing classifier using the question data that are annotated with the correct response rate. Then, with a simple binary search-like method based on the predicted difficulty values, we build a CAT system and have it tried out by human subjects.

As this is one of the earliest attempts in applying machine learning methods to such a task, we set out with a simple binary classification. We did not employ regression, as some of the readers may wonder, which outputs numerical values. The reason was that 1) it is expected to be unworthy to predict the correct response rate that is observed from subject groups, since such observation usually contains what is called *measurement error* in the literature of psychometrics. We train the classifier with the labels "easy" or "difficult," letting the computer to try to grasp a rough notion of difficulty.

3.1 Training Data

The training data set is obtained from a series of TOEIC preparation books (a total of 702 questions from Part 5, which are MC-FIBs.) Each question is annotated with the correct response rates, ranging from 0.0 to 98.5. The figures are based on the tests in

a TOEIC preparation school in Japan and reportedly based on the results of about 300 testees. Table 6 (in Appendix) shows samples from the training data.

All questions consist of a stem sentence of 20-30 words and four alternatives. Seemingly, all questions are intended to be of the same difficulty, rather than being increasingly difficult according to the question number. We have labeled the top 305 easiest questions as "easy" and the top 305 difficult questions as "difficult," based on the correct response rates, leaving out 8% around the average value [4].

3.2 Features

On deciding the feature set, we take a similar approach to Kunichika et al., who designed the factors of difficulty depending on the complexity of the question and of the answer. We assume the difficulty of a question to be composed of 1) the difficulty of the stem sentence, 2) the difficulty of the correct answer, and 3) the similarity between the correct answer and the distractors. As MC-FIB questions are often criticized for being possible to obtain the right answer just by reading a few words before and after the blanks, we have also added as a feature as 4) the tri-gram including the correct answer or a distractor. Each feature, with its notation used in this paper, is explained as follows:

1) Sentence features
The sentence features consist of **sentencelength**, which is the number of words in the original sentence, **maxdepth**, which is defined as the *depth*, or number of brackets/levels from the root to the deepest word in a parse result [5], and **avr_wlen**, which is the average number of characters in a word.

2) Answer features
The answer features provide information on the right answer, consisting of **blanklength**, which is the number of words in the correct answer, and an array of binary features on the POS (Part Of Speech) of the right answer (**pos_V**, **pos_N**, **pos_PREP**, **pos_ADV**, and **pos_ADJ**), which indicate inclusion of the part of speech in the right answer. For example, **pos_V** is true if any form of a verb is included in the correct alternative.

3) Distractor similarity features
The distractor similarity features are obtained from the analysis using the technique of modified edit distance, which have been used to extract a lowest-cost conversion path from one string to another. In our version of edit distance, we have applied the algorithm on a word basis, as opposed to the character based application as seen in spelling-error correction. While three kinds of operations, *insert*, *delete*, and *change*, are used in a standard edit distance, we have additionally defined three operations: *change_inflection*, *change_lexical_item*, and *change_suffix*. *Change_inflection* is an operation where a word is substituted by the same vocabulary item, but in a different inflectional form. *Change_lexical_item* is a substitution of a word with the same vocabulary item in the same inflectional form. *Change_suffix* is a substitution of a word

[4] This 8% was decided in a cross validation; we took a breaking point that maximizes the accuracy.
[5] We used the Charniak parser `http://www.cs.brown.edu/~ec/`.

Table 1: Results with different learning algorithms

Algorithm	Accuracy
SVM	**62.7960%**
SMO	58.1481%
Logistic	57.4074%
VotedPerceptron	57.0370%
IB5	57.0370%
NaiveBayes	56.6667%
SimpleLogistic	56.4198%
IB10	55.5556%
MultilayerPerceptron	53.9506%
J48	53.7037%
IB3	52.8395%
RBF Network	52.2222%
IB1	51.8519%

with another word with the same stem, which can be of a different part of speech. We set the cost of operation so the *change_inflection* (cost: 2), *change_lexical_item* (3), *change_suffix* (4), and standard *change* (5) are preferred in this order. The cost of *insert* and *delete* is set to 3, so the combination of deletion and insertion is never used when one change yelds the same result.

The features derived from the analysis are: **pathlength** is defined as the number of operations, and an array of binary features (**include_insert**, **include_delete**, **include_changeinflection**, **include_changelexicalitem**, **include_changesuffix**, **include_change**). For example, **include_insert** is true if any of the conversions from the right answer to the three distractors include an operation *insert*.

4) Tri-gram features
The tri-gram features are obtained from a search engine's hit score, assuming that the figure reflects the familiarity of a given tri-gram. Tri-gram features on the correct answer are **hit_correct**, **hit_pre2_corr**, **hit_pre2_corr**, **hit_pre_corr_post**, and **hit_corr_post2**, where **hit_correct** is the google hit score of the correct answer, **hit_pre2_corr** is the same score of the correct answer with the previous two words, **hit_pre_corr_post** is the hit score of the correct answer with the previous word and the following word, and **hit_corr_post2** is the hit score of the correct answer with the following two words. The same set of features is defined for the distractors, where **hit_distractor** is the average of the google hit score of the three distractors. All queries are posted in parenthesis.

All features are turned into numerical values, normalized[6] before being fed to the learner.

3.3 Learning Algorithms

We conduct the 10-fold cross validation to compare the performance of different learning algorithms from weka machine learning toolkit [12] and SVM$_{light}$ applied to this task. Table 1 shows the accuracy of each learning algorithm.

To our disappointment, many of the learning algorithms performed no different from sheer randomness (50%). The best accuracy is achieved by SVM$_{light}$, noted as

[6]We use a normalization filter in ***weka*** package (`http://www.cs.waikato.ac.nz/~ml/weka/`). We have also tried standardization, but did not obtain better performances than normalization.

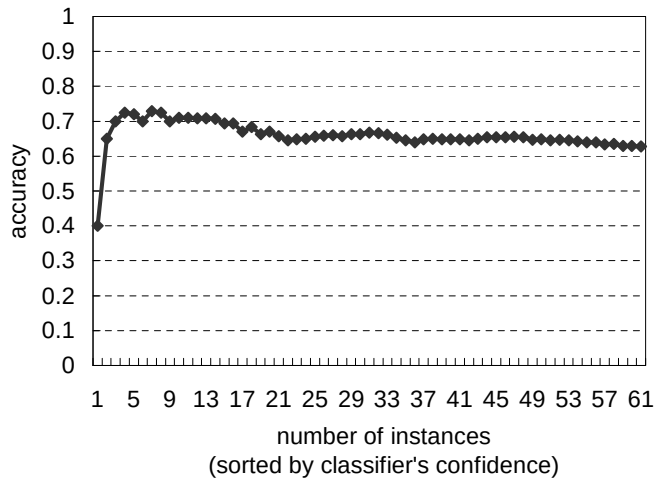

Figure 1: Accuracy on top n instances

SVM in Table 1, outperforming complex algorithms such as Voted Perceptron. Although, some of the classifiers have unexplored parameters (where we have used default values) that could change the resulting accuracy, we concluded that SVM is the most suited algorithm for this task. The fact that SMO (Sequential Minimal Optimization), which is also a version of SVM, performs second best supports the hypothesis that the high-dimensional maximum margin approach is effective for this problem.

One can observe that the parameter *k* optimizes IBk, or *k*-nearest neighbor algorithm, at around 5, though the gain from IB1 remains about 5%.

3.4 Top-N Accuracy

Using SVM$_{light}$, we conducted another cross validation with top N evaluation. Valuing more precision than recall, top N evaluation is an evaluation where the test instances are sorted by the confidence [7] of the classifier and the accuracy is calculated on the top N confidently classified instances. The figure in top N accuracy can be interpreted as the performance of the classifier when it is allowed to skip less confident instances. The result of top-N evaluation is shown in Figure 1.

The line indicates the mean accuracy of 10 split, averaged over the results of 1,000 runs. The x-axis shows the number of instances the classifier has labeled, and of which the classifier is most confident. The mean accuracy draws a sharp curve at first, achieving 72% with about 10 instances at the top. The accuracy is kept above 70%, up until the top 18 instances. Then, it slopes down marking 65% at top 50 instances and gradually slopes down reaching 0.629% in labeling all instances. The overall standard deviation was about 0.005, which indicates the classifier's stability. On the condition

[7] The confidence value in the context of the SVM algorithm is the distance to the separating hyperplane. We use this index as *difficulty value*, which ranges from negative to positive values, and in our case, smaller value signifies more difficult; if the classifier is more confident of a question's being difficult, it is a more difficult question.

Table 2: Accuracy gain of each feature (Left: those with positive contribution, Right: those with zero or negative contribution)

Feature f	w/o f	Gain
all features	**0.70769**	-
pos_N	0.65385	0.05385
pos_V	0.66154	0.04615
sentencelength	0.66923	0.03846
hit_pre2_dist	0.66923	0.03846
include_change	0.67692	0.03077
maxdepth	0.68462	0.02308
avr_wlen	0.68462	0.02308
hit_pre2_answer	0.69231	0.01538
hit_pre_dist_post	0.69231	0.01538
include_changesuffix	0.70000	0.00769
hit_dist_post2	0.70000	0.00769
hit_pre_answer_post	0.70000	0.00769
include_insert	0.70769	0.00000
answer_wlen	0.70769	0.00000
pos_ADV	0.70769	0.00000
blanklength	0.70769	0.00000
include_delete	0.71538	-0.00769
include_change _lexicalitem	0.71538	-0.00769
include_change _inflection	0.72308	-0.01538
pos_PREP	0.72308	-0.01538
hit_dist_count	0.72308	-0.01538
pathlength	0.73077	-0.02308
pos_ADJ	0.73077	-0.02308
hit_answer_post2	0.73077	-0.02308

that the predictor works similarly well on the automatically generated questions, the predictor could label the question with over 70% accuracy skipping three out of four instances.[8]

Note that binary classification can be extended to comparing and ranking two or more instances. SVM algorithm can be used for comparing two or more candidate instances, which is the way a classifier will work in an actual CAT system.

3.5 Feature Analysis

We investigate which of the features have contributed to the accuracy of the classifier, by removing one feature at a time. The difference from the accuracy with all features signifies the gain of accuracy caused by the feature. The experimental settings was the same as the above cross validation. Table 2 shows the accuracy and the performance gain with the top 15 instances, where the performance is kept above 70% with the largest number of labeled instances.

The most contributing feature was **pos_N**, with the gain of 5% of accuracy. **Pos_V** provides the next largest gain. These two features exceed the contribution of the sentence length, which is assumed to be an undoubted predictor for sentence difficulty. Then, information on the hits scores and the operations in the paths follows, with **include_change** and **include_changesuffix** providing larger contributions. The four features (**include_insert**, **blanklength**, **pos_AD** and **hit_answer**), however, do not affect the accuracy at the point of top 15. Several features, such as **hit_answer_post2** and **pos_ADJ**, exhibit negative contributions, though those features contribute by large margins as more instances are labeled. **Avr_wlen** provides a larger contribution as more instances are counted as top n.

[8]Note that skipping unconfident instances does not cause a problem in our AQG+CAT system, since unlimited number of automatically-generated questions are available.

To summarize the results of the above experiments: 1) the overall performance is higher with the instances with higher confidence values than all instances, 2) path features contribute to the accuracy, 3) **include_change** and **include_changesuffix** contribute more than the other path-related features, while features such as **include_insert** don't provide much information at the point of $n = 15$, and 4) the POS information on the answer phrase helps the accuracy boost, depending on which POS to look at. Information on the verb or noun inclusion in the answer phrase significantly contributes to the accuracy. However, adverb, preposition, and adjective do not result in positive accuracy gains. 5) Some of the features do not look effective with fewer number of confident instances, although many of them prove effective with the larger number of instances.

The results of cross validation with different features removed also provide an analysis on the group of testees. In the case of our data, the feature contribution reflects the tendencies of the testees who go to the TOEIC preparation school. It could also be possible to assume that the group of testees is a good sample of adult Japanese learners of English. Since the method of cross validation allows us to repeat the experiments with different features, the researcher of SLA (Second Language Acquisition) can perform analysis with their own devised features, without the need to collect the data with a carefully designed subject experiment. Also, provided with the sufficient amount of the data with a given learner, the analysis can provide a personal diagnosis of their tendencies.

3.6 Human Performance

The difference among the difficulties of the questions taken from the same series of books was quite subtle. To see the difficulty of this task for human judges, we asked two Japanese assistant professors in computer science to perform the binary classification. They were presented with a mixed set of 40 "difficult" questions and 40 "easy" questions and guessed the label of each instance. The accuracy of the two human judges was 70% and 72.5%. Considering the fact that the performance of human subjects is normally deemed to be the upper limit of an NLP task, this not-too-high performance of SVM classifier makes sense. The subjects pointed the type of question was a clue they used to decide the labels; the grammar questions tended to seem easy, while vocabulary questions were more difficult.

4 Subject Experiment

In order to see the efficacy of a trained difficulty predictor in the context of AQG+CAT testing, we conducted a subject experiment. The questions we use in the experiment are automatically generated from online news articles, which are administered by a simple algorithm based on the predicted difficulty values.

The entire evaluation was conducted through the Internet. The subjects were called for and volunteered through the department's email list. The participants were instructed by email, tried out the CAT system through their Internet browser, and answered the post-task questionnaire by e-mail.

Twenty students responded to our call. They are master's and doctoral students majoring in information studies (with either literature or science background.) Their first languages are Japanese (12) and Chinese (6) and other languages (2).

4.1 Automatic Question Generation

With our in-house AQG system, we generated grammar and vocabulary questions on articles from several online news websites: NHK (Japan), BBC (U.K.), and DongA (Korea). The method of AQG we employed was Coniam's frequency-based method [13] for vocabulary distractors and hand-written patterns for grammar distractors. Table 7 (in Appendix) shows samples from the questions used in the experiment, along with their predicted difficulty value (first column) and the correct response rates (second column). In this subject experiment, a set of automatically-generated questions were labeled with a classifier trained in an abovementioned method, then administered to the subjects.[9] For more information on our AQG method, see [14].

4.2 Administration algorithm

Assessing a testee with a CAT system is done in a similar way as finding their position on a number line. The system starts with the questions of a mean difficulty, then it jumps a pointer (representing a participant's position) to go up or down depending on the participant's response. The width of the jump is reduced as more questions are administered, following an inverse logarithmic function as defined below:

$$C(top - bottom)/log(n + 1)$$

where top is the maximum and $bottom$ is the minimum difficulty value in the question pool, which, at the beginning of evaluation, contained 3,000 automatically-generated questions. The value n is the number of questions attempted so far. The constant C is set by the simulation experiments. In this experiment, our system excludes the sentences that have previously been exposed to the participant.

4.3 Experiment results

We have conducted a three-session experiment, where the participants took part in two or all three sessions. The number of participants were 12 in the first, 15 in the second, and 17 in the third session. Fifty questions were administered at each session, where two sessions were with random administration and one session was with an adaptive administration. The basic information of the test results is summarized in Table 3.

[9] When applying the SVM$_{light}$'s classifier trained with the aforementioned data, the resulting values of the test data are extremely skewed. In fact, at most of the time, the same value is gained for all test data. This could be attributed to the difference between the training data and the test data. For example, the sentences from the news articles tend to be longer than the ones in TOEIC MC-FIB questions. Thus, the feature values (e.g., **sentencelength**) of the test data range outside the ones of the training data, resulting all instances being more difficult. We have re-run the training process with the option of preference ranking, with input training data (ranking) being all combinations of the "difficult" and "easy" instances. (About the option *preference ranking*, see the website of SVM$_{light}$ `http://svmlight.joachims.org/`) With this setting, the resulting difficulty value with the test data distributed much like a normal distribution.

Table 3: Summary of the test results

	First session	Second session	Third session
Average correct response rate (stdev.)	0.785 (0.097)	0.741 (0.075)	0.758 (0.079)
highest/lowest	0.980 / 0.627	0.860 / 0.600	0.920 / 0.660
Average total time	0:28:33	0:30:45	0:31:13
longest/shortest	1:10:53 / 0:15:32	0:54:58 / 0:12:41	0:49:51 / 0:11:39

Table 4: Average of the two indices in two groups based on the observed difficulty

	sentence length		predicted difficulty	
average	difficult: 26.14	easy: 24.91	difficult: -1.626	easy: -1.512
variance	difficult: 72.09	easy: 115.89	difficult: 0.326	easy: 0.404
p value	0.6425		0.2183	

Table 5: Correct response rate by part of session

part	1-10	11-20	21-30	31-40	41-50	variance
random1	0.753	0.846	0.741	0.792	0.725	0.00188
random2	0.716	0.799	0.724	0.758	0.696	0.00131
adaptive	0.729	0.724	0.727	0.790	0.788	0.00094

The questions that have been administered to the participants were automatically-generated, hence were not always the errorless ones. Still, the correct response rate of the participants were quite high with 75% on average with the highest being 86-98%.

4.4 Information gain by difficulty prediction

There were 103 questions that were solved by more than three participants. At a first look, disappointingly, there was no significant correlation between predicted difficulty values and the correct response rates. We further took a look at the distribution of the correct response rates, and sampled the “difficult” questions and “easy” questions. We call them *observed difficulty* as opposed to predicted difficulty. There were 44 questions that were answered correctly by all participants who had been administered them. We labeled those questions as “observed easy,” and labeled the questions whose correct response rate was 0.6 or below as “observed difficult.” Table 4 compares the two indices 1) sentence length (a baseline), and 2) the predicted difficulty value, in their relation to the observed difficulty.

The results show that both of the indices differ in the two groups as expected; the average sentence length is larger, and the predicted difficulty value is smaller, in the observed “difficult” group. The p-value is smaller on the predicted difficulty, which means that the predicted difficulty value differentiates the two groups better than the sentence length.

A weak level of significance ($p = 0.2$) is observed on the predicted difficulty, despite

of the difference of the two sets of questions, as well as a rather diverse subject group. The predicted difficulty was calculated on the test results of an English school, whose students are mostly Japanese learners. On the other hand, the participants we gathered included many international students whose first language was different from the others. Also, the difficulty must have reflected the difference in nature of the professionally written TOEIC preparation questions and the automatically-generated questions. The former is free of context and generally very well-written, while the latter tends to be context-dependent, and although it has 10 different patterns, still it gives an impression of being pattern-generated. These differences will be incorporated to further improve the current system. For example, we can incorporate the observed data into the training data to better tune the difficulty prediction.

4.5 Transition of the correct response rate

Finally, we took a look at transitional changes in correct response rate in the three sessions to see the system's adaptivity to the human users. We have split a session into five parts by the order of administration. Table 5 shows the correct response rate calculated on each part along with their variances.

First, it is observed that the correct response rate is more stable in the adaptively administered session. It is generally a good sign for a test, since stability of the correct response rate can be attributed to the system's administrating questions of similar difficulty, rather than moving to and from the extremities. Second, adaptive session was the only one where the performance of the participant rose in the last half of the session[10], which was unexpected, since we were hoping that the correct response rate should fall as the CAT system administeres more suited and thus challenging questions to the user.

The rise of the correct response rate can be attributed to the users' habituation to the patterns, since the adaptive session was the third session for all participants. Also, it is known that a CAT system needs fewer questions than conventional pen-and-paper tests to reach the true values with minimized errors. As Urry reports, only about 20 questions are necessary in English grammar and vocabulary tests for adult native speakers. In our data, the adaptive session was the only one where the correct response rate did not rise on the second split. We speculate that the adaptive administration actually chose more difficult questions to the response to the high correct response rates of the users, achieving the near value after 11 to 20 questions, and then drifted away to the easier questions.

5 Conclusions

We have investigated an application of the machine learning techniques to the problem of difficulty prediction for a CAT system. The SVM classifier shows a performance on par with human judges, and the predicted values show some evidence of efficacy, showing more information gain than sentence length index alone, and stable correct response rates than random administration. Future direction includes more investigation

[10] Whose difference from the other sessions was significant in t-test with ($0.5<p<1.0$)

with the features, such as the use of syllable numbers and other difficulty measures for words. The problem of the difference of subject groups would be alleviated by re-training and updating the classifier as the data from the targeted group is obtained. Combining this prediction with standard IRT procedure also is an interesting avenue towards a more effective assessment system.

References

[1] Baker, F.B., Kim, S.H.: Item Response Theory: Parameter Estimation Techniques, Second Edition (Statistics, a Series of Textbooks and Monographs). Marcel Dekker, Inc., New York, USA (July 2004)

[2] Urry, V.W.: Tailored testing: A successful application of latent trait theory. Journal of Educational Measurement **14**(2) (1977) 181–196

[3] Sumita, E., Sugaya, F., Yamamoto, S.: Measuring non-native speakers' proficiency of english by using a test with automatically-generated fill-in-the-blank questions. In: Proceedings of the Second Workshop on Building Educational Applications Using Natural Language Processing, Ann Arbor, Michigan, U.S., Association for Computational Linguistics (June 2005) 61–68

[4] Liu, C.L., Wang, C.H., Gao, Z.M., Huang, S.M.: Applications of lexical information for algorithmically composing multiple-choice cloze items. In: Proceedings of the Second Workshop on Building Educational Applications Using Natural Language Processing, Ann Arbor, Michigan, U.S., Association for Computational Linguistics (June 2005) 1–8

[5] Brown, J., Frishkoff, G., Eskenazi, M.: Automatic question generation for vocabulary assessment. In: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Vancouver, British Columbia, Canada, Association for Computational Linguistics (October 2005) 819–826

[6] Chen, C.Y., Liou, H.C., Chang, J.S.: Fast: An automatic generation system for grammar tests. In: Proceedings of the COLING/ACL on Interactive presentation sessions, Morristown, NJ, U.S., Association for Computational Linguistics (2006) 1–4

[7] Lee, J., Seneff, S.: Automatic generation of cloze items for prepositions. In: Proceedings INTERSPEECH 2007, Antwerp, Belgium (August 2007) 2173–2176

[8] Kunichika, H., Urushima, M., Hirashima, T., Takeuchi, A.: A computational method of complexity of questions on contents of english sentences and its evaluation. In: ICCE 2002: Proceedings of the International Conference on Computers in Education. (2002) 97–101

[9] Segler, T.M.: Investigating the Selection of Example Sentences for Unknown Target Words in ICALL Reading Texts for L2 German. PhD in Informatics, School of Informatics, University of Edinburgh, Edinburgh, U.K. (2005)

[10] Terada, H., Tanaka-Ishii, K.: Sorting texts by relative readability. In: Proceedings of Empirical Methods on Natural Language Processing (EMNLP) 2008, Honolulu, Hawaii, U.S., Association for Computational Linguistics (October 2008) 127–133

[11] Miyazaki, Y., Norizuki, K.: Developing a computerized readability estimation program with a web-searching function to match text difficulty with individual learners' reading ability. In: Proceedings of WorldCALL 2008, Fukuoka, Japan, CALICO (August 2008) d–111

[12] H.Witten, I., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. The Morgan Kaufmann Series in Data Management Systems, Jim Gray, Series Editor (October 1999)

[13] Coniam, D.: A preliminary inquiry into using corpus word frequency data in the automatic generation of english language cloze tests. CALICO Journal **16**(2-4) (1997) 15–33

[14] Hoshino, A., Huan, L., Nakagawa, H.: A framework for automatic generation of grammar and vocabulary questions. In: Proceedings of WorldCALL 2008, Fukuoka, Japan, WorldCALL (August 2008)

Appendix. Train data and test data

Table 6: Sample questions with different CRR (Correct Response Rates) in (%)

CRR	instance (TOEIC preparation questions)
98.5	If you would like to learn more about [ ] to use this advanced copy machine, simply call the number on the front of the pamphlet and we will send out one of our representatives. a. how b. which c. who d. what
64.1	Workers must [ ] the parcels on to a conveyor belt that carries them to the delivery trucks. a. load b. wrap c. fill d. enter
50.0	Contract negotiations between the union and Pacific Shipping Inc. [ ] in Long Beach after a three-week break. a. have resumed b. has resumed c. is resumed d. resumes
9.7	The purchasing manager is trying to [ ] a deal with the supplier, which could reduce the total cost of materials significantly. a. strike b. discount c. place d. drive

Table 7: Sample questions with predicted difficulty value and CRR in the subject experiment

difficulty	CRR	instance (automatically-generated questions)
-2.95	33.3	He was discharged after the hospital cited no external problems and [ ]. a. negative CT scan results b. results scan negative CT c. results scan negative concussion d. negative concussion scan results
-2.33	0.4	The researchers found for the first time that [ ] pylori reduces the risk of a relapse of stomach cancer by two-thirds. a. removing helicobacter b. removing gastritis c. to remove helicobacter d. to remove gastritis
-1.36	1.0	Koumura later told reporters that Rice's response to his question on North Korea was what Japan [ ] expected. a. had b. has c. was d. were
-1.28	1.0	However, disagreement persisted over which side should act first [ ] the fighting. a. to stop b. stop c. to pull d. pull