Wong, L.-H. et al. (Eds.) (2013). Proceedings of the 21st International Conference on Computers in Education. Indonesia: Asia-Pacific Society for Computers in Education

Calculating Test Item Similarity Using Latent Dirichlet Allocation

Teruhiko TAKAGI[a, b*], Masanori TAKAGI[c], Yoshimi TESHIGAWARA[d] & Kenji TANAKA[a]
[a] *Graduate School of Information Systems, University of Electro-Communications, Japan*
[b] *Research Fellow of Japan Society for the Promotion of Science, Japan*
[c] *Faculty of Software and Information Science, Iwate Prefectural University, Japan*
[d] *School of Science and Technology for Future Life, Tokyo Denki University, Japan*
*takagi@tanaka.is.uec.ac.jp

Abstract: In previous studies, we proposed methods for calculating similarity between test items to automatically retrieve similar test items in e-testing, and conducted experiments and evaluations of those methods. Test item similarity data is applicable to tasks such as automatically retrieving similar test items, automatically constructing item banks, visualizing structure between test items, optimizing amounts of test information, estimating the difficulty of unanswered test items, conducting computer adaptive testing, and creating test items. To improve the accuracy of retrieving similar test items, we propose a new method for calculating test item similarity that applies latent Dirichlet allocation (LDA), a generative probabilistic document model. We assume that each test item is represented by a vector using topics estimated by LDA, and the similarity between test items is calculated by cosine similarity. Applying LDA to calculate similarity between test items lowers the number of retrieved dissimilar test items, and creates vectors based on the relation between extracted terms. To accurately estimate topics in each test item, we perform preprocessing by identifying where important terms occur and enhancing the co-occurrence relation between terms. We use 250 test items from the Systems Administrator Examination to test the effectiveness of retrieving similar test items. The results indicate the effectiveness of the preprocessing steps, and of applying LDA to calculating test item similarity. We furthermore demonstrate the improvement in accuracy of retrieving similar test items by the proposed method in comparison with existing methods.

Keywords: e-testing, item bank, similar test item, LDA, similarity

1. Introduction

The prevalence of online testing has increased in recent years, resulting in a need for large item banks (Ueno, 2005; Ueno & Okamoto, 2008). Test items in item banks are often hierarchically classified, based on the knowledge they test. In the integrative e-testing system developed by Songmuang and Ueno (2008), for example, test items are classified according to subject, and according to broad and midrange scope, then classified into a multi-hierarchy by class subject according to the System of Intelligent Evaluation Using Tests for TeleEducation developed by Guzman and Conejo (2005). These systems are implemented using a computer adaptive testing (CAT)-based system or an automated test construction system. Test items in these systems are metadata such as correct response rate and item response theory parameters (Hambleton, Swaminathan, & Rogers, 1991). Learning management systems such as Moodle (Dougiamas, 1999) or Blackboard (Blackboard Inc., 1997) also hierarchically manage test items.

For fields in which knowledge is disorganized, however, more test items mean increased difficulty in appropriate classification. Appropriate classification may also require extensive familiarity with the knowledge of subject field or area. To address these problems, we have proposed a method of calculating similarity between test items to allow for automatic retrieval of similar test items (Takagi et al., 2009). In this study, we examine the calculation of similarity between multiple-choice items. We say that two test items are "similar" when they test the same knowledge (i.e., when the knowledge needed to provide the correct answer is similar for each question). This knowledge

includes areas such as sector-specific concepts, laws, figures, and history (targeted knowledge).

Test item similarity data is applicable to the following types of tasks:

(1) Automatically retrieving similar test items
(2) Automatically constructing item banks using clustering techniques (Manning & Schutze, 1999)
(3) Visualizing structure between test items using multidimensional scaling (Young & Hamer, 1987)
(4) Optimizing the amount of test information (Hambleton et al., 1991) when constructing tests
(5) Estimating the difficulty level of unanswered test items (Ikeda, Takagi, Takagi, & Teshigawara, 2012)
(6) Performing CAT for iterative learning (Ikeda et al., 2012)
(7) Creating test items in consideration of difficulty level

In previous studies, we proposed a procedure for automatically identifying parts (question, correct choice, or incorrect choice) where the targeted knowledge occurs based on the results of analyzing the type of test item and the features of terms which occur in the question or correct choice using natural language processing (Takagi, Takagi, & Teshigawara, 2009). Single nouns and compound nouns are extracted from the part identified by this procedure, and the similarity between test items is calculated based on the vector space model (Manning & Schutze, 1999). We also targeted test items created in computer networking classes, and experimentally retrieved similar test items. This experiment showed improved accuracy with our method in comparison with existing methods, and the effectiveness of automatically identifying item parts where targeted knowledge occurred.

However, this experiment also showed that noise such as superfluous terms and spelling variations in the extracted terms results in the retrieval of dissimilar test items. In addition, representing test items as vectors by only extracted terms is limiting, indicating a need to consider the relation between terms. To address these problems and improve the accuracy of retrieving similar test items, here we propose a new method of calculating similarity between test items using latent Dirichlet allocation (LDA) (Blei, Ng, & Jordan, 2003), a generative probabilistic document model. LDA estimates single or multiple topics that represent test item content based on the co-occurrence of terms in the test item. Fundamentally, we assume that each test item is represented by a vector using topics estimated by LDA, and the similarity between test items is calculated by cosine similarity (Manning & Schutze, 1999). Doing so provides two advantages:

(1) Since there are fewer topics than extracted terms, the number of vector dimensions, and thus the number of retrieved dissimilar test items, can be decreased.
(2) Since topics are probabilistically estimated based on the co-occurrence of extracted terms, topics can be estimated using key terms that are orthographically different yet semantically similar.

2. Latent Dirichlet Allocation

In the topic model, documents are described as a distribution of topics and each topic is described as a distribution of words. Hofmann (1999) proposed probabilistic latent semantic indexing (PLSI) in a pioneering study of topic modeling. LDA extends PLSI, and is a generative probabilistic document model where the multinomial distribution of each topic $Mult(\boldsymbol{\theta})$ is assumed to follow the Dirichlet distribution $Dir(\boldsymbol{\theta}|\boldsymbol{\alpha})$, which is the prior distribution conjugate to the multinomial distribution. LDA thus overcomes the overfitting problem, which can prevent generation of new documents for PLSI.

Figure 1 shows the LDA model represented as a probabilistic graphical model, denoting dependency among random variables or parameters as a directed graph. In the figure, the black circle

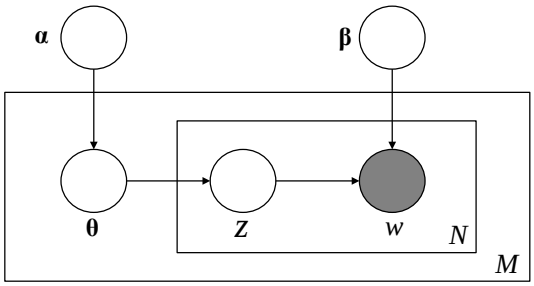

Figure 1. Graphical Model Representation of LDA.

indicates an observed variable and others circles indicate latent variables and unknown parameters. Rectangular areas indicate repeated sampling, with the number in the lower right corner indicating the number of repetitions. Here, N is the total number of words and M is the total number of documents. We next describe the generative process for a document corresponding to the graphical representation of LDA in Figure 1. Set of documents D is generated by repeating the following process M times:

1. Sample N words from a document.
2. Sample the generative probability of each topic $\boldsymbol{\theta}$ from the Dirichlet prior distribution $Dir(\boldsymbol{\theta}|\boldsymbol{\alpha})$.
3. For each of the N words w_n,
   (a) Sample a topic z_n from the multinomial distribution $Mult(\boldsymbol{\theta})$.
   (b) Sample the word w_n from the multinomial probability, conditioned by the topic z_n $p(w_n|z_n,\boldsymbol{\beta})$.

In LDA, the value of latent variable z probabilistically varies based on $\boldsymbol{\theta}$, and multiple topics are generated from a document. Since the model parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are located outside the rectangular area in Figure 1, LDA can generate unseen documents, too. It is assumed that the dimension k of the Dirichlet distribution (the dimension of the topic variable z) is known and fixed. In addition, the generative provability of a word to a topic is indicated by $\boldsymbol{\beta}$, which is a $k \times V$ matrix represented by $p(w^j=1|z^i=1)=\boldsymbol{\beta}_{ij}$. Given the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, the joint distribution is given of a topic mixture $\boldsymbol{\theta}$, a set of N topics $\mathbf{z}$, and a set of N words $\mathbf{w}$ as follows:

$$p(\boldsymbol{\theta}, \mathbf{z}, \mathbf{w} \mid \boldsymbol{\alpha}, \boldsymbol{\beta}) = p(\boldsymbol{\theta} \mid \boldsymbol{\alpha}) \prod_{n=1}^{N} p(z_n \mid \boldsymbol{\theta})\, p(w_n \mid z_n, \boldsymbol{\beta}) . \quad (1)$$

Integrating over $\boldsymbol{\theta}$ and summing over $\mathbf{z}$, we obtain the marginal distribution of a document as follows:

$$p(\mathbf{w} \mid \boldsymbol{\alpha}, \boldsymbol{\beta}) = \int p(\boldsymbol{\theta} \mid \boldsymbol{\alpha}) \left(\prod_{n=1}^{N} \sum_{z_n} p(z_n \mid \boldsymbol{\theta})\, p(w_n \mid z_n, \boldsymbol{\beta}) \right) d\boldsymbol{\theta} . \quad (2)$$

Here, the parameter $\boldsymbol{\alpha}$ is a k-vector with components $\boldsymbol{\alpha}_i>0$, and the parameter of the Dirichlet distribution is as follows:

$$p(\boldsymbol{\theta} \mid \boldsymbol{\alpha}) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \theta_1^{\alpha_1 - 1} \cdots \theta_k^{\alpha_k - 1}, \quad (3)$$

where $\Gamma()$ is the gamma function. The model parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are generally learned using an approximation based on variational Bayesian inference (Blei et al., 2003) or a Markov chain Monte Carlo (MCMC) method (Griffiths & Steyvers, 2004). Teh, Newman, and Welling (2007) proposed collapsed variational Bayesian inference, which improves the inference accuracy. Gibbs sampling (Griffiths et al., 2004) is another kind of MCMC method applied to learning parameters. While LDA requires specification of the number of topics, nonparametric Bayesian methods such as the hierarchical Dirichlet process do not (Teh, Jordan, Beal, & Blei, 2006). This study uses variational Bayesian inference to learn $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$.

In recent years, LDA has been widely applied in fields such as information retrieval and document clustering (Chemudugunta, Smyth, & Steyvers, 2007; Cao, Li, Zhang, & Tang, 2007; Wang, Zhang, & Zhang, 2008; Iwata, Yamada, & Ueda, 2008). To our knowledge, no research has targeted test items, so it is unclear whether these methods can be applied. Unlike the documents examined in the studies mentioned above, test items consist of questions with correct and incorrect choices, and sentences or terms differ by form. Therefore, in this study, we propose a method of applying LDA to capture test item features.

3. Previous Study

In this chapter, we describe a procedure for automatically identifying parts where the targeted knowledge occurs, and a procedure for calculating similarity between test items, both of which were proposed in our previous study (Takagi et al., 2009).

3.1 Automatically Identifying the Knowledge Occurrence Part

Figure 2 shows a procedure for automatically identifying the knowledge occurrence part. The five term

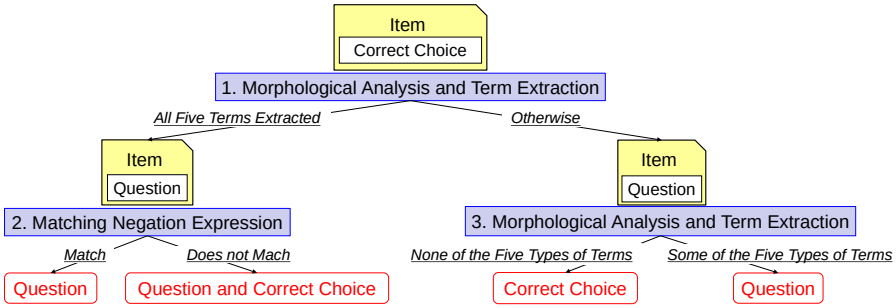

Figure 2. Procedure for Automatically Identifying the Knowledge Occurrence Part.

Table 1: Targeted knowledge types and examples.

Targeted Knowledge Type		Example
Single Noun	Japanese	*ru-ta* (router)
	English	Telnet, UDP
Compound Noun	Japanese	*denshiteki komyunike-shon* (electronic communication)
	English	TCP IP, B-to-B
	Japanese and English	OSI *sanshou moderu* (OSI reference model)

types in Figure 2 are types of targeted knowledge, which can be classified into single nouns and compound nouns. A single noun is a noun that cannot be further divided into shorter, more basic nouns. Targeted knowledge can be further classified into five types of terms, according to whether they are Japanese, English, or a combination of Japanese and English (Takagi et al., 2009). Table 1 shows the targeted knowledge types and examples. For morphological analysis we use the Chasen software package developed by the Nara Institute of Science and Technology (Matsumoto, 2000), the most popular and widely used Japanese morphological analysis program. To extract the five types of terms, we use the TermExtract Perl module developed by Nakagawa, Maeda, and Kojima (2003a), which is used as the "automatic technical term extracting system".

First, the correct choice is morphologically analyzed and the five types of terms are extracted from it ("1" in Figure 2). If the number of extracted terms is one and there are no other terms (e.g., particles or auxiliary verbs), the question is analyzed as to whether the sentences in question matches a negation expression, such as "*machigatta* (wrong)" or "*ayamatteiru* (incorrect)" ("2" in Figure 2). If so, the knowledge occurrence part is identified as the question. If these sentences do not match the negation expression, the knowledge occurrence part is identified as both the question and the correct choice. On the other hand, if multiple terms are extracted from the correct choice, or if there are other terms ("1" in Figure 2), the question is morphologically analyzed and the five types of terms are extracted from the question ("3" in Figure 2). If there are no extracted terms, the knowledge occurrence part is identified as the correct choice. If there is more than one extracted term, the knowledge occurrence part is identified as the question.

3.2 Calculating Test Item Similarity

Figure 3 shows the procedure of calculating test item similarity based on automatically identifying the knowledge occurrence part. First, the knowledge occurrence part is automatically identified by the procedure as described in the previous section ("1" in Figure 3). There are three parts where the targeted knowledge occurs. Second, sentences and their parts are morphologically analyzed, and the five types of terms shown in Table 1 are extracted ("2" in Figure 3). Third, stop words are eliminated from the extracted terms ("3" in Figure 3). Examples of stop words are the broader term and the term occurring in

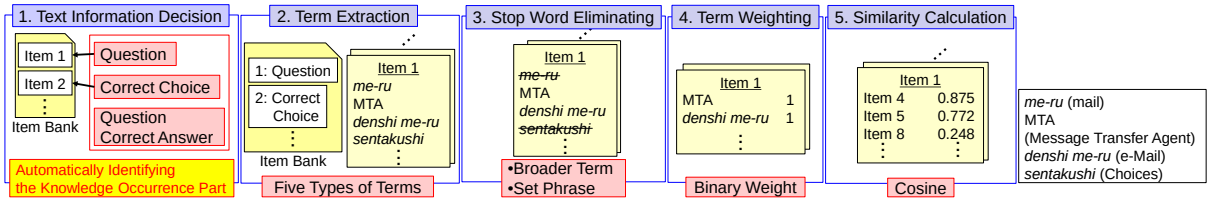

Figure 3. Procedure for Calculating Test Item Similarity.

a set phrase used in the question. In case of "3" in Figure 3 "*me-ru* (mail)" is a broader term against "*sentakushi* (choices)" and "*ika* (the following)") are considered unrelated to test item content. Fourth, the remaining terms in each test item are binary weighted (Manning & Schutze, 1999), adding weight 1 to all terms ("4" in Figure 3). Finally, the test item similarity represented as a vector featured by the term weights is calculated by cosine similarity ("5" in Figure 3).

Here, if the weight or binary variable to term i in documents d_x and d_y are x_i and y_i, respectively, the total number of terms i is T, and the measure of similarity (d_x, d_y) between these documents by cosine similarity is as follows:

$$\sigma(d_x, d_y) = \frac{\sum_{i=1}^{T} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{T} x_i^2 \times \sum_{i=1}^{T} y_i^2}} \cdot \quad (4)$$

4. Issues to be Considered and Solving Approach

We assume that each test item is represented by a vector using topics estimated by LDA. When doing so, each test item is represented by a co-occurrence matrix of terms. Thus, the accuracy of the topic estimation depends on the co-occurrence relation between terms in each test item. For example, LDA classified the following two test items as being on the same topic, despite their having no shared terms and being related to very different topics. The targeted knowledge of Test Item 1 is "copyrighted works (*chosakubutsu*)". The targeted knowledge of Test Item 2 is "swapping (*suwappingu*)". However, since the terms "copyrighted works" (Test Item 1) and "program (*puroguramu*)" (Test Item 2) co-occurred in other related test items, the co-occurrence between these terms is increased, and the item was estimated as belonging to the same topic.

Test Item 1

How long is the term of protection for personal "copyrighted works" after the author dies?
(1) 25 years (2) 50 years (3) 75 years (4) 100 years

Test Item 2

A user installed and ran a new "program" on a server with virtual memory storage, resulting in "swapping" and lowered processing efficiency of previously installed programs. Which of the following is an appropriate solution to this problem?
(1) Upgrading the CPU (2) Adding a magnetic disk device to expand auxiliary storage
(3) Upgrading to faster main memory (4) Increasing the amount of main memory

To prevent such misidentification of topics, the co-occurrence relation between the targeted knowledge and terms relating to the targeted knowledge should be considered in LDA. However, unlike general documents, test items consist of questions with correct and incorrect choices, and the targeted knowledge will not necessarily occur in the same location in each form. We therefore take the two following preprocessing steps before using LDA:

(a) Identifying the part (question, correct choice, or incorrect choice) where the targeted knowledge occurs
(b) Enhancing the co-occurrence relation between terms occurring in the part identified in preprocessing step (a)

To perform preprocessing step (a), we automatically identify the knowledge occurrence part as described in section 3.1. Doing so helps to decrease the number of terms unrelated to the targeted knowledge. Here, we focus on single nouns constituting a compound noun. The meanings of these single nouns often contain the meaning of the compound noun (e.g., "*denshi me-ru* (e-mail)", composed of "*denshi* (electronic)" and "*me-ru* (mail)", and "TCP *purotokoru* (TCP protocol)", composed of "TCP" and "*purotokoru* (protocol)"). Thus, a compound noun is created from semantically related single nouns, and there is a semantic relation among these single nouns. Therefore, to realize preprocessing step (b), we create a co-occurrence relation among single nouns, compound nouns, and the single nouns constituting compound nouns that occur in the part identified in preprocessing step (a). For example, when "*jiki disuku souchi* (magnetic disk device)" and "*konpyu-ta* (computer)" both occurred in a part, we extract "*jiki disuku souchi* (magnetic disk device)", "*jiki* (magnet)", "*disuku* (disk)", "*souchi* (device)", and "*konpyu-ta* (computer)". Extracting co-occurrences in this way allows for the correct evaluation of items, even when terms related to the targeted knowledge are written differently (e.g., "*jiki disuku souchi* (magnetic disk device)" and "*jiki disuku* (magnetic disk)").

5. Calculating Test Item Similarity Using LDA

In this section, we propose a method for calculating test item similarity using LDA, based on the preprocessing described in the previous section. In section 5.1, we describe the procedure of the proposed method. In section 5.2, we describe a method for managing test items using test item similarity.

5.1 Calculating Test Item Similarity Using LDA

Figure 4 shows the procedure of calculating test item similarity using LDA. First, the part where the targeted knowledge occurs is automatically identified by the procedure shown in Figure 2 ("1" in Figure 4). There are three parts where the targeted knowledge occurs. Second, sentences in their parts are morphologically analyzed, and single nouns, compound nouns, and single nouns constituting the compound nouns are extracted ("2" in Figure 4). Third, terms commonly occurring in set phrases used to pose questions are eliminated from the extracted terms ("3" in Figure 4), as described in section 3.2.

Fourth, co-occurrence matrices are created based on the extracted terms for each test item, and the topic of each test item is estimated based on LDA ("4" in Figure 4). The model parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are learned based on variational Bayesian inference. At this stage, we also give the number of similar test item groups as the initial number of topics k. Fifth, estimated topics are binary weighted ("5" in Figure 4), adding weight 1 to all topics. Finally, the similarity between test items represented as a vector featured by the topic weight is calculated by cosine as in equation (4) ("6" in Figure 4).

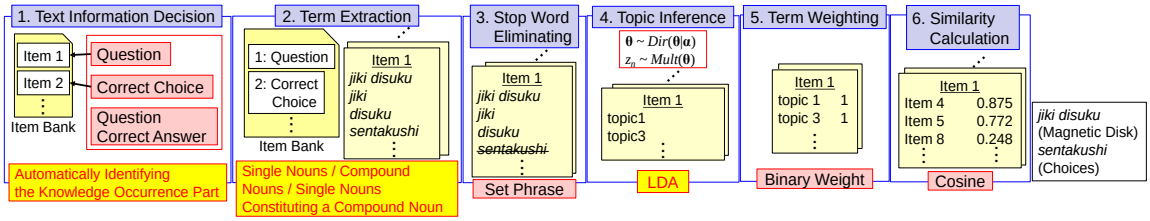

Figure 4. Procedure for Calculating Test Item Similarity Using LDA.

5.2 Automatically Classifying Test Items

Figure 5 shows the assumed procedure of automatically classifying test items into similar test item groups. We assume that test items are managed as an item bank and hierarchically classified according to subject or test categories and subcategories. Also, test items include metadata such as content, correct answer rate, and difficulty.

First, similarities among all test items created in the past are calculated (“1” in Figure 5). Test

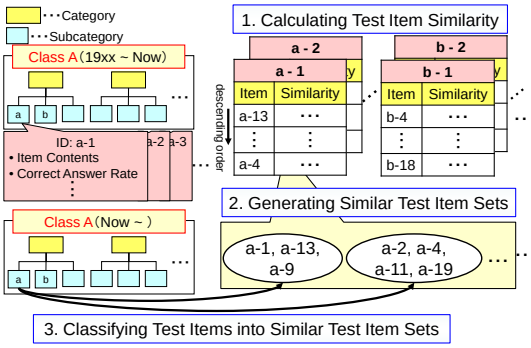

Figure 5. Procedure for Automatically Classifying Test Items.

items targeted for calculating similarity with another test item are assumed to be in the same category as the comparative one. Next, the resulting similarities generate similar test item groups using a clustering technique (“2” in Figure 5). In addition, the similarities between new test items and each similar test item group are calculated. From this result, new test items are classified into an appropriate similar test item group (“3” in Figure 5). As mentioned above, test items can be automatically classified into similar test item groups.

6. Experiment and Evaluation

6.1 Overview of the Experiment

We conducted an experiment where we retrieved similar test items to validate the effectiveness of the following three features, and to measure improvement in the accuracy of retrieving similar test items by the proposed method.

(1) The effectiveness of automatically identifying the knowledge occurrence part (“1” in Figure 4)
(2) The effectiveness of extracting single nouns, compound nouns, and single nouns constituting a compound noun (“2” in Figure 4)
(3) The effectiveness of using topics estimated by LDA (“3” in Figure 4)

To retrieve the similar test items, we used seven LDA methods, using the method described in section 3.2 and the termmi software (Nakagawa et al. 2003b), an existing text mining tool. Table 2 shows an overview of the LDA methods. LDA 1 through LDA 6 differ from the proposed method in terms of their text information determination (“1” in Figure 4) and term extraction (“3” in Figure 4). There are three types of text information from which terms are extracted. LDA 1 and LDA 2 extract terms from the part identified by the procedure, as described in section 3.1. LDA 3 and LDA 4 extract terms from the question and its correct answer. LDA 5 and LDA 6 extract terms from questions, correct choices, and incorrect choices. There are also two types of extracted terms; LDA 1, LDA 3, and LDA 5 extract terms as SN (single nouns) and SNs (multiple single nouns constituting a compound noun), while LDA 2, LDA 4, and LDA 6 extract terms as SN (single nouns) and CN (compound nouns).

In termmi, single nouns and compound nouns are first extracted from the question, correct choice, and incorrect choices. Next, the extracted terms are weighted based on occurrence and concatenation frequency of terms, called the *FLR* method (Nakagawa, Yumoto, & Mori, 2003). If a compound noun formed by adjoined single nouns N_1, N_2,…, N_L (in this order) is *CN*, the weight by concatenation frequency *LR*(*CN*) is defined as follows:

$$LR(CN) = \left(\prod_{i=1}^{L} (LN(N_i)+1)(RN(N_i)+1)\right)^{\frac{1}{2L}}. \quad (5)$$

$LN(N_i)$ is the number of all single nouns which adjoin the left of N_i and $RN(N_i)$ is the number of all single nouns which adjoin the right of N_i. If the occurrence frequency of *CN* is *f*(*CN*), the weight of *CN*, *FLR*(*CN*), is defined as follows:

$$FLR(CN) = f(CN) \times LR(CN). \quad (6)$$

Finally, similarity between test items is calculated as the cosine similarity (equation (4)).

We targeted 250 test items from the "Systems Administrator Examination" (Information Technology Promotion Agency, Japan, 2006) from 2004 to 2008. These test items have at least three similar test items. To evaluate the accuracy of retrieving similar test items, we use the micro-averages of

Table 2: Overview of methods using LDA in experiment.

Method	Text Information Decision ("1" in Figure 4)	Term Extraction ("2" in Figure 4)
Proposed method	Automatically identify the knowledge occurrence part	SN, CN, & SNs
LDA 1		SN & SNs
LDA 2		SN & CN
LDA 3	Question & correct choice	SN & SNs
LDA 4		SN & CN
LDA 5	Question, correct choice & incorrect choice	SN & SNs
LDA 6		SN & CN

the recall and the precision (Manning & Schutze, 1999). Given test item number i, the number of similar test items A_i, the number of retrieved test items B_i, and the number of similar test items among retrieved test items C_i, the micro-averages of recall and precision are defined as follows:

$$\overline{R} = \frac{\sum_{i=1}^{n} C_i}{\sum_{i=1}^{n} A_i} \tag{7}$$

$$\overline{P} = \frac{\sum_{i=1}^{n} C_i}{\sum_{i=1}^{n} B_i}. \tag{8}$$

The relationship between recall and precision is a trade-off, and it can be convenient to combine recall and precision into a single measure of overall performance. Therefore, we use the F-measure (Manning & Schutze, 1999), which converts recall and precision into a scalar value that takes into account both measures. The F-measure is the harmonic average of both values, and the F-measure of the micro-averages of recall and precision is defined as follows:

$$\overline{F} = \frac{1}{\frac{1}{2\overline{P}} + \frac{1}{2\overline{R}}}. \tag{9}$$

To evaluate whether similar test items are retrieved higher, we also set a threshold for the result of retrieved test items. In this case, test items exceeding the threshold are regarded as retrieved test items, and the micro-averages of recall, precision, and F-measure are calculated.

6.2 Retrieving Similar Test Items

The experimental procedure is given below. Steps (1) and (3) are conducted for 250 test items, and finally recall, precision, and F-measure are calculated for each method.

(1) Classifying test items into similar test item groups
The targeted knowledge of each test item was determined by one of the authors. We regarded similar test items as those that covered the same targeted knowledge, and we created similar test item groups. Two hundred and fifty test items were classified into 62 similar test item groups.

(2) Retrieving similar test items
Similarity between test items was calculated based on nine methods. The targeted test items for calculating similarity were the 250 test items targeted in this experiment. It is necessary to give the number of topics to use LDA. In this experiment, we regarded the number of topics as the number of similar test item groups, set as 62 for LDA. Test items were then arranged in order of high similarity.

(3) Extracting test items by threshold
Since test items targeted this experiment have at most seven similar test items, we set the threshold to the top seven test items, and extracted from the result of step (2).

(4) Calculating recall, precision, and F-measure

From the result of steps (2) and (3), the micro-averages of recall and precision were calculated by equation (7) and (8), and F-measure was calculated by equation (9). In this calculation, a similar test item is one determined to have the same targeted knowledge as test item *i*.

Table 3 shows the results of the recall, precision, and F-measure calculations. The left column values are calculated from the result of step (2), and the right column values are calculated from the result of step (3). Compared with the other methods, the proposed method improves the F-measure.

Experimental Results and Discussion

The results of the experiment indicate that the proposed method improves F-measure in comparison with other methods. The results clearly show that the proposed method improved accuracy of retrieving similarity test items. In this section, based on the experimental results shown in Table 3, we will discuss the effectiveness of three features described in section 6.1.

First, comparing LDA 1, LDA 3, and LDA 5, we find that LDA 1 improves the micro-averages of recall, precision, and F-measure in comparison with LDA 3 and LDA 5. These results indicate that

Table 3: Result of recall, precision, and F-measure (No threshold | Top seven test items).

Method	Recall		Precision		F-measure	
Proposed method	0.711	0.675	0.326	0.369	0.448	0.477
LDA 1	0.647	0.615	0.282	0.350	0.393	0.446
LDA 2	0.394	0.378	0.188	0.216	0.255	0.275
LDA 3	0.595	0.544	0.263	0.308	0.365	0.394
LDA 4	0.353	0.343	0.205	0.237	0.259	0.280
LDA 5	0.510	0.481	0.263	0.297	0.347	0.367
LDA 6	0.055	0.052	0.039	0.040	0.046	0.045
Method of previous study[a]	0.722	0.568	0.191	0.491	0.303	0.526
Termmi[b]	0.888	0.593	0.059	0.298	0.110	0.397

[a] Takagi et al., 2009. [b] Nakagawa et al. 2003b.

the procedure of automatically identifying the knowledge occurrence part is effective for improving the accuracy of calculating similarity using LDA. Comparing the proposed method, LDA 1, and LDA 2, the proposed method best improves the micro-averages of recall, precision, and F-measure. These results indicate that extracting terms as single noun, compound noun, and single nouns constituting a compound noun is effective for improving the accuracy of calculating similarity using LDA. Finally, comparing the results for the case with no threshold in the proposed method and existing methods, we find that the proposed method improves the micro-average of precision and the F-measure in comparison with the existing methods. These results indicate that using topics estimated by LDA is effective for decreasing false retrieval caused by extracting unnecessary terms.

As mentioned above, we showed the effectiveness of the proposed method for test items from the information technology field targeted in these experiments. However, in the method using LDA, similarities of retrieved test items were about the same values. This is because all topics estimated by LDA in each test item were binary weighted. Therefore, when more than one topic is estimated for a test item, each topic should be weighted by a different value.

7. Conclusion

We proposed a method for calculating similarity between test items for automatic classification. In the proposed method, test items are represented by a feature quantity vector as estimated by LDA. To accurately estimate topics, we performed the following preprocessing steps before using LDA:

(a) Identify the part (question, correct choice, or incorrect choice) where the targeted knowledge occurs

(b) Enhance the co-occurrence relation between terms occurring in the part identified in preprocessing step (a)

In preprocessing step (a), we automatically identified the knowledge occurrence part. In preprocessing step (b), terms were extracted as single nouns, compound nouns, and single nouns constituting a compound noun from the identified parts.

We furthermore conducted an experiment in which we retrieved similar test items. The result of this experiment showed the following three effects and accuracy improvements of the proposed method in comparison with existing methods:

(1) The effectiveness of automatically identifying the knowledge occurrence part
(2) The effectiveness of extracting single nouns, compound nouns, and single nouns constituting a compound noun
(3) The effectiveness of using topics estimated by LDA

In the future, we would like to work further on the tasks mentioned in section 1, making it possible to automatically manage, create, and group test items.

Acknowledgements

This work was supported by JSPS KAKENHI Grants (Numbers 24700904 and 25-8284).

References

Blackboard Inc. (1997). *Blackboard*. Retrieved from http://www.blackboard.jp

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. *J. Mach. Learn. Res., 3*, 993--1022.

Cao, J., Li, J., Zhang, Y., & Tang, S. (2007, September). LDA-Based Retrieval Framework for Semantic News Video Retrieval. *Proceedings of the First IEEE International Conference on Semantic Computing (ICSC '07)*, 155-160. Irvine, California, USA: IEEE Computer Society.

Chemudugunta, C., Smyth, P., & Steyvers, M. (2007, December). Modeling General and Specific Aspects of Documents with a Probabilistic Topic Model. *Proceedings of Advances in Neural Information Processing Systems 19 (NIPS '07)*, 241-248. Vancouver, B.C., Canada: Curran Associates, Inc.

Dougiamas, M. (1999). *moodle*. Retrieved from https://moodle.org

Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Science, 101*, 5228-5235.

Guzman, E., & Conejo, R. (2005). Self-Assessment in a Feasible, Adaptive Web-Based Testing system. *IEEE Trans. Education, 48* (4), 688-695.

Hambleton, R. K., Swaminathan, H., & Rogers, H. J. (1991). *Fundamentals of item response theory*. London: Sage Publications.

Hofmann, T. (1999, August). Probabilistic latent semantic indexing. *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '99)*, 50-57. New York, NY, USA: ACM.

Ikeda, S., Takagi, T., Takagi, M., & Teshigawara, Y. (2012, November). Proposal and Evaluation of a Method of Estimating the Difficulty of Items Focused on Item Types and Similarity of Choices. *Proceedings of the 20th International Conference on Computers in Education (ICCE2012)*, 254-261. Singapore.

Information-technology Promotion Agency, Japan. (2006). *Systems Administrator Examination*. Retrieved from http://www.jitec.ipa.go.jp/1_11seido/h13/ad.html

Iwata, T., Yamada, T., & Ueda, N. (2008, August). Probabilistic Latent Semantic Visualization: Topic Model for Visualizing Documents. *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '08)*, 363-371. Las Vegas, Nevada, USA: ACM.

Manning, C. D., & Schutze, H. (1999). *Fundamentals of Statistical Natural Language Processing*. MIT Press.

Matsumoto, Y. (2000). Morphological Analysis System ChaSen(<Special Features>Easy to Use Practical Freeware for Natural Language Processing). *IPSJ Magazine, 41* (11), 1208-1214.

Nakagawa, H., Maeda, A., & Kojima, H. (2003a). *TermExtract*. Retrieved from http://gensen.dl.itc.u-tokyo.ac.jp/termextract.html

Nakagawa, H., Maeda, A., & Kojima, H. (2003b). *Text Mining Tool for Windows, "termmi"*. Retrieved from http://gensen.dl.itc.u-tokyo.ac.jp/termmi.html

Nakagawa, H., Yumoto, H., & Mori, T. (2003). Term Extraction Based on Occurrence and Concatenation Frequency. *Journal of Natural Language Processing, 10* (1), 27-45.

Songmuang, P., & Ueno, M. (2011). Bees Algorithm for Construction of Multiple Test Forms in E-Testing. *IEEE Trans. Learn. Technol, 4* (3), 209-221.

Songmuang, P., & Ueno, M. (2008). Development and practice of an integrative e-testing system. *Journal of The Japan Association for Research on Testing, 4* (1), 53-64.

Takagi, T., Takagi, M., & Teshigawara, Y. (2009, March). A Proposal and Evaluation of a Method of Calculating Similarity between Quizzes Created by Students. *Proceedings of the Eighth IASTED International Conference on Web-Based Education (WBE2009)*, 360-366. Phuket, Thailand.

Teh, Y. W., Jordan, M. I., Beal, M. J., & Blei, D. M. (2006). Hierarchical Dirichlet Processes. *Journal of the American Statistical Association*, *101*, 1566-1581.

Teh, Y. W., Newman, D., & Welling, M. (2007, December). A Collapsed Variational Bayesian Inference Algorithm for Latent Dirichlet Allocation. *Proceedings of Advances in Neural Information Processing Systems 19 (NIPS '07)*, 1353-1360. Vancouver, B.C., Canada: Curran Associates, Inc.

Ueno, M. (2005). Web based Computerized Testing System for Distance Education. *Educational Technology Research*, *28* (1, 2), 59-69.

Ueno, M., & Okamoto, T. (2008, July). System for online detection of aberrant responses in e-testing. *Proceedings of the 2008 Eighth IEEE International Conference on Advanced Learning Technologies (ICALT '08)*, 824-828. Santander, Cantabria, Spain: IEEE Computer Society.

Wang, C., Zhang, L., & Zhang, H.-J. (2008, July). Learning to Reduce the Semantic Gap in Web Image Retrieval and Annotation. *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '08)*, 355-362. Singapore, Singapore: ACM.

Young, F., & Hamer, R. (1987). *Multidimensional Scaling--history, theory, and applications.* L. Erlbaum Associates.