A Comparison of Features for the Automatic Labeling of Student Answers to Open-ended Questions

Jesus Gerardo Alvarado Mantecon
University of Ottawa
800 King Edward Avenue.
Ottawa, Canada.
+1 613 668 7214
jalva061@uottawa.ca

Hadi Abdi Ghavidel
University of Ottawa
800 King Edward Avenue.
Ottawa, Canada.
+1 613 890 1389
habdi018@uottawa.ca

Amal Zouaq
University of Ottawa
800 King Edward Avenue.
Ottawa, Canada.
+1 613 562 5800 ext.6227
azouaq@uottawa.ca

Jelena Jovanovic
University of Belgrade
Jove Ilica 154.
11000 Belgrade, Serbia.
+381 11 3950 853
jelena.jovanovic@fon.bg.ac.rs

Jenny McDonald
University of Auckland
Private Bag 92019.
Auckland, NZ.
+64 9 9238140
j.mcdonald@auckland.ac.nz

ABSTRACT

The automatic evaluation of text-based assessment items, such as short answers or essays, is an open and important research challenge. In this paper, we compare several features for the classification of short open-ended responses to questions related to a large first-year health sciences course. These features include a) traditional n-gram models; b) entity URIs (Uniform Resource Identifier) and c) entity mentions extracted using a semantic annotation API; d) entity mention embeddings based on GloVe, and e) entity URI embeddings extracted from Wikipedia. These features are used in combination with classification algorithms to discriminate correct answers from incorrect ones. Our results show that, on average, n-gram features performed the best in terms of precision and entity mentions in terms of f1-score. Similarly, in terms of accuracy, entity mentions and n-gram features performed the best. Finally, features based on dense vector representations such as entity embeddings and mention embeddings obtained the best f1-score for predicting correct answers.

Keywords

Short open-ended responses, N-gram models, Entity URIs, Entity Mentions, Entity embeddings, Mention embeddings.

1. INTRODUCTION

Due to the growth of Massive Open Online Courses (MOOCs) and increased class sizes in traditional higher education settings, the automatic evaluation of answers to open-ended questions has become an important challenge and one which has yet to be fully resolved. On the other hand, it has been shown that open-ended assessments are better able to capture a higher level of understanding of a subject than other machine-scored assessment items [24]. Still, MOOCs usually rely on multiple-choice questions since the evaluation of open-ended assessments requires more resources in massive online courses [32]. The human effort required to manually evaluate students' answers has escalated with the spread of large-scale courses that enroll several hundred, or even thousands of students. To tackle this challenge, we analyze textual responses to a set of open-ended questions designed to encourage deep responses from students. We explore the use of vector space models (VSMs) that represent each answer with a real-valued vector, and evaluate those models on the task of classifying student responses into correct and not-correct. In particular, we examine and evaluate different feature sets that can be automatically derived from students' answers and used to represent those answers as vectors in a high dimensional space. The examined features do not require handcrafting based on the particularities of specific questions. Our main objective is to examine and compare the predictive power of different text features, automatically extracted from a corpus of answers to open-ended questions, on multiple classification algorithms.

We build VSMs using different text representations that result in either a sparse VSM (e.g., n-gram based VSM) or a dense VSM (e.g., VSM based on word embeddings). For sparse VSMs, we explore traditional n-gram features (unigrams, bigrams, trigrams, and n-grams that combine all of the previous features). We also investigate the usefulness of semantic annotations of students' responses for the classification task. Semantic annotation adds machine-readable meaning in the form of entities [21]. Hence, it enables the association of students' answers with vectors of domain-specific entities. Semantic annotators often rely on open Web-based knowledge bases such as DBpedia [13], an RDF representation of Wikipedia's semi-structured content. For example, given the entity *Aorta*, identified by a semantic annotator, we obtain its associated Web resource from DBpedia (http://dbpedia.org/page/Aorta), which further links to other related entities and properties from DBpedia. We make use of two semantic annotators: DBpedia Spotlight [19] and TAGME [6]. We query each annotator with the students' responses to obtain entities mentioned in the response. For each entity, we take the entity label and use it as *entity mention feature*, whereas the entity's Uniform Resource Identifier (URI) is used as *entity URI* feature.

To build a dense VSM, we rely on the entity mentions identified through semantic annotation and pre-trained word and entity embeddings. In particular, we retrieve vector representations of entity mentions using a GloVe model pre-trained on Wikipedia dumps [23]. Thus, our fourth feature set consists of entity mention embeddings based on GloVe. Finally, we represent entity URIs using a Wiki2Vec model trained on Wikipedia dumps to obtain another dense VSM. Hence, entity URI embeddings extracted from Wikipedia constitute our fifth feature set.

Given the short length of most answers and large vocabularies providing sparse vectors, we decided to include the last two sets of features to produce dense vector representations. In fact, dense vectors have shown an increase in performance for several natural language processing tasks [15]. Both GloVe [23] and Word2vec models [20] learn vector representations of words (called word embeddings) based on context. In total, we compare five types of features (n-gram, entity mentions, entity URIs, mention embeddings and entity embeddings) to train classification models to automatically label each student answer as correct or incorrect.

The rest of the paper is structured as follows: In Section 2, we present related work on automatic short answer grading. Then, we introduce our methodology, including the corpus description, our analysis pipeline and an in-depth description of our features. Section 5 describes the results of our experiments followed by the analysis of the effect of feature selection on our classifiers in Section 6. Finally, we discuss our findings and conclude the paper in Section 7 and 8.

2. RELATED WORK

One of the hot topics in the field of educational data mining is automatic short answer (response) grading (ASAG). In general, there are two kinds of ASAG approaches: response-based and reference-based [27]. In this paper, we analyze students' answers based on the response-based approach, which focuses only on students' answers. In contrast, reference-based ASAG also rely on the comparison of the student answer to the model answer.

Burrows et al. [4] classified all types of approaches to ASAG into five categories (eras): Concept mapping [8, 10, 12], Information extraction [5], Corpus-based methods [11], Machine learning, and Evaluation [28, 30]. In the Machine Learning approach, which is the approach followed in this study, the trend is to build models (supervised or unsupervised) through data mining and natural language processing techniques in order to assess students' answers.

ASAG systems can also be categorized into semi-automatic (teacher-assisted) and fully-automatic systems. In semi-automatic systems, students' answers are processed (clustered) to facilitate the grading process. For example, Basu [1] applied k-medoids clustering to students' answers to ease the grading process. In another work, Jayashankar [9] proposed an integration of data mining and word clouds to help teachers evaluate student answers through visualization.

Fully-automatic systems produce grades for each student, with or without additional feedback. Several features are considered in training these systems: lexical features (e.g. word length), syntactic features (e.g. sentence length and part-of-speech), semantic features (e.g. semantic annotations and triples), discursive features (e.g. referential expressions), statistical features (e.g. language modelling like n-grams and embeddings), and similarity features (e.g. cosine similarity).

McDonald et al. [17, 18] evaluated Naive Bayes and Max Ent classifiers using a number of features like bag of words, word length, and word and character n-grams. Madnani et al. [14] used these types of features in combination with triples to examine the performance (accuracy) of 8 different classifiers and regressors (linear and nonlinear). In another work, Riordan et al. [26] combined n-gram features, answer length, and word and character embeddings to compare the performance of SVM (as a baseline) with neural architectures. In several approaches, features based on the similarity between the students' responses and the teacher's response were used together with n-grams. For example, Sakaguchi et al. [27] used stacked generalization [31] to integrate response-based and reference-based models. In particular, Sakaguchi et al. first built a classifier based on sparse response-based features (e.g. character n-gram and word n-gram); the obtained predictions were combined with dense reference-based features (e.g. BLEU [22]) to build another stacked classifier. Both classifiers were built as support vector regression (SVR) models. Zhang et al. [33] compared Deep Belief Networks (DBN) [2] to five classifiers such as Naive Bayes and Logistic Regression. The classifiers were trained on features extracted from three models, namely the Question model (e.g. question difficulty), the Student model (e.g. probability that a student learned a concept based on the student's past performance), and the Answer model (e.g. length difference between student answer and model answer). The DBN performed better than the other classifiers in terms of accuracy, precision, and F-measure, but not recall. Roy et al. [25] developed an ASAG system that can grade answers in different domains. They relied on an ensemble classifier of student answers (question-specific approach) and a numeric classifier based on the similarity score between the model answer and students' answers (question-agnostic approach). Their features were words, n-grams, and similarity scores between student answers and model answer. Finally, Tack et al. [29] used ensemble learning of five classifiers based on lexical features (e.g., word length), syntactic features (e.g., sentence length), discursive features (e.g., number of referential expressions), and a number of psycholinguistic norms.

In this work, we follow the response-based approach as we build classifiers based on students' answers. Our approach differs from previous works in that we carry out ASAG (and more specifically classification) by comparing six classifiers trained with both sparse vector representations (based on n-grams and entities) and dense vectors representations (GloVe, Word2Vec). One additional difference is the use of semantic annotations (entity mentions and entity URIs) to build some of our vector space models. Finally, the features used in this work do not necessitate a huge feature engineering effort as they come directly from text or from the use of a semantic annotation API and an embedding model.

3. METHODOLOGY

We first give a description of the corpus used in our experiments, then we detail our overall approach as well as the metrics used in the evaluation phase. This is followed by an in-depth explanation of our features.

3.1 Corpus Description

Our data set is extracted from a corpus of student short-answer question (SAQ) responses drawn from a first-year human biology course (McDonald [16]). Among multiple elements in our data set, our experiments are based only on the labeled student responses to the survey and model answers (expected answers to the questions). Student SAQ responses and associated metadata were collected through a dialog system.

From the initial data set, we selected a sub-set of student answers based on the following criteria:

- Answers from the year 2012 only as this year is the one with the highest participation; out of 15,758 answers collected over 4 years, 7,548 originate from 2012.
- Out of the 42 different unique questions, we only use 6 questions that provide a reasonable number of responses as well as lengthy (deep) responses. We avoided questions that do not encourage answers that display deep understanding of the topic (e.g., yes-no questions, calculation questions or multiple choice questions).

The questions asked are designed to encourage deep responses from students [3]. The students are expected to explain or describe the knowledge obtained during the course in their own words rather than giving answers by the book. Table 1 presents the questions used in the study and their expected answers.

Table 1. Survey questions

ID	Question	Model Answer
Q.1	HR or heart rate is the number of times the heart beats each minute. A normal adult HR is around 72 beats/min. How would you check someone's HR?	You could measure their pulse.
Q.2	What is the pulse?	The pulse is a pressure wave or a pulsatile wave generated by the difference between systolic and diastolic pressures in the aorta.
Q.3	Inotropic state is a term that is sometimes used to describe the contractility of the heart. Can you describe what is meant by contractility?	Contractility is the force or pressure generated by the heart muscle during contraction.
Q.4	If you were 'building' a human being and you wanted to position receptors in the body to monitor blood pressure, where would you put them?	You'd probably want to put them near vital organs and at the main outflow from the heart. It turns out that the main human baroreceptors are located in the carotid sinuses and aortic arch.
Q.5	What feature of artery walls allows us to feel the pulse?	Artery walls are thick and strong and not very compliant
Q.6	Can you explain why you cannot feel a pulse in someone's vein?	You cannot feel a pulse in veins because the blood flow in veins is not pulsatile

The resulting sub-set amounts to 1,876 answers from 218 students to 6 questions. Note that not all students answered all the questions. Completing responses was voluntary, which accounts for the variability in the number of responses received to each question. In addition, the nature and quality of the responses are not necessarily representative of the class as a whole. Table 2 presents descriptive statistics on the students' answers to the selected subset of questions used in all the experiments.

Table 2. Statistics on students' answers per question

Question	Avg. words	Min. words	Max. words	Answers	Correct (%)
Q.1	6	1	36	243	65.43%
Q.2	9	1	82	422	17.54%
Q.3	6	1	31	316	33.86%
Q.4	4	1	34	151	54.97%
Q.5	3	1	27	171	25.15%
Q.6	9	1	34	361	31.86%

Each of these questions is associated with a set of students' answers. As an example, for question Q.6, we present the expected answer (i.e. Model answer), a deep response (Student Answer 1), and a simpler response (Student Answer 2):

Model Answer: You cannot feel a pulse in veins because the blood flow in veins is not pulsatile

Student Answer 1: The wave motion associated with the heart beat is stopped by the arteries and capillaries. Therefore, the vein has no pulse.

Student Answer 2: The blood flow is continuous.

Both student answers were labeled as *correct* by the human markers. Student Answer 1 would be considered a deeper answer than Student Answer 2, because it makes explicit the reasoning behind the answer, thus suggesting a better understanding of the topic.

The students' responses were manually evaluated by human markers with expertise in the domain of human biology. The annotators assigned a label negotiated through discussion. Such labels describe different aspects of an answer like quality of the response or correctness [16]. For example, answers may be labelled as incorrect, incomplete, and display disinterest in responding (*dont-know* label), among others. Further details on the labels used can be found in McDonald [16]. Table 3 displays some of those answers and the assigned labels.

Table 3. Student Answers sample

Question	Student Answer	Label
Q.5	Lack of elastic tissue	incorrect
Q.6	idk lol	dont-know
Q.4	In major arteries of the body, such as the common carotid or the aortic arch	ok
Q.3	ability to change volume	incomplete
Q.6	Ventricle contracts blood ejected into aorta, expanding vessel and increase pressure in vessel, wave of pressure cane felt is pulse	correct

For all of our experiments, we used model answer (expected answer) and student answers and re-labeled them as *correct* or *not-correct. Correct* answers comprise model answers plus all answers labeled as *correct* or *ok*. All other answers were re-labeled as *not-correct*. The resulting data set is composed of 65% *not-correct* answers and 35% *correct* answers.

3.2 Overall Approach

Our general approach can be described as follows:

1. ***Data pre-processing:*** in this step, we perform lemmatization and removal of punctuation marks and stop words (NLTK[1] stop words list) from the selected answers.
2. ***Feature extraction***: We consider five types of features: n-gram, entity URIs, entity mentions, URI embeddings, and mention embeddings, which are detailed in section 4. We extract n-grams, entity URIs and entity mentions from student responses. Then, entity mentions are used to query a pre-trained GloVe model [23] to obtain mention embeddings. Likewise, entity URIs are used to query a pre-trained Wiki2Vec model [34] to obtain entity embeddings. Both GloVe and Wiki2Vec are pre-trained on Wikipedia.
3. ***Vector space model (VSM)***: For n-gram features, entity mentions, and entity URIs, we compute a vector representation of each answer by extracting a vocabulary from all students' answers and using TF-IDF as the relevance metric to weight each feature in an answer. As for mention embeddings and entity embeddings, we generate VSMs by averaging embeddings over all mentions or URIs appearing in an answer. The output from this step is one VSM representation of all answers for each feature type.
4. ***Classification task***: we run several classification algorithms: the ZeroR algorithm as our baseline, Logistic regression, K-nearest neighbors (IBK), Decision trees (J48), Naïve Bayes, Support vector machine (SVM), and Random forest. We train each classifier using the entire data set of answers regardless of the question to which they belong. The rationale is that all answers belong to the same domain, and thus can be expected to be in a shared semantic space.

3.3 Evaluation Metrics

The evaluation is performed through 10-fold cross validation on each classifier. The metrics used for this purpose include:

- Accuracy: Percentage of correctly classified answers.
- Area Under the Curve (AUC): Probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance.
- Precision: Fraction of correctly classified answers within all classified instances.
- Recall: Fraction of relevant answers successfully retrieved.
- F1-score: Weighted harmonic mean of the precision and recall. It represents how precise and complete a classifier is.

4. FEATURE DESCRIPTION

4.1 N-gram Features

We create a vector representation for each answer based on n-grams. Table 4 shows some descriptive statistics on the obtained n-grams. We perform four experiments using different n-grams: unigrams, bigrams, trigrams, and the combination of all of them. To that end, four VSMs are built, one per n-gram group. Each vector holds the TF-IDF value of each item found in the answers. TF-IDF is calculated with the formula:

$$tf\text{-}idf_{i,j} = tf_{i,j} \times \left(\log \frac{1+n_d}{1+df_i} + 1\right)$$

Where $tf_{i,j}$ is the total number of occurrences of the term i in the student answer j, n_d is the total number of documents (i.e. answers) and df_i is the number of documents (i.e. answers) containing the term i.

Table 4. Total number of n-grams in answers for all questions

Answers	Unigrams	Bigrams	Trigrams
Unique	700	2114	4750
Total	6364	2589	3383

4.2 Entity URI Features

These features are based on entity URIs extracted from answers using two semantic annotators: DBpedia Spotlight and TAGME (see Sect. 1). The basic unit in the built VSM is the URI of a DBpedia resource (e.g. http://dbpedia.org/page/Baroreceptor). We send get requests to both annotators with the answers to be analyzed, and receive, for each answer, a list of entity mentions and their associated URIs. Table 5 shows statistics on the number of entity URIs and mentions (lowercase) retrieved by each of the two annotators.

Table 5. Number of entity URIs and mentions on all answers

Semantic Annotator	Entities		Mentions	
	Unique	Total	Unique	Total
Spotlight	143	1620	188	1620
TAGME	876	5054	806	5054

Table 6 provides an example of retrieved entity mentions and URIs for an answer to Q.2.

Table 6. Sample of retrieved entity URIs

Answer	Semantic Annotator	Mention	URI
Recoil caused by pressure in arteries	Spotlight	Recoil	dbpedia.org/page/Recoil
		Arteries	dbpedia.org/page/Artery
	TAGME	Recoil	dbpedia.org/page/Recoil
		Pressure	dbpedia.org/page/Pressure
		Arteries	dbpedia.org/page/Artery

We build a vector representation of each answer for each of the following configurations (i.e., vocabularies):

- Spotlight_URI: Set of entity URIs retrieved from all answers using DBpedia Spotlight.
- TAGME_URI: Set of entity URIs retrieved from all answers using TAGME.
- Intersection: Set intersection between the entity URIs retrieved from all answers with both tools.
- Union: Set union between the entity URIs retrieved from all answers with both tools.

This produces four VSMs based on entity URIs. The resulting VSMs use TF-IDF as the metric for estimating the value of each entity URI for each answer.

[1] https://www.nltk.org/

4.3 Entity Mention Features

We use the annotations retrieved by Spotlight and TAGME, selecting entity mentions as the basic units for building VSMs. A mention is a sequence of words spotted in an answer and associated to a URI. This means that an entity mention can be a unigram, but also a bigram or trigram. We compute the TF-IDF of each entity mention present in an answer to build its vector representation. As in entity URI features, we have one vocabulary per configuration with four VSMs as the final output. The available configurations, based on mentions, used to build a vector representation of each answer, are analogous to those described for entity URIs, except that they are based on mentions (Spotlight_Mention, TAGME_Mention, Intersection and Union).

4.4 Entity Embedding Features

For this set of features, we rely on the Wiki2Vec[2] model, a Word2Vec implementation pre-trained on Wikipedia, where Wikipedia hyperlinks are replaced with DBpedia entities (URIs). The model was presented by Zhou et al. [34] and is based on 100-dimensional vectors. Word2Vec models can either learn to predict a word given its context (CBOW) or predict a context given a target word (Skip-gram) [20]. This creates a vector space in which similar words or entities are close to each other. Likewise, Wiki2Vec creates a vector space model in which similar DBpedia entities are close to each other. Given that our entity URIs reference DBpedia resources, we consider it a suitable match. For each configuration, we query the Wiki2Vec model with the entity URIs found in each answer to obtain their corresponding embeddings. Table 7 shows the percentage of entity URIs that are associated with an embedding vector in the Wiki2Vec model per configuration. We also show the percentage for the GloVe model which is presented in section 4.5.

Table 7. Coverage of entity URIs and mentions on their corresponding models (Wiki2Vec and GloVe)

Configuration	% of entity URIs in Wiki2Vec	% of entity mentions in GloVe
Spotlight_URI	97.5 %	50.46%
TAGME_URI	93.94 %	62.16%
Intersection	97.11 %	49%
Union	94.63 %	65.10%

For each configuration, we have one VSM. In each VSM, we aggregate the entity embeddings per answer by calculating the average of the entity URI vectors. This produces a single embedding that represents the answer.

4.5 Mention Embedding Features

For the mention embedding features, we rely on word embeddings, where each word is an entity mention instead of an entity URI. We use the GloVe model [23] trained using Wikipedia dumps from 2014 and build vectors with 100 dimensions (as for entity URI embeddings). Unlike Word2Vec, GloVe is a count-based model derived from a co-occurrence matrix. We query the GloVe model with the entity mentions found in each answer. The coverage of the model is given in Table 7. For each configuration, we have one VSM where each answer is represented as the average of the entity mention vectors.

5. RESULTS

For each feature set, we trained six classification algorithms, and evaluated 120 different models. Due to the space limit, we present only the top two performing classifiers (Random forest and SVM) in terms of overall accuracy for each of our feature sets. ZeroR is also included as the baseline.

5.1 N-gram Results

Table 8 shows the accuracy (ACC) and AUC obtained using n-gram features. Overall, the accuracy and AUC obtained with Random forest were always higher than with SVM. In particular, unigrams obtained the best accuracy of 88.40% as well as the highest AUC (.95) using Random forest.

Table 8. Accuracy & AUC using n-gram features

N-gram	Random Forest		SVM		ZeroR	
	ACC %	AUC	ACC %	AUC	ACC %	AUC
Unigrams	**88.40**	**.95**	84.44	.82	65.1	.50
Bigrams	81.97	.87	79.93	.73	65.1	.50
Trigrams	72.84	.68	72.12	.61	65.1	.50
N-grams	85.58	.93	84.25	.80	65.1	.50

Table 9 shows additional results for models cross-validated with n-gram features. For the *correct* label, our best classifier was Random forest using unigrams for the f1-score (.82) and trigrams or n-grams for best precision (.93). For the *not-correct* label, again, Random forest got the best results, using unigrams for both f1-score (.91) and precision (.88).

Table 9. Precision, recall & f1-score using n-gram features

Label	Classifier	N-gram	Precision	Recall	F1
Correct	Random Forest	Unigrams	.89	**.77**	**.82**
		Bigrams	.92	.53	.67
		Trigrams	**.93**	.24	.38
		N-grams	**.93**	.63	.75
	SVM	Unigrams	.79	**.76**	**.77**
		Bigrams	.85	.51	.64
		Trigrams	**.88**	.23	.37
		N-grams	.86	.65	.74
	ZeroR	Unigrams	0	0	0
		Bigrams	0	0	0
		Trigrams	0	0	0
		N-grams	0	0	0
Not-correct	Random Forest	Unigrams	**.88**	.95	**.91**
		Bigrams	.79	.98	.88
		Trigrams	.71	**.99**	.83
		N-grams	.83	.98	.90
	SVM	Unigrams	**.87**	.89	.88
		Bigrams	.79	.95	.86
		Trigrams	.71	**.98**	.82
		N-grams	.84	.95	**.89**
	ZeroR	Unigrams	.65	1	.79
		Bigrams	.65	1	.79
		Trigrams	.65	1	.79
		N-grams	.65	1	.79

[2] https://github.com/idio/wiki2vec

A visible drop in recall from unigrams to trigrams (difference of .53) can be spotted for the correct label in both SVM and Random Forest. Based on the number of elements in each n-gram feature (Table 4), we observe that the amount of bigrams and trigrams is notably lower than unigrams. This can, at least partially, explain the lower recall using these features. Another noticeable result is that while the results obtained with Random Forest and SVM exceed the baseline for the *correct* label in terms of precision, recall and f1-score, the results for the *not-correct* label are closer to the baseline.

5.2 Entity Mention Results

The highest accuracy among these feature sets was achieved by Random Forest with the Union configuration (88.58%), as shown on Table 10. Again, Random forest outperformed SVM in terms of accuracy and AUC for each configuration.

Table 10. Accuracy & AUC using Entity mentions

Tool	Random Forest		SVM		ZeroR	
	ACC %	AUC	ACC %	AUC	ACC %	AUC
Spotlight Mention	78.61	.78	75.05	.67	65.1	.50
TAGME Mention	88.52	**.95**	85.22	.83	65.1	.50
Intersection	78.48	.77	75	.67	65.1	.50
Union	**88.58**	**.95**	85.34	.83	65.1	.50

Given that our Random forest classifier performed better in general for entity mentions, we based our following analysis on its results (Table 11). For the *correct* label, the use of TAGME_Mention or Union provided the highest f1-score (.83), but the use of TAGME_Mention alone provided slightly better precision (.87). On the *not-correct* label, once again, TAGME_Mention and the Union achieved the highest f1-score (.91), but this time the Union alone gave slightly better precision (.90).

Table 11. Precision, recall & f1-score using Entity mentions

Label	Classifier	Tool	Precision	Recall	F1
Correct	Random Forest	Spotlight Mention	.85	.47	.61
		TAGME Mention	**.87**	.79	**.83**
		Intersection	.84	.48	.61
		Union	.86	**.80**	**.83**
	SVM	Spotlight Mention	.77	.40	.53
		TAGME Mention	**.81**	.75	**.78**
		Intersection	.77	.40	.53
		Union	**.81**	**.76**	**.78**
	ZeroR	Spotlight Mention	0	0	0
		TAGME Mention	0	0	0
		Intersection	0	0	0
		Union	0	0	0
Not-correct	Random Forest	Spotlight Mention	.77	**.96**	.85
		TAGME Mention	.89	.94	**.91**
		Intersection	.77	**.95**	.85
		Union	**.90**	.93	**.91**
	SVM	Spotlight Mention	.75	**.94**	.83
		TAGME Mention	**.87**	.91	**.89**
		Intersection	.74	**.93**	.83
		Union	**.87**	.91	**.89**
	ZeroR	Spotlight Mention	.65	1	.79
		TAGME Mention	.65	1	.79
		Intersection	.65	1	.79
		Union	.65	1	.79

An explanation for the difference in performance between Spotlight_Mention and TAGME_Mention is the amount of mentions retrieved by each of the semantic annotators. Spotlight provided fewer annotations for the same answers than TAGME. In addition, our manual inspection of annotations revealed that TAGME tended to produce more accurate annotations than Spotlight. This suggests that higher quantity and quality of semantic annotations leads to a feature set that successfully differentiates between *correct* and *not-correct* answers.

5.3 Entity URI Results

The results presented in Table 12 show that Random forest provided highest accuracy and AUC on each configuration. The best accuracy and AUC were achieved by Random forest with TAGME_URI (86.60% and .94, respectively).

Table 12. Accuracy & AUC using Entity URIs

Tool	Random Forest		SVM		ZeroR	
	ACC %	AUC	ACC %	AUC	ACC %	AUC
Spotlight URI	80.55	.84	77.60	.75	60.8	.45
TAGME URI	**86.60**	**.94**	84.74	.82	65.1	.45
Intersection	77.03	.82	76.44	.74	59	.45
Union	86.50	.94	82.80	.80	63.6	.45

We notice that in terms of accuracy and AUC, TAGME_URI and Union on Random forest are slightly lower than TAGME_Mention and Union for Entity mention features.

Focusing on Random forest as the best performing classifier, we observe that for the *correct* label, the use of TAGME_URI and union of entity URIs provided the best f1-score of .80 (Table 13). In terms of precision, the union of entity URIs had a better performance (.86). For the *not-correct* label, again on Random forest, TAGME_URI and the Union configurations get better f1-score (.90). This time TAGME_URI alone provided the best precision (.88) for this label.

We observed that in some cases, the same mention was associated to different entity URIs in two different answers and that only one of the URIs was correct. When this happens, it affects the quality of the vector representation of student answers by increasing the number of URIs in the VSM vocabulary, thus making the representation even sparser.

Table 13. Precision, recall & f1-score using Entity URIs

Label	Classifier	Tool	Precision	Recall	F1
Correct	Random Forest	Spotlight URI	.82	.64	.72
		TAGME URI	.84	**.76**	**.80**
		Intersection	.77	.63	.69
		Union	**.86**	**.76**	**.80**
	SVM	Spotlight URI	.77	.62	.68
		TAGME URI	**.81**	**.73**	**.77**
		Intersection	.77	.61	.68
		Union	.80	.71	.75
	ZeroR	Spotlight URI	0	0	0
		TAGME URI	0	0	0
		Intersection	0	0	0
		Union	0	0	0
Not-correct	Random Forest	Spotlight URI	.80	.91	.85
		TAGME URI	**.88**	.92	**.90**
		Intersection	.77	.87	.82
		Union	.87	**.93**	**.90**
	SVM	Spotlight URI	.78	.88	.83
		TAGME URI	**.86**	**.91**	**.89**
		Intersection	.76	.87	.81
		Union	.84	.90	.87
	ZeroR	Spotlight URI	.61	1	.76
		TAGME URI	**.65**	1	**.79**
		Intersection	.59	1	.74
		Union	.64	1	.78

5.4 Entity Embedding Results

Among models trained using entity embeddings, the highest accuracy and AUC were achieved by Random forest with the TAGME_URI configuration, as shown in Table 14. For this feature set, we observe that Random forest has higher accuracy with TAGME_URI and Union than SVM on the same configurations; but SVM gets higher accuracy than Random forest using Spotlight_URI and Intersection. However, the AUC for Random forest is still higher than for SVM in all the configurations. We can also observe an increase in accuracy and in AUC (although modest) for the baseline.

Table 14. Accuracy & AUC using Entity embeddings

Tool	Random Forest		SVM		ZeroR	
	ACC %	**AUC**	**ACC %**	**AUC**	**ACC %**	**AUC**
Spotlight URI	80.13	.86	81.13	.71	73.5	.50
TAGME URI	**82.67**	**.90**	75.46	.70	63.7	.50
Intersection	76.43	.81	79.64	.66	74.6	.49
Union	82.45	.89	80.79	.69	73.5	.50

Further inspection of the results obtained on cross-validated models (Table 15) reveals that this time, the highest results differ between classification algorithms. For the *correct* label, we obtained better f1-score with Random forest using the union of entity embeddings (.89). However, SVM provided better precision using Spotlight (.88). The *not-correct* label had both the best precision (.85 using the union of entity embeddings) and f1-score (.87 using TAGME_URI) results using Random forest.

Table 15. Precision, recall & f1-score using Entity embeddings

Label	Classifier	Tool	Precision	Recall	F1
Correct	Random Forest	Spotlight URI	.83	.92	.87
		TAGME URI	**.85**	.64	.73
		Intersection	.81	.90	.85
		Union	.82	**.97**	**.89**
	SVM	Spotlight URI	**.88**	.92	**.88**
		TAGME URI	.74	.50	.60
		Intersection	.82	.93	**.88**
		Union	.82	**.94**	**.88**
	ZeroR	Spotlight URI	.73	1	.85
		TAGME URI	0	0	0
		Intersection	**.75**	1	**.86**
		Union	.73	1	**.86**
Not-correct	Random Forest	Spotlight URI	.68	.47	.56
		TAGME URI	.82	**.93**	**.87**
		Intersection	.56	.37	.44
		Union	**.85**	.41	.55
	SVM	Spotlight URI	.70	.50	.58
		TAGME URI	**.76**	**.90**	**.82**
		Intersection	.67	.39	.50
		Union	.72	.45	.55
	ZeroR	Spotlight URI	0	0	0
		TAGME URI	**.64**	**1**	**.78**
		Intersection	0	0	0
		Union	0	0	0

Even though TAGME_URI provided better precision for the correct label, the union of entity embeddings got better f1-score and recall. The increase in f1-score can be related to the amount of entity URIs provided by the Union (set union of entity URIs from DBpedia Spotlight and TAGME). This suggests that more entities have a positive effect on performance. Similarly, as in accuracy, there was an increase in precision and f1-score on both labels for the baseline classifier.

5.5 Mention Embedding Results

Table 16 shows that when mention embeddings were used as features, SVM achieved the highest accuracy of 81.79% with the Union configuration. This is the first time that Random forest is surpassed by SVM in terms of accuracy, However, Random forest is still outperforming SVM in terms of AUC.

Table 16. Accuracy & AUC using mention embeddings

Tool	Random Forest		SVM		ZeroR	
	ACC %	**AUC**	**ACC %**	**AUC**	**ACC %**	**AUC**
Spotlight Mention	74.83	.74	75.83	.63	73.50	.50
TAGME Mention	80.85	.85	79.66	.76	63.69	.50
Intersection	78.50	.73	79.52	.68	74.40	.48
Union	79.80	**.86**	**81.79**	.71	73.50	.49

As in entity embeddings, the highest results differ between classification algorithms. Table 17 presents detailed results for the performance of Random forest and SVM using mention embeddings. For the *correct* label, the Random forest classifier with the union of mention embeddings had f1-score of .88 (the highest F1 value). For precision, SVM did better with either the intersection or union of mention embeddings (.83). The *not-correct* label had both best precision (.86 using TAGME_Mention) and f1-score (.83 using the union of mention embeddings) with the SVM classifier.

Table 17. Precision, recall & f1-score using mention embeddings

Label	Classifier	Tool	Precision	Recall	F1
Correct	Random Forest	Spotlight Mention	.78	.91	.84
		TAGME Mention	**.82**	.61	.70
		Intersection	.81	.93	.87
		Union	.80	**.98**	**.88**
	SVM	Spotlight Mention	.80	.90	.85
		TAGME Mention	.78	.61	.69
		Intersection	**.83**	.92	.87
		Union	**.83**	**.94**	**.88**
	ZeroR	Spotlight Mention	.73	1	.85
		TAGME Mention	0	0	0
		Intersection	**.74**	1	.85
		Union	.73	1	.85
Not-correct	Random Forest	Spotlight Mention	.55	.30	.39
		TAGME Mention	.81	**.92**	**.86**
		Intersection	.64	.36	.46
		Union	**.83**	.30	.44
	SVM	Spotlight Mention	.57	.36	.44
		TAGME Mention	**.80**	**.90**	**.85**
		Intersection	.65	.44	.52
		Union	.75	.48	.58
	ZeroR	Spotlight Mention	0	0	0
		TAGME Mention	**.64**	**1**	**.78**
		Intersection	0	0	0
		Union	0	0	0

6. FEATURE SELECTION

In this section, we describe the results obtained when applying two feature selection methods to our dataset: mean decrease impurity (MDI) and mean decrease accuracy (MDA). Both methods employ random trees to measure the importance of a feature [7]. We trained different classifiers with the selected features and compared their results to the same classifiers without feature selection.

First, we calculated the MDA and MDI scores for each feature in our data set and kept only features with scores strictly higher than 0. Negative or zero MDA/MDI values were either detrimental or unhelpful to the performance of the classifiers. Table 18 shows the number of features before and after feature selection.

Table 18. Number of remaining features with and without (WFS) feature selection

	Technique		
Features	**WFS**	**MDA**	**MDI**
N-gram	700	90	205
Entity Mention	665	99	179
Entity URI	875	109	236
Entity Embedding	100	83	84
Mention Embedding	300	161	117

Then, we trained and evaluated Random Forest, SVM, and ZeroR classifiers using each of the top performing configurations per feature set in terms of f1-score to compare the results obtained with and without feature selection. The obtained results (Figure 1) show that in most cases, feature selection led to a slight increase in the accuracy of our classifiers. Specifically, MDA improved the accuracy of the classifiers in every case by as much as 4.9 for SVM using mention embeddings as features. However, overall Random forest generally remained the best, in terms of accuracy, with and without feature selection.

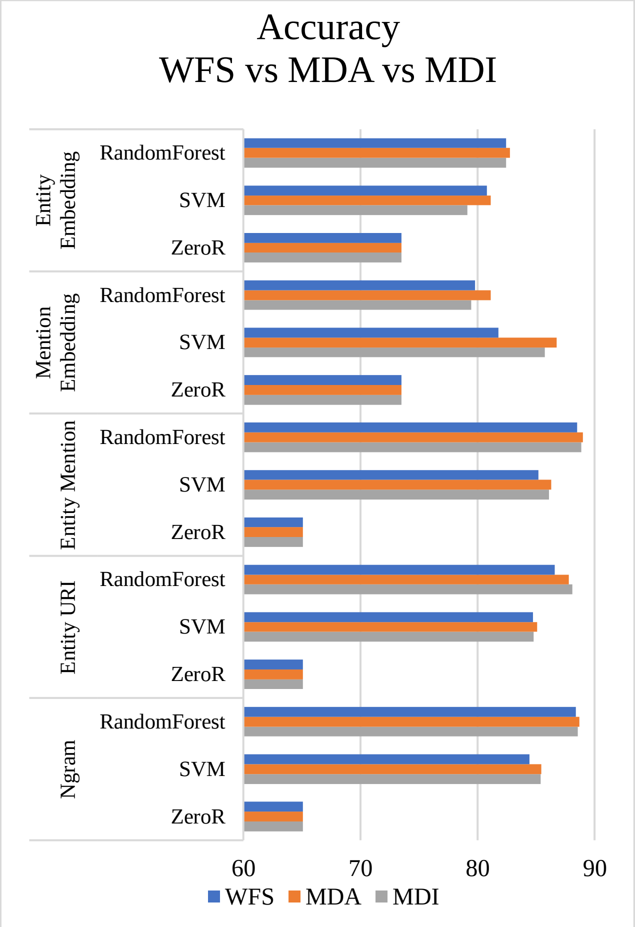

Figure 1. Accuracy without feature selection (WFS) versus MDA & MDI

7. DISCUSSION

Overall, our Random forest classifiers proved the best in terms of accuracy and AUC. The only exception is with mention embeddings in which SVM did better in terms of accuracy by at most 1 percentage point. Therefore, we base our conclusions only on Random forest.

In terms of accuracy, there was not much difference between several feature sets as shown on Figure 2. The two best feature sets for accuracy were entity mentions with the union (88.58%) or TAGME configurations (88.52%) and n-gram features with the unigrams configuration (88.40%); these feature sets achieved the highest AUC (.95), as well.

In terms of precision (Figure 3), n-grams outperformed other feature sets for the *correct* label (.93) and entity mentions obtained the best results for the *not-correct* label using the union and TAGME configurations (.90, .89).

For F1-score (Figure 4), entity embeddings achieved the highest score for the *correct* label (.89 using the union configuration) closely followed by mention embeddings (union). Entity mentions (using the union or TAGME configurations) and unigrams did better for *not-correct* (.91) followed by entity URIs (.90 with TAGME and union) and n-grams.

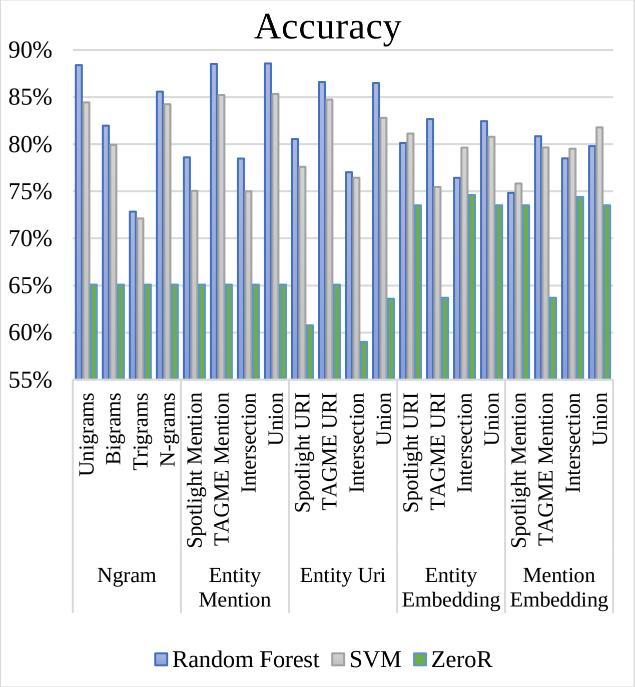

Figure 2. Accuracy results

When considering which class (*correct*, *not-correct*) we were best able to predict in terms of precision (Figure 3), we found that the detection of *correct* answers was better than *not-correct answers*, with differences ranging from .01 to .25 with Random forest. N-gram features were better at detecting *correct* answers than *not-correct* ones; while entity mentions did better for the *not-correct* (using union or TAGME) label. In 14 out of our 20 possible configurations, the classifiers were more precise in detecting *correct* answers. This is the case despite the unbalanced ratio of 35% *correct* answers and 65% *not-correct* answers used for training. When we focus on the f1-score (Figure 4) we obtain better results for the *not-correct* label. We observe that the union configuration for entity embeddings and mention embeddings is the best for *correct* answers while entity mentions (TAGME or union) followed by unigrams outperform the other features for the *not-correct* answers.

On average, unigrams are the best at differentiating between correct and not-correct labels in terms of precision while entity mentions (either with TAGME or Union) is preferred in terms of f1-score.

The best configuration based on semantic annotations depends on the considered evaluation metric. Based on accuracy, features that use mentions (entity mentions and mention embeddings) performed better with either union or TAGME. The feature sets that use URIs (entity URIs and entity embeddings) performed better with URIs obtained using TAGME. In both cases, the use of TAGME alone obtains either the best result or is very close to the highest value. For f1-score, the use of TAGME for entity mentions and entity URIs provided the same results as the union for both labels; additionally, TAGME and union are also the best configurations for both entity mentions and entity URIs. Entity embeddings and mention embeddings had their best f1-score on the *correct* label using the union, but better f1-score for *not-correct* using TAGME alone. When we average the f1-score for both labels, we obtain higher results with TAGME. The reason for very similar results with TAGME and the union is that the annotations provided by Spotlight were often a subset of those provided by TAGME.

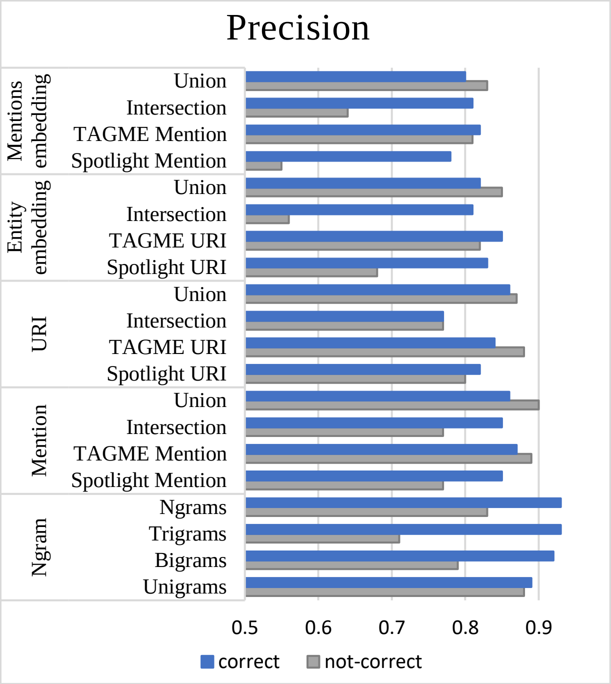

Figure 3. Precision results for Random forest

Both entity and mention embeddings performed worse than n-gram features and semantic annotations models based on accuracy. However, one interesting observation is that, for the *correct* label, entity and mention embeddings outperformed all features on f1-score (Figure 4). Entity embeddings obtained slightly better results (precision, f1-score and accuracy) compared to mention embeddings.

Our feature selection efforts show that MDI did not consistently improve the overall accuracy of our classifiers. It was the MDA feature selection technique which provided improvement in all the cases. The increase in accuracy ranged from .3% to 4.9%.

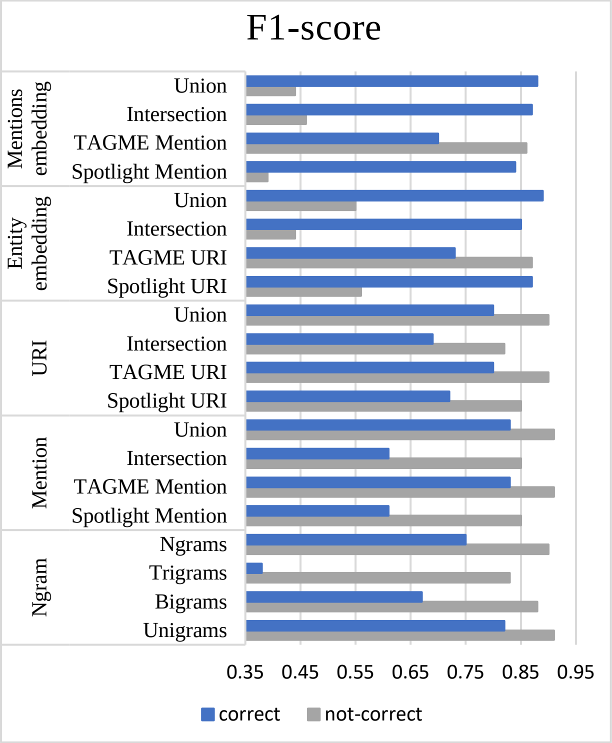

Figure 4. F1-score results for Random Forest

8. CONCLUSION

In this paper, we compared several vector-based feature sets coupled with classifiers for the ASAG task.

In general, we showed that on average, entity mention features (TAGME or union) are the top features in terms of f1-score while n-gram features (unigrams) are the best in terms of precision. For the detection of *correct* answers, we showed that n-gram features (trigrams and n-grams) and features based on embeddings (entity and mention embeddings with the union configuration) are the most effective in terms of precision and f1-score respectively. In terms of semantic annotations, TAGME provided the best accuracy for each feature with the exception of entity mentions, where the union configuration slightly outperformed TAGME alone. Finally, the MDA feature selection technique slightly improved the accuracy of all the classifiers.

One of the main limitations of this study is the unbalanced set of labeled answers available in the corpus. Another limitation is associated with the configuration of semantic annotators as we only tested the default level of confidence for each annotator. One additional limitation, for mention embeddings specifically, is the relatively low coverage obtained using GloVe. We plan to address these limitations in future work by testing the proposed features against other available ASAG datasets. We also intend to experiment with varying the level of confidence and similar parameters of the semantic annotators. Another important step will be to exploit a combination of the current features to benefit from their respective strengths for the correct and not correct labels. Finally, we will explore other methods for response classification using additional features that exploit model answers and deep learning architectures.

9. ACKNOWLEDGMENTS

This research is supported by an INSIGHT grant from the Social Sciences and Humanities Research Council of Canada (SSHRC).

10. REFERENCES

[1] Basu, S., Jacobs, C., and Vanderwende, L. 2013. Powergrading: a clustering approach to amplify human effort for short answer grading. *Transactions of the Association for Computational Linguistics*, 1, 391-402.

[2] Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. 2007. Greedy layer-wise training of deep networks. In *Proceedings of Advances in Neural Information Processing Systems Conference*, 153-160.

[3] Biggs, J., and Tang, C. 2011. *Teaching for quality learning at university (4th ed.).* London, UK: McGraw-Hill International.

[4] Burrows, S., Gurevych, I., and Stein, B. 2015. The eras and trends of automatic short answer grading. *International Journal of Artificial Intelligence in Education*, 25, 1, 60–117.

[5] Dessus, P., Lemaire, B., and Vernier, A. 2000. Free-text assessment in a virtual campus. In *Proceedings of the 3rd International Conference on Human System Learning*, 61-75.

[6] Ferragina, P., and U. Scaiella. 2010. TAGME: On-the-fly Annotation of Short Text Fragments (by Wikipedia Entities). In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, 1625–1628.

[7] Gilles Louppe, Louis Wehenkel, Antonio Sutera, and Pierre Geurts. 2013. Understanding variable importances in forests of randomized trees. In *Proceedings of Advances in Neural Information Processing Systems Conference*. 431–439.

[8] Heilman, M., and Madnani, N. 2013. ETS: Domain adaptation and stacking for short answer scoring. In Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval)*, 2, 275-279.

[9] Jayashankar, S., and Sridaran, R. 2017. Superlative model using word cloud for short answers evaluation in eLearning. *Education and Information Technologies*, 22, 5, 2383-2402.

[10] Jordan, S., and Mitchell, T. 2009. e-Assessment for learning? The potential of short-answer free-text questions with tailored feedback. *British Journal of Educational Technology*, 40, 2, 371–385.

[11] Klein, R., Kyrilov, A., and Tokman, M. 2011. Automated assessment of short free-text responses in computer science using latent semantic analysis. In G. Roßling, T. Naps, C. Spannagel (Eds.), In *Proceedings of the 16th annual joint conference on innovation and technology in computer science education*, 158–162. Darmstadt: ACM. (CAPS'3), 61-76.

[12] Leacock, C., and Chodorow, M. 2003. C-rater: automated scoring of short-answer questions, *Computers and the Humanities*, 37, 4, 389–405.

[13] Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., van Kleef, P., Auer, Sö. and Bizer, C. 2015. DBpedia - A Large-scale,

Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web Journal*, 6, 167-195.

[14] Madnani, N., Loukina, A., and Cahill, A. (2017). A Large Scale Quantitative Exploration of Modeling Strategies for Content Scoring. *In Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, 457-467.

[15] Magooda, A. E., Zahran, M. A., Rashwan, M., Raafat, H. M., and Fayek, M. B. 2016. Vector Based Techniques for Short Answer Grading. In *Proceedings of Florida Artificial Intelligence Research Society Conference* (*FLAIRS)*, 238-243.

[16] McDonald, J., Bird, R. J., Zouaq, A., and Moskal, A. C. M. 2017. Short answers to deep questions: supporting teachers in large-class settings, *Journal of Computer Assisted Learning*. 33, 4 306–319.

[17] McDonald, J., Knott, A., and Zeng, R. 2012. Free-text input vs menu selection: exploring the difference with a tutorial dialogue system. In *Proceedings of the Australasian Language Technology Association Workshop 2012*, 97-105.

[18] McDonald, J., Knott, A., Zeng, R., and Cohen, A. 2011. Learning from student responses: A domain-independent natural language tutor. In *Proceedings of the Australasian Language Technology Association Workshop*, 148-156.

[19] Mendes, P. N., Jakob, M., García-Silva, A., and Bizer, C. (2011, September). DBpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, ACM, 1-8.

[20] Mikolov, T., Chen, K., Corrado, G. and Dean, J. 2013. Efficient Estimation of Word Representations in Vector Space, In *Proceedings of Workshop at International Conference on Learning Representations ICLR.*

[21] Oren, E, Moller K, Scerri, S, Handschuh, S and Sintek, M 2006, What are Semantic Annotations? *Relatório técnico. DERI Galway*, 9, 62-75.

[22] Papineni, K., Roukos, S., Ward, T., and Zhu, W. J. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 311-318.

[23] Pennington, J., Socher, R., and Manning, C. 2014. GloVe: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532-1543.

[24] Reilly, E. D., Stafford, R. E., Williams, K. M., and Corliss, S. B. 2014. Evaluating the validity and applicability of automated essay scoring in two massive open online courses, *The International Review of Research in Open and Distributed Learning*, 15, 5, 83-98.

[25] Roy, S., Bhatt, H. S., and Narahari, Y. 2016. Transfer Learning for Automatic Short Answer Grading. In *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI)*, Hague, Netherlands, 1622-1623.

[26] Riordan, B., Horbach, A., Cahill, A., Zesch, T., and Lee, C. M. 2017. Investigating neural architectures for short answer scoring. *In Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, 159-168.

[27] Sakaguchi, K., Heilman, M., and Madnani, N. 2015. Effective feature integration for automated short answer scoring. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1049-1054.

[28] Shermis, M. D. 2015. Contrasting state-of-the-art in the machine scoring of short-form constructed responses. *Educational Assessment*, 20, 1, 46-65.

[29] Tack, A., François, T., Roekhaut, S., and Fairon, C. 2017. Human and Automated CEFR-based Grading of Short Answers. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, 169-179.

[30] The Hewlett Foundation. 2012. The Automated Student Assessment Prize: Short Answer Scoring. http://www.kaggle.com/c/asap-sas

[31] Wolpert, D. H. 1992. Stacked generalization. Neural Networks, 5, 2, 241–259.

[32] Yuan, L., and Powell, S. 2013. MOOCs and open education: Implications for higher education. Centre for Educational Technology & Inoperability Standards. Retrieved from http://publications.cetis.ac.uk/wp-content/uploads/2013/03/MOOCs-and-Open-Education.pdf

[33] Zhang, Y., Shah, R., and Chi, M. 2016. Deep Learning + Student Modeling + Clustering: a Recipe for Effective Automatic Short Answer Grading. In *Proceedings of the 9th International Conference on Educational Data Mining* (*EDM)*, 562-567.

[34] Zhou, H., Zouaq, A., and Inkpen, D. 2017. DBpedia Entity Type Detection Using Entity Embeddings and N-Gram Models. In *Proceedings of International Conference on Knowledge Engineering and the Semantic Web*, 309-322.