# Challenge Balancing for a Kanji E-Tutoring System

Winkels, M.; Roijers, D.M.; van Someren, M.; Yamamoto, E.; Pronk, R.; Odijk, E.; de Jonge, M.

**Publication date**
2018
**Document Version**
Final published version
**Published in**
30th Benelux Conference on Artificial Intelligence

# 30th Benelux Conference on Artificial Intelligence

## BNAIC 2018 Preproceedings

## November 8-9, 2018

## Jheronimus Academy of Data Science (JADS), 's-Hertogenbosch, The Netherlands

**Editors:**
*Martin Atzmueller*
CSAI/JADS, Tilburg University
*Wouter Duivesteijn*
Eindhoven University of Technology

# Challenge Balancing for a Kanji E-Tutoring System

Marysia Winkels [1], Diederik M. Roijers [2], Maarten van Someren [1], Emi Yamamoto [3], Richard Pronk [1], Edwin Odijk [1], and Maarten de Jonge [1]

[1]University of Amsterdam, [2]Vrije Universiteit Amsterdam, [3]Leiden University

**Abstract.** In this paper, we investigate the potential of direct challenge balancing in e-tutoring, especially in domains where there are many skills to acquire. As a case study, we create an e-tutoring system for *kanji*. Our system estimates the perceived challenge level using both the correctness of the answers of the students and implicit feedback, and adapts accordingly. In order to make this estimation we train a classifier on labelled data collected via the same system. We show empirically that the perceived challenge can be estimated well using implicit feedback, and that the adaptive system based on challenge balancing is preferred over a system in which the student selects a difficulty setting, indicating that direct challenge balancing is a promising research direction for e-tutoring.

**Keywords:** Challenge Balancing · E-Tutoring · Machine Learning · Kanji

## 1 Introduction

Students often need to acquire large skill sets. Depending on their mode of studying, this may involve lectures and/or practice sessions. However, the learning process by which students acquire the necessary skills can vary a lot from person to person. For example, one student may be able to acquire a skill by listening to a single lecture, while other students may require prolonged periods of practice. Because of the variance between people, the average curriculum is typically aimed at the average students, which may lead students who need more practice and instruction to drop out, while students who acquire the skills very fast might experience very little challenge and do not reach their full potential. Personalised e-tutoring systems can help mitigate this situation.

E-tutoring systems instruct students by training their skills, e.g., by providing examples, asking questions, and providing feedback. For the above-mentioned reasons, it is highly desirable to adapt the way e-tutoring systems instruct to the individual student. In this paper, we consider the domain of learning kanji — the logographic characters borrowed from Chinese used to write Japanese — as a running example. Kanji is a notoriously large and difficult domain to instruct, and requires a lot of effort from students of Japanese [6]. The amount

of kanji characters ranges in the thousands, with a little over 2000 being in the so-called jōyō kanji list: the list of kanji designated for daily use. Kanji can be complex, as they may consist of smaller components (radicals) and can be used in conjunction with other kanji to form new meanings. However, knowing all the components does not imply being able to read the character, i.e., each kanji has its own associated skill, which is influenced in part by the skills required for the individual components. Because the kanji-domain is so large, e-tutoring systems are often used to supplement, or even instead of, lectures.

Classic e-tutoring systems typically rely on the same strategy applied in classrooms, which is to divide the skill set into separate subsets of similar difficulty and presenting them in order of increasing difficulty. There are two problems with this approach. Firstly, the difficulty of each skill may be hard to quantify, and many metrics may exist that provide some, but not absolute, information about the difficulty. Secondly, a curriculum-based approach does not take the differences between students into account, leading to the same problems as with lecture-based instruction, namely that it may be too hard for some, while being too easy for others. Therefore, a mismatch in difficulty may result in students abandoning learning out of frustration. However, domains like kanji, in which students often indicate that they quickly forget studied material [2], require a lot of repetition in order to learn the skills. Engagement is therefore critical to the success of the e-tutoring system, and selecting the appropriate challenge level is an important prerequisite for student engagement.

One approach to attaining the appropriate challenge level, is using competence models. A competence model includes parameters for the difficulty of the skills to acquire, as well as the current level w.r.t. each skill for each student. Furthermore, in an e-tutoring system, there also need to be parameters to model student learning. After estimating the parameters of a competence model for a domain, an e-tutoring system can choose an appropriate next learning object (such as a question or example) for each student individually. A major limitation of competence models however, is that they tend to be highly data-intensive [7]. Even in small domains, the amount of data required to estimate the parameters with the required accuracy for an e-tutoring system, is often more than available [8].

In this paper, we exploit the key observation, that it is often not necessary to explicitly model competence, in order to provide the appropriate challenge to students in an e-tutoring system. Instead, by creating a mapping from implicit user feedback to experienced challenge level, the e-tutoring system can personalise the experience. This insight is used in the field of digital gaming in order to maximise engagement of players [1]. By creating a kanji e-tutoring system that uses implicit user feedback to tailor to the individual user, we attempt to provide a more engaging system. Thereby we aim to provide a system that is more comfortable to use for an extensive period of time. Our approach, inspired by Bakkes et al.'s approach for challenge balancing in gaming, is based on using both the properties of the kanji [4] (i.e., skills) to learn and reuses a

variety of known difficulty metrics for kanji, as well as implicit feedback from the user — such as the time spent per question and whether hints were used — in order to directly estimate the perceived challenge of the student. In order to be able to estimate this perceived challenge we collected labelled data using a five-point challenge scale, and learned a classification model. We show that using this implicit feedback is key to estimating the perceived challenge. We use the classification model to personalise the kanji tutor. We indicate that this approach fulfills the requirements of a proper kanji trainer outlined in [5]. We show empirically that this personalisation is preferred by students over a system in which they can select a fixed difficulty setting. Our approach is relatively simple, exploits existing metrics and is data-efficient. We therefore conclude that challenge balancing (rather than competence-difficulty balancing) is a promising direction for e-tutoring systems.

## 2 Method

Our aim is to create a system that estimates the perceived challenge level and personalises accordingly in order to improve engagement and minimise frustration. We aim to providing a reasonable challenge for the user at all times. Therefore, we need to be able to estimate this perceived challenge level. We train a classifier for perceived challenge in the offline learning phase. Using this classifier, the system can adapt the challenge level accordingly in the online adaptation phase, corresponding to the final system.

We first create the learning environment with multiple-choice questions. The number of possible choices varies. A 'hint' option is available for each question. mmClicking 'hint' results in a breakdown of the question in the kanji's radicals with their meanings, as taken from Alan R. Miller's personal homepage at New Mexico Tech[1]. Additionally, for 188 kanji, an illustration can be provided (an example of which is given in Figure 1) that can be helpful for the user to determine the meaning of the kanji in question. These images are taken, as permitted, from the Japan Foundation London[2]. The database of kanji and their associated meanings is taken from the `kanjidic2` project[3]. Upon first use of the system, users are asked to indicate what their current level regarding kanji: "beginner", "intermediate" or "expert".

Fig. 1: The hint provided for a question on '山' (mountain).

The system uses short series of five questions which we call a *chunk*. Each chunk is generated using a difficulty vector $\boldsymbol{v}$, which we define later using the infor-

[1] `http://infohost.nmt.edu/~armiller/japanese/kanjiradical.htm`
[2] `http://www.jpf.org.uk/language/kanjifiles/kanjicard.html`
[3] `http://www.edrdg.org/kanjidic/kanjd2index.html`

mation in Table 1. The adaptation takes place by changing $\boldsymbol{v}$. For each chunk the system observes total number of correctly answered questions and implicit feedback (Table 2) to estimate the challenge level the chunk posed to the user. The challenge level is then changed accordingly. In the offline phase, each chunk is followed by a request for feedback regarding the challenge level in a five-point scale. In the online phase, the system estimates the challenge level automatically and adapts the difficulty, $\boldsymbol{v}$, accordingly in order to provide the appropriate challenge level for each user.

### 2.1 Difficulty Parameters

Questions in a chunk are assured to be of a similar challenge level because they are generated by the same difficulty vector $\boldsymbol{v}$. The parameters, i.e., elements of $\boldsymbol{v}$, and the possible values they can take are detailed in Table 1. There are 1440 possible combinations of the parameter values, 1404 of are suited to generate (at least) five questions.

*Grade* An ordering of kanji and associated readings that should be learned per school year was created by the Japanese Ministry of Education. The lower the grade, the more simplistic the kanji and associated reading is.

*Multiple Choice options* The probability of giving a correct answer is in general higher for a question with fewer alternatives.

| **Parameter** | **Range** |
|---|---|
| Grade | 1–5 |
| MC options | 2–5 |
| Minimum stroke count | 1–9 |
| Answer similarity | 0.0, .5, 1.0 |
| Reversed Questions | True/False |
| Question Type | Kanji/Vocab |

Table 1: Parameters and their ranges

*Minimum stroke count* Kanji with a lower number of strokes are more recognisable and easily remembered than those with a high number of strokes. To avoid a large range of stroke count within one chunk, a maximum stroke count is determined based on the minimum stroke count[4].

*Answer similarity* The higher the similarity between the correct answer and the alternatives, the more difficult the question. Similarity between questions was measured in terms of shared radicals, difference in stroke count, overlap in meaning and readings.

*Reversed question* Boolean parameter to indicate whether kanji or the meaning was given. Either a kanji is presented, and the user is required to answer the associated meaning, or the other way around.

*Question type* The parameter question type had two possible values: *kanji* and *vocab*. The distinction is that the system will either only give a single kanji or a vocabulary word which includes one of the kanji.

[4] Minimum stroke count +3

We make the assumption that combinations of parameter settings, i.e., possible values of $\boldsymbol{v}$, can be ranked in terms of difficulty and that this ranking is user-independent (objective), i.e., if domain experts (or sufficiently competent users) would be asked to compare the difficulty of two questions, they would be highly likely to give the same answer. This is a common assumption in competence modelling. Note that: a) such rankings would be relative, and would not provide any information of how much more difficult the questions would be, and b) there is no one-on-one correspondence between the (difficulty of the) skill and the difficulty of the question, i.e., it is possible to ask an easy question about a difficult skill. Therefore, a ranking provides only limited information about the difficulty of the skills and questions. We ranked the 1404 options approximately, using expert opinion, on the basis of a simple formula.

The difficulty of a question is different from a user's competence level. In the competence model, the competence of the user for a given skill is on the same scale as the difficulty of that skill. The probability of a correct answer and the perceived challenge depend on the difference between the difficulty and the competence. Competence-based e-tutoring systems (e.g. [8]), estimate both the competence in a skill for a user and the difficulty of a skill. This is often a prohibitively data-intensive approach, especially when there are many skills to learn [7] (as in the kanji domain). A key insight underlying this paper is that we do not need to model difficulty and competence explicitly, but that we can estimate the perceived challenge level directly, corresponding to estimating the difference between competence and difficulty directly. We show experimentally that this is possible, and leads to a relatively high accuracy, even with a limited amount of data. Furthermore, because it does not require modelling difficulty explicitly, our ranking suffices.

### 2.2 Offline learning Phase

The offline learning stage is aimed at creating a classifier that maps observations (Table 2) to the perceived challenge level. In order to gather data for this learning task, the system is set up to provide a user with a chunk generated by a single difficulty vector $\boldsymbol{v}$ (and therefore approximately the same level). The entries in $\boldsymbol{v}$ correspond to the parameters described in Section 2.1. The initial $\boldsymbol{v}$ depends the users own estimation of her level of expertise when starting the system, to prevent presenting questions relatively far from the user's actual level of expertise.

For each chunk, the system records *implicit* and *explicit feedback* that can be observed from user interaction. This feedback — described in Section 2.2 — is used as sole input to the classifier.

After a chunk of questions is completed, the user is asked to indicate the challenge level posed by the just completed chunk on a five-point scale: "way too easy" (1), "easy" (2), "just right" (3), "hard" (4), and "way too hard" (5). These perceived challenge level labels are the class labels for the classifier. In the online adaptation phase, the system will aim for "just right" as the perceived challenge level.

| **Feature** | **Explanation** |
|---|---|
| Correctness | The count of questions that were answered correctly of the set of five questions. |
| Mean duration | The average time spent per question in the chunk. |
| Std duration | The standard deviation of the mean duration. |
| Hint requested | The count of question that the user requested a 'hint' for. |
| Mean hint time | The arithmetic mean of the amount of time it took per question to request a hint (if a hint was indeed requested). |
| Std hint time | The standard deviation of mean hint time. |
| Mean remaining time | The average time it took for the user to answer a question after the hint was requested. |

Table 2: Features used in the offline learning task

In total, 581 chunks of questions were completed and labelled by various participants. For each chunk, the observations (as described in Table 2) and indicated challenge level are stored. We train a model that maps the *implicit feedback* to the perceived challenge level, using 3-fold cross-validation. In order to prove the importance of the implicit feedback — which is a key element of our proposed method — we also train a model based solely on correctness of the answers and compared to a model that uses the implicit feedback as described in Table 2.

We train different classifiers including random forest, support vector machine, logistic regression and neural networks [3]. The selection of these techniques is based on the assumption that random forests and neural networks outperform linear function-based algorithms, as the former are particularly well-suited to model data with complex interactions and dependencies between features. Intuitively, this seems applicable to our set of features; we hypothesised, for example, that a short duration of the chunk with a high correct answer rate indicates perceived simplicity, whereas a similar observation of duration with a low correct answer rate suggests the current challenge level of the questions is too high. However, logistic regression is regularly used in intelligent tutoring systems and support vector machines have often proven themselves useful, so both are considered as well.

### 2.3 Online Adaptation Phase

During the offline learning stage, data is collected by offering participants a simple system which presented them with questions and allowing them to indicate the experienced challenge level. This system is expanded upon to include online adaptation; instead of requesting the user to label the experienced challenge of a chunk, this is predicted using the observations made and the model created in the offline learning stage.

Based on this prediction, the new parameter combination is determined to generate the next set of questions. The estimated experienced challenge level indicates how similar the current user level is to the previously presented parameter combination, and the new parameter combination is determined accordingly. The outer values (1 and 5) cause a large jump up or down the parameter ranking ($40-60$ positions), the closer values (2 and 4) cause a small jump ($5-15$ positions) and a classification of 'just right' (3) results in the next chunk being very similar to the previously seen one (a jump of only $0-1$ positions). In order to ensure that the parameters for the first generated chunk are relatively close to the skill level of the user, the user is asked to indicate their expertise w.r.t. kanji (beginner, intermediate, expert) at the first use of the system.

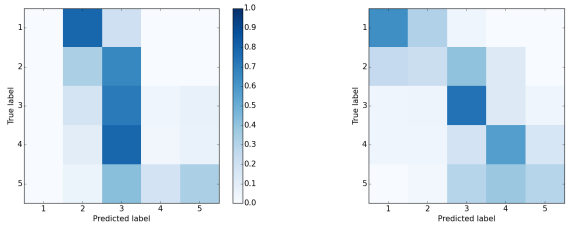
The ranking of parameter combinations is created by consulting expert opinion. The most significant factor in determining the challenge level is judged to be the grade from which the kanji are sampled. In order of most influence on objective question difficulty to least are grade, multiple choice options, question type, stroke count, and reversed question. All 1404 combinations of parameters that are capable of generating five questions are ranked according to this principle.

## 3 Experiments

In this section we discuss the experimental results. First, we perform the machine learning task of learning a classifier, and demonstrate clearly that using implicit feedback is necessary to come to an accurate estimation of challenge level. Secondly, in order to test whether automatic challenge balancing in a kanji e-tutoring system is preferable over a non-adaptive system, we perform an A/B test comparing the adaptive system against a baseline where the user can set her own challenge level once, when entering the system.

### 3.1 Offline learning task

To be able to train a challenge level classifier we collected 581 data points (i.e., chunks). Each data point contained the information as specified in Table 2 and a class label provided by the user, as specified in Section 2.2.




(a) Without implicit feedback

(b) With implicit feedback

Fig. 2: Confusion matrix random forest

To determine which type of model was best, we used 3-fold cross-validation as a model selection technique. Of the different possible models of Section 2.2, the random forest classifier achieved the highest accuracy of 0.526. Therefore, we selected this model.

In order to test whether implicit feedback provides extra information besides the correctness of the answers of the users, we trained two classifiers on the same data: a classifier using only the correctness of the answers, and one using all the information of Table 2. We compare the confusion matrices. Note that the correct classifications are on the diagonal. Figure 2a shows the confusion matrix of a decision tree's ability to correctly estimate a user's skill level based on correctness of their answers alone, and Figure 2b shows the confusion matrix of the decision tree's ability to correctly estimate the user's skill level based on all the information of Table 2. In Figure 2a we observe that the classifier without the implicit feedback tends to misclassify the more extreme classes ("way too easy" or "way too hard"), and classifies too many data points as "just right". This is a clear indication of lack of information. On the other hand, when we use all available feedback, the shape of the confusion matrix is much more diagonal.

The total accuracy of the classifier using implicit feedback is 0.526. Although an accuracy of 0.526 is not very strong in itself, it should be noted the type of errors are much more favourable than for the classifier that does not use implicit feedback. As can be read from the confusion matrix, large misclassifications (such as classification of a 1 as a 5) hardly ever occur, making the model usable for our predictive purposes. We therefore conclude that using implicit feedback is essential for correct perceived challenge level classification.

The classifier we learned in the offline phase — using implicit feedback — is used as input for the online adaptation phase. In this phase the classifications made by the model are used as input to adapt the chunks to an appropriate challenge level for the individual user.

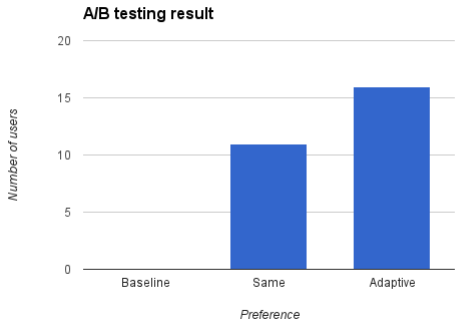



Fig. 3: A/B test result: adaptive system vs. non-adaptive system

### 3.2 Online A/B testing

To test whether automatic challenge balancing in a kanji e-tutoring system is preferable over a non-adaptive system, we perform an A/B test comparing the adaptive system against the baseline. Test subjects were presented with both systems in a random order, with 30 questions each (6 chunks). The participant was then asked to indicate which, if any, system they preferred.

The presented challenge balancing system was as described in section 2.3. The non-adaptive system was created to take the user's initial chosen level of expertise (beginner / intermediate / expert), determine an associated rank in the fixed ranking, and randomly sample around that rank using a Gaussian distribution. A standard deviation was decided upon to ensure the user would be presented with multiple types of questions and would not prefer the adaptive solely on a higher rate of question type variation. With our ranking of questions of 1404 possibilities, a standard deviation of 20 was found to be just enough to provide the user with variation while not deviating too far from the initial rank.

In total, 27 users participated in the A/B test. Figure 3 shows the results of this test. Our results show that 11 users noticed no difference between adaptive and non-adaptive system, and 16 preferred the adaptive system. Interestingly, none of the users reported a preference for the non-adaptive system.

Further feedback from users who were questioned afterwards (10 users) included comments where users reported preferring the adaptive system as it seemed to better match their skill level (as intended), but two people voiced a preference because one system provided them with different kinds of questions. The latter was not as intended, and we had attempted to prevent this by sampling using a standard deviation in the non-adaptive system; this observation was caused due to random effects.

Because our system was never deemed worse, and preferred by a majority of the participants, we conclude that adaptation via direct perceived challenge balancing is viable way to personalise e-tutoring systems in the kanji domain.

## 4 Conclusion

In this paper, we investigated whether challenge balancing offers an alternative to competence-challenge balancing in e-tutoring systems. With 58% of users preferring the adaptive system over the non-adaptive against 42% having no preference, we have shown empirically that an adaptive challenge balancing system is viable. We have shown that using implicit feedback in order to estimate the challenge level is key challenge level estimation. Because of the relative simplicity of our method, and the favourable results, even in a domain with a very large skill set, we conclude that challenge balancing, using implicit feedback as input, is a promising direction for e-tutoring systems that merits further investigation.

In future work, we would like to test on a wide range of domains that have large skill sets to learn, and try to attain a higher accuracy by adding more features. We also note that in this version of our system, we used a simple ranking, based on expert opinion. This takes effort (though in this case minimal effort), is domain dependent, and might be inaccurate. In future research we aim to create a method that automatically learns the ranking from any $\boldsymbol{v}$, (containing the basic available features that might concern difficulty) for any domain via implicit feedback and perceived challenge level labels.

## References

1. Bakkes, S., Whiteson, S., Li, G., Visniuc, G.V., Charitos, E., Heijne, N., Swellengrebel, A.: Challenge balancing for personalised game spaces. In: IEEE GEM. pp. 1–8 (2014)
2. Banno, E., Ikeda, Y.: 非漢字圏学習者の漢字学習意識とストラテジー使用 (Students' attitudes and strategies toward learning kanji: A survey of learners from non-kanji background ). Japan International Student Education **14**, 13–21 (2009)
3. Bishop, C.M.: Pattern recognition and machine learning (2006)
4. Chen, H.C., Hsu, C.C., Chang, L.Y., Lin, Y.C., Chang, K.E., Sung, Y.T.: Using a radical-derived character e-learning platform to increase knowledge of chinese characters (2013)
5. Komori, S., Zimmerman, E.: A critique of web-based kanji learning programs for autonomous learners: Suggestions for improvement of wwkanji. Computer Assisted Language Learning **14**(1), 43–67 (2001). https://doi.org/10.1076/call.14.1.43.5786, `http://dx.doi.org/10.1076/call.14.1.43.5786`
6. Ōkita, Y.: 漢字学習ストラテジーと学生の漢字学習に対する信念 (Kanji learning strategies and student beliefs on kanji learning). Japanese Education in the World **5**, 105–124 (1995)
7. Rekker, L.: Parameter estimation in competence models (2014), university of Amsterdam
8. Roijers, D.M., Jeuring, J., Feelders, A.: Probability estimation and a competence model for rule based e-tutoring systems. In: LAK'12. pp. 255–258 (2012)