Feature Engineering and Ensemble-Based Approach for Improving Automatic Short-Answer Grading Performance

Archana Sahu and Plaban Kumar Bhowmick

***Abstract*—In this paper, we studied different automatic short answer grading (ASAG) systems to provide a comprehensive view of the feature spaces explored by previous works. While the performance reported in previous works have been encouraging, systematic study of the features is lacking. Apart from providing systematic feature space exploration, we also presented ensemble methods that have been experimentally validated to exhibit significantly higher grading performance over the existing papers in almost all the datasets in ASAG domain. A comparative study over different features and regression models toward short-answer grading has been performed with respect to evaluation metrics used in evaluating ASAG. Apart from traditional text similarity based features like WordNet similarity, latent semantic analysis, and others, we have introduced novel features like *topic models* suited for short text, *relevance feedback* based features. An ensemble-based model has been built using a combination of different regression models with an approach based on *stacked regression*. The proposed ASAG has been tested on the University of North Texas dataset for the regression task, whereas in case of classification task, the student response analysis (SRA) based ScientsBank and Beetle corpus have been used for evaluation. The grading performance in case of ensemble-based ASAG is highly boosted from that exhibited by an individual regression model. Extensive experimentation has revealed that feature selection, introduction of novel features, and regressor stacking have been instrumental in achieving considerable improvement in performance over the existing methods in ASAG domain.**

***Index Terms*—Automatic short answer grading, topical similarity, relevance feedback, stacked ensemble, lexical overlap, semantic similarity.**

I. Introduction

THE rapid proliferation of Massive Open Online Courses (MOOCs) has spawned different challenges to scale out to a wider mass of students. One such challenge is to grade free text answers provided by the students. Manual grading mode is infeasible to adopt given the operational scale. This particular need rekindled the interest in programmatic or automatic grading of free text answers by adopting text processing techniques. *Automatic Short-Answer Grading* (ASAG) is the procedure of assigning grades to student provided free-text answers either by comparing it with corresponding model answers or pattern-based answers extracted from student answers [1], [2].

The primary challenge in ASAG is to deal with variations in surface representations of key concepts in student answer and model answer pairs. In many cases, the student answers are syntactic and lexical variations of model answers. The answer pairs may contain synonyms, polysemous words and statements that are paraphrases of each other.

This can be shown by the following illustration of a question:

Example I.1: **Q:** How do you delete a node from a binary search tree?

MA: Find the node, then replace it with the leftmost node from its right subtree (or the rightmost node from its left sub-tree).

SA: If the node has no children, delete it right away, otherwise, put either the furthest right node on the left side or the furthest left node on the right side in that place and perform the above on that node to guarantee that its children get handled properly.

The problem of finding semantic similarity of a pair of strings has been well studied in NLP literature [3]. Consequently, a variety of such similarity measures have been used in different ASAG Systems[4], [5].

In this paper, we present a comprehensive and structured study of different text similarity measures applied in ASAG problem. With the objective of improving state-of-art performance in this research domain, a set of novel features and methods have been proposed. The ASAG problem can be viewed in two ways:

- **Regression task:** The student answer is assigned a mark/grade based on the extent of similarity with the corresponding model answer.
- **Classification/labeling task:** The student answer is classified into one of the categories from among the set of categories i.e., *'correct'*, '$partially\_correct\_incomplete$', *'contradictory'*, *'irrelevant'*, 'non_domain', based on its similarity with the corresponding model answer.

With reference to the challenges posed by the ASAG problem, following are our contributions:

(1) **Feature grouping:** Systematic study of different classes of text similarity measures in the context of short answer grading has been presented.

Manuscript received May 21, 2018; revised January 2, 2019; accepted January 25, 2019. Date of publication February 6, 2019; date of current version March 18, 2020. *(Corresponding author: Archana Sahu.)*

The authors are with the Centre for Educational Technology, Indian Institute of Technology Kharagpur, Kharagpur 721302, India. (e-mail: sahuarchana7@gmail.com; plaban@cet.iitkgp.ac.in).

Digital Object Identifier 10.1109/TLT.2019.2897997

(2) **Novel feature set**: A set of new features for the task of ASAG has been introduced which includes:
- Relevance feedback based features with feedback from all student answers and feedback from the least scoring student answers
- Topic-modelling features relevant to short-answer grading
- Information retrieval motivated feature
- Inverse document frequency (IDF) based overlap feature

(3) **Feature and Model analysis:**
- Optimal feature selection has been achieved through univariate feature selection strategies.
- A comparison of the performance of grading models, developed with different regression techniques has been carried out.
- A stacked ensemble of grading models based on different regression techniques has been used for ASAG in regression task. Its capability to create out-of-sample predictions proves useful to capture distinct data regions where each individual regression model performs the best.
- Feature ablation study is performed to indicate effect of the novel features in grading/labeling model for regression and classification task.

II. Related Work

A number of approaches have been proposed for automatic grading of short answers.

C-rater[6] generates a canonical representation through Predicate Argument Structures (PAS) for each of the student answers and the corresponding model answers that is free from specific syntactic or morphological forms. The canonical representations of the student answer and model answer generated are then compared to each other using rule-based algorithm. Accordingly, C-rater generates a score for the student answer, which represents the ability of the student to cover maximum number of concepts discussed in the model answer.

To grade short-answers, different approaches rely on measuring the semantic overlap between the student answer and model answer using semantic similarity measures such as knowledge-based and corpus-based measures [4], [5], [7].

A machine-learning model with features like alignment of dependency graphs of student answer and model answer pairs and lexical semantic similarity has been employed for short-answer grading [5].

Features such as Stemmed matching, Levenshtein matching, Parts of Speech (POS) matching, WORDNET relationships such as synonymy, antonymy, hypernymy, hyponymy are taken into account while computing the alignment scores. The syntax-aware alignment scores have been combined with scores obtained from semantic similarity measures such as: Knowledge based measures namely, the WORDNET based similarity measures and Corpus-based measures such as Latent Semantic Analysis (LSA) [7].

A neural architecture has been proposed for ASAG in [8] that comprises of three neural network blocks: Siamese bidirectional LSTMs (bi-LSTMs), EMD (Earth mover's distance) pooling layer and regression layer. A student answer is assigned a grade/score by solving an optimization problem using the above mentioned neural network blocks.

An ASAG technique based on ensemble of two classifiers and domain adaptation principle is designed to grade student answers [9]. Canonical Correlation Analysis (CCA) algorithm based on domain adaptation ensures there is maximum correlation between the projected features of student answers belonging to both the source and target domains. The projected features from the source domain train the second classifier which further predicts the labels for target domain student answers. TF-IDF features of a pseudo-training data pool created from confidently predicted student answers from target domain, is used to train the first classifier. An ensemble of the first classifier and second classifier is used to predict the labels of the remaining student answers from the target domain.

Domain adaptation technique has also been used sparingly in [10]. The ASAG system is mainly a logistic regression model that is trained using lexical similarity features [11].

III. Research Gaps

As reviewed in the previous section, prior research works have followed a text-similarity approach which focus on the standard measures of semantic similarity namely knowledge-based measures, corpus-based measures (LSA) [5], relevance feedback-based features [4]. Recently, word-embedding features, lexical overlap-based approaches have also been explored [12], [13]. In this section, we point out the research gaps based on discussion in previous section.

- A number of text-similarity approaches have been employed for ASAG, but without any category-wise grading performance analysis of features.
- Typical answer size being small, previous works do not consider traditional topic models to be that effective. However, topic modeling over short text has not been explored.
- Though relevance feedback based features have been used for ASAG, variants of the same have not been explored further.
- Though different features have been proposed, feature engineering concerning selection of appropriate feature set has not been explored much.

IV. Research Design

In the research framework, we discuss various text-similarity measures used as features, the asssociated feature extraction techniques and the grading models used in this work.

A. Problem Definition

Automatic Short-answer grading ($ASAG$) problem has been modelled as a supervised machine-learning problem with the following input and output specifications:

Input: A pair of short-answers (MA, SA) representing model answer (MA) and student answer (SA) to a question (Q)

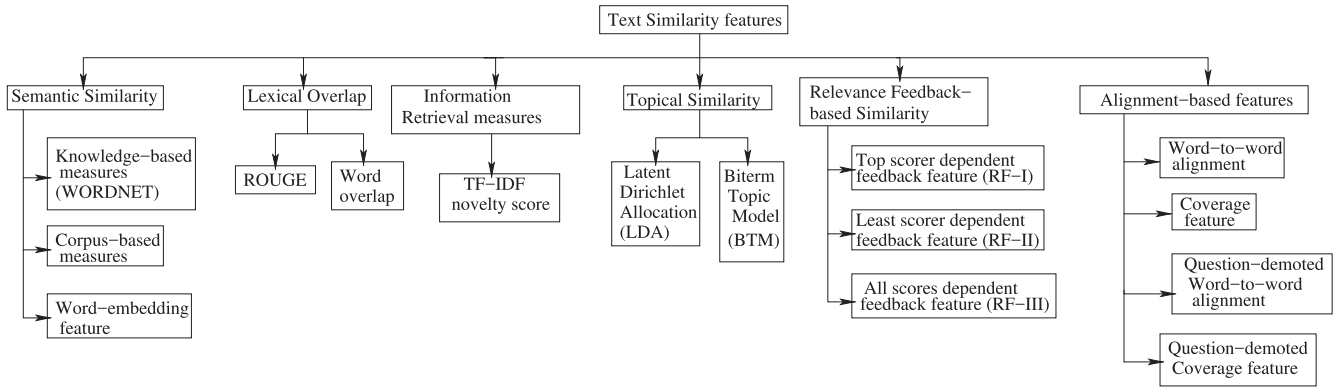


Fig. 1. Classification of text smilarity measures (features) for an answer-pair to train the ASAG model.

Output: Assignment of grade to student answer (Regression task) or assignment of label (Classification task) based on the extent of similarity between the student answer and the respective model answer

The problem of $ASAG$ (regression task) can be mathematically formulated as follows:

- The problem of short-answer grading has been modeled as a supervised regression problem, where the regression model is trained with a data set of <S, M, g> tuple where S = Student answer, M = Model answer, $g \in \mathbb{R}$ = grade assigned to S.
  Definition 1 (Similarity Vector): $\vec{x} = x_1, x_2, ..., x_n$ is defined to be an n-dimensional similarity vector obtained by computing similarity between <S,M> based on n similarity metrics where $x_i = \mu_i(S, M)$.
  Based on the similarity values between every student answer and its corresponding model answer considering different similarity metrics, every student answer in training data is converted to a vector representation.
  The objective is to learn a regression model.
  $Y = f(\vec{X}, \vec{w})$
  where $\vec{X}$ refers to the input feature vector obtained by considering vectors for all the student answers in the training set and $\vec{w}$ are the regression coefficients/ model parameters to be estimated.
- **Evaluation metrics:** The grading performance of the regression model is measured using the following evaluation metrics:
  1) Root Mean Squared Error (RMSE) [14]
  2) Pearson Correlation Coefficient ρ [15]

The problem of $ASAG$ (Classification Task) can be modeled as follows:

- A number of machine learning (ML) algorithms for classification involve representation of a score to a category *k* for every data instance *i*, as shown below:

$$score(X_i, k) = \beta_k \cdot X_i \tag{1}$$

  where
  X_i: feature vector for data instance i
  β_k: vector of weights corresponding to category $kscore(X_i, k)$: score obtained on assigning instance i to category k.
  The training phase involves training a classifier with the features of various data instances in the training set. This leads to determination of optimal weights β_k for each category k, that help in prediction of category k for each of the data instances in the test set. The category for which the score obtained is the highest is predicted for the concerned data instance.

$$k^* = \underset{i}{\operatorname{argmax}}\, score(X_i, k) \tag{2}$$

  Evaluation metrics that are used here are:
  1) Weighted-average F1 score [16]
  2) Macro-average F1 score[17]

B. Feature Extraction

A set of text-similarity features ranging from conventional Knowledge-Similarity measures to the more recent Biterm Topic Model [18], word-embedding features [19] are extracted for each of the pairs of student answer and model answer. Classification of the similarity measures used in the current study is shown in Fig. 1.

1) Semantic Similarity Features: The semantic similarity measures considered in this study can be classified into three categories as described below:

- **Knowledge-based measures:** Knowledge-based measures are the measures that quantify the degree of similarity using information drawn from semantic networks. WordNet [3] has been used widely for detecting the semantic similarity of words.
  The eight knowledge-based measures of semantic similarity namely shortest path (PATH), Leacock & Chodorow (LCH), Wu & Palmer (WUP), Resnik (RES), Lin, Jiang & Conrath (JCN), Lesk and Hirst & St.Onge (HSO) are computed between a pair of short-answers using a scoring function as mentioned in [7].
- **Corpus-based features:** In contrast to knowledge-based measures, the corpus-based measures [5], [7], for example, Latent Semantic Analysis (LSA) [20] do not require any encoded understanding of either the vocabulary or the grammar of a text's language. They prove effective in scenarios where robust language-specific resources (e.g. WordNet) may not be available.

The LSA model in the current study has been trained using Wikipedia articles relevant to the domains of the student answers in the data set to be described in Section V. It is used to derive the vector representations for each pair of student answer and model answer.
Cosine similarity is used to determine the similarity between those vector representations.

- **Word-embedding feature:** This feature refers to the vector representation of words, considering the words occurring in the context to be the elements of the vector. There are two approaches regarding construction of context vectors of a word [21], namely, 1) Context Count modeling and 2) Context Predictive modeling.
The context count modeling approaches such as LSA, are not resilient to data sparsity problem and thus may not be suitable for short length of answers. This motivates us to use a Context Predictive model such as $CBOW$ or Skip-gram.
The training corpus that is used to develop CBOW model is collected by forming sequences of sentences from each of the Wikipedia articles aligned to the grading/labeling corpus. Word vectors are constructed for each of the words in a pair of answers belonging to the dataset using the trained $CBOW$ or *Skip-gram* model. The mean of the word vectors in each of the answers in a pair is obtained and cosine similarity between the pair of mean vectors is used as the semantic similarity score between the pair of student answer and model answer.

2) Lexical Overlap Features: These features are computed for a pair of answers based on common occurrence of lexical units. There are certain cases where the student answer contains redundant points in addition to the points relevant to the answer (as in model answer) of the question asked.

Example IV.1: **Q:** What is a queue and what are the main operations in queue?

MA: A data structure that stores elements following the first-in-first-out principle. The main operations in a queue are enqueue and dequeue.

SA1: A queue is a linear, first-in-first-out data structure. Data must be accessed in the same order it was put in the queue, so only the oldest item in the queue is accessible at any time. Main functions defined are enqueue and dequeue.

SA2: It is a particular set of entities that are put into a certain order by the enqueue and the dequeue functions.

It is observed that *SA1* contains more number of words common to *MA* than *SA2*. Also, the keywords of *MA* are '*first-in first-out*' , '*enqueue*' and '*dequeue*' which appear in a more spread-out manner in *SA1*. In order to grade such student answers, it is necessary to measure their overlap with the respective model answers, according to content discussed in each of them. There are various ways in which lexical overlap score between a pair of answers is computed. Lexical overlap features are categorized as follows:

- **Word-overlap features:** These features [22] are computed based on a number of words common to both the model answer and student answer. The word-overlap features used in this study are: Jaccard Similarity Coefficient, Simple word overlap, IDF overlap, Phrasal overlap.
- **Summary evaluation measures:** In general, the model answers tend to be precise and to the point as compared to the student answers. The reciprocal scenario, though rare, may also exist. So, it may be assumed that between student answer and model answer, the answer having shorter length is a summary of the other. This motivated us to use measures such as Recall Oriented Understudy for Gisting Evaluation (ROUGE) namely ROUGE-N, ROUGE-L, ROUGE-W, ROUGE-S and ROUGE-SU [23] that evaluate the quality of the computer generated summaries in this context.
  - **ROUGE-N:** N-gram Co-occurrence Statistics
  - **ROUGE-L:** Longest Common Subsequence (LCS)
  - **ROUGE-W:** Weighted Longest Common Subsequence (WLCS)
  - **ROUGE-S:** Skip-bigram Co-occurrence Statistics
  - **ROUGE-SU:** Skip-bigram plus unigram-based co-occurrence statistics

  We have considered only ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-W measures for the requirement of the proposed short-answer grading system. ROUGE-S and ROUGE-SU have not been taken into consideration due to the limitation of data sets which contain pairs of short-answers in which overlap of skip-grams, overlap of skip-grams plus unigrams is minimal. ROUGE has been used in [24] for evaluation of learner generated summaries.

3) Information Retrieval Measures: TF-IDF measures [25] based on Information retrieval are used for estimating the relevance and similarity between a pair of answers. *TF-IDF* novelty measure is one of the *TF-IDF* measures used for obtaining similarity between answers. It has been previously used for the TREC Novelty track for finding topically similar sentences in the dataset [26]. The research pointed out that the performance of novelty measures was quite sensitive to the presence of non-relevant sentences, which prompted us to consider it as a good candidate for identifying similar student answers to the model answer.

- **TF-IDF novelty measure:** The relevance of the student answer given the corresponding model answer is quantified using the *TF-IDF* novelty score. *TF-IDF* novelty score is computed as follows:

$$TF-IDF(A,B) = \sum_{w \in A \cap B} \log(tf_{w,A}+1)\log(tf_{w,B}+1)\log\left(\frac{N+1}{df_w+0.5}\right) \quad (3)$$

where $tf_{w,A}$ is the number of occurrences of a word w in a student answer. A represents the set of words in the student answer; $tf_{w,B}$ is the number of occurrences of a word w in the corresponding model answer. B represents the set of words in the model answer; N is the total number of student answers in the dataset; and df_w

is the number of student answers that the word w appears in.

4) Topical Similarity Features:

- **Latent Dirichlet Allocation:** Latent Dirichlet Allocation (LDA) [27] is an example of a topic model that is based on the assumption that each text discusses a mixture of topics and each topic is indicated by a subset of words that appear in the text. LDA is useful in handling polysemy, i.e., it is capable of modeling different meanings of the words.
  This can be illustrated by the following example:
  Example IV.2:
  SA1: A **variable** that contains the address in memory of another variable.
  SA2: The linked lists can be of **variable** length.
  We observe that the word '$variable$' is common in both *SA1* and *SA2*. But the '$variable$' in *SA1* refers to the noun form indicating a location in computer memory where a value may be stored. On the other hand, the '$variable$' in *SA2* concerns the adjective form indicating the ability of an object to change or be changed.
  In LDA, '$variable$' in *SA1* will be assigned to a different topic than the '$variable$' in *SA2*. In this way, the topic distribution of answer *SA1* will vary from that of answer *SA2*. This will be useful for distinguishing between answers and assigning grades to them.
  Since an answer is a distribution over the topics contained in it, the topical similarity between the student answer and the model answer corresponds to the similarity between their respective topic distributions. The *Hellinger Distance* [28] is one of the measures that determines the dissimilarity between the topic distributions. It is transformed into a similarity measure by subtracting it from 1.
- **Biterm Topic Model:** Biterm Topic Model (BTM) [18] is another topic model which has been observed to be more effective in modeling topic in short text as compared to traditional LDA.
  The topic modeling of the corpus in BTM is described as follows:
  – The biterms are obtained by extraction of word-pairs co-occurring within the context window. A short text document is represented by an individual context unit/context window. The biterms obtained from all the short text documents represent the training corpus for BTM model.
  – Gibbs Sampling is used to obtain topic-word distribution in the corpus and topic distribution of the entire training corpus [18].

  The topic proportions for each of the answers in a pair are obtained by computation of the expectation of the topic proportions of biterms generated from each of the answers. Similar to LDA, these topic proportions are used to compute a dissimilarity score between the pair of answers in the form of Hellinger distance [28]. This distance when subtracted from 1, represents the topical similarity between the pair of answers using BTM.

5) Relevance Feedback-Based Features: In many cases, model answers may fail to cover all the appropriate concepts and their semantic variations that are required to answer a given question. At the same time, it is also observed that student answers that are close to the model answer in semantic space may contain such left-out concepts and semantic variations. This observation forms the basis of the three different relevance feedback based features. The basic idea is to update a seed model answer with feedback from semantically similar student answers. The updating technique is similar to the pseudo-relevance feedback used in information retrieval [29]. This leads to enhancement of vocabulary of the model answer due to the paraphrasing observed across the student answers.

Computation of these features are performed in two distinct steps: 1) Similarity computation 2) Model answer update.

Generic Similarity Computation Step: The *similarity computation* step is generic to all relevance feedback based features. For a given question Q, the LSA similarity[1] of the model answer for Q (M_Q) and each student answer ($S_Q(j)$) is computed.

$$\mathcal{F}^1_Q(j) = \texttt{LSA-Sim}(S_Q(j), M_Q) \quad \texttt{where} \quad 1 \leq j \leq N_Q \quad (4)$$

N_Q is the number of student answers for a given question Q.

The *model answer update* step varies with relevance feedback based feature category each of which adopt a different strategy to update the baseline or original model answer.

Top Scorer Dependent Feedback Feature (RF-I): Computation of this feature involves selection of top students answers based on similarity with the model answer. A model answer is then updated with best P similar student answers based on $\mathcal{F}^1_Q(j)$ value in Equation (4). P being decided according to the number of student answers corresponding to a question Q. Let T_Q be the set of top P scoring student answers and $W(S)$ is the set of words extracted from a student answer $S \in T_Q$. The update received from the top student answers is represented as follows:

$$U_Q = \bigcup_{S_Q(j) \in T_Q} W(S_Q(j)) \quad (5)$$

Let M^O_Q be the words in model answer, i.e., $W(M_Q)$. The updated model answer is represented as

$$M^u_Q = M^O_Q \cup U_Q \quad (6)$$

With the updated model answer, a second round computation of LSA Similarity scores of each student answer is performed.

$$\mathcal{F}^2_Q(j) = \texttt{LSA-Sim}(S_Q(j), M^u_Q) \quad \texttt{where} \quad 1 \leq j \leq N_Q \quad (7)$$

The scores of the best P similar student answers are kept unchanged, whereas the final scores of the remaining student answers are computed by multiplying their second round LSA similarity scores with the P^{th} maximum score achieved in the

[1]LSA-Sim: Cosine similarity of two vector each represented with LSA.

first round. This is done to ensure that the scores of top P answers do not become artificially high. Also, none of the lower-scored answers achieve a new score higher than the best answers. The computation of final RF-I score $F_Q^f(j)$ of a student answers $S_Q(j)$ is performed using Equation (8).

$$F_Q^f(j) = \begin{cases} F_Q^1(j) & \text{if } S_Q(j) \in T_Q \\ d_{QP} \times F_Q^2(j) & \text{otherwise} \end{cases} \quad (8)$$

where d_{QP} represents the P^{th} maximum score obtained in the first round computation.

Least Scorer Dependent Feedback Feature (RF-II): This feature tries to capture the effect of words from least scoring student answer in grading. Computation of this feature is similar to that of RF-I except that the model answer is updated with least P similar student answers. Let L_Q be the set of least P scoring student answers for a question Q based on $F_Q^1(j)$ score. The model answer update is performed similar to the update strategy adopted in RF-I except that L_Q is used in place of T_Q.

The final RF-II score $F_Q^f(j)$ of a student answer $S_Q(j)$ is computed as follows:

$$F_Q^f(j) = \begin{cases} F_Q^1(j) & \text{if } S_Q(j) \in L_Q \\ a_{QP} \times F_Q^2(j) & \text{otherwise} \end{cases} \quad (9)$$

P^{th} minimum score obtained in the first round computation is represented as a_{QP}.

All Scores Dependent Feedback Feature (RF-III): This feature considers contributions from each student answer for a given question in updating corresponding baseline model answer. The rationale behind considering all the student answers stems from two boundary conditions ignored in computation of RF-I and RF-II.

- If all student answers to a particular question are poorly scored in the first round, then taking the words from the top P student answers might end up with degrading the original model answer.
- Variance in similarity scores does not play any role in the update procedure. Consequently, student answer with maximum similarity value will contribute with same strength as the 2^{nd} ranked student answer with significant difference in similarity value.

To address the issues mentioned, RF-III consider contributions from all the student answers where the contribution from a student answer is a function of the extent of similarity between the original model answer and the student answer.

The first step of update procedure involves extracting words from each student answer ($S_Q(j) \in A_Q$) and model answer (M_Q) for a given question Q. A_Q is the set of all answers for a question Q.

$$M_Q^u = W(M_Q) \bigcup_{S_Q(j) \in A_Q} W(S_Q(j)) \quad (10)$$

In the second step, confidence score of each of the terms $t_i \in M_Q^u$ is computed based on its presence in different student answers. The confidence score of t_i primarily depends on two factors:

- *Abundance:* If t_i appears in multiple answers, its confidence score should be boosted.
- *Answer Strength:* If t_i appears in the student answers that are highly similar to the model answer, high confidence score should be attributed to t_i

The second step of the update procedure is presented in Algorithm 1. The algorithm considers both the mentioned factors to compute aggregated confidence score of a term. A *bias* value is added to the confidence score if the term is present in the original model answer. Finally, the confidence score is passed through a logistic sigmoid function to restrict the range to [0 1].

Algorithm 1. Confidence Score Computation

Input: M_Q, M_Q^u, A_Q
Output: M_Q^s where $M_Q^s[i] = Score(t_i)$ for each $t_i \in M_Q^u$
foreach $t_i \in M_Q^u$ **do**
  $M_Q^s[i] := 0$
  foreach $S_Q(j) \in A_Q$ **do**
    // `Abundance Factor: Compute the importance of` t_i `in` $S_Q(j)$ `using TF-IDF measure`
    $p :=$ TF-IDF$(t_i, S_Q(j))$
    // `Answer Strength Factor: Computed as LSA-based similarity between` $S_Q(j)$ `and` M_Q
    $q :=$ `LSA-Sim`$(S_Q(j), M_Q)$
    $M_Q^s[i] \; += \; p \times q$
  if $t_i \in W(M_Q)$ **then**
    bias = 1
  else
    bias = 0
  $M_Q^s[i] :=$ `Sigmoid`$(M_Q^s[i] + bias)$
return M_Q^s

The updated model answer is represented as a vector containing the contributing score of each word in it whereas the student answer is represented by the tf-idf weights of each word in it. Cosine similarity between these vector-pairs indicates similarity between the respective answer-pair and represents the All Scores Dependent Feedback Feature (RF-III) for the student answer.

6) Alignment-Based Features: Alignment features are based on finding the best combination of pairs of similar semantic units having similar contexts. An example is shown below, for which alignment-based features may be useful:

Example IV.3: **Q:** What are the elements typically included in a class definition?

MA: The elements typically included in a class definition are function members and data members.

SA: The elements typically included in a class definition are the function prototypes, usually declared public, and the data members used in the class, which are usually declared private.

Following key pairs of words should be aligned between SA and MA as shown below:

`function` → `function`, `prototypes` → `members`, `data` → `data`, `members` → `members`

Due to context similarity between the pair of answers that would be considered while using word-to-word aligner [30], it

is expected that 'prototypes' will be rightly aligned to 'members' with the help of context around these words in the pair of answers.

Following are the types of Alignment-based features that are explained in detail in [31].

- Word-to-word alignment using word-aligner
- Coverage feature
- Question-demoted word-to-word alignment using word-aligner
- Question-demoted coverage feature

C. Answer Grading Models

We have worked on the available data sets in ASAG domain. Classification as well as regression models have been used based on the datasets. We have explored different individual regression and classification models. We have also investigated to leverage the best of individual grading models to improve overall grading performance through ensemble construction.

- **Individual models:**
  - **Regression:** The individual regression models that are trained and used to predict the scores of student answers are mentioned as follows: Least squares linear regression (LR) [32], Support vector regression (SVR) [33], Kernel ridge regression (KRR) [34], Least Absolute Shrinkage and Selection Operator (LASSO) [35], ElasticNet [36], Tree [37], Bagging Tree and Boosting Tree [37].
  - **Classification:** A classifier such as Random-forest [38] with a specified number of base estimators is trained with the features discussed in the previous section. The trained classifier is hence used to predict the labels of the student answers in the test sets.
- **Ensemble learning:** Ensemble learning [39] refers to the process of training multiple models and using combination of such models to make predictions, such that a better generalization performance is achieved than that obtained using the individual models. It aims to minimize the risk of an unfortunate selection of a specific poorly performing individual model. The process of ensemble learning for regression consists of two phases namely:
  - **Ensemble generation:** It refers to the generation of the individual regression models (also referred to as base models) in the ensemble using different machine learning algorithms. Hence, it is also indicated as heterogeneous ensemble generation.
  - **Ensemble integration:** Ensemble integration [40] refers to various methods employed for integration of the predictions using the base models (obtained in the previous step). Integration may take place either as a combination of output predictions from the base models or as a selection of output prediction from any base model based on some criteria.

    Ensemble integration can be represented by the following equation:

$$\hat{f}(x) = \sum_{k=1}^{K} h_k(x) f_k(x) \tag{11}$$

where x refers to the feature vector corresponding to a sample from the test set. K is the number of base models considered in the ensemble. $h_k(x)$ indicates the weight assigned to the prediction $f_k(x)$ by the individual base models. $\hat{f}(x)$ is the prediction obtained after taking into account ensemble of base models.

In case of ensemble integration involving combination of predictions from base models,

$$h_k(x) = \text{constant or vary according to input value x} \tag{12}$$

We have used ensemble method based on Stacked Regression [41], [42] to combine base learners such as Tree, Bagging tree, Boosting tree, Support vector regression, Linear regression, LASSO, ElasticNet, Kernel ridge regression. Cross-validation technique is used for tuning of hyperparameters ($h_k(x)$) of the individual models.

V. Evaluation

A. Test Bed

The test bed used for the purpose of evaluation comprises of three types of datasets:

- **University of North Texas dataset (UNT) [5]**: This dataset comprises of 2442 student answers provided by a class of undergraduate students and corresponding model answers to around 87 questions spread across ten assignments and two examinations of a Data Structures course at the University of North Texas.[2] The answers were graded independently by two human graders from 0 (completely incorrect) to 5 (perfect answer). The average of the two grades has been taken as the gold standard against which we compare the system output. The agreement between the two human graders (Inter Annotator Agreement (IAA)) is Pearson Correlation Coefficient (ρ) $= 0.586$.
- **Subsets of Student Response Analysis (SRA) corpus:** The SRA corpus [11] is comprised of two distinct subsets as follows:
  - **ScientsBank:** This corpus has around 10000 student answers to around 197 assessment questions belonging to around 15 different science domains [43]. The answers have been annotated as explained in [43].
  - **Beetle:** This corpus contains student responses extracted from interactions with the Beetle-II Tutorial Dialogue system [44]. It contains around 56 questions in Basic Electricity and Electronics domains with approximately 3000 manually labeled student answers. The student responses may refer to single or multiple model answers.

[2]Expanded version of the data set used by [4].

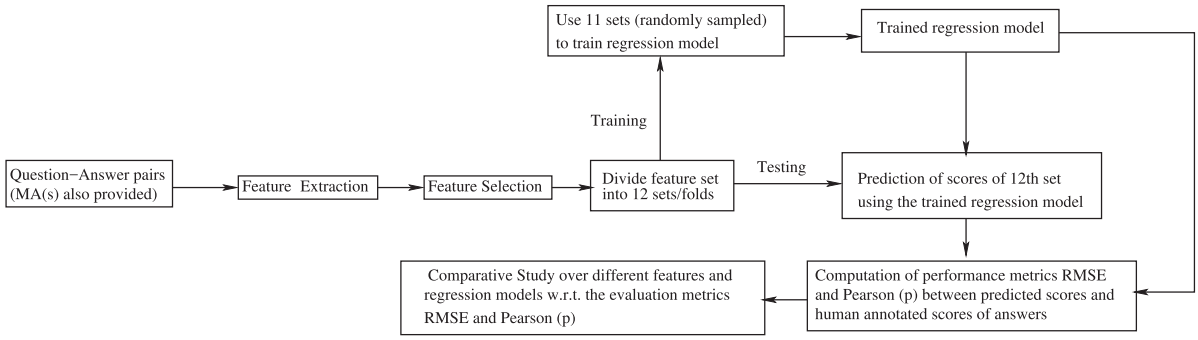


Fig. 2. Flow of experimental work in regression task.

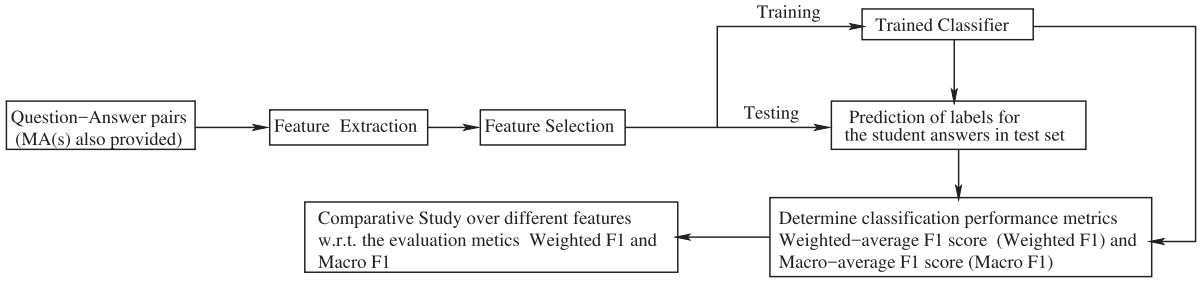


Fig. 3. Flow of experimental work in classification/labeling task.

The SRA corpus [11] consists of manually labeled student responses that are categorized into any of the five categories namely, *Correct*, *Partially_Correct_Incomplete*, *Contradictory*, *Irrelevant*, *Non_domain*. In SRA corpus, there are three kinds of test sets (explained in [43]): 1) Unseen-answers (UA), 2) Unseen-questions (UQ) and 3) Unseen-domains (UD). In case of ScientsBank corpus, all the above test sets are considered for evaluation, whereas for the Beetle corpus, only UA and UQ are considered for evaluation as the Beetle Corpus involves data from a single domain.

B. Experimental Design

The general flow of experimental work for regression-based grading model is shown in Fig. 2 and that for classification-based grading model is shown in Fig. 3.

In classification task, the results reported using performance metrics Weighted-average F1 score consider all 5 classes, and Macro-average F1 score consider all classes except *Non_domain* due to minimal percentage of samples of student answers having *Non_domain* labels in the training set as well as test sets in comparison to that for other labels.

1) Experimental Set-Up: Following experiments are conducted for the purpose of evaluation. The performance metrics for experiments on ScientsBank in the proposed system have been reported using the evaluation procedure in [9].

(1) **Experiment 1 (Performance analysis of feature groups):** For the regression task of grading, a model corresponding to each of the regression techniques is trained on each individual category of features to determine the performance of each feature-group. In case of classification task, a similar procedure is performed.

(2) **Experiment 2 (Feature significance tests):** These tests are needed to determine whether the relationship between a feature corresponding to an answer pair and the actual grade assigned to the student answer in the pair is significant.
One of such type of tests is based on univariate feature selection. It is performed for the regression task so as to obtain the *f-statistic*[3] [45] and the corresponding *p-values* for each of the features extracted for a pair of answers.
In case of classification task, χ^2 statistic and the corresponding p-values are computed for each feature.

(3) **Experiment 3 (Optimal feature set selection):** The whole set of features are arranged in the descending order of their f-statistic scores. Feature groups are formed by taking top N features at a time, where N = 10, 15, 20, 25, 30, 36. In case of regression-based grading task, a model corresponding to each of the regression techniques is trained with those feature groups. An optimal set of top N features is decided for each regression model corresponding to the maximum value of Pearson ρ and minimum value of RMSE obtained for the concerned regression model.
For the classification task, similar procedure is performed as above considering χ^2 statistics.

[3]An F-statistic score indicates if the means between two populations (the two populations being values for a particular feature and the corresponding true labels) are significantly different. This score is a measure of how informative each feature is for a particular dataset.

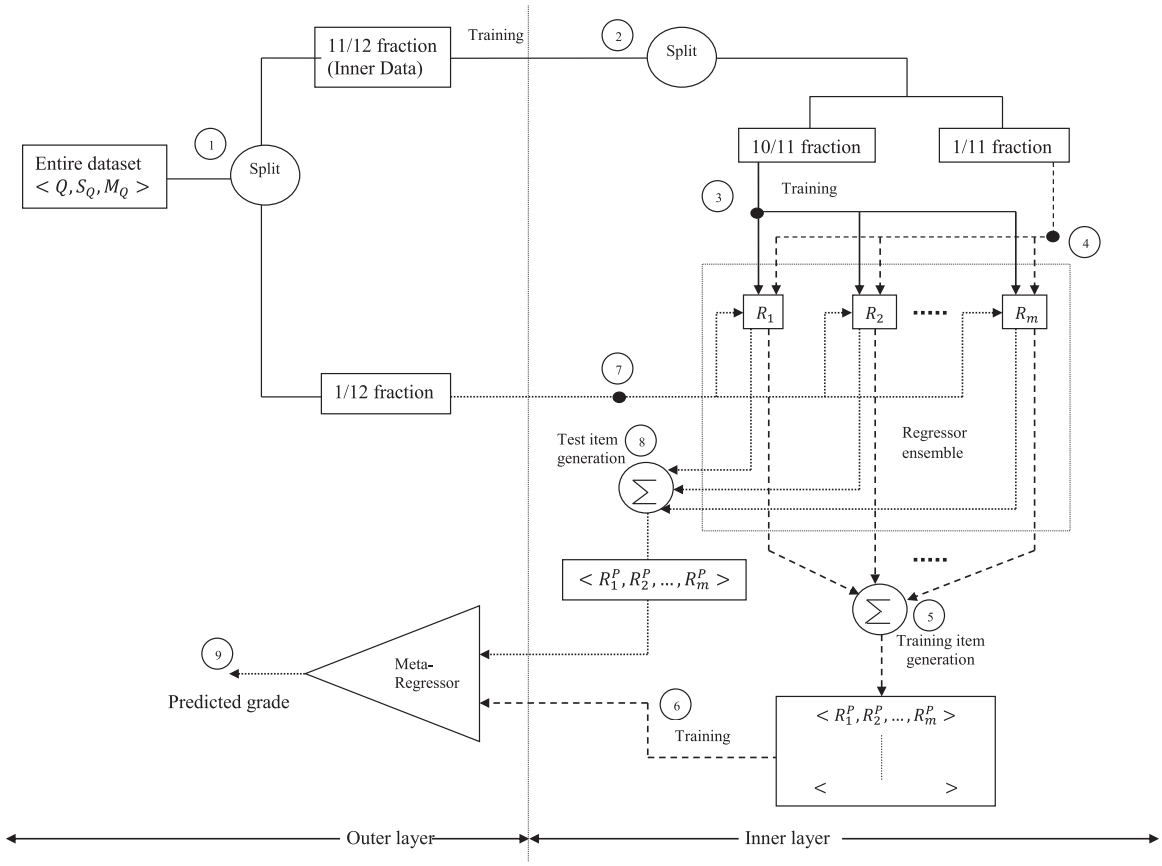


Fig. 4. Schematics of nested cross-validation over proposed stacked ensemble. Each individual step is assigned a sequence number: 1) Splitting of dataset in outer nesting layer. 2) Splitting of the *Inner Data* into training data for individual models and training item generation data for the meta-regressor. 3) Training of individual model. 4) Individual model prediction. 5) Training data generation for meta regressor (R_i^p is the prediction by i^{th} regression model). 6) Training the regressor with vectors generated from step 5. 7) Generating predictions from individual modes for outer layer test data. 8) Test data generation for meta-regressor. 9) Prediction by meta-regressor.

(4) **Experiment 4 (Ensemble-based regression- University of North Texas dataset):** In this experiment, eight regression models namely SVR, KRR, LR, LASSO, ELASTIC, TREE, BAGGING TREE (BAG), BOOSTING TREE (BOOST) have been trained using the training set. Each of the eight trained regression models (also known as base models) predicts a grade called as predicted grade for each student answer in the test set.
As mentioned earlier, an ensemble learning technique based on stacked regression is employed to predict final grades of each of the student answers in the test set. The stacked regression ensemble technique combines multiple regression models via a meta-regressor that has been implemented with a multi-layer perceptron. Base level models represented by the individual regression models are trained using entire training set. Predictions of the base level models against test data as input are used as features to train a meta-regressor.
Training and testing of the ensemble is performed through nested cross-validation (depicted in Fig. 4). In the first level, a 12-fold cross-validation is used where $\frac{11}{12}$ fraction of data items (will be referred as *Inner data* henceforth) are used to train the inner level of the ensemble and rest $\frac{1}{12}$ fraction of data items are used to test the entire ensemble set-up. *Inner Data* is used to train individual regression model with $\frac{10}{11}$ fraction of *Inner Data* and rest $\frac{1}{11}$ fraction is used to generate training sample for the meta-regressor. A training data item for the meta-regressor is 8-length vector where each dimension represents predicted score by the corresponding individual regression model. The stacked ensemble uses 12 -fold cross-validation to maintain parity with existing Short-answer grading system discussed in [5].

(5) **Experiment 5 (Ablation test):** Ablation test is performed to measure the impact of each feature in the grading/labeling performance. In this test, a single feature in turn is removed from the whole feature set and effect in the grading performance is observed.
We have performed ablation tests to measure the impact of only the novel features (RF-II, RF-III, BTM, IDF Overlap, TF-IDF Novelty score) which are a part of the best feature set to train the proposed ASAG system for both the regression as well as classification/ labeling tasks.

2) Experimental Results and Analysis:

(1) **Experiment 1:** The grading performance observed for UNT dataset, considering each feature group is

TABLE I
PERFORMANCE ANALYSIS REGARDING MODELS CORRESPONDING TO DIFFERENT REGRESSION TECHNIQUES, TRAINED ON EACH CATEGORY OF FEATURES (PEARSON (ρ): PEARSON CORRELATION COEFFICIENT, RMSE: ROOT MEAN SQUARED ERROR, PARAMETERS ARE: SVR: LINEAR KERNEL, KRR: (-5 $< \log_2(\gamma) <$ 7), LR: (-20 $< \log(\gamma) <$ 50), BAG, BOOST: NO. OF TREES = 200, AND ELASTICNET: α = 0.5)

	Semantic		IR		Topical		Lexical Overlap		Feedback		Alignment	
	Pearson (ρ)	RMSE	Pearson (ρ)	RMSE	Pearson (ρ)	RMSE	Pearson (ρ)	RMSE	Pearson (ρ)	RMSE	Pearson (ρ)	RMSE
SVR	0.4466	1.0317	0.2266	1.1648	0.1384	1.1308	0.5142	0.9953	0.2993	1.1446	**0.5671**	**0.9590**
KRR	0.4241	1.0130	0.2678	1.0715	0.1186	1.1059	0.5030	0.9616	0.3297	1.0438	**0.5920**	**0.8935**
LR	0.4383	1.0014	0.1783	1.1105	0.1417	1.0953	0.5006	0.9593	0.3232	1.0494	**0.5624**	**0.9185**
LASSO	0.4335	1.4121	0.0074	1.1057	0.020	1.1061	0.4897	1.1446	0.2982	1.1365	**0.5558**	1.1472
Elastic	0.4294	1.4237	0.0051	1.1053	0.0043	1.1054	0.4885	1.1284	0.2981	1.1253	**0.5625**	1.1552
Tree	0.3907	1.0311	0.2302	1.1075	0.1139	1.1035	0.3120	1.1963	0.1781	1.3360	**0.4366**	1.0620
Bag	0.4194	1.0134	0.2324	1.1243	0.1228	1.1294	0.4781	0.9864	0.3241	1.0523	**0.5796**	**0.9068**
Boost	0.4276	1.0060	0.2560	1.0860	0.1129	1.1142	0.4727	0.9886	0.3233	1.0669	**0.5734**	**0.9122**

TABLE II
PERFORMANCE ANALYSIS REGARDING CLASSIFIER MODEL, TRAINED ON EACH CATEGORY OF FEATURES (RANDOM-FOREST CLASSIFIER WITH NUMBER OF BASE LEARNERS/TREES = 500)

Feature groups	ScientsBank						Beetle			
	Weighted F1			Macro F1			Weighted F1		Macro F1	
	UA	UQ	UD	UA	UQ	UD	UA	UQ	UA	UQ
Semantic Similarity	**0.9024**	0.6074	0.5906	**0.8878**	0.4644	0.4549	**0.6021**	0.5010	**0.5178**	0.4773
IR	0.7964	0.6037	0.5808	0.7738	0.4754	0.4433	0.3291	0.3519	0.2015	0.2053
Topical Similarity	0.8232	0.5522	0.5305	0.8086	0.4077	0.3623	0.3767	0.3845	0.2982	0.2462
Lexical Overlap	0.8774	**0.6274**	0.6204	0.8616	**0.4913**	0.4822	0.5983	**0.5671**	0.4659	**0.5075**
Relevance Feedback	0.8574	0.5589	0.5508	0.8346	0.4141	0.3918	0.4341	0.4061	0.3491	0.2850
Alignment-based	0.8804	**0.6217**	**0.6501**	0.8776	0.4866	**0.4923**	0.5576	**0.5544**	0.4529	0.4947

presented in Table I. It is observed that Alignment-based features show the best grading performance for almost all regression modeling techniques (as shown by highlighted RMSE values in Table I), closely followed by Lexical Overlap features.

It is observed from Table II that the Semantic-similarity features exhibit the best classification performance for ScientsBank UA and Beetle UA test sets, whereas Alignment-based and Lexical Overlap features perform the best w.r.t. classification on the ScientsBank UQ and Beetle UQ test sets. The Alignment-based features show the best classification performance in case of ScientsBank UD test set. The feature group-wise best classification performance has been mentioned considering weighted F1 score only. The ScientsBank UA and Beetle UA test sets contain student answers to questions similar to those in their respective training sets. Hence the vocabulary in the test sets is closely similar to the vocabulary of training set leading to best classification performance by the semantic similarity features.

(2) **Experiment 2:** All the features are found to have high f-statistic scores highlighting their substantial significance towards prediction of grade in the regression task. These features also exhibit high χ^2 statistic scores, thus validating their significant contribution towards assignment of labels to student answers in classification task. The distribution of the features over the f-statistic scores in case of Regression task is shown in Fig. 5a and that over the χ^2 statistic scores obtained in case of Classification task is shown in Fig. 5b and Fig. 5c, respectively. Even though all features are substantially significant in the regression test as shown in Fig. 5a, top N features are selected with the objective to maximize grading performance.

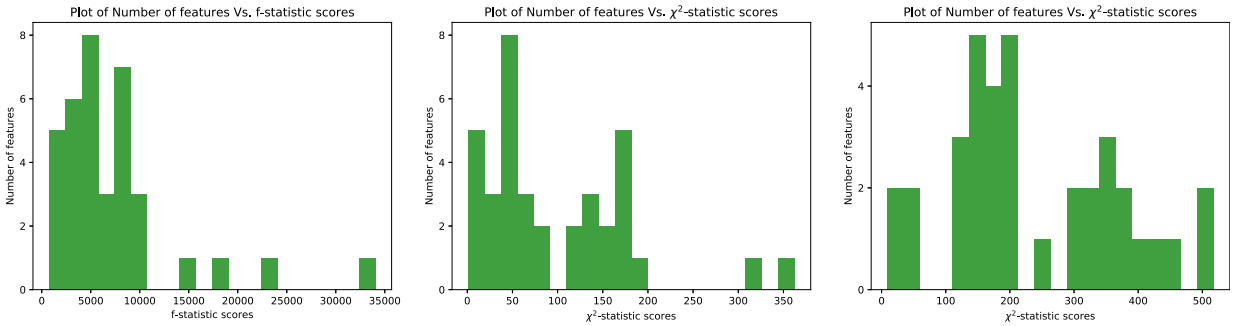
(3) **Experiment 3:** Features from top 10 to total 36 features having higher f-statistic scores and hence lower p-values are selected to train the regression models for the grading task involving UNT dataset. The evaluation performance measures in each case are shown in Figs. 6 and 7. It is observed that the optimal number of features varies with the regression models.

Features from top 10 to total 36 features having higher χ^2 statistics values are selected to train the random-forest classifier for the classification/labeling task involving ScientsBank and Beetle.

Figs. 8 and 9 show that the best classification performance is achieved with top N features, N varies for each of the datasets as well as the performance measures. It is observed that the proposed ASAG system achieves the best labeling performance in case of the Beetle Dataset with N = 36, i.e. all the features have significant contribution towards the best performance.

(4) **Experiment 4:** The best grading performance by an individual regression model for ASAG has been obtained by Bag which is ρ = 0.6179 and RMSE = 0.8738, with top N = 30 features used to train Bag as shown in Figs. 6 and 7.

It is observed from Table IV that the ensemble approach for regression exhibits better grading performance than Bag (N = 30 features) (current work with individual regression model) with ρ = 0.703 and RMSE = 0.793. The improvement of grading performance of the stacked ensemble approach as compared to the best individual model can be explained as follows:



(a) Regression task involving University of North Texas dataset (UNT)

(b) Classification task involving SciEntsBank dataset

(c) Classification task involving Beetle dataset

Fig. 5. Range of f-statistic/χ^2 statistic scores over the features in regression/classification task.

1) The heterogeneous stacked ensemble approach adopted here leads to introduction of model diversity, which reduces the variance of resulting ensemble. As a result, there is less overfitting, resulting in an overall better model.
2) Individual regression models may not work well for grading of some data items simultaneously. The stacked ensemble method combines the performance of all the individual models in such a way that almost all data items are suitably graded.

(5) **Experiment 5:** Ablation test (Regression task)/ (Classification task) involves assessment of strength of each of the novel features belonging to the best feature set used for regression/classification tasks.

In cases where the novel features namely RF-II, RF-III, BTM, IDF Overlap, TF-IDF Novelty score belong to the best performing feature set ($best_f$), they are each removed in turn from $best_f$. The remaining features are used to train the Stacked Ensemble in case of Regression task or the Random-forest classifier in case of Classification task. The grades/labels of the testing set are predicted using the trained Stacked Ensemble/ Random-forest classifier. The experimental results are shown in Table III. Following are some important observations in regard to ablation test:

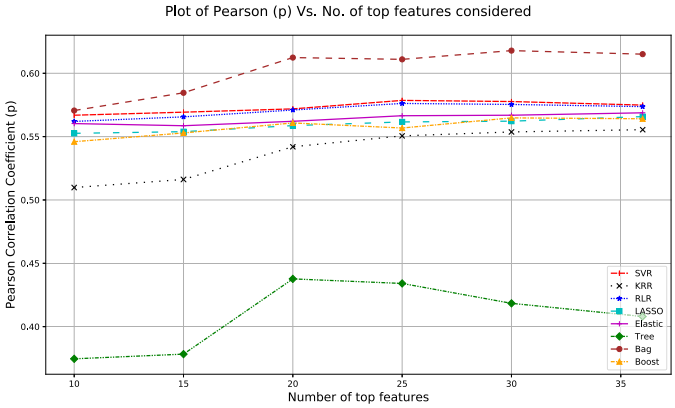


Fig. 6. Pearson (ρ) vs. no. of top features.

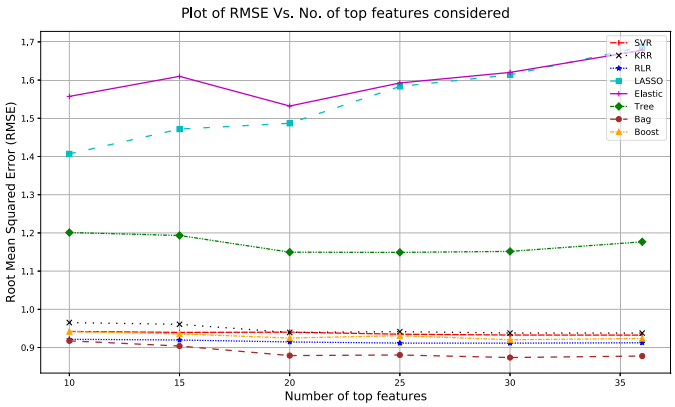


Fig. 7. RMSE vs. no. of top features.

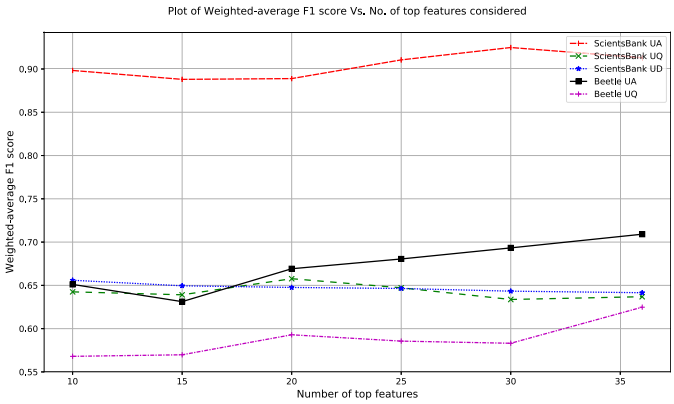


Fig. 8. Weighted-average F1 score vs. no. of top features.

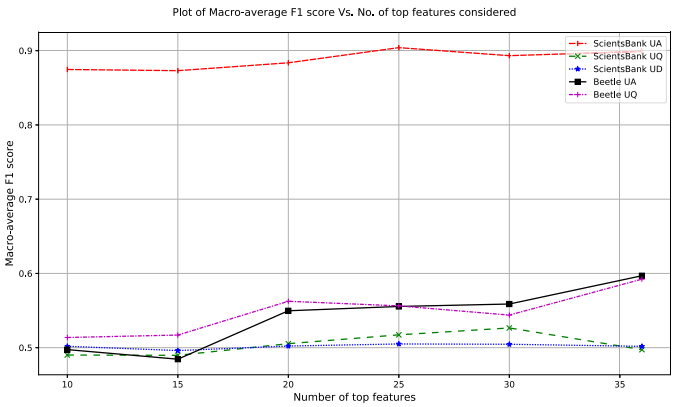


Fig. 9. Macro-average F1 score vs. no. of top features.

TABLE III
RESULTS OF ABLATION TEST PERFORMED TO TEST IMPORTANCE OF THE FEATURES PROPOSED IN THE PRESENT WORK

Features	UNT		ScientsBank						Beetle			
	Pearson (p)	RMSE	Weighted F1			Macro F1			Weighted F1		Macro F1	
			UA	UQ	UD	UA	UQ	UD	UA	UQ	UA	UQ
Least scorer dependent feedback feature (RF-II)	0.6995	0.7948	0.9063	0.6494	0.6478	0.9004	0.5040	0.5004	0.7177	0.6132	0.6060	0.5867
All scores dependent feedback feature (RF-III)	0.6863	0.8098	0.9118	0.6473	0.6445	0.8875	0.5210	0.4983	0.7005	0.6176	0.5865	0.5942
Biterm topic Model (BTM)	0.6996	0.7947	0.9043	0.6475	0.6507	0.8960	0.5064	0.5012	0.7050	0.6299	0.5899	0.6051
IDF Overlap	0.7015	0.7934	0.9060	0.6486	0.6506	0.8911	0.5205	0.5004	0.7043	0.6165	0.5890	0.5934
TF-IDF Novelty Measure	0.7044	0.7894	0.9149	0.6521	0.6472	0.8920	0.5167	0.5013	0.7060	0.6172	0.5907	0.5933

- There is a visible decrease in grading performance over UNT dataset (from ρ = 0.703 to 0.6863 and RMSE = 0.793 to 0.8098) due to exclusion of RF-III from $best_f$ as compared to exclusion of other novel features namely, RF-II and BTM.
  In case of ScientsBank UA and Beetle UA test sets, this leads to reduction in performance as compared to the best labeling performance for the above mentioned test sets.
  This may be due to the following reason: RF-III is based on LSA similarity feature which is corpus-based. Corpus-based features have the highest performance when trained on a domain-specific corpus [4]. UNT dataset contains student-answers based on *'Data Structures'* domain only. ScientsBank UA contains student answers from a limited number of Science domains and Beetle UA contains answers of a single domain only.
- Elimination of TF-IDF Novelty score from $best_f$ for grading student answers in ScientsBank UQ leads to decrease in classification performance on comparison with the best classification performance of the proposed system (Macro F1-score reduces from 0.527 to 0.5167). Also, there is a slight increase (ρ = 0.703 rises to 0.7044 and RMSE = 0.793 falls to 0.7894) in grading performance of UNT dataset when TF-IDF Novelty score is added to $best_f$.
- The classification performance degrades from Macro F1-score = 0.658 to 0.6486 when IDF overlap feature is removed from $best_f$ in case of ScientsBank UQ test set.
- Exclusion of RF-II from $best_f$ leads to decrease of classification performance from Weighted F1-score = 0.6248 to 0.6132 and Macro F1-score = 0.5923 to 0.5867 in case of Beetle UQ test set. A similar phenomenon is observed when it is removed from $best_f$ of ScientsBank UA test set, with decrease in Weighted F1-score from 0.925 to 0.9063. The reasons for good impact of RF-II on performance in Classification task are similar to those for RF-III as explained previously.
- It is observed that there is reduction in classification performance from Weighted F1-score = 0.925 to 0.9043, on removal of BTM from $best_f$ of ScientsBank UA test set. Since ScientsBank UA comprises of answers to same questions as that in training set, ScientsBank UA and ScientsBank training sets are expected to contain similar distribution of topics. Hence topics learnt by BTM during training prove useful in gauging topical similarity between pairs of answers in ScientsBank UA test set.

3) Comparison With State-of-Art in ASAG: In this section, performance comparison of the proposed system with that of the state-of-art ASAG system is presented. The ASAG systems that have been considered are Mohler [5], Earth Mover's Distance-based ASAG System [8], Iterative Ensemble (ITE) [9] and ETS_2 [10]. Table IV shows the comparison of all recent ASAG systems with the proposed ASAG system, both for Regression Task and Classification/Labeling task.

It is observed that the proposed system surpasses all the current ASAG systems for all the tasks. It shows highly improved performance than [8] in case of Regression task and much better performance than [9], in case of Classification Task on ScientsBank. There is marginal improvement in case of classification of student answers in Beetle as compared to ETS_2 in [10].

The best grading performance by the proposed system for regression task is obtained with the top 30 features from a wide spectrum of domains as Pearson (ρ) = 0.703 and RMSE = 0.793. The best labeling performance for classification task in case of ScientsBank Dataset is obtained as Weighted F1-score = 0.925 (UA test set with top 30 features), Weighted F1-score = 0.658 (UQ test set with top 20 features) and Weighted F1-score = 0.0.656 (UD test set with top 10 features). The best labeling performance for classification task in case of Beetle Dataset is obtained using all 36 features with Weighted F1-score = 0.7091 (UA test set) and Weighted F1-score = 0.6248 (UQ test set).

C. Discussion

In this section, we present analysis of the error cases in different grading models. The error cases have been categorized into several categories as presented below:

- **Wrong assignment of label due to extremely short student answer:** In Example V.1, no word overlaps between SA and MA and no matching of contexts due to SA being very short lead to failure in capturing lexical overlap and alignment between SA and MA. Due to mismatched contexts, no significant word-vector similarity is determined. No topics can be extracted from extremely short SA, hence topical similarity measures

TABLE IV
COMPARISON OF PERFORMANCE OF EXISTING ASAG SYSTEMS WITH RESPECT TO THE PROPOSED SYSTEM

ASAG Systems	Mihalcea		ScientsBank						Beetle			
	Pearson (p)	RMSE	Weighted F1			Macro F1			Weighted F1		Macro F1	
			UA	UQ	UD	UA	UQ	UD	UA	UQ	UA	UQ
Mohler [5]	0.518	0.978	-	-	-	-	-	-	-	-	-	-
Earth Mover's Distance-based ASAG System [8]	0.649	0.830	-	-	-	-	-	-	-	-	-	-
Iterative Ensemble (ITE) [9]	-	-	0.672	0.518	0.507	0.612	0.415	0.402	-	-	-	-
ETS_2 [10]	-	-	0.625	0.356	0.434	0.581	0.274	0.339	0.705	0.614	**0.619**	0.552
Proposed System	**0.703**	**0.793**	**0.925**	**0.658**	**0.656**	**0.899**	**0.527**	**0.505**	**0.7091**	**0.6248**	0.5969	**0.5923**

are unable to contribute towards proper labeling of SA.

Example V.1:

Q: What does a voltage reading of 1.5 tell you about the connection between a bulb terminal and a battery terminal?

MA: The terminals are not connected. the terminals are separated by a gap. The terminals are separated. There is a gap. The terminals are in different electrical states.

SA: its working

Thus, SA seems to be from a completely different domain than MA. Hence, SA is wrongly assigned 'Non_domain' label by the proposed ASAG system. It should be assigned *'Contradictory'* since it completely refutes all the statements in MA.

- **Failure to capture sequence information:** In Example 5.2, all key words of MA are present in SA, though not in a similar sequence. So, the word-overlaps captured by lexical overlap features mislead the ASAG to inappropriately assign a higher score to SA.

  Example V.2:

  Q: What is the inorder traversal of a binary tree ?

  MA: Traverse the left subtree, then the root, then the right subtree.

  SA: Inorder starts with the root then does right child then left child recursively.
- **Inability to find connected elements:** The contents of SA in Example V.3 do not directly or distinctively refer to 'Linked lists' or some dynamic structure for storage purpose about which the question is asked. SA appears to be a generalized statement, even though its contents answer the question in an implicit manner.

  Example V.3:

  Q: What is the advantage of linked list over arrays?

  MA: Linked lists are dynamic structures which allow for a variable number of elements to be stored.

  SA: There is no predetermined length.

  In such a case, generally, none of the words in SA match with those in MA, leading to failure of lexical overlap measures and alignment-based features to extract similarity information.

VI. CONCLUSION

In this work, a comparative study of various text similarity measures along with the different regression techniques has been presented. The proposed stacked-regression based ensemble model showed a huge improvement over not only the ASAG based on a single regression model (Bagging tree) but also the more recent ASAG discussed in [8]. The heterogeneity of the proposed stacked ensemble model ensures that variance of the resulting ensemble is reduced leading to less overfitting. The performance of all the individual models is combined in such a way that almost all data items are suitably graded. The alignment-based and lexical overlapping features from the optimal feature set, in addition to the semantic similarity features employed in the proposed system contribute a lot towards enhancement in labeling performance as compared to previous ASAG systems. The relevance feedback features that aims to augment original model answers with cues from the student answers have proved to be effective significantly. Each of the novel features added to the feature set for training the proposed ASAG model has some impact, towards its improved grading /labeling performance over the presently popular ASAG systems. Apart from the ensemble models discussed in the proposed work, we plan to consider and compare various other ensemble methods such as Deep Belief Networks comprising of stacked Restricted Boltzmann machines (RBM) [46], AdaBoost (Adaptive Boosting), GBM (Gradient Boosting) [47] and other variants of Boosting, for ASAG systems.

REFERENCES

[1] J. Z. Sukkarieh, S. G. Pulman, and N. Raikes, "Auto-marking 2: An update on the ucles-oxford university research into using computational linguistics to score short, free text responses," in *Proc. Int. Assoc. Educational Assessment*, Philadephia, 2004, p. 15.

[2] P. Thomas, "The evaluation of electronic marking of examinations," in *Proc. 8th Annu. Conf. Innov. Technol. Comput. Sci. Edu.*, 2003, vol. 35, no. 3, pp. 50–54.

[3] T. Pedersen, S. Patwardhan, and J. Michelizzi, "Wordnet: similarity—Measuring the relatedness of concepts," in *Proc. Demonstration Papers HLT-NAACL.*, Association for Computational Linguistics, 2004, pp. 38–41.

[4] M. Mohler and R. Mihalcea, "Text-to-text semantic similarity for automatic short answer grading," in *Proc. 12th Conf. Eur. Chapter Assoc. Comput. Linguistics.*, Association for Computational Linguistics, 2009, pp. 567–575.

[5] M. Mohler, R. Bunescu, and R. Mihalcea, "Learning to grade short answer questions using semantic similarity measures and dependency graph alignments," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Human Lang. Technol.*, Association for Computational Linguistics, 2011, pp. 752–762.

[6] C. Leacock and M. Chodorow, "C-rater: Automated scoring of short-answer questions," *Comput. Humanities*, vol. 37, no. 4, pp. 389–405, 2003.
[7] R. Mihalcea, C. Corley, and C. Strapparava, "Corpus-based and knowledge-based measures of text semantic similarity," in *Proc. 21st Nat. Conf. Artif. Intell.*, 2006, vol. 6, pp. 775–780.
[8] S. R. S. Kumar and S. Chakrabarti, "Earth mover's distance pooling over siamese LSTMs for automatic short answer grading," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 2046–2052.
[9] S. Roy, H. S. Bhatt, and Y. Narahari, "An iterative transfer learning based ensemble technique for automatic short answer grading," *2016, arXiv:1609.04909*.
[10] M. Heilman and N. Madnani, "ETS: domain adaptation and stacking for short answer scoring," in *Proc. 7th Int. Workshop Semantic Eval.*, Atlanta, GA, USA, Jun. 14–15, 2013, pp. 275–279.
[11] M. O. Dzikovska, R. D. Nielsen, and C. Brew, "Towards effective tutorial feedback for explanation questions: A dataset and baselines," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol.*, Association for Computational Linguistics, 2012, pp. 200–210.
[12] T. Kenter and M. de Rijke, "Short text similarity with word embeddings," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, 2015, pp. 1411–1420.
[13] K. Sakaguchi, M. Heilman, and N. Madnani, "Effective feature integration for automated short answer scoring," in *Proc. Human Lang. Technologies, Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2015, pp. 1049–1054.
[14] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *Int. J. Forecasting*, vol. 22, no. 4, pp. 679–688, 2006.
[15] G. R. Streiner and D. L. Streiner, *PDQStatistics.* Hamilton, London: BC Decker Inc, 2003.
[16] D. M. Powers, "Evaluation: From precision, recall and f-measure to ROC, informedness, markedness and correlation," *J. Mach. Learn. Technol.*, no. 1, pp. 37–63, 2011.
[17] H. Narasimhan, W. Pan, P. Kar, P. Protopapas, and H. G. Ramaswamy, "Optimizing the multiclass f-measure via biconcave programming," in *Proc. 16th IEEE Int. Conf.* Data Mining, 2016, pp. 1101–1106.
[18] X. Yan, J. Guo, Y. Lan, and X. Cheng, "A biterm topic model for short texts," in *Proc. 22nd Int. Conf. World Wide Web.*, International World Wide Web Conferences Steering Committee, 2013, pp. 1445–1456.
[19] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.
[20] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *J. Amer. Soc. Inf. Sci.*, vol. 41, no. 6, pp. 391–407, 1990.
[21] M. Baroni, G. Dinu, and G. Kruszewski, "Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, pp. 238–247, 2014.
[22] P. Achananuparp, X. Hu, and X. Shen, "The evaluation of sentence similarity measures," in *Proc. 10th Int. Conf. Data Warehousing Knowl. Discovery*, Berlin, Heidelberg: Springer-Verlag, 2008, pp. 305–316.
[23] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Proc. Text Summarization Branches Out Workshop*, 2004, pp. 74–81.
[24] N. Madnani, J. Burstein, J. Sabatini, and T. O'Reilly, "Automated scoring of a summary writing task designed to measure reading comprehension," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics 2013*, vol. 163, pp. 163–168.
[25] D. Metzler, Y. Bernstein, W. B. Croft, A. Moffat, and J. Zobel, "Similarity measures for tracking information flow," in *Proc. 14th ACM Int. Conf. Inf. Knowl. Manage.*, 2005, pp. 517–524.
[26] J. Allan, C. Wade, and A. Bolivar, "Retrieval and novelty detection at the sentence level," in *Proc. 26th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2003, pp. 314–321.
[27] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.
[28] M. Nikulin ,"Hellinger Distance," *Encyclopedia of Mathematics*, M. Hazewinkel, Ed. Berlin, Germany: Springer, 2001.
[29] J. J. Rocchio, "Relevance feedback in information retrieval," In G.Salton (Ed.), The SMART Retrieval System, Englewood Cliffs, N.J.: Prentice Hall, Inc. pp. 313–323, .
[30] M. A. Sultan, S. Bethard, and T. Sumner, "Back to basics for monolingual alignment: Exploiting word similarity and contextual evidence," *Trans. Assoc. Comput. Linguistics*, vol. 2, pp. 219–230, 2014.
[31] M. A. Sultan, C. Salazar, and T. Sumner, "Fast and easy short answer grading with high accuracy," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol.*, 2016, pp. 1070–1075.
[32] S. J. Miller, "The method of least squares," Mathematics Dept. Brown Univ., Providence, RI, USA, 2006, pp. 1–7.
[33] A. Smola and V. Vapnik, "Support vector regression machines," in *Proc. Adv. Neural Inf. Process. Syst.*, 1997, vol. 9, pp. 155–161.
[34] K. Vu, J. Snyder, L. Li, M. Rupp, B. F. Chen, T. Khelif, K.-R. Müller and K. Burke, "Understanding kernel ridge regression: Common behaviors from simple functions to density functionals," *Int. J. Quantum Chemistry*, vol. 115, pp. 1115–1128, 2015.
[35] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statistical Soc. B (Methodological)*, vol. 58, pp. 267–288, 1996.
[36] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Roy. Statistical Soc. B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.
[37] W.-Y. Loh, "Classification and regression trees," *Wiley Interdisciplinary Rev., Data Mining Knowl. Discovery*, vol. 1, no. 1, pp. 14–23, 2011.
[38] T. K. Ho, "Random decision forests," in *Proc. 3rd Int. Conf. Document Anal. Recognit.*, 1995, vol. 1, pp. 278–282.
[39] J. Mendes-Moreira, C. Soares, A. M. Jorge, and J. F. D. Sousa, "Ensemble approaches for regression: A survey," *ACM Comput. Surv.*, vol. 45, no. 1, p. 10:1–10:40, 2012.
[40] N. Rooney, D. Patterson, S. Anand, and A. Tsymbal, "Dynamic integration of regression models," in *Proc. Int. Workshop Multiple Classifier Syst.*, Springer, 2004, pp. 164–173.
[41] D. H. Wolpert, "Original contribution: Stacked generalization," *Neural Netw.*, vol. 5, no. 2, pp. 241–259, Feb. 1992.
[42] L. Breiman, "Stacked regressions," *Mach. Learn.*, vol. 24, no. 1, pp. 49–64, 1996.
[43] M. O. Dzikovska *et al.*, "Semeval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge," in *Proc. 7th Int. Workshop Semantic Eval., NAACL-HLT*, Atlanta, GA, USA, Jun. 14–15, 2013, pp. 263–274.
[44] M. O. Dzikovska, A. Isard, P. Bell, J. D. Moore, N. Steinhauser, and G. Campbell, "Beetle II: An adaptable tutorial dialogue system," in *Proc. SIGDIAL Conf.*, Association for Computational Linguistics, 2011, pp. 338–340.
[45] E. M. Cabaña, *F Distribution*. Berlin, Germany: Springer, 2011, pp. 499–501.
[46] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
[47] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, 1997.

Archana Sahu received the B.E. degree from the University of Pune, Pune, India, and the M. Tech degree in electronic engineering from the Shri Guru Gobind Singhji Institute of Engineering and Technology, Nanded, India. She is currently working toward the Ph.D. degree with the Centre for Educational Technology, Indian Institute of Technology Kharagpur, Kharagpur, India. Her research areas include natural language processing for E-learning and artificial intelligence in education.

Plaban Kumar Bhowmick received the B.Tech degree from Calcutta University, Kolkata, India, 2002, and the master's and Ph.D. degrees in computer science and engineering from the Indian Institute of Technology (IIT) Kharagpur, India, in 2006 and 2011, respectively. He is an Assistant Professor with the Centre for Educational Technology, IIT Kharagpur. He has been Co-Principal Investigator and Technical Lead of the National Digital Library of India project. His research areas include text processing, artificial intelligence in education, semantic web technology, knowledge graphs, and digital library technologies.