Automatic Assessment of Complex Assignments using Topic Models

Saar Kuzi, William Cope, Duncan Ferguson, Chase Geigle, ChengXiang Zhai
University of Illinois at Urbana-Champaign
{skuzi2,billcope,dcf,geigle1,czhai}@illinois.edu

ABSTRACT

Automated assessment of complex assignments is crucial for scaling up learning of complex skills such as critical thinking. To address this challenge, one previous work has applied supervised machine learning to automate the assessment by learning from examples of graded assignments by humans. However, in the previous work, only simple lexical features, such as words or n-grams, have been used. In this paper, we propose to use topics as features for this task, which are more interpretable than those simple lexical features and can also address polysemy and synonymy of lexical semantics. The topics can be learned automatically from the student assignment data by using a probabilistic topic model. We propose and study multiple approaches to construct topical features and to combine topical features with simple lexical features. We evaluate the proposed methods using clinical case assignments performed by veterinary medicine students. The experimental results show that topical features are generally very effective and can substantially improve performance when added on top of the lexical features. However, their effectiveness is highly sensitive to how the topics are constructed and a combination of topics constructed using multiple views of the text data works the best. Our results also show that combining the prediction results of using different types of topical features and of topical and lexical features is more effective than pooling all features together to form a larger feature space.

INTRODUCTION

Assessment is an essential part of instruction as it is needed to determine whether the goal of education has been met. It is also very useful for providing helpful feedback to learners based on their work, which can often be an effective way to improve the efficiency of learning. Furthermore, the assessment may also potentially help instructors improve the materials that they teach or the way they teach those materials; for example, uniformly poor performance of all students may indicate either the materials are too hard for the students or the teaching of those materials was not very effective.

L@S '19, June 24–25, 2019, Chicago, IL, USA

 ISBN 978-1-4503-6804-9/19/06…$15.00
DOI: https://doi.org/10.1145/3330430.3333615

Traditionally, assessment has been done manually by instructional staff such as instructors or teaching assistants and is known to be labor-intensive. Such a manual way of performing assessment can thus not keep up with the pace of the growing demand for education at a larger scale. To scale up education without causing increased cost from manual assessment, researchers have studied how to automate the assessment by using computational methods. In addition to reducing human effort, thus also education cost, the automated assessment also has the potential to leverage data mining and machine learning to provide detailed feedback to students in order to help them learn from their mistakes.

There is a large body of work on automated assessment, focusing mainly on simple assignments, including computer programs [1], essays [19], and short answer questions [12]. Indeed, the current Massive Open Online Course (MOOC) platforms, such as Coursera and edX, can only support the automated assessment of simple assignments which mostly involve multiple-choice questions. The current methods for automated assessment have limited capability in detecting sophisticated qualities of learning in complex assignments such as critical thinking, creative thinking, development of broad conceptual constructs, and self-reflection on the limits of one's understanding. As a result, current online courses are less effective for teaching sophisticated subjects and skills that require assessment using complex assignments.

Complex assignments are very common and essential for teaching students sophisticated skills. For example, in medical education it is typical to use assignments designed based on sample medical cases to assess students' ability of critically analyzing a case, reflecting on any missing knowledge that they might need to acquire, and suggesting appropriate actions. Complex assignments are often in the form of unstructured long natural language text, which makes it quite challenging to automate their assessment as it must be done based on some semantic analysis of the content written by the student. As a result, these assignments are currently graded manually by the instruction staff. However, manual grading of such complex assignments is labor-intensive, making it hard to scale up the teaching of these important skills.

To address this problem, a previous study [5] has proposed to apply machine learning to automate the assessment of complex assignments, where the authors have demonstrated the feasibility of using supervised learning to automate the assessment of complex assignments and recommended using

supervised learning to score and rank assignments and have a human grader assign cutoffs to generate grades. However, this work has used primarily simple lexical features to represent the text data generated by students, which included the number of occurrences of vocabulary words in the student assignment.

Using simple lexical features for automated assessment of complex assignments, however, has two inherent deficiencies. First, words are often ambiguous (polysemy) and some may be synonymous (synonymy). Second, single words or a few words may not be able to capture complex semantics in student writing, which may require many words to describe. To illustrate these limitations, we present in Figure 1 a paragraph which was extracted from a complex assignment in the data set that was used for our experiments[1]. The data set consists of clinical case analyses performed by students in the school of veterinary medicine. One of the goals of this assignment is to assess the ability of students to identify a major differential diagnosis in a clinical case and support it with relevant evidence. We can see in Figure 1 that the student identifies the "Tetralogy of Fallot" as a possible diagnosis and supports it by using evidence from the case (e.g., "pulmonic stenosis" and "over-riding aorta"). Thus, in order to accurately assess the student analysis, words that describe both the diagnosis and the relevant evidence must be taken into account. However, this may not be easily achieved when only simple lexical features are used since single words are treated as individual and independent features.

In this paper, we address the limitations of simple lexical features and propose using topics as features for automated assessment of complex assignments. A topic is represented as a distribution over words with the high probabilities assigned to words that are most important for characterizing a topic.

Compared with the simple lexical features, topical features are potentially advantageous because of their ability to better capture semantics via clustering of words. Specifically, a topic represented as a word distribution would address the problem of polysemy by allowing a word to have non-zero probabilities for multiple topics and the problem of synonymy by involving all the synonyms of a word in the same topic representation with non-zero probabilities. Moreover, because of the use of potentially all the words in the vocabulary and the flexibility in assigning different weights (i.e., probabilities) to them, the topic representation can help distinguish subtle differences between assignments and is more interpretable compared to simple lexical features. For example, topical features correlated with grades may inform the instructor of conceptual understanding and misunderstanding by students better than correlated lexical features.

Topics can be automatically learned from the text data using probabilistic topic models such as the Probabilistic Latent Semantic Analysis model (PLSA) [7] and the Latent Dirichlet Allocation model (LDA) [3]. Topic models have already been successfully used in various text mining applications. For example, they were shown to be effective for prediction of time series variables [20], information retrieval [22], and text analysis [23]. Our paper can be regarded as a novel application of topic models for automated assessment of complex assignments.

> "The animal presents with multiple cardiac defects: ventricular septal wall defect (VSD), right ventricular hypertrophy, over-riding aorta, and pulmonic stenosis. These four defects make up the congenital malformation that is the Tetralogy of Fallot (Figure 1). This is further supported by the shunting of blood from the right ventricle to the left ventricle, the location of the ventricular septal defect, and the gas exchange imbalance, as this disease can cause cyanosis from hypoxemia of blood in systemic circulation [2,4,5,6]."

Figure 1: A paragraph extracted from a complex assignment in the data set that was used in our experiments.

We propose and study multiple ways to extract topics and apply topical features to automated assessment. Specifically, we propose to generate different kinds of topics using multiple views of the text data, including: (1) Topics learned in an unsupervised manner, using all available data (both training and test sets). (2) Topics learned using the guidance of assignment grades in a training set: using only the high scoring assignments may result in topics that capture good practices, while using low scoring ones may result in topics that capture common mistakes in student works. (3) Topics learned using different granularities of text segments which can capture different levels of semantic meaning.

Since the basic lexical features and topical features provide two complementary perspectives of the complex assignment, we also explore how to combine topical features with the simple lexical features as well as how to combine different kinds of topical features. We propose and study two approaches for feature combination: (1) Combining the prediction results of the models learned for each group of features separately. (2) Pooling all features together to form a larger set and learning a single prediction model.

We evaluate the proposed methods using clinical case assignments performed by veterinary medicine students. Assignments were assessed using a rubric which was designed by the instructor with the goal of assessing different facets of the student complex assignment. Our goal then is to automatically assess the quality of an assignment in each of the rubric dimensions. Our experimental results show that topical features are interpretable and highly effective for the assessment of complex assignments. Moreover, combining topical features with lexical features substantially outperform the baseline of only using lexical features, and combining the prediction results of using lexical features and topical features respectively is more effective than pooling the lexical features and topical features. Finally, we show that different rubric dimensions benefit from different models, learned using different views of the text.

RELATED WORK

The current technology for automatic grading is mostly limited to multiple choice questions, short answers [12], and essay

[1]Thanks to Allison Dianis for allowing us to use this part of her work.

scoring [2]. A recent work explored automated assessment of more complex assignments [5]. Our work is a direct extension of this work by exploring more sophisticated features, i.e., topical features for automated grading.

There is a large body of work on using Latent Semantic Analysis (LSA) for automated assessment of simple essays. One idea is to apply LSA to parts of the course material which are relevant to the writing assignment [11, 4]. Then, essays are graded based on their similarity (in the induced LSA space) with human-graded essays, or with essays written by experts. Compared with probabilistic topic models, LSA is less interpretable.

LDA and PLSA were also used for automated grading of essays [10, 9]. In both works, essays were directly compared with the course material using a topic model, learned from the relevant course material. In other works, only the course material was used (i.e., no human-graded assignments). For instance, one work used course material annotated by the instructor to indicate topics and notions within each topic [13]. Then, essays were graded based on their similarity with the topic that served as the essay subject. Our work goes beyond using topics to measure similarity and systematically studies how to construct effective topical features using multiple views of data and how to optimally combine topical features with lexical features, which have not been studied in the previous work. Another difference of our work from this previous work is our focus on grading of complex assignments, which is more challenging than essay grading.

Topic models were also applied to other educational applications, including, e.g., using LSA for generating automated semantic feedback to children compositions [21] and measuring similarity of student answers and hand-crafted list of expectations and misconceptions in an interactive tutoring system [6].

Topic models have frequently been used for classification tasks since their inception [3]. The most traditional approach is to use a topic model to infer topics on a set of training documents and then at classification time use the model to infer topic proportion vectors for the unseen documents, which are then used as an input to a classifier [15].

Another approach is to integrate the topic modeling and the classification tasks into one unified framework [17, 24] where both the topics and the labels are modeled directly through an augmented LDA/PLSA topic model. Then, topics are learned across the entire corpus at once, including documents from all labels to be predicted. By contrast, the approach we propose in this paper generates an independent set of topics for each label, which can then be combined into a larger feature vector. Furthermore, compared to previous works in which specialized topic models with supervision were developed, our approach is completely general and can thus be combined with any existing topic models to achieve the effect of supervision.

Topic models have also been modified to more directly support other specific tasks. One example is the author-topic model [18], which attempts to model topical preferences within documents as well as among individual authors of documents. In these cases, the underlying graphical model itself is adapted to address the new task, which often necessitates the derivation of a new sampling algorithm. In this paper, we instead focus on approaches that can leverage *existing* topic models more optimally without directly changing the underlying graphical model itself. In this sense, our work is completely orthogonal to the existing work on topic models.

SUPERVISED LEARNING FOR AUTOMATED ASSESSMENT

Our main goal is to improve the feature representation of student works in the supervised learning approach to the automated assessment of complex assignments [5] by constructing more effective features based on topics. One way of applying supervised learning to automated assessment is to use the student assignment data to learn a function that takes a feature representation of the student's work as input and outputs a grade (e.g., by using a regression model or a multiclass classification model). However, as argued in the work [5], when using such a technique for automating the grading with humans in the loop, the problem is better framed as a ranking problem. That is, we would not use the prediction score directly as a (final) grade, but instead use it to rank all student works and let the instructor decide where to place a cutoff to convert the ranking scores into final grades. With this setup, the problem from the computational perspective is to take all student works as input and generate a ranking of them. We will learn such a ranking function based on a set of training examples where we are given the correct ranking of many training instances. Our main hypothesis is that when we add topical features to the baseline approach, which uses simple lexical features, we would improve the ranking accuracy of the ranking function.

TOPIC DISCOVERY AND CONSTRUCTION OF TOPICAL FEATURES

The main approach we explore in this paper is to use topic models to learn topics and construct features based on the learned topics, which would then be used in a supervised machine learning framework to predict the ranking of student works. In this section, we describe the technical approaches in detail, starting with an introduction to topic models.

Topic models background: A topic model is a probabilistic generative model for text data. The underlying premise is that text in a document is generated by a mixture of several topics, which represent different themes. In this paper, we use the LDA model [3] in order to learn topics. LDA can be applied to any set of documents to learn k topics, where each topic is a multinomial distribution over the vocabulary words, $\theta_j \forall j \in \{1, 2, ..., k\}$. For example, in a topic model built using a collection of news articles, a topic about sports is expected to attribute high probabilities to words such as *football*, *basketball*, and *tennis*, but very small probabilities to words like *congress*, *party*, and *bill*, which may have high probabilities in a topic about politics. Furthermore, each document in the training set is assigned with a multinomial distribution over the k topics, π_d; that is, $\pi_{d,j}$ is the probability that a word within the document d was drawn from topic j. The generative process according to LDA goes as follows: (1) A multinomial

distribution over topics, π_d, is sampled from a Dirichlet prior distribution. (2) For each position in the document, a topic j is selected by sampling from π_d. (2) A word is sampled according to θ_j.

Rubric-guided topic modeling: In this paper, our goal is to generate features, using topic models, that can be effectively used for automatic assessment of complex assignments. Our assumption is that topics can be viewed as textual patterns in assignments, which correlate with performance in the different rubric dimensions used for grading. The standard approach for learning topics would be to use all available assignments (from both the training and the test set) in a fully unsupervised manner. This type of model would benefit from using the maximum amount of data. We refer to this model as **StandardModel**.

However, such a model may not necessarily pick up topics that have high correlations with grades. To potentially obtain such more discriminative topics, we propose an alternative approach, which is to use guidance from the assignment grades in the different rubric dimensions. Specifically, for each rubric dimension, we learn two topic models. One topic model is learned using the high scoring assignments, whereas the other one is learned using the low scoring assignments. The idea here is that the topics learned using high scoring assignments can be expected to capture common patterns present in them, which may serve as useful indicators of a good grade. Similarly, the topics learned using low scoring assignments may pick up common patterns in assignments where the students have made mistakes, and the patterns can capture the commonly made mistakes. We note that while supervision was used for splitting the data, the topic modeling algorithm remains unsupervised[2]. We refer to these models as **RubricGuided**.

For each assignment we generate topical features using all RubricGuided models (two models for each rubric dimension), regardless of the rubric dimension grade to be predicted. Note that the topical features generated from modeling one rubric dimension may also be useful for predicting grades in another dimension; indeed, our experimental results show that highly important topical features for predicting the performance in one dimension are often those generated using the guidance of other dimensions.

Multi-scale topic modeling: We may further learn different models by using text segments extracted from the original assignments in different granularities. By doing so, we expect to capture semantics at different levels, which may be necessary for supporting automated assessment. For example, topic models learned using low granularity of text (i.e., long text segments) may be able to capture high-level patterns, while models with high granularity (i.e., short text segments) may capture more implicit ones. Furthermore, prediction of performance in different rubric dimensions may rely on different granularities of information. Technically, we split each assignment into n segments[3]. Then, we feed the model with the text segments, treating them as individual and independent documents.

Generating topic model features: As discussed earlier, we use the data set in order to learn various topic models so as to obtain multiple views of the text data. Specifically, for each of our suggested approaches for topic modeling (StandardModel and RubricGuided), we learn several topic models by varying the level of text granularity (n) and the number of topics (k). Once we obtained those topic models, the next step is to define topical features and their values.

The multinomial distribution of the j'th topic in a topic model with k topics and a granularity level of n is denoted $\theta_j^{n,k}$. The coverage of a topic in an assignment segment d_i (the i'th segment of an assignment d, $i \in \{1,2,...,n\}$) is measured using an approximation of the Kullback-Leibler divergence (KL) between the distribution of the topic ($p(\cdot|\theta_j^{n,k})$) and of the assignment segment ($p(\cdot|d_i)$) over the vocabulary terms[4].

$$score(\theta_j^{n,k}, d_i) = \sum_{w \in V: p(w|\theta_j^{n,k})>0} p(w|d_i) \log \frac{p(w|d_i)}{p(w|\theta_j^{n,k})}; \quad (1)$$

where w is a word in the vocabulary V. $p(w|d_i)$ is estimated using the maximum likelihood approach, that is $p(w|d_i) = \frac{tf(w \in d_i)}{|d_i|}$; $tf(w \in d_i)$ is the number of occurrences of w in d_i and $|d_i|$ is the total number of words in d_i. In order to generate a topical feature for each assignment, the scores of the different assignment segments are aggregated as follows:

$$f_j^{n,k}(d) = \log\left(1 + \max_{i \in \{1,...,n\}} score(\theta_j^{n,k}, d_i)\right); \quad (2)$$

$j \in \{1,...,k\}$, i.e., for a single topic model we generate k topical features per assignment. We use the max aggregation function in order to capture for each assignment the most salient features. Indeed, our experiments showed that this approach performs better than other approaches such as taking the average or using all features; we do not report the actual results as they do not convey further insight.

An alternative approach for estimating $score(\theta_j^{n,k}, d_i)$ would be to directly use the distribution of documents over topics, $\left\{\pi_{d_i}^{n,1}, \pi_{d_i}^{n,2}, ..., \pi_{d_i}^{n,k}\right\}$; $\pi_{d_i}^{n,j}$ is the coverage of topic j in the i'th segment of assignment d. This distribution is learned for assignments in the training set and can be easily inferred for unseen documents. However, our experimental results showed that using this distribution is not as effective for autometed grading as using the KL-divergence measure as in Equation 1. Thus, we have mainly used the KL-divergence measure in most of our experiments. (We further discuss this issue in the experimental results section.)

[2]This is in sharp contrast to some existing supervised topic models [17] where a specific topic modeling algorithm is tied to labeled data for supervision.

[3]In this paper, we split the assignments into equal length segments; length is defined by the number of sentences.

[4]This is an approximation as the summation is only over terms with positive probabilities in $\theta_j^{n,k}$ so as to avoid division by zero.

Simple lexical features (Unigrams): In previous work, raw frequencies of vocabulary words in the assignments were used as features for automated assessment [5]. We use this approach as a baseline in our experiments[5]. We denote this type of features **Unigrams**, and compute them for each word w in the vocabulary as follows:

$$f_w^{unigram}(d) = \log(1 + tf(w \in d)); \quad (3)$$

$tf(w \in d)$ is the number of occurrences of w in assignment d.

Feature combination: Due to the "soft" matching inherent in a topic (word distribution), topic features may not be as discriminative as some lexical features. In general, they may be complementary to each other. Thus intuitively, it may be desirable to combine them. We explore different approaches for combining the features extracted using different topic models, and combining topical features with lexical features. In one approach, denoted **FeatureComb**, we pool all features together to form a larger feature set. Then, the weights for each feature can be learned using any supervised machine learning algorithm (e.g., [14]). We will further discuss the specific algorithms we used in our experiments in the next section.

One potential limitation of the FeatureComb approach is that when we have a large number of features, i.e., counts of vocabulary words and topical features, the machine learning program may not necessarily be able to assign the optimal relative weights among all the topics since the topics would be mixed together with words.

To address this limitation, we propose a second approach where we learn several models corresponding to different groups of features. Then, we combine for each assignment the prediction results according to each model. In such an approach, the relative weights among the topical features can be optimized when we train a separate ranking or classification model. We propose two methods to that end: (1) **ScoreComb**: summing up the prediction scores of an assignment in the different models (soft-max normalization is first applied to the scores in each model). (2) **RankComb**: transforming the prediction scores into ranks and summing them up. It is unclear which method works better because it depends on whether the scores are actually comparable; if so, ScoreComb probably would make more sense, whereas RankComb can be expected to be more robust, though it cannot take advantage of the magnitude of the difference in scores.

EVALUATION

Our goal of the evaluation is to assess the effectiveness of the proposed topical features for automated assessment. To this end, we followed a similar experimental setup as in [5] and used a data set of medical case analysis assignments. We fixed the machine learning framework and varied the ways we construct features to be used in the framework. We compare the performance of using different features and different ways to combine features with the main goal being to understand whether topical features can outperform the lexical features or add value on top of them.

[5]This previous work also used similarity-based features, generated using the similarity between the student work and an instructor crafted work. Those features were computed assuming some structure of the assignment. In our case, however, we were not able to make these assumptions since students were not confined to any structure. Thus, we leave the incorporation of similarity-based features for future work.

Experimental setup

Data set: As part of their training in clinical problem-solving, first year veterinary students were given the assignment of analyzing a clinical case by the development of a multimedia-containing text document. Students were asked to provide answers to specific questions about the case and also to connect the animal's problems to their basic physiology and anatomical understanding. The exercise was designed to parallel their concurrent didactic training (lectures and laboratories), but to also challenge the students to reflect upon elements of the case that forced deeper self-study and/or review, including elements of future clinical training, such as the development of a differential diagnosis list. Furthermore, in the tradition of evidence-based medicine, students were asked to identify and justify the references that they chose.

Clinical case assignments performed by students in two consecutive years were used for the evaluation. Both classes were given the same case with the same instructions; students in both classes were at the same level of their studies. We used the assignments of the most recent class as a test set (134 assignments), and of the other class as a training set (160 assignments). The data set is not as large as we would like, but this is the best data set that is available for studying this problem; indeed, the main reason why no larger data sets are available is precisely due to the difficulty in scaling up this kind of classes, the main motivation for our study. In general, the analyses ranged from about 1,400 to 3,400 words in length, with the average being about 2,400 words; students were not confined to a specific structure. Students were also provided with the rubric that was later used for peer evaluation and which is composed of the following dimensions:

(1) *Problems (Pro.)*: The student should list the three most serious clinical problems in the case, and defend his/her reasoning. (2) *Differentials (Dif.)*: The student should identify at least two major differential diagnoses for the animal and defend his/her choices with evidence from the case and information from the literature. (3) *Evidence (Evi.)*: The student should identify the clinical observations in the case to support his/her problem list and differential diagnosis list. (4) *Understanding (Und.)*: The student should respond to various questions in order to evaluate his/her understanding of the case (for example: "If unmanaged, what kind of additional clinical signs would you expect?"). (5) *Conclusions (Con.)*: The student should identify and explain at least 2 personal learning issues from the exercise. (6) *References (Ref.)*: The student should provide references that helped him/her to understand the case. (7) *Overall (Ove.)*: The overall impression of the reviewer from the analysis.

Assignments were graded in each of the rubric dimensions by three peer-reviewers (also students in the class) as part of the formative feedback phase between a first and final draft of the authoring student's work. Therefore, the reviewer comments

Table 1: Inter-reviewer agreement. The average standard deviation of grades (Stdv) and the average number of reviewers who agreed on a grade (Agreements) are reported.

		Pro.	Dif.	Evi.	Und.	Con.	Ref.	Ove.
Training	Stdv	.476	.530	.611	.616	.479	.513	.617
	Agreements	2.6	2.3	2.1	2.0	2.8	2.6	2.1
Test	Stdv	.539	.550	.614	.679	.517	.574	.511
	Agreements	2.4	2.4	2.0	1.8	2.8	2.4	2.4

were about the first draft of a student's work with the goal of leading to improvement of the final draft. We use the average grade of the reviewers in each dimension (each reviewer selects a grade for each dimension from $\{0,1,2,3,4\}$). In Table 1 we report the inter-reviewer agreement in terms of the standard deviation of grades (Stdv) and the number of reviewers who agreed on a specific grade (Agreements); the average score over all assignments in the entire data set is reported.

Implementation details: Text was extracted from the assignments and then was pre-processed, including stopword removal and Porter stemming, using the MeTA toolkit [16]. We also used the MeTA implementation for LDA (Collapsed Variational Bayes was used for inference). We learn topic models using two levels of text granularity (n): (1) *FullText*: documents are not split into paragraphs ($n = 1$), (2) *Paragraphs*: documents are split into three paragraphs ($n = 3$). Unless stated otherwise, we use 5 topics. For the Unigrams baseline, we use only words that appear in at least 5 assignments in the training set. We report the Kendall's-τ correlation between the ranking of student assignments based on their known grades and the ranking of the same set of student assignments based on the scores produced using our automated assessment method. The correlation takes values between [-1,1] where -1 and 1 correspond to a perfect negative and positive correlation, respectively.

Learning framework: The proposed topical features can be used in any machine learning framework, but to focus on studying the effectiveness of various features, we want to fix the learning framework. Specifically, in our experiments, we used the learning-to-rank (LTR) approach from information retrieval [14]. The goal in LTR is to learn a function that accurately ranks assignments based on their performance in a specific rubric dimension. Specifically, we use the pair-wise LTR algorithm, SVMrank [8] (cs.cornell.edu/people/tj/svm_light/svm_rank.html). The goal of this algorithm is to learn a linear ranking function while minimizing the number of incorrectly ordered pairs of assignments; two assignments are considered incorrectly ordered if one is ranked higher than the other while having a lower performance grade.

Experimental results

Main result: Our main result is presented in Table 2. We report the performance of using only topical features (Topics) and of using the combination of topical and simple lexical features (Topics+Unigrams). We compare these approaches with the baseline of using only simple lexical features (Unigrams). According to the upper block of the table, topical features are highly effective compared to Unigrams. Specifically, for all rubric dimensions, except for one, at least one of the three methods for feature combination outperforms the baseline; the improvements are by a very large margin for some dimensions (e.g., 94% and 65% improvement for Differentials and References, respectively, when the RankComb approach is used). An additional observation, based on the upper block of Table 2, is that Unigrams and Topics often find the prediction of certain dimensions challenging to a similar extent. For example, the lowest correlation for both approaches is obtained for Conclusions. This may be because Conclusions evaluates the authors' self-reflection on their knowledge base and is thus harder to judge based on textual features. Other dimensions, on the other hand, can be more based upon concrete concepts and facts and thus may be easier to judge (e.g., Evidence). Comparing the different methods for feature combination, we can see that the prediction result combination methods (ScoreComb and RankComb) outperform the FeatureComb approach for all rubric dimensions. Comparing ScoreComb to RankComb, we observe that RankComb is more effective for most rubric dimensions, albeit not by a large margin.

Table 2: Performance (Kendall's-τ) of using simple lexical features (Unigrams), topical features (Topics), and their combination (Topics+Unigrams). Three methods for feature combination are compared. Boldface: best result in a column.

Method	Pro.	Dif.	Evi.	Und.	Con.	Ref.	Ove.
Unigrams	.268	.185	.312	.360	.194	.259	.260
Topics							
FeatureComb	.266	.188	.147	.295	.085	.294	.134
ScoreComb	.293	.357	.312	.383	.210	**.439**	.238
RankComb	.294	**.359**	.317	.392	.191	.429	.249
Topics + Unigrams							
FeatureComb	.260	.177	.303	.355	.177	.278	.246
ScoreComb	**.319**	.334	.328	.402	**.218**	.431	.240
RankComb	.314	.355	**.348**	**.417**	.208	.433	**.269**

We next analyze the performance of using both Unigrams and topical features (lower block). According to the results, the combination of topical features and Unigrams does not result in further improvements over the individual components in the case of FeatureComb for the majority of rubric dimensions. However, when prediction result combination approaches are used, we can see that the combination is of merit. Specifically, in the case of ScoreComb the combination outperforms its individual components in four rubric dimensions. As for RankComb it is the case for six dimensions. This suggests that although topics are very useful features, they cannot replace the baseline lexical features, presumably due to the lack of subtle discriminations in topical features. We conclude, based on Table 2, that our best performing approach is RankComb using both Unigrams and Topics. We would refer to this approach in the rest of the paper as RankComb(Topics+Unigrams).

An interesting finding, based on Table 2, is that combining features using their prediction results is much more effective than the standard approach of pooling all features together both for combining different types of topical features and for combing topical features with Unigrams. These results

Table 3: Performance (Kendall's-τ) of StandardModel vs. RubricGuided. Unigrams are also combined in the model using the RankComb method. Note: the last line is identical to last line in Table 2 as both describe the same method. Boldface: best result in a column.

Model	Pro.	Dif.	Evi.	Und.	Con.	Ref.	Ove.
Unigrams	.268	.185	.312	.360	.194	.259	.260
RubricGuided+	.294	.275	.314	.385	**.231**	.388	.221
RubricGuided–	.306	.285	**.349**	.395	.207	.377	.257
RubricGuided	**.320**	.336	.335	.404	.226	.408	.242
StandardModel	.263	.287	.324	.356	.163	.405	**.300**
RubricGuided and StandardModel	.314	**.355**	.348	**.417**	.208	**.433**	.269

suggest that when we pool different kinds of features together, the machine learning algorithm may not be able to optimize the relative weights on the same type of features as well as when we train a separate classifier for each type of features separately. It would be interesting to further investigate this issue in future work.

StandardModel vs. RubricGuided: In Table 3 we compare the performance of the different topic models used to generate the topical features. Specifically, we report the performance of the RankComb(Unigrams+Topics) approach, the best performing one according to Table 2, when different topic models are used for topical feature generation. RubricGuided topics built using only the high scoring assignments and only the low scoring assignments are denoted RubricGuided+ and RubricGuided–, respectively. According to the results, both components of RubricGuided are highly effective. Specifically, using these topical features outperforms the baseline in a vast majority of relevant comparisons. Comparing RubricGuided+ with RubricGuided–, we can see that the latter is more effective for most rubric dimensions. One possible explanation for this may be because there are only a few ways to be "right" but many ways to be off-track. Furthermore, we can see in Table 3 that both RubricGuided models are complementary. That is, their combination outperforms the individual components in most dimensions. StandardModel is also very effective, outperforming the baseline for most dimensions. Yet, its performance is dominated by those of the RubricGuided model. Combining StandardModel with RubricGuided (as in our best performing approach in Table 2) results in further improvements in four out of seven dimensions. Thus, we conclude that using multiple topic models, generated using multiple views of the text data, is beneficial for automated assessment of complex assignments. We note that the generalization of this finding, using data sets from different domains, is an interesting avenue for future work.

Free parameters analysis: In Figure 2 we analyze the performance of RankComb(Unigrams+Topics) (our best performing approach) as a function of the text granularity level, n (FullText vs. Paragraphs), and the number of topics, k ($\in \{5, 15, 25\}$). Due to space limitations, we present the result only for 4 rubric dimensions; the main findings for the other dimensions are similar. According to the results, different rubric dimensions benefit from different levels of text granularity. For example, topic models generated using Paragraphs achieve better performance in the Problems and Conclusions dimensions. A possible explanation for that is that the information relevant for grading some dimensions is concentrated in specific paragraphs of the assignment. For example, we observed that many assignments begin with the description of the problems and naturally conclude with a clear conclusion section. On the other hand, information regarding other dimensions can be spread throughout the entire assignment. This might be the case, for example, in the Overall and Evidence dimensions. We can see that for most dimensions the performance of using both levels of granularity is dominated by either one of the individual levels. Yet, using both levels results in a more robust performance. As for the number of topics used in a topic model, we observe that the optimal value depends on the rubric dimension and the level of text granularity. In most cases, FullText models benefit from a relatively low number of topics, where the opposite holds for the Paragraphs models. Yet, when combining both levels of granularity we can see that the model performance is relatively stable with respect to the number of topics. We conclude that our approach of combing all levels of granularity and fixing the number of topics is reasonable in the case where no prior knowledge is given. An alternative approach would be to select the values for these parameters on a per-dimension basis (e.g., using prior knowledge or a development set). Such approaches can be further explored in future work[6].

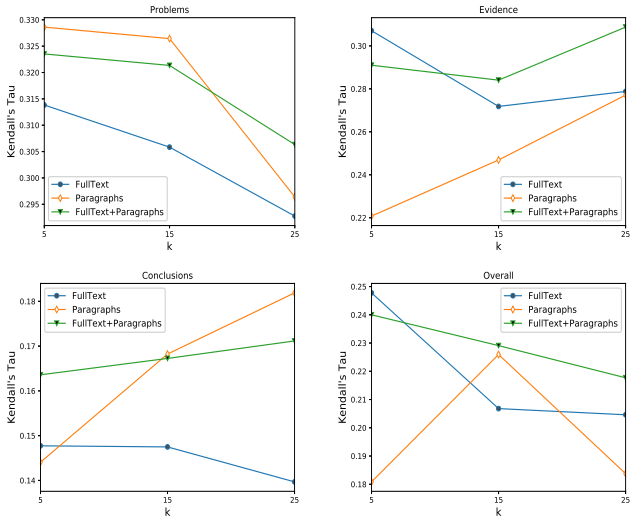

Figure 2: Performance of RankComb(Unigrams+Topics) as a function of the number of topics (x-axis), k, and the level of text granularity (different lines), n. Each graph corresponds to a single dimension. Note: figures are not to the same scale.

Feature analysis: In Table 4 we report the most salient topical features for each rubric dimension. Specifically, for each rubric dimension, we present features that achieved the highest Kendall's-τ correlation when only their values were used for

[6]Using a development set was not practical in our case due to the small size of data set.

Table 4: The most correlative (Kendall's-τ) features for each rubric dimension. Correlation is reported in the brackets.

Pro.	Dif.	Evi.	Und.	Con.	Ref.	Ove.
+Ref-FullText-4 [.283]	+Ref-FullText-4 [.346]	–Ref-FullText-3 [.300]	–Dif-FullText-1 [.353]	+Ref-FullText-4 [.236]	+Und-Paragraphs-3 [.398]	+Ref-FullText-4 [.265]
+Ove-FullText-2 [.271]	–Und-FullText-2 [.303]	+Pro-FullText-5 [.299]	+Ref-Paragraphs-4 [.344]	+Und-FullText-3 [.234]	+Ove-Paragraphs-1 [.386]	–Und-FullText-2 [.256]
+Und-Paragraphs-3 [.255]	–Con-FullText-5 [.300]	+Und-Paragraphs-3 [.299]	–Pro-FullText-3 [.332]	+Ref-FullText-2 [.228]	+Dif-Paragraphs-3 [.373]	FullText-5 [.239]
+Ove-Paragraphs-1 [.253]	+Und-Paragraphs-3 [.299]	+Ove-Paragraphs-1 [.297]	+Und-FullText-2 [.332]	+Dif-FullText-4 [.223]	+Ref-Paragraphs-4 [.373]	+Und-FullText-3 [.235]

ranking. Topical features, extracted from the RubricGuided model, are named according to the following format: "(+/–)Dimension-GranularityLevel-TopicNumber". For example, the topical feature named "+Ref-FullText-4" corresponds to the fourth topic in the RubricGuided model, built using the high scoring ('+') assignments in the References dimension with the FullText granularity level. As for the StandardModel, we simply mention the text granularity and the topic number (e.g., "FullText-4"). As can be seen in Table 4, RubricGuided features are the most effective ones. Specifically, StandardModel-based features are among the most correlative ones only in the case of the Overall dimension. An interesting observation is that some RubricGuided models are highly effective for dimensions that are different from the dimension that was used for building the model. For example, topical features generated using the RubricGuided models, which were learned with the guidance of the References or the Understanding dimensions, are among the top correlated features for all other rubric dimensions.

Case study: We use a case study in order to gain further understanding on why using topical features is effective for automated assessment of complex assignments. For the simplicity of discussion, we focus on the Overall dimension. In Table 5 we present 15 representative terms for each of the 4 most correlative topics according to Table 4. Terms in each topic were extracted as follows. We first extract 100 terms from each topic based on their probability in the topic's multinomial distribution. Then, for each topic, we leave only the terms that do not appear in the other three topics. Finally, we use the 15 terms with the highest probabilities. We do that in order to better distinguish between the different topics. We also present the 15 most correlative terms in the Unigrams baseline ("Unigrams+" and "Unigrams–" are the most positively and negatively correlated terms, respectively).

The first topic presented in Table 5 is built using the high scoring assignments in the References dimension (+Ref-FullText-4). Terms in this topic include general verbs such as "relate", "consider", and "mean". Positive correlation with such terms might suggest that the author has tried to explain his/her thinking. Some other terms are appropriate for the references part of the work. For example, "Merck" refers to the common general reference "Merck Veterinary Manual". The second topic is learned using the low scoring assignments in the Understanding dimension (–Und-FullText-2). This topic tends to include general terms taken directly from the case provided to the student such as "systolic", "wave", and "QRS". Such terms were used in the case to describe the animal's condition and the tests that were performed. Positive correlation with these terms might suggest that the author has put emphasis on details given in the case in order to better support the analysis and diagnosis. The third topic in the table was learned using all assignments (FullText-5). This topic contains terms that reflect novel findings of the work (for example: the term "sibling" might refer to the finding that the animal's weight could be compared with the weight of his healthy sibling). On the other hand, this topic also contains more general terms such as "note", and "found", suggesting that positive attributes were given to those analyses trying to explain signs, history and diagnostic findings in the case. The last topic in the table is learned using the high scoring assignments in the Understanding dimension (+Und-FullText-3). Words in this topic mostly include generic anatomical terms that most students might have needed to use in high scoring explanations, such as: "procedure", "infection", and "septum".

Next, we examine the terms in the Unigrams baseline. Unigrams+ contains the most positively correlated terms in the assignments. Most of the terms, except for "CO", "II", and "infection", are fairly general and might reflect the author integrating and explaining well. Finally, we examine the most negatively correlated unigrams (Unigrams–). Surprisingly, the terms reflect recognizable features of the pathophysiology and correct diagnosis in this case. One explanation for that would be that students scored it negatively because less explanation of alternatives was included by these authors.

Based on this analysis, we can see that topics get a different interpretation in the case of complex assignments. Specifically, they represent patterns in student works, rather than themes. This is the case as all assignments have a common theme.

Methods for construction of topical features: As mentioned previously, a natural way to use topics as features in supervised learning would be to use the distribution of a document over the topics. Yet, we found that in our case using such an approach is not as effective as measuring the distance between the topic distribution and the document distribution over terms. In Table 6 we further explore this finding by comparing different approaches for the construction of topical features. We experiment with the following approaches: (1) Distribution (Dist.): using the distribution of a document over topics. (2) KL: using KL-divergence between the topic distribution and the document distribution over terms (this approach was used throughout the paper). (3) KL-norm: sum

Table 5: Topic examples. 15 representative terms for the four most correlative topics with the performance in the Overall dimension. The two right-most columns correspond to positively and negatively correlated unigrams, respectively.

+Ref-FullText-4		–Und-FullText-2		FullText-5		+Und-FullText-3		Unigrams+		Unigrams–	
Vegas	level	work	obstruct	weight	neutrophils	small	Jan	occur	weak	work	ventricle
understand	echocardiogram	web	negative	VI	neonatal	sign	infection	need	evaluation	hole	serious
TOU	contraction	wave	narrow	ultrasound	IV	sever	function	heard	II	major	change
consider	resistance	tract	lung	space	found	septum	dilation	back	reduce	determine	ST
relate	anesthesia	systolic	large	sibling	elevated	return	differential	get	partial	congenital	result
reason		shift		respiratory		report		sufficient		obstruct	
number		reverse		range		procedure		infection		like	
Merck		QRS		pleural		patent		CO		bypass	
mean		pump		PCO		mild		sign		TOF	
manual		position		note		medical		mean		animal	

Table 6: Comparing the performance (Kendall's-τ) of different approaches for topical feature construction. Only topical features are used and are combined using either FeatureComb or RankComb. Boldface: best result in a column in an inference method block.

Inference	Method	Pro.	Dif.	Evi.	Und.	Con.	Ref.	Ove.
				FeatureComb (Topics)				
CVB	Dist.	.166	.128	**.205**	.272	.055	**.368**	.047
	KL	**.266**	**.188**	.147	**.295**	**.085**	.294	**.134**
	KL-norm	.138	.181	.157	.213	.015	.303	.007
Gibbs	Dist.	.149	.141	**.191**	.110	**.172**	.221	−.028
	KL	**.171**	**.194**	.165	**.281**	.147	.277	**.120**
	KL-norm	.055	.127	.153	.053	.104	**.294**	−.044
				RankComb (Topics)				
CVB	Dist.	.133	.119	.273	.229	.053	.375	.071
	KL	**.294**	**.359**	**.317**	**.392**	**.191**	**.429**	**.249**
	KL-norm	.136	.220	.237	.235	−.005	.386	.081
Gibbs	Dist.	.165	.147	.198	.230	**.227**	.259	−.005
	KL	**.182**	**.257**	.170	**.320**	.154	.315	**.211**
	KL-norm	.102	.173	**.213**	.231	.165	**.327**	.066

normalizing the KL-divergence-based features so that all topical features per topic model would sum up to 1. Furthermore, we examine the performance of these techniques in different scenarios. Specifically, we report the performance of using only topical features (no Unigrams) when combined by using the FeatureComb, and the RankComb methods. We also report the performance of using different approaches for inference of the LDA model, based on different implementations. The first approach is Collapsed Variational Bayes (CVB), implemented as part of the MeTA toolkit. The second approach is Gibbs sampling, implemented as part of the Python Gensim library (radimrehurek.com/gensim/models/ldamodel.html).

Comparing the performance of using document distributions to that of using KL, we can see that the latter is overall more effective. Specifically, using KL outperforms the document distribution approach in the majority of rubric dimensions for all inference and combination methods. A possible explanation for this finding might be attributed to the comparability of features. That is, in the case of document distribution over topics, a non-negative probability must be assigned to each topic, and all probabilities must sum up to 1. Thus, it might be the case where a document is assigned a probability for a topic just in order to satisfy this condition. This is of course not the case when KL is used. We experiment with the KL-norm approach in order to test this hypothesis. Indeed, we can see in Table 6 that the performance of KL drops for most rubric dimensions when normalization is applied. Moreover, it is often the case where using normalization degrades the performance of KL in dimensions where the performance of using topic distributions is lower, and improves the performance of it in the opposite case. Yet, we note further exploration must be done in order to reach conclusive findings regarding this issue. This should include, for instance, exploration of other data sets and other supervised learning algorithms. Such exploration is out of the scope of this paper and is left for future work.

CONCLUSIONS AND FUTURE WORK

We studied the problem of automated assessment of complex assignments and extended a previous work by exploring a new kind of features based on topics. We proposed and studied multiple ways to construct topical features and different ways to combine them with lexical features. Evaluation results on a data set of medical case assignments show that the topical features are generally very effective, however the best approach is to use them in combination with lexical features. Furthermore, the use of labels to guide the extraction of topics is effective and in general, combining topics learned using multiple views of the text data appears to be the most effective and robust. The findings of our study can be directly useful for improving the current methods for automated assessment of complex assignments, thus potentially helping scale up education in subject areas that require them. As our approaches to topic feature construction and combination are all general, they can be used in many different domains.

One limitation of our work is that we used a relatively small data set. Yet, the main reason for which we were not able to obtain a larger data set, as already noted in a previous study [5], is that because without automated assessment tools, it is harder to teach a very large class with such an assignment. An important future work is thus to leverage the proposed methods to automate the grading of complex assignments and thus enable the teaching of larger classes, which in turn will help us collect larger data sets for further evaluation of the proposed techniques. Another limitation of our work is that the evaluation is only limited to one data set. In the future, we should also leverage the generality of the proposed methods to further evaluate them using multiple other data sets of complex assignments to further verify our findings. The ultimate evaluation of the proposed technique will have to be done by building and deploying an intelligent assessment

assistant in a real education environment and obtain feedback from the instructors and learners, an important future direction to further explore.

Acknowledgments. We thank Mary Kalantzis, Maureen McMichael, Matthew Montebello, and Duane Searsmith for their helpful comments. Thanks to Duane Searsmith for helping with the preparation of the data set. This material is based upon work supported by the National Science Foundation under grant numbers 1629161 and 1801652.

REFERENCES

1. Kirsti M Ala-Mutka. 2005. A survey of automated assessment approaches for programming assignments. *Computer science education* 15, 2 (2005), 83–102.
2. Stephen P Balfour. 2013. Assessing Writing in MOOCs: Automated Essay Scoring and Calibrated Peer Review. *Research & Practice in Assessment* 8 (2013), 40–48.
3. David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.
4. Peter W Foltz, Darrell Laham, and Thomas K Landauer. 1999. Automated essay scoring: Applications to educational technology. In *EdMedia: World Conference on Educational Media and Technology*. Association for the Advancement of Computing in Education (AACE), 939–944.
5. Chase Geigle, ChengXiang Zhai, and Duncan C Ferguson. 2016. An exploration of automated grading of complex assignments. In *Proceedings of the Third (2016) ACM Conference on Learning@ Scale*. ACM, 351–360.
6. Arthur C Graesser, Patrick Chipman, Brian C Haynes, and Andrew Olney. 2005. AutoTutor: An intelligent tutoring system with mixed-initiative dialogue. *IEEE Transactions on Education* 48, 4 (2005), 612–618.
7. Thomas Hofmann. 2017. Probabilistic latent semantic indexing. In *ACM SIGIR Forum*, Vol. 51. ACM, 211–218.
8. Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 133–142.
9. Tuomo Kakkonen, Niko Myller, Erkki Sutinen, and Jari Timonen. 2008. Comparison of dimension reduction methods for automated essay grading. *Journal of Educational Technology & Society* 11, 3 (2008).
10. Tuomo Kakkonen, Niko Myller, Jari Timonen, and Erkki Sutinen. 2005. Automatic essay grading with probabilistic latent semantic analysis. In *Proceedings of the second workshop on Building Educational Applications Using NLP*. Association for Computational Linguistics, 29–36.
11. Thomas K Landauer, Darrell Laham, Bob Rehder, and Missy E Schreiner. 1997. How well can passage meaning be derived without using word order? A comparison of Latent Semantic Analysis and humans. In *Proceedings of the 19th annual meeting of the Cognitive Science Society*. 412–417.
12. Claudia Leacock and Martin Chodorow. 2003. C-rater: Automated scoring of short-answer questions. *Computers and the Humanities* 37, 4 (2003), 389–405.
13. Benoit Lemaire and Philippe Dessus. 2001. A system to assess the semantic content of student essays. *Journal of Educational Computing Research* 24, 3 (2001), 305–320.
14. Tie-Yan Liu and others. 2009. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval* 3, 3 (2009), 225–331.
15. Yue Lu, Qiaozhu Mei, and ChengXiang Zhai. 2011. Investigating task performance of probabilistic topic models: an empirical study of PLSA and LDA. *Information Retrieval* 14, 2 (2011), 178–203.
16. Sean Massung, Chase Geigle, and ChengXiang Zhai. 2016. MeTA: a unified toolkit for text retrieval and analysis. *Proceedings of ACL-2016 System Demonstrations* (2016), 91–96.
17. Jon D Mcauliffe and David M Blei. 2008. Supervised topic models. In *Advances in neural information processing systems*. 121–128.
18. Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. 2004. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*. AUAI Press, 487–494.
19. Salvatore Valenti, Francesca Neri, and Alessandro Cucchiarelli. 2003. An overview of current research on automated essay grading. *Journal of Information Technology Education: Research* 2 (2003), 319–330.
20. Yiren Wang, Dominic Seyler, Shubhra Kanti Karmaker Santu, and ChengXiang Zhai. 2017. A Study of Feature Construction for Text-based Forecasting of Time Series Variables. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 2347–2350.
21. Peter Wiemer-Hastings and Arthur C Graesser. 2000. Select-a-Kibitzer: A computer tool that gives meaningful feedback on student compositions. *Interactive learning environments* 8, 2 (2000), 149–169.
22. Xing Yi and James Allan. 2009. A comparative study of utilizing topic models for information retrieval. In *European conference on information retrieval*. Springer, 29–41.
23. Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. 2011. Comparing twitter and traditional media using topic models. In *European conference on information retrieval*. Springer, 338–349.
24. Jun Zhu, Amr Ahmed, and Eric P Xing. 2009. MedLDA: maximum margin supervised topic models for regression and classification. In *Proceedings of the 26th annual international conference on machine learning*. ACM, 1257–1264.