A Fluctuation Smoothing Approach for Unsupervised Automatic Short Answer Grading

Shourya Roy[†] **Sandipan Dandapat**[‡] **Y. Narahari**[§]
[†]Xerox Research Centre India, Bangalore
[‡]Microsoft India, Hyderbad,
[§]Indian Institute of Science, Bangalore
shourya.roy@xerox.com, sadandap@microsoft.com, hari@csa.iisc.ernet.in

Abstract

We offer a fluctuation smoothing computational approach for unsupervised automatic short answer grading (ASAG) techniques in the educational ecosystem. A major drawback of the existing techniques is the significant effect that variations in model answers could have on their performances. The proposed fluctuation smoothing approach, based on classical sequential pattern mining, exploits lexical overlap in students' answers to any typical question. We empirically demonstrate using multiple datasets that the proposed approach improves the overall performance and significantly reduces (up to 63%) variation in performance (standard deviation) of unsupervised ASAG techniques. We bring in additional benchmarks such as (a) paraphrasing of model answers and (b) using answers by k top performing students as model answers, to amplify the benefits of the proposed approach.

1 Introduction

In this paper, we deal with the problem of automatic assessment of students' constructed answers in natural language in an educational environment. In particular, we are interested in *short answers*: a few words to a few sentences long (everything in between fill-in-the-gap and essay type answers (Burrows et al., 2015)) and refer to the task as *Automatic Short Answer Grading* (ASAG). An example ASAG task is shown in Table 1. This is a non-trivial task owing to: linguistic variations (a given answer could be articulated in many different ways); subjective nature of assessment (multiple possible correct answers or no correct answer); lack of consistency in human rating (non-binary scoring on an ordinal scale within a range); etc.

Question	How are overloaded functions differentiated by the compiler?
Model Ans	Based on the function signature. When an overloaded function is called, the compiler will find the function whose signature is closest to the given function call.
Stud#1	It looks at the number, types, and order of arguments in the function call
Stud#2	By the number, and the types and order of the parameters.

Table 1: A question, model answer, and student answers from a computer science course (Mohler and Mihalcea, 2009). These will be used as a running example in the paper.

The number and type of its parameters.
The compiler selects the proper functions to execute based on number, types and order of arguments in the function call.
It selects the proper function to execute based on number, types and order of arguments in the function call.
The compiler selects proper function to execute based on number, types and order of arguments in the function call.
Is based on number, types, and order of arguments in the function call.
Compiler selects proper function to execute based on number, types and order of arguments in the function call.

Table 2: A few other possible model answers to the same question shown in Table 1.

Two recent survey papers by Roy et al. (2015) and Burrows et al. (2015) provide comprehensive views of research in ASAG. Both papers have grouped prior research based on the types of approaches used as well as the extent of human supervision needed. They report that a large fraction of prior work in ASAG uses various textual similarity measures (lexical, semantic etc.) to obtain a similarity value between student and model answers and convert those values to grades appropriately. Such *unsupervised ASAG techniques* largely reduce the need of instructor involvement either for training the ASAG system (as done in supervised ASAG (Sukkarieh et al., 2011; Madnani et al., 2013; Ramachandran et

Proceedings of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications,
pages 82–91, Osaka, Japan, December 12 2016.

al., 2015)) or for providing all possible variations of model answers (e.g. concept mapping based approaches (Burstein et al., 1999; Leacock and Chodorow, 2003)). However, the unsupervised ASAG techniques suffer from two shortcomings which motivate the main contribution of this work. First, variations in model answers can significantly affect their performance. Consider a few *equivalent* model answers in Table 2 of our running example question (shown in Table 1). Replacing the instructor provided model answer with any of these is not expected to change human evaluation of student answers. However doing so make unsupervised ASAG techniques exhibit alarming fluctuations. An intrigued reader may make a forward reference to Figure 3 to see the fluctuation in performance of four unsupervised ASAG techniques against equivalent model answers to the question. This is a pragmatic concern as it shows that unsupervised ASAG techniques' performance significantly fluctuate depending on how model answers are specified. We will address this shortcoming in this paper. The second shortcoming is that there is no standardization in respect of how model answers are written across datasets or even within a dataset. The model answer in Table 1 is more detailed and self-contained than the model answer `"Abstraction and reusability"` for another question `"What are the main advantages associated with object-oriented programming?"` from the same dataset. Owing to such variation and key additional factors such as types of questions (*definition*, *interpretation*, *comparison* etc.), it is unlikely that any one of these ASAG techniques that rely solely on model answers would work well across all types of questions. While this is an important problem to be addressed, we do not deal with it in this paper.

There are two reasons behind the first shortcoming of the unsupervised ASAG techniques. The first is, all words (generally, n-grams) in the model answer are given equal importance to compute the score of a student answer, whereas instructors typically give emphasis on certain key words (or concepts). Second, there can be diverse ways to express a correct answer but the model answer may not capture all possible ways. These lead to the primary contribution of this work, which is to propose a *fluctuation smoothing* approach to make unsupervised ASAG techniques more consistent and effective. The proposed approach is generic; it works across lexical, knowledge based, and vector space based ASAG techniques. It is based on an intriguing finding that short answers to a question typically contain significant lexical overlap among them and such overlapping text portions are usually coupled to the correct answer to the question.[1] Based on similar motivation, Ramachandran et al. demonstrated effectiveness of using students' answers to a question to extract patterns for short answer scoring (2015). They used a graph based approach to extract patterns from groups of questions and their answers towards constructing regular expression alike patterns for short answer scoring. While they used a supervised approach using the extracted patterns as features, our approach is completely unsupervised - hence easier to test on new datasets and in real life. Secondly, we opine that regular expression based features can be constraining towards generalization and real life usage for free text answers.

Our approach is implemented in two steps. In the first step, we propose a variant of the sequential pattern mining problem (Agrawal and Srikant, 1995) to identify n-grams with high *support* that are more common (than the rest of the n-grams) among student answers. In the second step, we deduce our fluctuation smoothing approach by re-weighing different parts of the model answers in a manner proportional to their support values which in turn are used by the unsupervised ASAG techniques. Intuitively, the proposed technique brings in diversity by upweighing certain *scoring* words (n-grams) in the model answer (for our running example, the upweighed words include *call* and *overload*) and introducing new words (n-grams) which are not present in the model answer (e.g. *parameter*, *argument*, *type*) but have possibly affected instructors' scores. In the sequel, we use the words *approach*, *method*, and *technique* synonymously. We also use the phrases *fluctuation smoothing* and *stabilization* synonymously.

Our contributions: The contributions and novelty of this work are summarized below.

- We present vulnerabilities of unsupervised ASAG techniques arising from their sole dependence

[1] We hasten to add that this finding may not always be true. For example, occasionally, instructors design difficult and tricky questions to mislead students towards incorrect answers. However, we have empirically found lexical overlap to pervade the answers in our datasets. Additionally, we propose a *conservative* variant of our fluctuation smoothing approach that does not get affected by common mistakes made by a large number of students.

on the way model answers are written by instructors. These are potential roadblocks for practical adoption of these techniques and motivate our work in this paper.

- We propose a fluctuation smoothing approach, based on sequential pattern mining (Agrawal and Srikant, 1995), for unsupervised ASAG techniques by leveraging textual overlap between students' short answers. (§ 2.2)
- We empirically demonstrate on multiple datasets that the proposed approach improves the overall performance and significantly reduces the variation in performance. We bring in additional benchmarks such as (a) paraphrasing of model answers and (b) using top-k best answers by students as model answers, to convincingly demonstrate the efficacy of the proposed approach (§ 3.3)
- We create and offer a new dataset on high-school English reading comprehension task in a Central Board of Secondary Education (CBSE) school in India. (§ 3.1)

2 Proposed Approach for ASAG

In this section, we first describe unsupervised ASAG techniques building upon popular lexical, knowledge based, and vector space based methods. In the following section, we describe the proposed fluctuation smoothing approach.

2.1 Unsupervised ASAG Techniques

The basic premise of unsupervised ASAG techniques is: higher the similarity between the model and a student answer, higher the score the latter receives. Typically these measures output a score between 0 and 1 which are subsequently scaled by the maximum attainable score for a question. Building on these measures, we present an unsupervised technique leveraging the asymmetric nature of ASAG tasks i.e. student answers to be evaluated against the model answer and not the other way. Given two texts, model answer M and a student answer S, we conduct standard pre-processing operations such as stopword removal and stemming. The score of S with respect to M is then defined as:

$$asym(M, S) = \frac{1}{k}\sum_{i=1}^{k} \max_{\mathbf{s}_j \in S}(sim(\mathbf{m}_i, \mathbf{s}_j)) \tag{1}$$

where $\mathbf{m}_i$ and $\mathbf{s}_j$ are pre-processed n-grams of M and S respectively and k is the number of n-grams in M. For $n = 1$, $\mathbf{m}_i$ and $\mathbf{s}_j$ are words of M and S; and k is the length of M with respect to number of words. $sim(.,.)$ is a textual similarity measure of one of the following types:

- **Lexical**: In this category, we consider lexical overlap (**LO**) between model and student answers. It is a simple baseline measure which looks for exact match for every content words (post pre-processing e.g. stopword removal and stemming).
- **Knowledge based**: These measures employ a background ontology to arrive at word level semantic similarity values based on various factors such as distance between two words, lowest common ancestor, etc. Mohler and Mihalcea (2009) compared eight different knowledge-based measures to compute similarities between words in the model and student answers using Wordnet. We select the best performing one in this category, the measure proposed by Jiang and Conrath (**JCN**) (Jiang and Conrath, 1997) as shown below:

  $$sim(m_i, s_j) = [IC(m_i) + IC(s_j) - 2 \times IC(LCS(m_i, s_j))]^{-1}$$

  where, $LCS(m_i, s_j)$ is the least common subsumer of m_i and s_j in Wordnet (Miller, 1995), $IC(w) = -log\ P(w)$ and $P(w)$ is the probability of encountering an instance of word w in a large corpus.
- **Vector space based**: In this category, we have chosen one of the most popular measures of semantic similarity, namely, Latent Semantic Analysis (**LSA**) (Landauer et al., 1998) trained on a Wikipedia dump. We also consider the recently popular word2vec tool (**W2V**) (Mikolov et al., 2013) to obtain vector representation of words which are trained on 300 million words of Google news dataset and are of length 300. Both LSA and W2V build on several related ideas towards capturing importance of context to obtain vector representation of words e.g. the distributional hypothesis "Words will

occur in similar contexts if and only if they have similar meanings" (Harris, 1968). Similarity between words is the measured as the cosine distance between corresponding word vectors in the resultant vector space using the well known dot product formula.

2.2 Fluctuation Smoothing

2.2.1 Intuition

In unsupervised ASAG techniques, each student answer is compared against the model answer **independently** to arrive at a score indicating goodness of the student answer. These techniques give equal importance to each word of the model answer, whereas an instructor would often look for certain key words (equivalently, concepts, phrases etc.) such as the word *signature* in the model answer shown in Table 1. Second, there could be alternate ways of expressing equivalent correct answers different from the instructor provided model answer, such as the notion of *(function) signature* being rightly expressed by *number, types and order of arguments* for our running example. Both of these phenomena contribute towards fluctuations in performance of unsupervised ASAG techniques.

Towards addressing this issue, we exploit a fact that student answers to a question, as a collection, are expected to share more **commonalities** than any random collection of text snippets. Furthermore, we observe that such commonalities are likely to influence instructor given scores irrespective of whether or not they are a part of the model answer. Hence, extracting and incorporating these commonalities in an ASAG technique should help smoothing the fluctuation exhibited by unsupervised ASAG techniques relying solely on instructor given model answers. In the first step of the proposed two-step fluctuation smoothing approach, we apply a variant of the *sequential pattern mining* algorithm (Agrawal and Srikant, 1995) to identify frequent common n-grams in student answers. In the second step, we either upweigh n-grams present in the model answer or add new n-grams, with weights proportional to their support.

2.2.2 Technique

Sequential patterns in the context of text has been used to capture non-contiguous sequence of words for classification and clustering (Jaillet et al., 2006). Prior work has reported that for such tasks, sequential patterns have more reliable statistical properties than commonly used lexical features e.g. n-grams in NLP domain (Sebastiani, 2002). For short answers too, our observation was that sequential patterns are more statistically significant and less noisy than n-grams and hence developed our approach based on sequential patterns. The following two steps, namely, *mining high support n-grams* and *updating unsupervised ASAG scoring* are repeated for all questions.

Step 1: Mining High Support n-grams

The objective of this step is to extract commonly occurring patterns and quantify the notion of commonalities using *support*:

1. A student answer (s_i) is converted to a sequence of words ($w^i_1, w^i_2, \ldots, w^i_k$) by performing standard pre-processing operations such as stopword removal and stemming as well as task specific pre-processing viz. question word demoting.

2. An n-gram p, is a sequence of n consecutive tokens from s_i i.e. $p = w^i_j, w^i_{j+1}, \ldots, w^i_{j+n-1}$. It is imperative to note that these n-grams are not n consecutive words from the original model answer. Hence they may not make semantic sense when considered in isolation. For example, a few frequent n-grams for various n of our running example are `(number, type, order, argument, call)`,`(base, number, type, order)`, `(proper, execute, base)`.

3. The support of p is defined as $sup(p)=\frac{|\{s_i:p\in s_i\}|}{|\{s_i\}|}; \forall i$ i.e. the fraction of student answers containing p. Connecting to our intuition, n-grams with high support are commonalities among answers we are looking for.

Step 2: Updating Unsupervised ASAG Scoring

1. Sort n-grams in decreasing order of their support.

2. For every n-gram p assign a weight w, where $w = sup(p) \times f + count(p, M)$ and the function $count(.,.)$ returns the number of times the n-gram p appears in M (0, if it does not). The multiplicative factor f ensures differential weighing of the same n-gram appearing in multiple questions. The intuition being in the longer model answer, an n-gram obtained the same support amidst larger number of n-grams hence should have higher weights. Experimentally we find that average length of answers gives the best performance across various datasets.

3. Update the asymmetric similarity measure as below by incorporating the weights and with appropriate normalization.

$$asym(M, S) = \frac{1}{\sum_{i=1}^{k'} w_i} \sum_{i=1}^{k'} w_i \max_{\mathbf{s}_j \in S}(sim(\mathbf{m}_i, \mathbf{s}_j)) \tag{2}$$

where w_i is the weight of $\mathbf{m}_i$ and k' is the new length of M with respect to the number of n-grams.

To illustrate, for our running example's model answer, some of the words which get higher weights are (in decreasing order of weights) are `function, overloaded, call, compiler` etc. whereas new words such as `type, argument, number, order, parameter, proper` etc. get added to the model answer. In a *conservative* smoothing variant, we increase the support of n-grams which has $count(.,.) > 0$ i.e. only the word which appear in the instructor provided model answer. We call this conservative as it prevents introducing n-grams which might arise from a common misconception among large number of students.

3 Performance Evaluation

In this section, we present empirical answers to the following questions. Given n *valid* model answers to a question, what is the extent of fluctuation exhibited by the unsupervised ASAG techniques and smoothing achieved by the proposed approach (**Fluctuation Smoothing Performance**)? Secondly, are there alternate ways of bringing in diversity instead of relying on single instructor provided model answer? How do they compare against the proposed approach of leveraging student answers as a source of diversity (**Aggregate Performance**)?

3.1 Datasets

We evaluate the proposed fluctuation smoothing technique on three datasets. (i) **CSD:** This is one of the earliest ASAG datasets consisting of 21 questions with 30 student answers evaluated each on a scale of 0-5 from an undergraduate computer science course (Mohler and Mihalcea, 2009). Student answers were independently evaluated by two annotators and automatic techniques are measured against their average. (ii) **X-CSD:** This is an extended version of CSD with 81 questions by the same authors (Mohler et al., 2011). (iii) **RCD:** We created a new dataset on a reading comprehension assignment for Standard-12 students in Central Board of Secondary Education (CBSE) in India. The dataset contains 14 questions answered by 58 students. The answers were graded by two expert human raters based on model answers, again on a scale of 0-5 and optional scoring scheme.

All datasets have less than (total number of questions × total number of students) answers as presumably some students did not answer some questions. We mark such missing entries as "No Answer" and corresponding groundtruth scores as zero.

3.2 Performance Metric

A wide variety of metrics have been used in the ASAG literature with no standard one. Pearson's r and quadratic weighted Cohen's κ are the two most popular one which we use, though the suitability of the former for ASAG has been questioned (Mohler and Mihalcea, 2009). For every question we compute r and κ and reported values are average over all questions for each dataset.

3.3 Quantitative Results

Fluctuation Smoothing Performance: Towards generating valid model answers for each question, we select those student answers which were graded as perfect 5/5 by the instructor with respect to the model answer. We consider each of them (and the instructor provided model answer) as a model answer in turn and score all student answers using the unsupervised ASAG techniques - with and without fluctuation smoothing (i.e. Equation (1) and Equation (2) respectively). For our running example, there are 20 perfect scoring student answers (some of those are shown as examples in Table 2). Resultant Pearson's r of scores and associated fluctuations can be seen in Figure 3. Firstly, it is clear that all techniques suffer from fluctuation in performance as they rely solely on the model answers to score student answers (red dashed lines). Secondly, it is visually evident that the fluctuation smoothed technique (blue solid lines) shows significantly less variability than respective non-smoothed ones. Corresponding standard deviation (SD) values clearly support the observation. Table 3 shows aggregate SD in performance of all methods for three datasets under no smoothing and the proposed smoothing technique along with its conservative variant. Each big cell, bordered with double lines, shows the SD numbers for a (technique, metric) combination. It is evident that the proposed smoothing technique reduces fluctuations across all settings, often significantly, going up to 63% for (LSA,CSD) combination. This is a clear demonstration of the efficacy of the proposed fluctuation smoothing approach. Expectedly, the conservative approach gives less smoothing effect supporting our intuition that student answers bring in diverse views not captured in instructor provided model answers.

Aggregate Performance: At the core of it, the proposed fluctuation smoothing technique is essentially synthesizing a *super* model answer for every question. It does so by bringing in additional and diverse snippets of possible correct answers from the corresponding student answers. Next we ask the question if there are other possible ways of bringing in diverse views of correct answers. Towards that we consider the following two possible variants:

Paraphrasing: Using freely available paraphrase generators [2] we created four different paraphrases for each model answer and applied the proposed smoothing technique. For example, for the model answer of our running example (Table 1), one paraphrased model answer is "`Predicated on the function signature. When an overloaded function is called, the compiler will find the function whose signature is most proximate to the given function call.`"

Answers of top-k Students: We selected five best performing students for each dataset and their answers are chosen as model answers.Note that their answers to all questions may not be necessarily correct, but their overall performances were better than the rest of the students.

Each of these options is used as a source of diversity to reweigh different parts of the model answer (as shown in Equation 2). Figures 1 and 2 show Pearson's r and Cohen's κ for CSD with unsupervised ASAG techniques. Default refers to the scenario when no diversity source is used (Equation 1) and the remaining are with various diversity options described. Across all settings, we note that average performance of unsupervised techniques are comparable. The "stabilized-proposed" option emerges as the best, albeit jointly in some cases, though the differences are significant. However, by bringing in diversity we can reduce fluctuation in performance as we already demonstrated in Table 3 and Figure 3. We make similar observations on XCSD and RCD.

Finally, we obtain better r value for CSD compared to the values reported by Mohler and Mihalcea in their paper (2009) for these two techniques. We believe that this is owing to bigger size of Wikipedia corpus on which LSA was trained (compared to what it was in 2009) as well as our asymmetric similarity measure (compared to their symmetric measure) and possibly difference in preprocessing (we used WordnetLemmatizer instead of more commonly used Porter stemmer etc.).

4 Prior Art

Recent survey papers by Roy et al.(2015) and Burrows et al. (2015) provide comprehensive views of research in ASAG. Both of them have grouped prior research based on types of approaches used as

[2] http://paraphrasing-tool.com/; http://www.goparaphrase.com/; https://spinbot.com/; http://paraphrase.generalconnection.com/

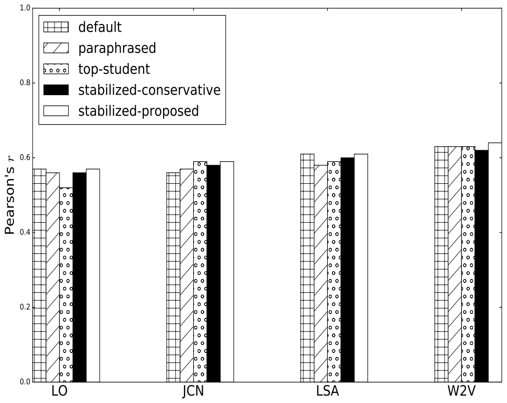

Figure 1: Aggregate Pearson's r for CSD.

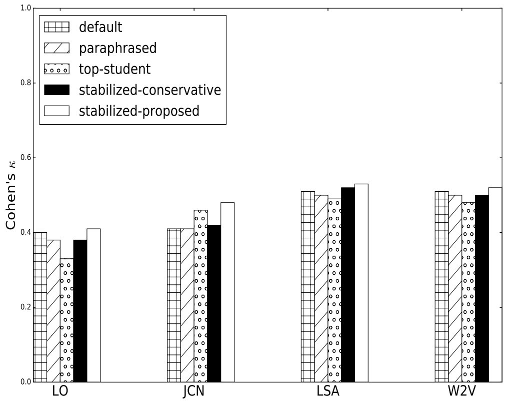

Figure 2: Aggregate Cohen's κ for CSD.

		Pearson's r			Quadratic κ		
		CSD	XCSD	RCD	CSD	XCSD	RCD
LO	no smoothing	0.11	0.11	0.10	0.12	0.14	0.06
	all students - cons	0.11	0.11	0.10	0.12	0.14	0.06
	all students	0.05	0.05	0.06	0.07	0.09	0.04
JCN	no smoothing	0.10	0.10	0.12	0.09	0.10	0.07
	all students - cons	0.09	0.09	0.11	0.09	0.10	0.06
	all students	0.05	0.04	0.06	**0.05**	0.05	0.02
LSA	no smoothing	0.11	0.08	0.09	0.12	0.12	0.03
	all students - cons	0.09	0.06	0.09	0.10	0.09	0.02
	all students	**0.04**	**0.03**	**0.05**	**0.05**	**0.04**	**0.01**
W2V	no smoothing	0.11	0.10	0.09	0.13	0.14	0.04
	all students - cons	0.09	0.08	0.08	0.10	0.11	0.04
	all students	0.05	0.04	**0.05**	0.06	0.06	**0.01**

Table 3: Aggregate standard deviation (SD) in performance (lower the better) of all techniques under the proposed smoothing technique (*all students*), its conservative variant (*all students-cons*) against the default *no smoothing* option. Best performances for each (metric, dataset) combination are emphasized (**bold**).

well as extent of human supervision needed. In this section, we cover relevant unsupervised ASAG techniques (e.g. lexical, knowledge-based, vector space etc.). Among the **lexical** measures, Evaluating Responses with BLEU (ERB) due to Perez et al. (2004) adapted a popular evaluation measure for machine translation, BLEU (Papineni et al., 2001) for ASAG with a set of NLP techniques such as stemming, closed-class word removal, etc. This work initially appeared as part of an ASAG system, *Atenea* (Alfonseca and Pérez, 2004) and later as *Willow* (Pérez-Marín and Pascual-Nieto, 2011). Mohler and Mihalcea (2009) conducted a comparative study of different semantic similarity measures for ASAG including **knowledge-based** measures using Wordnet as well as **vector space-based** measures such as LSA (Landauer et al., 1998) and Explicit Semantic Analysis (ESA) (Gabrilovich and Markovitch, 2006). LSA has remained a popular approach for ASAG and been applied in many variations (Graesser et al., 2000; Kanejiya et al., 2003; Klein et al., 2011). Lexical and semantic measures have been combined to validate natural complementarity of syntax and semantics for ASAG tasks (Perez et al., 2005). A combination of different string matching and overlap techniques were studied by Guetl on a small scale dataset (2008). Gomaa and Fahmy compared several lexical and corpus-based similarity algorithms and their combinations for grading answers in 0-5 scale (2012). This central reliance on instructor provided model answer of all these techniques leads to significant variation in performances even when it is replaced by another equivalent model answer.

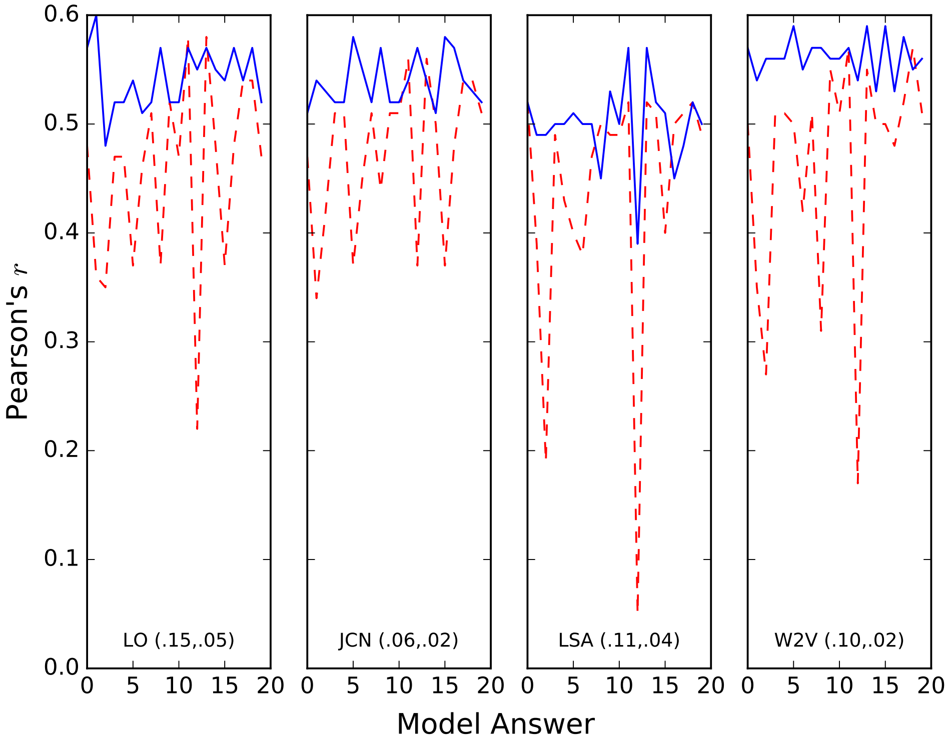

Figure 3: Fluctuation in Pearson'r of unsupervised ASAG techniques. It can be seen that unsupervised ASAG techniques with proposed fluctuation smoothing (blue solid lines) are significantly flatter than without smoothing (red dotted lines). Values within parenthesis are standard deviation (SD) value without and with smoothing respectively.

Dzikovska et al. conducted a 5-way Student Response Analysis challenge as a part of SemEval-2013 (2013). However, the task had more emphasis on giving feedback on student answers possibly using textual entailment techniques. ASAG can be seen as related to the task of textual entailment (Levy et al., 2013) especially when the elements of a model answer are compared against student answers. Textual entailment has seen a notable amount of work with a variety of shared tasks (Marelli et al., 2014; Xu et al., 2015) and vector space models have been considered as well (Zhao et al., 2015). Although the techniques are mostly supervised, the tasks and features used in such systems are relevant for consideration.

5 Conclusion and Future Work

In this paper, we introduced a fluctuation smoothing computational approach for unsupervised ASAG techniques in educational ecosystem. It addressed a major drawback - the significant effect that variations in model answers could have on the performance of these techniques. We empirically demonstrated with experimentation that the proposed approach at least retains (and in most cases, improves) the overall performance and significantly reduces the variation in performance of unsupervised ASAG techniques. We introduced additional benchmarks such as (a) paraphrasing of model answers and (b) using top-k best answers by students as model answers for comparing against the proposed approach. We intend to contribute a new dataset on high-school English reading comprehension task in a Central Board of Secondary Education (CBSE) school in India. In future, we would like to evaluate the proposed technique on other standard datasets such as SemEval-2013 dataset and Kaggle ASAP dataset (Kaggle, 2015).

6 Acknowledgment

We thank Oliver Adams and Ajay Nagesh for helpful discussions during ideation and early implementation of parts of this work.

References

Rakesh Agrawal and Ramakrishnan Srikant. 1995. Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*, pages 3–14, Washington, DC, USA. IEEE Computer Society.

Enrique Alfonseca and Diana Pérez. 2004. Automatic assessment of open ended questions with a bleu-inspired algorithm and shallow nlp. In *EsTAL*, volume 3230 of *Lecture Notes in Computer Science*, pages 25–35. Springer.

Steven Burrows, Iryna Gurevych, and Benno Stein. 2015. The eras and trends of automatic short answer grading. *International Journal of Artificial Intelligence in Education*, 25(1):60–117.

Jill Burstein, Susanne Wolff, and Chi Lu. 1999. Using lexical semantic techniques to classify free-responses. In *Breadth and depth of semantic lexicons*, pages 227–244. Springer.

Myroslava O Dzikovska, Rodney D Nielsen, Chris Brew, Claudia Leacock, Danilo Giampiccolo, Luisa Bentivogli, Peter Clark, Ido Dagan, and Hoa T Dang. 2013. Semeval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge. Technical report, DTIC Document.

Evgeniy Gabrilovich and Shaul Markovitch. 2006. Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge. In *AAAI*, pages 1301–1306. AAAI Press.

Wael H Gomaa and Aly A Fahmy. 2012. Short answer grading using string similarity and corpus-based similarity. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 3(11).

Arthur C. Graesser, Peter M. Wiemer-Hastings, Katja Wiemer-Hastings, Derek Harter, and Natalie K. Person. 2000. Using latent semantic analysis to evaluate the contributions of students in autotutor. *Interactive Learning Environments*, 8(2):129–147.

Christian Guetl. 2008. Moving towards a fully automatic knowledge assessment tool. *iJET*, 3(1).

Zellig Harris. 1968. *Mathematical Structures of Language*. John Wiley and Son, New York.

Simon Jaillet, Anne Laurent, and Maguelonne Teisseire. 2006. Sequential patterns for text categorization. *Intelligent Data Analysis*, 10(3):199–214.

J.J. Jiang and D.W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proc. of the Int'l. Conf. on Research in Computational Linguistics*, pages 19–33.

Kaggle. 2015. The hewlett foundation: Short answer scoring. `http://www.kaggle.com/c/asap-sas`. Online; accessed July 16, 2016.

Dharmendra Kanejiya, Arun Kumar, and Surendra Prasad. 2003. Automatic evaluation of students' answers using syntactically enhanced lsa. In *Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing - Volume 2*, pages 53–60.

Richard Klein, Angelo Kyrilov, and Mayya Tokman. 2011. Automated assessment of short free-text responses in computer science using latent semantic analysis. In *ITiCSE*, pages 158–162. ACM.

T. K. Landauer, P. W. Foltz, and D. Laham. 1998. An Introduction to Latent Semantic Analysis. *Discourse Processes*, 25:259–284.

Claudia Leacock and Martin Chodorow. 2003. C-rater: Automated scoring of short-answer questions. *Computers and the Humanities*, 37(4):389–405.

Omer Levy, Torsten Zesch, Ido Dagan, and Iryna Gurevych. 2013. Recognizing partial textual entailment. In *ACL (2)*, pages 451–455. The Association for Computer Linguistics.

Nitin Madnani, Jill Burstein, John Sabatini, and Tenaha OReilly. 2013. Automated scoring of a summary writing task designed to measure reading comprehension. In *Proceedings of the 8th workshop on innovative use of nlp for building educational applications*, pages 163–168. Citeseer.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 2014. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. *SemEval-2014*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Michael Mohler and Rada Mihalcea. 2009. Text-to-text semantic similarity for automatic short answer grading. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pages 567–575.

Michael Mohler, Razvan C. Bunescu, and Rada Mihalcea. 2011. Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In *ACL*, pages 752–762.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. Bleu: a method for automatic evaluation of machine translation. Technical report, IBM Research Report.

Diana Perez, Enrique Alfonseca, and Pilar Rodríguez. 2004. Application of the bleu method for evaluating free-text answers in an e-learning environment. In *LREC*. European Language Resources Association.

Diana Perez, Alfio Gliozzo, Carlo Strapparava, Enrique Alfonseca, Pilar Rodriguez, and Bernardo Magnini. 2005. Automatic assessment of students' free-text answers underpinned by the combination of a BLEU-inspired algorithm and latent semantic analysis. In *Proceedings of the Eighteenth International Florida Artificial Intelligence Research Society Conference, FLAIRS*, Clearwater Beach, FL, United states.

Diana Pérez-Marín and Ismael Pascual-Nieto. 2011. Willow: a system to automatically assess students free-text answers by using a combination of shallow nlp techniques. *International Journal of Continuing Engineering Education and Life Long Learning*, 21(2-3):155–169.

Lakshmi Ramachandran, Jian Cheng, and Peter Foltz. 2015. Identifying patterns for short answer scoring using graph-based lexico-semantic text matching. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 97–106.

Shourya Roy, Y. Narahari, and Om D. Deshmukh. 2015. A perspective on computer assisted assessment techniques for short free-text answers. In *Computer Assisted Assessment. Research into E-Assessment*, pages 96–109. Springer.

Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47.

Jana Z Sukkarieh, Ali Mohammad-Djafari, Jean-Franc¸ois Bercher, and Pierre Bessie´ re. 2011. Using a maxent classifier for the automatic content scoring of free-text responses. In *AIP Conference Proceedings-American Institute of Physics*, volume 1305, page 41.

Wei Xu, Chris Callison-Burch, and William B Dolan. 2015. Semeval-2015 task 1: Paraphrase and semantic similarity in twitter (pit). *Proceedings of SemEval*.

Jiang Zhao, Man Lan, Zheng-Yu Niu, and Yue Lu. 2015. Integrating word embeddings and traditional nlp features to measure textual entailment and semantic relatedness of sentence pairs. In *IJCNN*, pages 1–7. IEEE.