Automated Short Answer Grading: A Simple Solution for a Difficult Task

Stefano Menini[†], **Sara Tonelli**[†], **Giovanni De Gasperis**[‡], **Pierpaolo Vittorini**[‡]
[†]Fondazione Bruno Kessler (Trento), [‡]University of L'Aquila
`{menini,satonelli}@fbk.eu`
`{giovanni.degasperis,pierpaolo.vittorini}@univaq.it`

Abstract

English. The task of short answer grading is aimed at assessing the outcome of an exam by automatically analysing students' answers in natural language and deciding whether they should pass or fail the exam. In this paper, we tackle this task training an SVM classifier on real data taken from a University statistics exam, showing that simple concatenated sentence embeddings used as features yield results around 0.90 F1, and that adding more complex distance-based features lead only to a slight improvement. We also release the dataset, that to our knowledge is the first freely available dataset of this kind in Italian.[1]

1 Introduction

Human grading of open ended questions is a tedious and error-prone task, a problem that has become particularly pressing when such an assessment involves a large number of students, like in an Academic setting. One possible solution to this problem is to automate the grading process, so that it can facilitate teachers in the correction and enable students to receive immediate feedback. Research on this task has been active since the '60s (Page, 1966), and several computational methods have been proposed to automatically grade different types of texts, from longer essays to short text answers. The advantages of this kind of automatic assessment do not concern only the limited time and effort required to grade tests compared with a manual assessment, but include also the reduction of mistakes and bias introduced by humans, as well as a better formalization of assessment criteria.

In this paper, we focus on tests comprising short answers to natural language questions, proposing a novel approach to binary *automatic short answer grading* (ASAG). This has proven particularly challenging because an understanding of natural language is required, without having much textual context, while grading multiple-choice questions can be straightforwardly assessed, given that there is only one possible correct response to each question. Furthermore, the tests considered in this paper are taken from real exams on statistical analyses, with low variability, a limited vocabulary and therefore little lexical difference between correct and wrong answers.

The contribution of this paper is two-fold: we create and release a dataset for short-answer grading containing real examples, which can be freely downloaded at `https://zenodo.org/record/3257363#.XRsrn5P7TLY`. Besides, we propose a simple approach that, making use only of concatenated sentence embeddings and an SVM classifier, achieves up to 0.90 F1 after parameter tuning.

2 Related Work

In the literature, several works have been presented on automated grading methods, to assess the quality of answers in written examinations. Several types of answers have been addressed, from essays (Kanejiya et al., 2003; Shermis et al., 2010), to code (Souza et al., 2016). Here we focus on works related to short answers, which are the target of our tests. With short answers we refer to open questions, given in natural language, usually with the length of one paragraph, recalling external knowledge (Burrows et al., 2015). When assessing the grading of short answers we face two main issues, *i)* the grading itself and *ii)* the presence of appropiate datasets.

ASAG can be tackled with several approaches, including pattern matching (Mitchell et al., 2002), looking for specific concepts or keywords in the answers (Callear et al., 2001; Leacock and Chodorow, 2003; Jordan and Mitchell, 2009), using bag of words and matching terms (Cutrone et al., 2011) or relying on LSA (Klein et al., 2011). Some other solutions rely more heavily on NLP techniques, for example by extracting metrics and features that can be used for text classification such as the overlap of n-grams or POS between student's and teacher's answers (Bailey and Meurers, 2008; Meurers et al., 2011). Some attempts have been made also to use similarity between word embeddings as a feature (Sultan et al., 2016; Sakaguchi et al., 2015; Kumar et al., 2017).

Another aspect that can affect the performance of different ASAG approaches is the target of automated evaluation. We can for instance assess the quality of the text (Yannakoudakis et al., 2011), its comprehension and summarization (Madnani et al., 2013), or, as in our case, the knowledge of a specific notion. Each task would therefore need a specific dataset as a benchmark. Other dimensions affecting the approach to ASAG and its performance are also the school level for which an assessment is required (e.g primary school vs. university) as well as its domain, e.g. computer science (Gütl, 2007), biology (Siddiqi and Harrison, 2008) or math (Leacock and Chodorow, 2003). As for Italian, we are not aware of existing automated grading approaches, nor of available datasets specifically released to foster research in this direction. These are indeed the main contributions of the current paper.

3 Task and Data Description

The short grading task that we analyse in this paper is meant to automatize part of the exam that students of Health Informatics in the degree course of Medicine and Surgery of the University of L'Aquila (Italy) are required to pass. It includes two activities: a statistical analysis in R and the explanation of the results in terms of clinical findings. While the evaluation of the first part has already been automatized through automated grading of R code snippets (Angelone and Vittorini, 2019), the second task had been addressed by the same authors using a string similarity approach, which however did not yield satisfying results. Indeed, they used Levenshtein distance to compute the distance between the students' answer and a gold standard (i.e. correct) answer, but the approach failed to capture the semantic equivalence between the two sentences, while focusing only on the lexical one.

For example, an exam provided students with data about surgical operations, subjects, scar visibility and hospital stay, and asked to compute several statistical measures in R, such as the absolute and relative frequencies of the surgical operations. Then, students were required to comment in plain text on some of the analyses, for example state whether some data are extracted from a normal distribution. For this second part of the exam, the teacher prepared a "gold answer", i.e. the correct answer. Two real examples from the dataset are reported below.

Correct answer pair:

> (Student) *Poiché il p-value e maggiore di 0.05 in entrambi i casi, la distribuzione è normale, procediamo con un test parametrico per variabili appaiate.*
>
> (Gold) *Siccome tutti i test di normalità presentano un p>0.05, posso utilizzare un test parametrico.*

Wrong answer pair:

> (Student) *Siccome p<0.05,la differenza fra le due variabili è statisticamente significativa.*
>
> (Gold) *Siccome il t-test restituisce un p-value > di 0.05, non posso generalizzare alla popolazione il risultato osservato nel mio campione, e quindi non c'è differenza media di peso statisticamente significativa fra i figli maschi e femmine.*

The goal of our task is, given each pair, to train a classifier and label correct and wrong students' answers. An important aspect of our task is that the correctness of an answer is not defined with respect to the question, which is not used for classification. For the moment we also focus on binary classification, to determine whether an answer is correct or not, without providing a numeric score on how much it is correct or wrong. With the data organized into student-professor answers pairs, the classification is done considering *i)* the semantic content of the answers (represented through word embeddings *ii)* features related to the pair structure of the data such as the overlap or the distance between the two texts. The adopted features are explained in detail in Section 4.1.

3.1 Dataset

The dataset available at `https://zenodo.org/record/3257363#.XR5i8ZP7TLY` has been partially collected using data from real statistics exams spanning different years, and partially extended by the authors of this paper. The dataset contains the list of sentences written by students, with a unique sentence ID, the type of statistical analysis it refers to (if either given for the hypothesis or normality test), its degree in a range from 0 to 1, and its fail/-pass result, flanked with a manually defined gold standard (i.e. the correct answer). The degree is a numerical score manually assigned to each answer, which takes into account whether an answer is partially correct, mostly correct or completely wrong. Based on this degree, the pass/fail decision was taken, i.e. if degree < 0.6 then fail, otherwise pass.

In order to increase the number of training instances and achieve a better balance between the two classes, we manually negated a set of correct answers and reversed the corresponding fail/pass result, adding a set of negated gold standard sentences for a total of 332 new pairs. We also manually paraphrased 297 of the original gold standard sentences, so that we created some additional pairs. Overall the dataset consists of 1,069 student/gold standard answer pairs, 663 of which are labeled as *"pass"* and 406 as *"fail"*.

4 Classification framework

Although several works have explored the possibility to automatically grade short text answers, these attempts have mainly focused on English. Furthermore, the best performing ones strongly rely on knowledge bases and syntactic analyses (Mohler et al., 2011), which are hard to obtain for Italian. We therefore test for the first time the potential of sentence embeddings to capture pass or fail judgments in a supervised setting, where the only required data are *a)* a training/test set and *b)* sentence embeddings (Bojanowski et al., 2017) trained using fastText[2].

4.1 Method

Since we cast the task in a supervised classification framework, we first need to represent the pairs of student/gold standard sentences as features. Two different types of features are tested: **distance-based features**, which capture the similarity of the two sentences using measures based on lexical and semantic similarity, and **sentence embeddings** features, whose goal is to represent the semantics of the two sentences in a distributional space.

All sentences are first preprocessed by removing the stopwords such as articles and prepositions, and by replacing mathematical notations with their transcription in plain language, e.g. "$>$" with *"maggiore di"* (*greater than*). We also perform part of speech tagging, lemmatisation and affix recognition using the TINT NLP Suite for Italian (Aprosio and Moretti, 2018). Then on each pair of sentences the following distance-based features are computed:

- Token overlap: a feature representing the number of overlapping tokens between the two sentences normalised by their length. This feature captures the lexical similarity between the two strings.
- Lemma overlap: a feature representing the number of overlapping lemmas between the two sentences normalised by their length. Like the previous one, this feature captures the lexical similarity between the two strings.
- Presence of negations: this feature represents whether a content word is negated in one sentence and not in the other. For each sentence, negations are recognised based on the NEG PoS tag or the affix 'a-' or 'in-' (e.g. *indipendente*), and then the first content word occurring after the negation is considered. We extract two features, one for each sentence, and the values are normalised by their length.

Other distance-based features are computed at sentence level, and to this purpose we employ fastText (Bojanowski et al., 2017), an extension of word embeddings (Mikolov et al., 2013; Pennington et al., 2014) developed at Facebook that is able to deal with rare words by including subword information, and representing sentences basically by combining vectors representing both words and subwords. To generate these embeddings we start from the pre-computed Italian language model[3] trained on Common Crawl and Wikipedia. The latter, in particular, is suitable for our domain, since it includes also scientific content and statistics pages, therefore the language of the exam should be well represented in our model. The embeddings are created using continuous bag-of-word with position-weights, a dimension of 300, character n-grams of length 5, a window of size 5 and 10 negatives.

[2] https://fasttext.cc/

[3] https://fasttext.cc/docs/en/crawl-vectors.html

Then, the embedding of the sentences written by the students and the gold standard ones are created by combining the word and the subword embeddings with the fastText library. Each sentence is therefore represented through a 300 dimensional embedding. Based on this, we extract four additional distance-based features:

- Embeddings cosine: the cosine between the two sentence embeddings is computed. The intuition behind this feature is that the embeddings of two sentences with a similar meaning would be close in a multidimensional space

- Embeddings cosine (lemmatized): the same feature as the previous one, with the only difference that the sentences are first lemmatised before creating the embeddings

- Word Mover's Distance (WMD): WMD is a similarity measures based on the minimum amount of distance that the embedded words of one document need to move to reach the embedded words of another document (Kusner et al., 2015) in a multidimensional space. Compared with other existing similarity measures, it works well also when two sentences have a similar meaning despite having few words in common. We apply this algorithm to measure the distance between the solutions proposed by the students and the ones in the gold standard.

- Word Mover's Distance (lemmatized): the same feature as the previous one, with the only difference that the sentences are first lemmatised before creating the embeddings

The sentence embeddings used to compute the distance features are also tested as features in isolation: a 600 dimensional vector is indeed created by concatenating each sentence embeddings composing a student answer – gold standard pair. This representation is then directly fed to the classifier. We adopt this solution inspired by recent approaches to natural language inference using the concatenation of premise and hypothesis (Bowman et al., 2015; Kiros and Chan, 2018).

As for the supervised classifier, we use support vector machines (Scholkopf and Smola, 2001), which generally yield satisfying results in classification tasks with a limited number of training instances (as opposed to deep learning approaches).

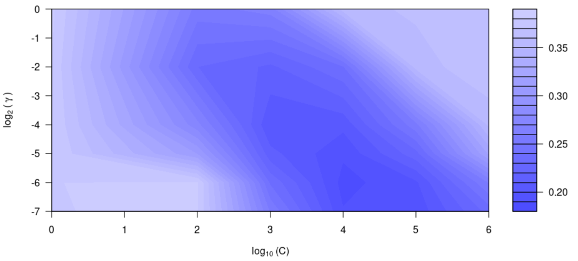

Figure 1: Plot for parameter tuning

We then proceeded to find the best C and γ parameters by means of grid-search tuning (Hsu et al., 2016), through a 10-fold cross-validation to prevent to overfit the model. Finally, with the parameters that returned the best performance, we finalised the classifier and calculated its accuracy and F1 score. The analyses were performed using R 3.6.0 with caret v6.0-84 and e1071 v1.7-2 packages (R Core Team, 2018).

4.2 Results

Figure 1 shows the plot summarising the tuning process. In summary, within the explored area, the best parameters were found to be $C = 10^4$ and $\gamma = 2^{-6}$. The resulting tuned model produced the following results:

- Accuracy = 0.891 (balanced accuracy = 0.876);

- F1 score = 0.914;

With a similar approach, we also tuned the classifier when fed with only the concatenated sentence embeddings as features (i.e., without distance-based features). With best parameters $C = 10^3$ and $\gamma = 2^{-3}$, the results were:

- Accuracy = 0.885 (balanced accuracy = 0.870);

- F1 score = 0.909;

To evaluate the quality of the model learned with these two configurations, and make sure that it does not overfit, we perform an additional test: we collect a small set of students' answers from a different statistics exam than the one used to create the training set. This is done on novel data by collecting students' answers from a small number of new questions, and manually creating new gold answers to be used in the pairs. Overall, we obtain

77 new answer pairs, consisting of 14 wrong and 63 correct answers. We then run the best performing model with all features and using only sentence embeddings (same C and γ as before). The results are the following:

- Accuracy using all features $= 0.7838$ (balanced accuracy $= 0.5965$);
- F1 score 0.8710;

while the results achieved using only sentence embeddings are:

- Accuracy $= 0.7973$ (balanced accuracy $= 0.6349$);
- F1 score $= 0.8780$;

5 Discussion

The results presented in the previous section show only a small increase in performance when using the distance-based features in addition to the sentence embeddings after tuning both configurations. This outcome highlights the effectiveness of using sentence embeddings to represent the semantic content of the answers in tasks where student's and gold solutions are very similar to each other. In fact, the sentence pairs in our dataset show a high level of word overlap, and the only discriminant between a correct and a wrong answer is sometimes only the presence of "$<$" instead of "$>$", or a negation.

The second experiment, where the same configuration is run on a test set taken from a statistics exam on different topics, shows an overall decrease in performance as expected, but the classification accuracy is still well above the most frequent baseline. In this setting, using only the sentence embeddings yields a slightly better performance than including the other features, showing that they are more robust with respect to a change of topic.

In general terms, despite the accurate parameter tuning, the classification approach seems to be applicable to short answer grading tests different from the data on which the training was done, provided that the student's and gold answer types are the same as in our dataset (i.e. limited length, limited lexical variability).

6 Conclusions

In this paper, we have presented a novel dataset for short answer grading taken from a real statistics exam, which we make freely available. To our knowledge, this is the first dataset of this kind. We also introduce a simple approach based on sentence embeddings to automatically identify which answers are correct or not, which is easy to replicate and not computationally intensive.

In the future, the work could be extended in several directions. First of all, it would be interesting to use deep-learning approaches instead of SVM, but for that more training data are needed. These could be collected in the upcoming exam sessions at University of L'Aquila. Another refinement of this work would be to grade the tests by assigning a numerical score instead of a pass/fail judgment. Since such scores are already included in the released dataset (the degrees), this would be quite straightforward to achieve. Finally, we plan to test the classifier by integrating it in an online evaluation tool, through which students can submit their tests and the trainer can run an automatic pass/fail assignment.

References

Anna Maria Angelone and Pierpaolo Vittorini. 2019. The Automated Grading of R Code Snippets: Preliminary Results in a Course of Health Informatics. In *Proc. of the 9th International Conference in Methodologies and Intelligent Systems for Technology Enhanced Learning*. Springer.

Alessio Palmero Aprosio and Giovanni Moretti. 2018. Tint 2.0: an all-inclusive suite for NLP in italian. In *Proceedings of the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018), Torino, Italy, December 10-12, 2018.*

Stacey Bailey and Detmar Meurers. 2008. Diagnosing meaning errors in short answers to reading comprehension questions. In *Proceedings of the Third Workshop on Innovative Use of NLP for Building Educational Applications*, pages 107–115. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, September. Association for Computational Linguistics.

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

David H Callear, Jenny Jerrams-Smith, and Victor Soh. 2001. Caa of short non-mcq answers.

Laurie Cutrone, Maiga Chang, et al. 2011. Autoassessor: Computerized assessment system for marking student's short-answers automatically. In *2011 IEEE International Conference on Technology for Education*, pages 81–88. IEEE.

Christian Gütl. 2007. e-examiner: towards a fully-automatic knowledge assessment tool applicable in adaptive e-learning systems. In *Proceedings of the 2nd international conference on interactive mobile and computer aided learning*, pages 1–10. Citeseer.

Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. 2016. A Practical Guide to Support Vector Classification. Technical report, National Taiwan University.

Sally Jordan and Tom Mitchell. 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.

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

Jamie Kiros and William Chan. 2018. Inferlite: Simple universal sentence representations from natural language inference data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4868–4874.

Richard Klein, Angelo Kyrilov, and Mayya Tokman. 2011. Automated assessment of short free-text responses in computer science using latent semantic analysis. In *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education*, pages 158–162. ACM.

Sachin Kumar, Soumen Chakrabarti, and Shourya Roy. 2017. Earth mover's distance pooling over siamese lstms for automatic short answer grading. In *IJCAI*, pages 2046–2052.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International Conference on Machine Learning*, pages 957–966.

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

Nitin Madnani, Jill Burstein, John Sabatini, and Tenaha O'Reilly. 2013. Automated scoring of a summary-writing task designed to measure reading comprehension. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–168.

Detmar Meurers, Ramon Ziai, Niels Ott, and Stacey M Bailey. 2011. Integrating parallel analysis modules to evaluate the meaning of answers to reading comprehension questions. *International Journal of Continuing Engineering Education and Life-Long Learning*, 21(4):355–369.

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

Tom Mitchell, Terry Russell, Peter Broomhead, and Nicola Aldridge. 2002. Towards robust computerised marking of free-text responses.

Michael Mohler, Razvan Bunescu, and Rada Mihalcea. 2011. Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 752–762, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ellis B Page. 1966. The imminence of grading essays by computer. *The Phi Delta Kappan*, 47(5):238–243.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP*.

R Core Team. 2018. R: A Language and Environment for Statistical Computing.

Keisuke Sakaguchi, Michael Heilman, and Nitin Madnani. 2015. Effective feature integration for automated short answer scoring. In *Proceedings of the 2015 conference of the North American Chapter of the association for computational linguistics: Human language technologies*, pages 1049–1054.

Bernhard Scholkopf and Alexander J Smola. 2001. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

Mark D Shermis, Jill Burstein, Derrick Higgins, and Klaus Zechner. 2010. Automated essay scoring: Writing assessment and instruction. *International encyclopedia of education*, 4(1):20–26.

Raheel Siddiqi and Christopher Harrison. 2008. A systematic approach to the automated marking of short-answer questions. In *2008 IEEE International Multitopic Conference*, pages 329–332. IEEE.

Draylson M Souza, Katia R Felizardo, and Ellen F Barbosa. 2016. A systematic literature review of assessment tools for programming assignments. In *2016 IEEE 29th International Conference on Software Engineering Education and Training (CSEET)*, pages 147–156. IEEE.

Md Arafat Sultan, Cristobal Salazar, and Tamara Sumner. 2016. Fast and easy short answer grading with high accuracy. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1070–1075.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading esol texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 180–189. Association for Computational Linguistics.