



Paraphrase Generation and Supervised Learning for Improved Automatic Short Answer Grading

Leila Ouahrani¹ · Djamal Bennouar¹

Accepted: 22 December 2023

© International Artificial Intelligence in Education Society 2024

Abstract

We consider the reference-based approach for Automatic Short Answer Grading (ASAG) that involves scoring a textual constructed student answer comparing to a teacher-provided reference answer. **The reference answer does not cover the variety of student answers as it contains only specific examples of correct answers.** Considering other language variants of the reference answer can handle variability in student responses and improve scoring accuracy. Alternative reference answers may be possible, but manually creating them is expensive and time-consuming. In this paper, we consider two issues: First, we need to automatically generate various reference answers that can handle the diversity of student answers. Second, we should provide an accurate grading model that improves sentence similarity computation using multiple reference answers. Therefore, our proposed approach to solve both problems highlights two components. First, we provide a sequence-to-sequence deep learning model that targets generating plausible paraphrased reference answers conditioned on the provided reference answer. Secondly, we propose a supervised grading model based on sentence embedding features. The grading model enriches features to improve accuracy considering multiple reference answers. Experiments are conducted both in Arabic and English. They show that the paraphrase generator produces accurate paraphrases. Using multiple reference answers, the proposed grading model achieves a Root Mean Square Error of 0,6955, a Pearson correlation of 88,92% for the Arabic dataset, an RMSE of 0,7790, and a Pearson correlation of 73,50% for the English dataset. While fine-tuning pre-trained transformers on the English dataset provided state-of-the-art performance (RMSE: 0.7620), our approach yields comparable results. Simple to construct, load, and embed into the Learning Management System question engine with low computational complexity, the proposed approach can be easily integrated into the Learning Management System to support the assessment of short answers.

Keywords Short Answer Grading · Paraphrase generation · Automatic reference answer generation · Encoder-decoder · Natural Language Processing

Extended author information available on the last page of the article

Published online: 26 January 2024

Springer

Introduction

E-learning is the process of learning using Information and Communication Technologies (ICTs). It has become popular in higher education. Assessment is one of its most fundamental parts. Unfortunately, online education is criticized for the lack of educational commitment because of a significant dropout rate (Qiu, 2019). Higher education institutions offer classes to a sizable number of students. Such a large environment presents particular challenges for instructors. One of these challenges is student assessment.

Teachers are expected to design effective support assessments face-to-face and online. Online tests provide additional flexibility in terms of when and where the tests take place, as well as when feedback is provided to students (Khan & Khan, 2019). The need for large-scale assessments and the cost of manual marking has led to the development of automated assessment (Whitelock & Bektik, 2018).

The various modes of accessing the learner's knowledge are objective and subjective tests. As the objective test usually comprises multiple-choice questions, true/false questions, matching questions, etc., non-objective tests focus on descriptive answers to open-ended questions namely short-answer questions and essay questions (Ashton et al., 2005; Jordan, 2013). With an increase in technology for learning and the shift to more student-centered approaches, open-ended questions are becoming more common in higher education (Beckman et al., 2019). Open questions provide students to develop their interpretations of requirements, and online technologies offer greater flexibility and allow new types of interactions with teachers and students (Sychev et al., 2020). According to Bloom's Taxonomy of Educational Objectives (Bloom, 1984), short answer questions capture the learner's ability to acquire, understand, and synthesize. Short answer questions require concise and focused written answers using acceptable vocabulary, related to the subject. Short answer questions are supposed to target only a few facts and concepts (Ziai et al., 2012). Automatic Short Answer Grading (ASAG) consists of "assessing constructed short free natural language responses using computational methods" (Burrows et al., 2015). Developing an effective scoring system for short answers for e-learning environments is a challenging task due to the subjectivity of the questions, linguistic variations, and topical variations in the answers. Until recently ASAG models have not provided human-like performance in scoring answers (Jayashankar & Sridaran, 2017; Schneider et al., 2023; Shermis, 2015).

The short answer scoring problem is considered in two ways: response-based and reference-based (Sakaguchi et al., 2015). In the response-based approach, a scoring function is learned from human-scored responses. The scoring function uses features extracted from the student response. The reference-based approach compares a student response and a reference teacher response using various text similarity methods. We deal with the problem using the reference-based approach, which consists of comparing the student's answer to the teacher's reference answer and determining how similar they are. A supervised regression model is trained to predict a grade using text similarities, term weighting, and

length ratio features. We suggest generating several reference answers to consider the coverage and diversity of the reference answers in order to improve the grading model.

The adoption of ASAG systems remains low in e-learning environments in practice despite the panoply of existing theoretical work (Adams et al., 2016). Typically, in Learning Management Systems, software for marking short answer questions uses regular expressions, templates, and logic expressions to determine matches between the student answer and the reference answer. Regular expressions can provide good scores, but developing them manually is time-consuming (Sychev et al., 2020). For example, Moodle, the widely used Learning Management System (LMS), provides the “Regular Expression Short Answer question-type” (Moodle, 2011) to code correct answers as regular expressions. It imposes to students many constraints on the formulation of their answers. For tutors, the challenge is twofold. First, it concerns the manual construction of grammar templates regular expression specifying the teacher’s reference answer. The second concerns students’ compliance with template constraints. Students are penalized for additional space or a spelling error. Open Mark’s PMatch system (Jordan & Butcher, 2013) developed at the Open University is considered the more developed ASAG system in e-learning environments. It is based on the matching of keywords and their synonyms and can score very short answers of up to one sentence in length. All required words, word stems, and synonyms for correct answers are matched by regular expressions against the reference answers using a word-level pattern matching. Good scoring accuracy is obtained, but questions of this type remain underused in LMS platforms. The need to collect and label several hundred-student responses to train each question and the time required to develop a matching of answers is still a critical challenge.

In a general context, the ASAG predicts similarity scores comparing to a provided teacher reference answer. The reference answer represents the most relevant formulation of the answer. So, whenever student answers have a high common text overlap with the reference answer, they are likely to get a higher score (Ramachandran et al., 2015). Since it relies on semantic similarity in the meaning, grading short answers automatically is problematic because short contexts rarely share many words in common (Ab Aziz et al., 2009). Although a typical reference answer is considered ideal (Kumar et al., 2017), it does not represent all possible correct answers. It does not include all alternative ways of expressing the correct answer. To construct a response, students combine synonyms, paraphrases, and different sentence structures. With one reference answer, some student answers may be accurate due to little or no similarity to the provided reference answer. Having multiple alternative reference answers for the same question may handle the diversity of student responses and improve then accuracy. Several formulations of the reference answer may be possible but difficult to generate manually by the teacher.

Data augmentation to improve short answer grading has received little attention in ASAG systems. Handcrafting to produce alternative reference answers can be time-consuming and inefficient because it takes skill and human effort to create paraphrases that capture the original reference answer’s meaning (Marvaniya et al., 2018). It might not be a scalable or trustworthy method for scoring short answers at scale.”

Therefore, there is a need to automate the generation of alternative reference answers. ASAG systems require precise notation and a thorough understanding of response text, making them difficult to implement in real-world settings. When the system fails to achieve reasonable performance, trust issues arise, and errors in automatic grading can have a significant impact on individuals (Azad et al., 2020; Hsu et al., 2021; Schneider et al., 2023). The issue of trust in the tool will remain a major challenge until ASAGs can provide trustworthy human performance. Because of their statistical nature, reliable ASAG systems are critical for trust and practical utility (Schneider et al., 2023).

We consider two issues in this paper. First, we need to generate automatically various reference answers that can handle the diversity of student answers. Second, we should provide an accurate grading model that improves sentence similarity computation using multiple reference answers. Therefore, our proposed approach to solve both problems highlights two components. First, we provide a sequence-to-sequence deep learning model that targets generating plausible paraphrased reference answers conditioned on the provided reference answer. Secondly, we propose a supervised grading model based on sentence embedding features. The grading model provides and enriches features to improve the score accuracy considering multiple reference answers. The diversity of a student's answer is managed by comparing it to all of the reference answers. The ASAG will return the grade corresponding to the pair with the highest score (reference answer, student answer).

The novelty of our work lies in the utilization of the paraphrase generation Natural Language Processing (NLP) task to improve the scoring accuracy. Paraphrases have the same meaning as the original sentences but use different formulations. The paraphrased answers will increase the vocabulary of the reference answer and its writing style, which can manage the diversity of student answers. The proposed approach can help improve the coverage and diversity of reference answers. We aim to answer this research question: Can paraphrase generation improve the ASAG system in the e-learning environment and make it feasible in practice and at scale?

Two main contributions are highlighted in this paper: The first contribution is the use of the paraphrase generation natural language processing task to improve the ASAG system. To the best of our knowledge, this is the first work that uses paraphrase generation for improved short answer scoring. The second concerns the task of generating paraphrases in the Arabic language. Limited papers have dealt with the task in Arabic. Our paper may respond to this challenging task in Arabic and establishes baselines.

The rest of the paper is structured as follows: in Sect. "[Related Work](#)", we discuss related work. Sect. "[Method and Data](#)" details our method and data involving the problem formulation (Sect. "[Problem formulation](#)"), the proposed model (Sect. "[Proposed Paraphrase Generation Model](#)"), and its intrinsic evaluation according to the paraphrase generation task (Sect. "[Bi-Lstm Baseline system](#)"). In Sect. "[Automatic Short Answer Grading \(ASAG\)](#)", we present the proposed grading model and detail the integration of paraphrase generation into the ASAG tool. In Sect. "[Evaluation and results](#)", we evaluate the overall system and discuss results and implications. We conclude in Sect. "[Conclusion and Future Work](#)".

Related Work

ASAG Approaches

Data augmentation using multiple reference answers to improve short-answer scoring has not been widely explored. Hand-crafting techniques have already been used (Kumaran & Sankar, 2015; Leacock & Chodorow, 2003; Noorbehbahani & Kardan, 2011; Sukkarieh & Blackmore, 2009) to generate alternative reference answers. Mohler and Mihalcea (2009) used the pseudo-relevance feedback method from information retrieval (Rocchio, 1971) to generate alternative reference answers automatically. They augmented the reference answer with student answers that received the best scores according to the similarity measure. Although the quality of the system improved a bit, the method required a time-consuming training process. The Alternative Sentence Generator Method (Omran & Ab Aziz, 2013) is based on a database of synonyms used to tag the synonym for each word in the reference answer with its synonyms, to cover all answers. Although this method can generate many sentences, it is not suitable for under-resourced languages such as the Arabic language where knowledge-based linguistic resources (ontologies, dictionaries, thesaurus, lexicons...) are not necessarily available or are not rich enough. Moreover, it did not allow new formulations of the reference answer. Dzikovska et al. (2014) used generalizable lexical representations of the reference answer and rules to cover semantic information. The graph-based cohesion technique (Ramachandran & Foltz, 2015) was used to generate alternative reference texts by summarizing the content of top-scoring student responses. The extracted responses were used as alternative reference answers. The method suggested the need for a summarization technique. The summarizers produced sentences that are more representative and useful for scoring than the sample reference answers. The use of Rocchio's method (Mohler & Mihalcea, 2009) was revised by Pribadi et al., (2018) to obtain alternative reference answers without the training process. They used the Maximum Marginal Relevance (MMR) method, proposed in (Carbonell & Goldstein, 1998). They proposed a GAN-LCS (Geometric Average Normalized-Longest Common Subsequence) unsupervised similarity method that removed non-contributive words in the reference answer. The results discussed by the authors did not suggest a significant improvement in the correlation against the gold standard on the English Mohler dataset (Mohler et al., 2011) for which much more accurate results are available. Koleva et al. (2014) developed a system for grading German foreign language learners' reading comprehension tests using paraphrase detection. They used word alignment of paraphrase fragments extracted from parallel corpora to determine semantic entailment relationships between student and reference answers. Extracted features from paraphrase fragments are used for linear regression learner, considering the strength of the semantic connection. False answers share no paraphrase fragment, while strongly linked fragments involve correct answers. On the German CREG corpus (Ott et al., 2012), the system reached an accuracy of 86.8%. The system did not use point grading. The evaluation, by true or false, does not allow for the detection of potential deep differences between

manual and automated scores. Although not using multiple alternative reference answer generation, recent works hold the best scoring accuracies on the Mohler Dataset widely used in English ASAG evaluation. Sultan et al. (2016) proposed a short answer grading system involving semantic similarity, text alignment, question demoting, term weighting, and length ratios. Kumar et al. (2017) used a Siamese bidirectional LSTMs applied to a reference and a student answer, based on earth-mover distance across all hidden states from both LSTMs and a final regression layer to output grades. Hand-crafted features and sentence embedding features were combined by Saha et al. (2018) to improve the accuracy. An end-to-end deep neural network was trained to learn embeddings and a neural network trained a grading classifier. This is challenging, as training embedding requires large, unlabeled data. Gomaa and Fahmy (2020) used a Skip-thought vector approach to convert reference and student answers into embeddings to measure their similarity.

More recent methods for ASAG explored sophisticated feature representations, computed by attention-based and transformer models (Vaswani et al., 2017), to better capture structural and semantic features. These models showed state-of-the-art results on various NLP tasks such as question answering, sentiment analysis, text summarization and more. Schneider et al., (2023) used a large dataset of 10 million question-answer pairs in various fields categorized in two classes correct and incorrect. They demonstrate the effectiveness of fine-tuning the BERT (Devlin et al., 2019) transformer model for automatic scoring. They achieved an accuracy of 86% by adjusting hyper-parameters for complex datasets. The study focused on trust and ethics while involving humans in automatic classification and improving score accuracy. According to the experimental analysis the authors conducted on a sample of difficult questions graded by a human, BERT performance fluctuates more when applied to small datasets. With a point grading system, it would be possible to detect potentially significant differences from expected accuracy that are not currently possible with the true/false evaluation used in their dataset.

(Gaddipati et al., 2020) used pre-trained embeddings of the transfer learning models ELMo (Peters et al., 2018), BERT (Devlin et al., 2019), GPT (Radford et al., 2018), and GPT-2 (Radford et al., 2019) to evaluate their effect on the ASAG task. Regression with cosine similarity features was used to compare sentence embeddings of the reference and student answers from these models using the Mohler's dataset (Mohler et al., 2011). Authors observe that ELMo model outperformed on domain-specific ASAG compared to other transfer learning models but the Sultan system (Sultan et al., 2016) achieved better on the Mohler dataset on all the pre-trained models. The pre-trained data of ELMo, BERT, GPT and GPT-2 models are extensive and the domain of the tested ASAG is comparatively very specific and smaller. These findings highlighted that a pre-trained model like BERT might not perform as well on a very diverse dataset as on a very domain-specific one (Schneider et al., 2023). These models for ASAG remain difficult in practice because the ASAG's dataset is typically small and cannot provide enough training or fine-tuning data. To capture more domain-specific features using transformers, Agarwal et al. (2022) applied short text matching using Multi-Relational Graph Transformer representation to incorporate relation-enriched structural information. They include the semantic representation of a relationship in the preparation of token embeddings,

which improved the model's overall performance and achieved the state of the art on the Mohler Dataset. Transformer-based language models, like BERT, are powerful but they are computationally prohibitive (Huang et al., 2022). When used on a large scale, as is the case in e-learning environments, they present challenges for practitioners. Because saving and loading model parameters require more memory and processing power.

Paraphrase Generation

Our study concerns the application of paraphrase generation in the ASAG task. Generating high-quality paraphrases is challenging in natural language processing tasks. It aims to synthesize the paraphrases of a given sentence automatically. Paraphrase generation techniques can be classified into two main categories: controlled paraphrase generation methods and deep learning methods. Controlled methods exploit handwritten rules and thesaurus-based alignments or use Statistical Machine Translation (SMT-based) techniques that consider paraphrase generation as a special case of monolingual machine translation (Wubben et al., 2010). These methods are controlled by templates or syntactic trees providing extra information such as phrasal and lexical dictionary (Huang et al., 2019), keywords (Zeng et al., 2019), sentential exemplar (Chen et al., 2020), syntactic trees and tree encoder (Kumar et al., 2020), retriever-editor (Kazemnejad et al., 2020), syntactic transformations (Goyal & Durrett, 2020), retrieval target syntax selection (Sun et al., 2021). Inspired by the success of deep learning networks, paraphrase systems use existing parallel corpora to train sequence-to-sequence models to get better performance. Prakash et al. (2016) used a technique that vertically stacks multiple Long Short-Term Memory (LSTM) layers with residual connections to achieve better performance from very deep neural networks. A deep generative framework VAE-SVG-eq (Variational Auto-Encoder for Sentence Variant Generation) was proposed by Gupta et al. (2018). The long short-term memory (LSTM) (Hochreiter & Schmidhuber, 1997) and the Variational Auto-Encoder (Kingma & Welling, 2013) were combined to develop an end-to-end deep learning paraphrase generator. The system achieved the best results of the state of the art in paraphrase generation. The Generative Adversarial Paraphrase model (GAP) (Yang et al., 2020) is an end-to-end conditional generative architecture for generating paraphrases via adversarial training. It does not depend on extra-linguistic information. More recently, large language models using transformer architectures and less supervised data were proposed for many NLP tasks with implementation codes and datasets made publicly available (on repositories such as Hugging Face, GitHub). Pre-trained language models such as GPT-2¹ (Radford et al., 2020) and BART² (Lewis et al., 2020) are utilized as encoder-decoder frameworks. The "Text-to-Text Transformer" T5³ (Raffel et al., 2020)

¹ <https://huggingface.co/gpt2>

² https://huggingface.co/docs/transformers/model_doc/bart

³ <https://github.com/google-research/text-to-text-transfer-transformer>

is a unified framework that converts all text-based language problems into a text-to-text format. Therefore, some researchers utilized these model frameworks and adapted the code to different NLP tasks including paraphrasing. Considering the Arabic language, few works have dealt with the paraphrase generation task. An Arabic metaphor paraphrasing approach (Alkhatib & Shaalan, 2018), used neural machine translation to paraphrase Arabic metaphors. The metaphor is first translated into a pivot language and translated then to English using a bilingual corpus. Al-Raisi et al. (2018a) trained a bidirectional LSTM neural network to generate paraphrases using their Arabic dataset (Al-Raisi et al., 2018b). Their system was evaluated by computing cosine similarity. In comparison to standard automatic metrics for paraphrase generation, no results were provided. As the authors concluded, *"the neural model has learned interesting linguistic constructs like phrases used for sentence opening but the output is still far from practical applicability"*. To fill this gap and improve the quality of paraphrase generation in Arabic, we explore sequence-to-sequence deep learning models. The generated paraphrases are used as alternative reference texts for ASAG improvement.

Method and Data

Problem Formulation

In paraphrase generation, we are interested in models that take in a source sentence (S) containing the words ($s_1, s_2, s_3 \dots s_n$) and generate an output sentence (G) with the same meaning but a different surface containing the words ($g_1, g_2, g_3 \dots g_m$). Formally, given an input sentence (source sentence) S where $S = \{s_1, s_2, s_3 \dots s_n\}$, the aim is to generate one or more sentences $G = \{g_1, g_2, g_3 \dots g_m\}$ where the sentence length of the generated sentence and the input sentence may vary.

Our goal is to find the sentence G such that the conditional probability $p(G|S)$ is maximized. We model $p(G|S)$ as a product of word predictions (formula (1)):

$$p(G|S) = \prod_1^M p(g_t | g_{1:t-1}, S) \quad (1)$$

This indicates that the probability of generating each current word (g_t) relies on the previously generated words ($g_{1:t-1} = g_1, g_2 \dots g_{t-1}$) and the source sentence S. This many-to-many sequence prediction problem predicts a sequence of words $\{g_1, g_2, g_3 \dots g_m\}$ from a sequence of words ($s_1, s_2, s_3 \dots s_n$). We note here that the term "paraphrase" means sentences that convey the same meaning. We take a broader perspective that considers using different words in terms of synonym substitution, lexical changes, grammar changes, verb/noun conversion, and semantic implications to be acceptable paraphrases. We do not consider fine-grained linguistic distinctions of meaning between sentences.

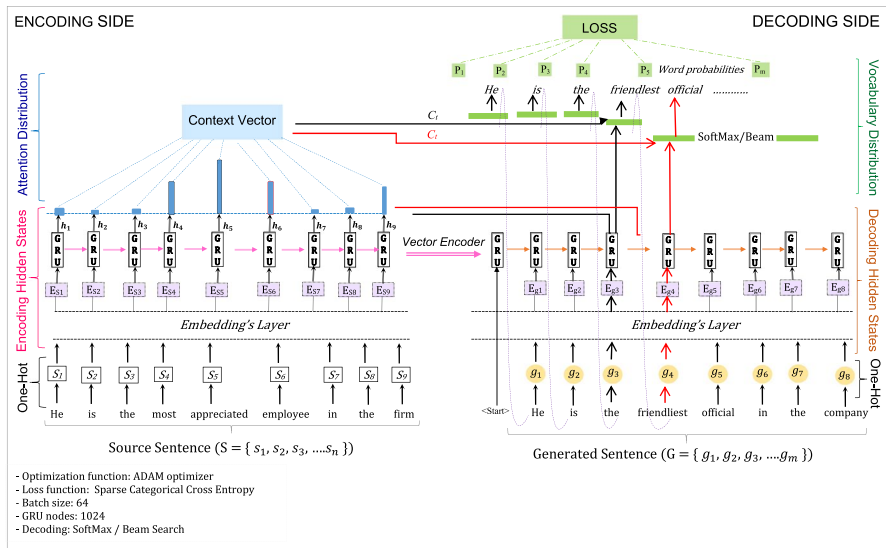


Fig. 1 Proposed Alternative Sentence Generator Encoder-Decoder attentional model, which consists of sentence encoder (Encoding side) and sentence decoder (Decoding side). The model uses relevant words in the source sentence to generate novel words. For example, to produce the word "friendliest" in the paraphrase "He is the friendliest official in the company", the model may attend to the words "most" and "appreciated" in the source sentence ("He is the most appreciated employee in the firm") that have high attention weights in the vector context. The Synonyms "official" and "company" are generated in influence with a high attention weight of "employee" and "firm" in the source sentence

Proposed Paraphrase Generation Model

For modeling the conditional probability $p(G|S)$, we propose ARAG-ED namely (Alternative Reference Answer Generator Encoder-Decoder), an Encoder-decoder which targets generating plausible alternative reference answers conditioned on the provided reference answer. The Encoder-Decoder (Cho et al., 2014) is a recurrent neural model that generates an output sequence from an input sequence. ARAG-ED is made up of two parts, the encoder, and the decoder. Each part uses deep neural networks (Gated Recurrent Unit (GRU) (Chung et al., 2014)) to handle variable-length sequence inputs. The main advantages of this model are the ability to train a single end-to-end model (directly on the source and target sentences) and to handle variable-length input and output text sequences. As illustrated in Fig. 1, two steps are involved in generating paraphrases: encoding and decoding. The encoder is used to encode the input sentence into an encoder vector corresponding to the last hidden state. Decoding uses the generated encoder vector as the initial hidden state of the decoder to guide the decoder in predicting words one after another in the generated sentence. We use the attention mechanism to assist the model in focusing on specific parts of the input sentence to further improve it. Transfer learning with a unified Transformer framework such as T5 (Raffel et al., 2020) was originally pre-trained for English. It may be an attractive transfer learning approach, but we chose to build a model of encoder-decoder from scratch for two reasons: First, although

a multilingual version of the T5 model namely mT5 (Xue et al., 2021) is available, it is not clear how well it can fare on non-English tasks in the absence of comparisons with pre-trained monolingual linguistic models that serve different non-English contexts. Moreover, (Kreutzer et al., 2022) raised systematic problems in multilingual corpora on which language models have been trained. Second, as we investigate paraphrase generation to improve the ASAG accuracy with an interest in the Arabic language, our goal for our model was to reflect typical practices to transfer learning in paraphrase generation. We avoided investigating in parallel mT5 on the generation of Arabic paraphrases and studying the impact of paraphrases on the ASAG accuracy. Transformers are prone to over fitting, particularly when trained on smaller datasets (Xu et al., 2021). Indeed, the self-attention mechanism requires a lot of computation, making it more resource-intensive than standard encoder-decoder models (Huang et al., 2022). It requires a significant amount of data to train effectively. This may pose a challenge in Arabic where data is scarce and may limit their applicability in under-resourced environments. We trained an attentional standard encoder-decoder and explored the performance on Arabic and English datasets.

Sentence Encoder The encoder consists of an embedding layer that represents words and a gating layer that embeds the input information.

Embedding layer. The objective is to construct embeddings from One-hot word representations. One-hot vector is a $1 \times N$ vector that distinguishes N words in the vocabulary. The vector consists of 0 s except for a single 1 in a cell used to index the word in the vocabulary. Once the dataset is coded, the « One-hot » vector for each word can be generated relative to its position in the vocabulary (of size N). As an input to the embedding layer, each word of the input reference text is introduced by its One-hot vector to the neural network to generate a reduced representation (the embedding) while keeping the semantic links between the words in the text. We retain 256, the dimension of the vector space in which the words will be embedded. In Fig. 1, the words of the source sentence $S = \{s_1, s_2, s_3 \dots s_n\}$ are represented by One-hot vectors which are coded into embeddings $\{E_{s_1}, E_{s_2} \dots E_{s_n}\}$. The generation of embeddings in the model involves the generation of similar word distribution which may correspond to synonyms also.

Gated Recurrent Unit Layer (GRU). Gated Recurrent Neural Networks (GRUs) (Chung et al., 2014) offer a solution to the gradient vanishing problem. GRU is a simplified version of LSTM with only two gates: a reset gate and an update gate for resetting and updating the cell hidden state. LSTM uses three gates (forget gate, input gate, and output gate). GRU is less computationally expensive than LSTM and requires fewer parameters to train. GRUs can outperform LSTM networks on low-complexity sequences (Cahuantzi et al., 2021; Chung et al., 2014). We chose GRU cells because they showed better results both for the convergence time and for the iterative efficiency. The number of GRU nodes used in the model is fixed at 1024. Hyper-parameters are defined as:

- *Optimization function.* We use the ADAM (Adaptive Moment Estimation) (Kingma & Ba, 2015) stochastic optimizer.
- *Loss function.* The "sparse Categorical Cross Entropy" function is used (formula (2)) to calculate the loss during the training process:

$$-\frac{1}{N} \sum_{i=1}^n \sum_{c=1}^C 1_{y_i \in C_c} \log P_{model}[y_i \in C_c] \quad (2)$$

Where n , C , and P are respectively the number of observations, the number of classes corresponding to the number of different words in the vocabulary dataset used in the One-hot representation, and the probability of the observation " i " relative to the class " c ".

- *Batch size.* The model trains progressively on dataset batches of the same size (64 pairs of sentences).

Attention mechanism. The performance of the encoder-decoder network degrades rapidly as the length of the input sentence increases (Cho et al., 2014). The problem resides at the decoder level where only the last hidden state generated by the encoder is used as a context vector. In the case of long input sequences, the encoder is unable to retain, until the final hidden state, all the information useful for generating the output. To overcome this lack, we integrate into the encoder an attention mechanism (Bahdanau et al., 2015). The attention mechanism considers all the information contained in the hidden states at different time steps. It uses all the hidden states of the encoder to generate a context vector at each time step. It calculates alignment scores between the previous hidden state of the decoder and all hidden states of the encoder. To generate a word g_i , attention is paid to each word in the input sequence. Attention is expressed by weights at the encoder that generates a score for each hidden state so that the hidden states for which attention should be given will have a high score. Attention weights (P_i) are produced by the SoftMax function and applied to the scores generated by the encoder. After having calculated all the attention weights, a context vector is calculated according to formula (3):

$$ContextVector = \sum_{i=1}^n P_i h_i \quad (3)$$

h_i Hidden state at time-step i ,

n represents the number of words in the source sentence, and

P_i Weight of hidden state h_i

Each node (GRU_i) of the decoder, except the first, has as input, the output of the previous node (g_{i-1}), and the context vector generated by the attention mechanism

(C_t) at time step t . The first node of the decoder (GRU_1) receives as input the last hidden state of the encoder with the first context vector (C_1) generated by the attention mechanism. As illustrated in Fig. 1, the model considers relevant words in the source sentence to generate novel words. For example, to produce the word "friendliest" (in the generated paraphrase), the model may attend to the words "most" and "appreciated" (in the source sentence) that have high attention weights in the context vector. The Attention mechanism enables the decoder to focus on some words that are of high relevance when generating a word ("most", "appreciated", "employee", "firm"). Weight for each word in the source sequence in each time step is computed to indicate the importance, emphasizing the essential information from the source sentence ("most", "appreciated", "employee", "firm") and de-emphasizing the unimportant information ("is", "The"). In the example (Fig. 1), the synonyms "official" and "company" are generated in influence with a high attention weight of "employee" and "firm" already encoded in the source sentence.

Sentence Decoder (Generator) On the decoding side, the contextualized representation is used at each decoding step with the vector embedding of previously generated words. A distribution over the vocabulary is obtained and the word with the highest probability is generated. The decoder is made of three layers, an embedding layer, a GRU layer, and an output layer:

Embedding layer. The decoder has an embedding layer that generates the corresponding embedding vector from the digital representation of each word (g_i) of the paraphrase.

The GRU layer. The decoder takes as its initial input the last hidden state generated by the encoder; the vector encoder. This hidden state contains the essential information contained in each word of the input sentence. To generate a word g_i at the time step t , the decoder has as input the hidden state, the output generated at the previous time step, and the embedding of the precedent-generated word of the paraphrase.

The output layer. The results of the GRU layer pass through the SoftMax function that acts as a classifier in the output layer of the decoder to predict a multinomial probability distribution over integers representing vocabulary words. For example, to generate the word "official", the decoder has as input the embedding of the last generated word "friendliest", the context vector C_t , and the last GRU hidden state. When calculating the distribution vocabulary (SoftMax), the C_t attention context vector will give a significant probability value to the word "official" which is a synonym of "employee". The word generated at the output is the one with the highest probability value. Therefore, we will generate only one paraphrase. While this approach is often effective, it is non-optimal. Indeed, we use the Beam Search Decoding Algorithm (Vijayakumar et al., 2018) as an approximate search that gives multiple paraphrases. The beam search algorithm expands all possible next steps, keeps the output sequences, and controls the number of beams (parallel searches) through the sequence of probabilities. Several paraphrases are generated according to the size of the beam.

Table 1 Pre-trained used word embedding

Word embedding	Model	Dimension	Words
English CoNLL17 corpus	Word2Vec Skip-gram	100	4,027,169
Arabic CoNLL17 corpus	Word2Vec Skip-gram	100	1,071,056

Bi-Lstm Baseline System

There are no existing public results on Arabic paraphrasing generation, we implement the Bi-LSTM neural network as a baseline for comparison. Bi-LSTM neural networks operate on sequential data using two LSTM sub-layers; one layer running forward and another layer running backward. The two hidden states of the two layers combined preserve information from the past and the future. Bi-LSTM can learn long-term dependencies without keeping duplicate context information. We use 256 LSTM nodes at each sub-layer. The model considers all the words of the input sentence for the prediction of an output word in both directions. It has three layers: the embedding layer which transforms the input words into embeddings, the Bi-LSTM layer for the generation of words, and the classification layer (SoftMax) for the definition of the output words. As illustrated in Table 1, in the Embedding layer, we used Word2Vec Continuous Skip-gram pre-trained word vectors from the (NLPL word embeddings repository).⁴ We used two pre-trained models of word embeddings, namely the English corpus CoNLL17 and the Arabic corpus CoNLL17. These Embeddings are of dimension 100 and contain the mapping of 4,027,169 words (for English) and 1,071,056 words (for Arabic) to capture the semantic and syntactic properties of the language. We use the ADAM optimizer with the Categorical Cross-Entropy loss function with a learning rate of 0,001. The model trains progressively on fixed batches of 16 pair size. The nodes are dropped by a dropout probability of 0.3 (30%). This regularizes the model to get the average predictions of all parameter settings and aggregate the final output. This ensures the model is generalized and hence reduces the overfitting problem. At inference time, all the units are considered. Therefore, the final weights will be larger than expected, and to deal with this problem, weights are first scaled by the dropout rate. The network is then able to make accurate predictions. So, if a unit is retained with the dropout probability during training, the outgoing weights of that unit are multiplied by the same dropout probability during the inference time. So, if a unit is retained during the training, the outgoing weights of that unit are multiplied by the same dropout probability during the inference time.

Evaluation of Generated Alternative Reference Answers

In this section, we perform an intrinsic evaluation of the paraphrase generator in relation to the paraphrase generation task. The effectiveness of the proposed approach is thus, verified empirically across a real educational environment. Experiments are conducted

⁴ <http://vectors.nlpl.eu/repository/>

Table 2 Dataset portions that are used to train and test the paraphrase generator

Dataset	Training (60%)	Validation (20%)	Test (20%)	Total (100%)
Quora dataset (English)	210,00	70,000	70,000	350,00
Al-Raisi Dataset (Arabic)	46,423	15,474	15,474	77,371

both in Arabic and English. We found it appropriate to use English as a reference language. This will allow us to compare our results with existing works and validate our model independently of the language. In the following, we describe the used datasets and evaluation results.

DATA

*Al-Raisi Arabic Dataset*⁵ (Al-Raisi et al., 2018b). It is the first parallel monolingual corpus of full sentences in Arabic. It contains 100,000 pairs of (original sentence, reference paraphrase). This dataset is the largest parallel Arabic corpus publicly available. It was generated automatically by translating a parallel bilingual corpus (English French) using Google translate APIs (English-Arabic, and French-Arabic).

The Quora English Dataset.⁶ It is one of the most widely used datasets in English paraphrasing. It consists of over 400,000 pairs of sentences (original sentence, reference paraphrase). Each pair is annotated with a binary value indicating whether the two sentences are paraphrases of each other. To train the generator, only sentence pairs that are paraphrases of each other are selected. Since we are interested in short answers, we have eliminated pairs of sentences that are too long, in the datasets, by setting the maximum size to 50 words. Therefore, we picked 350,000 pairs from the Quora Dataset and 77,371 pairs from the AL-Raisi dataset. We used 60% of each dataset for training, 20% for validation and 20% for testing as shown in Table 2. Dataset pre-processing transforms the raw data into a format more suitable and usable by the model: cleaning, elimination of long sentences, normalization, and conversion to lowercase (for English), and tokenization. To use a one-hot representation of the words, the dataset is prepared to extract the vocabulary composed of all the different words in the dataset. A sequential integer is assigned to each word of the extracted vocabulary. The result is that each input sentence is coded by a set of numbers corresponding to word positions in the extracted vocabulary. The training was performed on the cloud service Google Collaboratory (Carneiro et al., 2018) with Nvidia Tesla K80 GPU/12 GB of DDR5 memory provided.

Automatic Evaluation of Generated Paraphrases

Metrics. Viewing paraphrase generation of reference answer as a text-to-text machine translation, we could use classical machine translation metrics, such as

⁵ <http://www.cs.cmu.edu/~fraisi/arabic/arparallel/>

⁶ <https://github.com/jakartaresearch/quora-question-pairs>

Table 3 Examples of generated paraphrases using ARAG-ED

Arabic paraphrases	
<i>Original sentence</i>	ونحن لا نرغب في القيام بذلك وهكذا سنناقش مره اخرى هذه المسألة
<i>Reference sentence</i>	We do not wish to do so and so we will once again discuss this issue سنناقش مره اخرى هذه المسألة لأننا لا نرغب في القيام بذلك
<i>Sentence generated by ARAG-ED (Arabic)</i>	We will discuss this issue again because we do not wish to do so نحن لا نتوي القيام بذلك ونحن مره اخرى سنتحدث عن ذلك
	We do not intend to do that and we will talk about it again.
English paraphrases	
<i>Original sentence</i>	why do some people ask questions on Quora that could be asked directly to a search engine
<i>Reference sentence</i>	why do few people post questions on Quora check Google first
<i>Sentence generated by ARAG-ED (English)</i>	why do people ask questions on Quora that could simply be googled

BLEU (Papineni et al., 2002), GLEU (Napoles et al., 2015), and METEOR (Lavie & Agarwal, 2007) to measure the quality of the generated answers. A common finding in the machine translation community is that automatic metrics correlate well with human judgments at the system level (Wubben et al., 2010). This means that the correlation analysis between automatic evaluation metrics and human scores is consistent for the whole translation system. BLEU is quick and language-independent and assesses a generated paraphrase by measuring the fraction of n-grams that appear in a set of reference paraphrases (ground-truth paraphrases) that captures two aspects of translation: adequacy and fluency. GLEU (Google-BLEU) is a variant of the BLEU metric, used specifically to measure the grammatical error correction rate of n-grams generated with all the reference sentences. METEOR is based on unigram precision and recall. It significantly improves the correlation with human judgments. METEOR computes the similarity score of two texts by using a combination of unigram precision, unigram recall, and some additional measures like stemming and synonymy matching.

Results. An example of generated paraphrases in Arabic and English using ARAG-ED is presented in Table 3. Note that the training dataset is made up of pairs (Source sentence, Reference sentence). The generated sentence is the output produced by the trained model. We performed an automatic evaluation of the three models (Bi-LSTM Baseline, ARAG-ED without attention mechanism, and ARAG-ED with Attention Mechanism) using the (BLEU, GLEU, and METEOR) metrics. These metrics always take a value between 0 and 1. This value indicates how similar the predicted text is to the reference texts, with values closer to 1 representing more similar texts. Note that the Meteor metric is not calculated for the Arabic dataset since the calculation depends on the WordNet knowledge-based model. The Arabic version of WorldNet is not sufficiently rich not to distort the results. Evaluation results are reported in Table 4. For the Bi-LSTM model, the obtained results are very weak for all the metrics on the two datasets. The quality of the pre-trained Embeddings negatively influenced the results. During the training, we noticed that several words had no representation in Word Embedding. They are replaced with zeros. This is predictable. We consider it as a baseline to appreciate the improvements in the

Table 4 Proposed model evaluation for paraphrase generation task

	El-Raisi Arabic Dataset		Quora English Dataset		
	BLEU	GLEU	BLEU	GLEU	METEOR
Bi-LSTM (Baseline)	12	4	17	8	8
ARAG-ED (Without Attention Mechanism)	15	8	19	11	18
ARAG-ED (Beam = 1)	63	59	54	42	42

Table 5 Proposed ARAG-ED-beam search decoding evaluation

BEAM	El Raisi Arabic Dataset		Quora Dataset			
	Avg-BLEU	Best- BLEU	Avg-BLEU	Best- BLEU	Avg-METEOR	Best-METEOR
1	63	63	54	54	42	42
4	59	—	—	—	—	—
7	55	—	—	—	—	—
10	53	54	43	49	24	28

proposed model. An improvement is noted for the ARAG-ED model. Note here that the model constructs the embeddings in the embedding layer. The attention mechanism has significantly improved the results. For the Arabic dataset (BLEU=63, GLEU=59) and the English dataset (BLEU=54, GLEU=42, METEOR=42).

Generating multiple paraphrases with the beam search algorithm. In the following, we consider the ARAG-ED model with an attention mechanism. In Table 5, we report the obtained results using the beam search-decoding algorithm to generate multiple paraphrases. Trained on the Arabic Dataset, the model generates 1, 4, 7, and 10 for each source sentence by varying the beam size (1 & 10 for the English dataset). Avg-BLEU and Best-BLEU are calculated. They refer to the average and the best BLEU for the multiple-generated paraphrases. AVG-BLEU corresponds to the average of the averages of the BLEU scores calculated for the pairs (sentence generated, reference paraphrase). Best-BLEU corresponds to the best BLEU score of pairs (generated sentence, reference sentence) for each beam. As we can see, the BLEU score decreases each time the number of generated paraphrases increases (63% for Beam=1 up to 53% for Beam=10). This is predictable in the sense that the average includes the best case and the worst case. With Beam=1, the SoftMax function predicts the highest probability for the single generated paraphrase. By relying on the interpretation of the BLEU scores (Lavie, 2010), we can say that the generated paraphrases are of good quality for Arabic and English (For Beam=10: BLEU=53% for Arabic paraphrases and BLEU=49% for English paraphrases). The interpretation states that "Scores over 30 generally reflect understandable translations – Scores over 50 generally reflect good and fluent translations" (Lavie, 2010). Therefore, the most relevant aspect of our approach concerns implementing an attention mechanism with generated embeddings that increased the model's performance without a need for higher computational complexity. This is important since

Table 6 ARAG-ED vs. State-of-the-art on the Quora dataset-comparative analysis

	Beam = 1		Beam = 10			
			Average		Best	
	BLEU	METEOR	BLEU	METEOR	BLEU	METEOR
VAE-SVG-eq Gupta et al., (2018)	26,2	25,7	37,1	32	38	32,9
GAP Yang et al., (2020)	N/A	N/A	N/A	N/A	47,6	30,94
ARAG-ED	54	42	43	24	49	28

the paraphrase generator is integrated into the online assessment system to improve the ASAG accuracy. Factors such as server performance and practical efficiency must be considered when automatic scoring is applied to e-learning environments.

Comparison with previous work on the Quora dataset. In the literature review of paraphrase generation, English-related results are often the only ones that are available. In Table 6, the BLEU and METEOR obtained scores are compared with those obtained by recent related works using encoder-decoder variants trained on the same dataset we used. We compare with the work of VAE-SVG-eq (Gupta et al., 2018) and the GAP (Yang et al., 2020)) that used Quora Dataset for which results are available. For Beam=1, the GAP model does not report any evaluation. We discuss the results for Beam=10. Considering the BLEU metric, we notice that our model achieves the best scores (Avg-BLEU: +5,9%; Best-BLEU: +11%) compared to VAE-SVG-eq and (Best-BLEU: +1,4%) compared to GAP. This means that the decoder (generator) correctly reproduces the n-gram alignments of the reference sentence in the generated sentence. Moreover, syntactically, the generated sentences are well-formed and of good quality. Using METEOR, our model achieves similar scores compared to the best model but less good (Avgas-METEOR: -8%; Best-METEOR: -4.9%) compared to VAE-SVG-eq and (Best-METEOR: -2.94%) compared to GAP. The lack here could be due to the quality of the embeddings generated in the first phase. More training epochs could result in better-quality embeddings and improve the METEOR score. The obtained scores are subjected to human manual evaluation, which is discussed in the next section.

Manual Evaluation of Generated Paraphrases

Although widely used in NLP tasks, recent studies (Chaganty et al., 2018; Lai et al., 2022) have shown that automatic metrics such as BLEU, GLEU, and METEOR are biased which means that the correlation varies across systems. Automatic evaluation metrics focus on the n-gram overlaps instead of meaning. As a result, automatic metrics are likely to score certain systems better than others, irrespective of their actual human evaluation scores. When dealing with Arabic, due to its rich and complex morphological features, Natural Language Processing has always been more challenging. Arabic has different word forms and word orders, which make it possible to express any sentence in different forms.

Table 7 Average of three human evaluations of generated reference texts

	Relevance /5	Readability /5
Quora dataset	4,82	4,94
Al Raisi Dataset	N/A	N/A
Sample-Quora-dataset-VAE-SVG-eq Gupta et al. (2018)	3,57	4,08
Sample-Al Raisi Dataset-ARAG-ED (ours)	3,52	3,88
Sample-Quora-Dataset-ARAG-ED (ours)	3,58	4,51

Tokens in Arabic are problematic due to the rich and complex morphology of Arabic. The token thus requires knowledge of the concatenation constraints of affixes and accompaniments in Arabic words. Although BLEU scoring is a widely used method, it does not consider these concatenation constraints in Arabic. BLEU computes its score according to the matching n-grams in the texts it is comparing. It does not take into consideration the proper use of grammar and seems to score more accurately in the evaluation of long sentences. In our case, there is a substantial set of generated paraphrases that are correct according to human judgment but are scored poorly by these metrics, which means that the metrics have a poor recall. As the usefulness of generated paraphrases would depend on the application at hand (multiple reference answers for ASAG improvement), a qualitative evaluation by human experts is essential. Human evaluation is naturally more costly because of the human effort involved compared to automatic evaluation, but more representative of the quality of the generated paraphrases along multiple dimensions such as relevance, and readability (Babych, 2014). Relevance expresses the adequacy of the generated paraphrase with the reference sentence. The aim is to examine to what extent the generated sentence retains the same meaning as the reference sentence. The readability expresses the comprehensibility of the generated paraphrases in terms of form and grammar. Therefore, evaluations based on human judgments are complementary to automatic evaluations using metrics. Based on the human evaluation reported by (Gupta et al., 2018), the authors indicated that the paraphrases in the Quora dataset are not 100% accurate (relevance = 4.82/5 & readability = 4.94/5). For the Arabic dataset, no information for a manual qualitative evaluation is available. With this in mind, we manually evaluated 100 pairs of randomly sampled generated paraphrases using ARAG-ED trained on the Al-Raisi Arabic dataset, namely (Sample-Al-Raisi-Dataset-ARAG-ED in Table 7). Similarly, we conducted a manual evaluation on 100 randomly sampled generated paraphrases using ARAG-ED trained on the Quora English dataset, namely (Sample-Quora-Dataset-ARAG-ED in Table 7). Three human experts were asked to assign a score between 1 and 5 for the relevance and the readability aspects of the sample of pairs (source, paraphrase). We have retained these two aspects because they are available for the complete Quora dataset and VAE-SVG-eq (Gupta et al., 2018) on a random sample. We do the same evaluation for English and Arabic paraphrases. The human experts found assigning manual evaluations a challenging task. They noted that the difficulty lay not in deciding

whether the generated paraphrase was readable and relevant, but in determining the precise degree of readability and relevance compared to the input answer. To understand how consistent the human experts were with one another, evaluations are run using Pearson's correlation coefficient (r : the higher the better) between the annotators. For all manually evaluated paraphrases, we calculated the Pearson correlation between each pair of annotators. The Pearson correlation between annotators demonstrates related subjectivity. The experts with the highest correlation achieved an agreement of more than 68%. Those with the lowest correlation achieved an agreement of more than 63%. This is reasonable since subjectivity is related with any evaluative act (Brown & Glasner, 1999). We finally considered the average of the three human manual scores as reported in Table 7 to consider a sufficient margin of error due to subjectivity. For the English paraphrases, the relevance is equal to 3,58/5 (71.6% of the meaning is preserved). This result is like that of the sample generated by VAE-SVG-eq on the Quora dataset (3,57/5). We can say that the semantics of paraphrases are well-preserved overall. However, the sentences generated by our model are more syntactically correct with average textual readability of (4,51/5) against (4,08/5) for the VAE-SVG-eq model. This confirms well the obtained results previously using automatic metrics. Although we have no results from related works for the Arabic dataset to compare with, the generated sample preserved quite good relevance and readability overall (relevance = 3,52/5; readability = 3,88/5).

The proposed ARAG-ED generates accurate paraphrases, according to the intrinsic experiments we conducted for the task of paraphrase generation. It generates different and accurate paraphrases from the source sentences. The remainder of the paper discusses the impact of using the proposed ARAG-ED on improving the accuracy of the ASAG system.

Automatic Short Answer Grading (ASAG)

We use the results of the paraphrase generation task to improve automatic short answer scoring. In this section, we describe our model for short answer grading and the integration of the paraphrase generator into the grading tool.

Proposed Grading Model

We propose a supervised regression general-question model that combines text similarities, word weighting, and answer length statistics features. Features are detailed in the next section. **Word Embeddings, trained on large general domain corpora, are used to learn general domain knowledge. Unlike training the model separately for each question, various questions are trained in the same model.** As shown in Fig. 2, the ASAG model requirements highlight two processes; the training process and the scoring process.

The training process is performed once and generates the trained grading model:

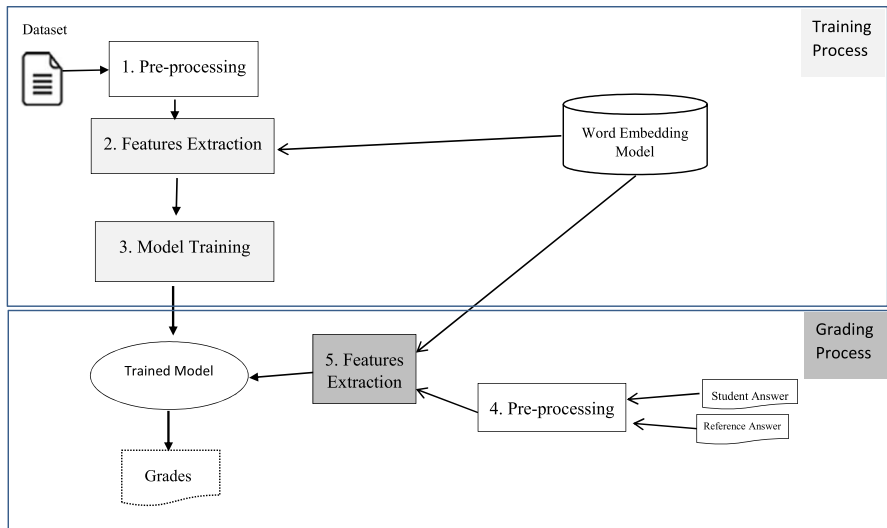


Fig. 2 Automatic short answer grading model overview

- (1) Pre-processing: normalize the dataset (Cleaning, Stop-word Removals, Tokenization, and Stemming).
- (2) Extraction of features that serve as inputs for the training of the model.
- (3) Training the model to fix feature weights in the trained model.

The scoring process ensures the automatic scoring of a student's answer by having as input the student's answer and the reference answer:

- (4) Pre-processing: Normalizes the answers using pre-processing techniques (Cleaning, Stop-word Removal, Tokenization, and Stemming).
- (5) Extraction of features related to the inputs: Features are introduced to the trained model to predict the score. The proposed model is trained on the gold standard human grades of the dataset with the Ridge Linear Regression (Ridge-LR) using the «Scikit-learn» library (Scikit-learn, 2019). Ridge Linear Regression (Ridge-LR) aims to regularize the complexity by introducing penalty factors to reduce fitting and handle multicollinearity. The model is trained with a second-degree "Polynomial Feature" method to fit a much higher range of data. Polynomial features are features created by raising existing features to an exponent (squaring for example). A matrix of the proposed features and all their polynomial transformations is created from the dataset to train the grading model. The advantage of the Ridge Linear Regression is that it regularizes complexity by introducing penalty factors to reduce over fitting and handle multicollinearity.

Proposed Features Features are used to train the supervised grading model introducing learning from lexical overlaps, sentence-embedding-based similarities, word weighting, and answer length features.

Lexical overlap features. We use the Jaccard distance (Real & Vargas, 1996) and the Dice coefficient (Dice, 1945). These similarities allow the measurement of word overlap between the two answers at the expense of word frequency. We use the Normalized Longest Common Subsequence (LCS) (Islam & Inkpen, 2008) to consider the length of both the shorter and the longer answer and the maximal consecutive longest common subsequence starting at any character.

Word weighting features. Semantic similarity is enhanced by TF-IDF weights (Salton & Buckley, 1988) to distinguish the discriminating words in the corpus from those which are less discriminating. Part-of-speech tagging (Toutanova et al., 2003) is used to consider the syntactic aspect of the word's vector space. As sentence meaning unfolds from the verb, the highest weight is assigned to the verb followed by nouns, adjectives, and adverbs.

Sentence-embedding-based similarity features. These features are obtained based on the sentence distribution where the student answers and the reference answers are encoded using embeddings. We use distributed vector representations from the Word2vec embedding model (Mikolov et al., 2013) to capture syntactic and semantic contextual word-to-word relationships. The Word2vec model has two components: Continuous Bag of Words (CBOW) and skip-gram. The CBOW model infers the target word knowing the context of the word, while the skip-gram model infers the context of an input word. We compose the distributional representations of the words in the reference answer and student answer to characterize the semantics of entire sentences in terms of the cosine vector similarity. Vector representations of answers are generated by calculating the sum of word vectors for all words in the answers. First, the answer is tokenized into words. Word vectors are retrieved from the Word Embedding model and then summed to get a single vector representing the sentence vector. The TF-IDF and part-of-speech tagging weights are applied to the word vectors before summing them. For learning Arabic word distribution, we use the pre-trained Skip-gram Word Embeddings shared by (Zahran et al., 2015). The model consists of a multidimensional word representation of dimension 300 for Modern Standard Arabic. It contains about 6.3 million entries and about 5.8 billion words. It was trained on a large amount of raw Arabic texts from diverse sources (Arabic Wikipedia, Arabic Giga-word Corpus, Arabic Wiktionary, BBC, and CNN Arabic news corpus, Microsoft crawled Arabic Corpus, Arabic books...). For learning English word distribution, we use a 300-dimensional pre-trained Fasttext⁷ Skip-gram model trained with sub word information (Bojanowski et al., 2017). The model provides 2-million-word vectors trained on Common Crawl to capture meanings of the general English domain.

⁷ <https://fasttext.cc/docs/en/english-vectors.html>

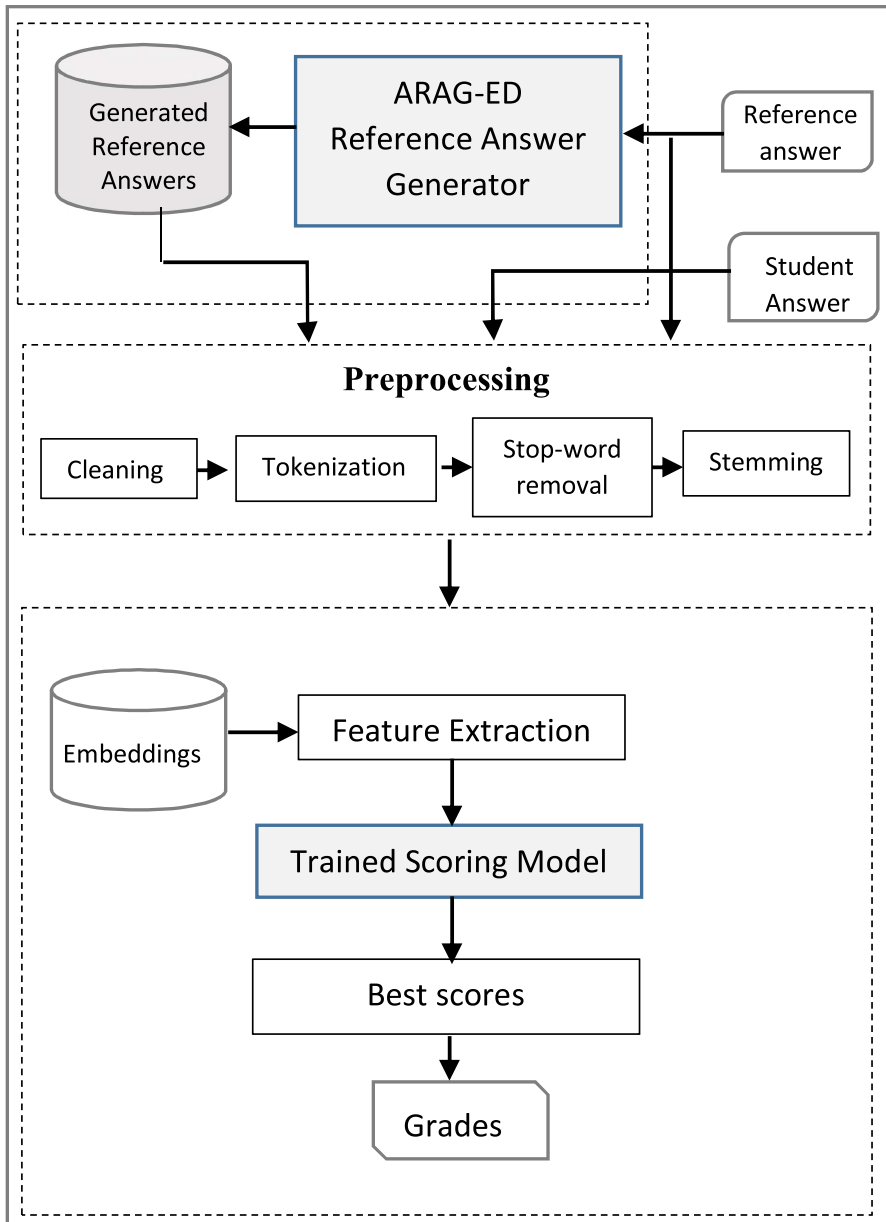


Fig. 3 Integrating paraphrase generator ARAG-ED into the ASAG tool

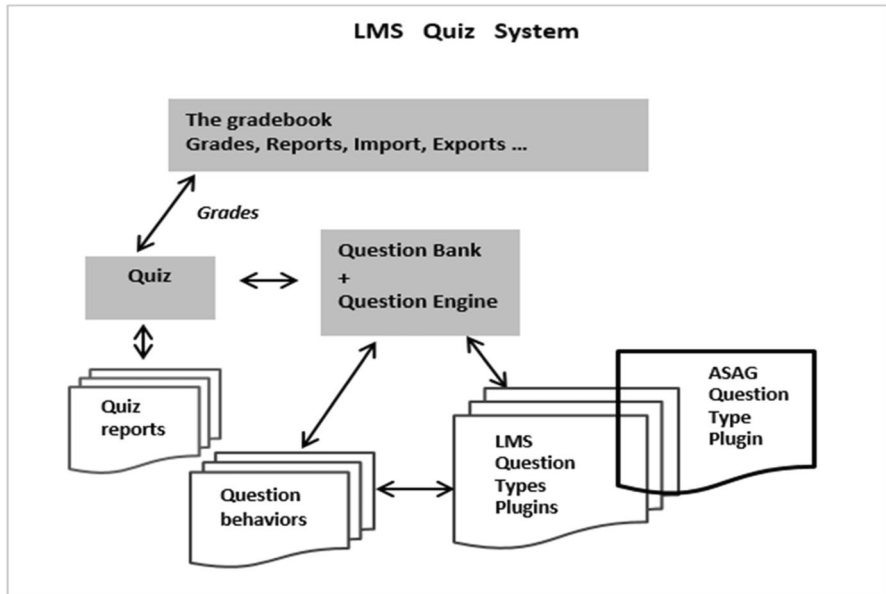


Fig. 4 The ASAG Question Type Plugin extends the question engine of the quiz system to the proposed ASAG model and controls communication with the LMS quiz system

Answer Length Features. Word length features play an important role when considering the similarity of sentences (Zhao et al., 2014). Longer sentences are more difficult to understand. To consider the role that word length features play in the similarity of sentences, we use length features in the training process: the length of the student's answer, the length difference between the student's answer and the reference answer, and the redundancy frequency of terms in answers. The redundancy frequency is the ratio of the number of words that repeat more than once in the student's answer to the total number of words in the answer. The redundancy frequency is considered as input features since the calculation of similarity is biased by word repetition.

Integrating the Paraphrase Generator ARAG-ED into the Grading Tool

As illustrated in Fig. 3, the proposed grading model is implemented to support the ASAG tool. The ASAG tool is integrated into the online quiz system of our university LMS. The code of the ASAG tool (English and Arabic- trained models) is shared here.⁸ The ARAG-ED generator is integrated into the ASAG tool. The Question Engine, the core module of the quiz system, is extended with the proposed grading tool. Given a reference answer, the ARAG-ED generates several alternative reference answers that are used as inputs with student answers to the trained model. The model is not retrained with multiple responses. Grade prediction for a student's

⁸ <https://github.com/Grader-ASAG>

response is done with the reference answer. It is recalculated with each generated reference answer. The best score is returned. Features are extracted to infer the trained model. The scoring processes is performed as presented in Fig. 2. To prepare the answers for feature extraction, we apply linguistic pre-processing techniques consisting of (sentence cleaning, stop-word removal, sentence tokenization, and sentence stemming). Sentence embedding is then performed using pre-trained word embedding to extract features that serve as inputs to the trained grading model.

Integrating the ASAG Tool into the LMS

Contextually, as in most of our country's public universities, our university is provided with an e-learning environment hosting a Moodle Learning Management System (LMS), as a part of a government initiative to reinforce face-to-face courses and to promote technology-enhanced learning in higher education. A large part of the courses and assessments are given in Arabic. The effectiveness of the proposed approach is thus, verified empirically across a real educational environment. The proposed model is implemented to support formative and summative short answer assessment tool. The ASAG tool (grader and generator) is integrated into the online quiz system of our LMS University as a plugin. The Question Engine of the LMS is extended with our new ASAG Question Type Plugin. As shown in Fig. 4, The ASAG Question Type Plugin extends the question engine of the quiz system to our ASAG model and ensures transparency with the LMS. This means that it is possible to create a quiz that seamlessly includes questions supported by our ASAG and existing question types already supported by the Learning Management System quiz system (essays, multi-choice...). The plugin controls communication with other modules of the quiz system and inherits from the LMS question behavior, question bank, quiz reports.... Teachers and students access the system and perform their specific role assessment tasks in the same way they access other activities. The smooth integration of the ASAG plugin into the quiz system allows inheriting the advantage that the question type and the question behavior are defined as separate concepts. The question behavior (for example, whether the question should be run in "interactive mode" with instant feedback and multiple trials or in "delayed mode" with a single attempt allowed and no comments until student responses have been submitted, ...) is left to the choice of the teacher according to the objective of the assessment (formative or summative). In conformity with the LMS interfaces, the tool provides interfaces in Arabic and English.

Evaluation and Results

Experiment Design

We conducted two types of experiments; quantitative and qualitative ones. Quantitative experiments measure the scoring accuracy using datasets. They are conducted both in Arabic and English on two datasets; the AR-ASAG Arabic dataset (Ouahrani & Bennouar, 2020) and the Mohler et al. (2011)'s English Short Answer Dataset

(Mohler et al., 2011). The qualitative experiments measure the impact of the integration of the ASAG on students' achievement according to formative and summative assessments. The approach is developed on a realistic case study conducted at our university. Experiments with students were conducted using the university web platform where students are already enrolled in different courses. Students follow a continuous assessment, which is conducted online to reinforce face-to-face courses in a hybrid teaching approach. Students access the LMS from the university and home. The developed ASAG plugin was integrated into the LMS Question Engine for the "cybercrimes" course taught in Arabic. Experiments are conducted with 30 students even if the ASAG is available for all students. In the context of formative assessments, several assignments were proposed to students. The assignments contained short answers questions and questions with objective answers (multiple choice, fill in blanks ...). The formative evaluation was led by tutors. Students have access to a history in which they can see their tests. The questions can be seen in detail, including the answer given by the student, the obtained grade, and the reference answers. A summative evaluation was conducted to assess student learning at the end of the course unit. It was carried out online in the university on secure internal workstations. Feedback is collected to evaluate the proposed approach.

Data Two datasets are used to train and experiment with the grading model: the Arabic AR-ASAG dataset (Ouahrani & Bennouar, 2020) and the English Mohler et al. (2011) dataset (Mohler et al., 2011). We randomly divide the datasets into a training set (70%), an evaluation set (10%), and a test set (20%) stratified into the question types (for the Arabic dataset). Features are extracted from the dataset. The features and grades of the dataset are loaded in a data frame to train the scoring models and compute the regression analysis. Each model is executed several times to find the optimal parameters that influence the model's efficiency for the best accuracy.

*AR-ASAG Dataset (2020)*⁹ (Ouahrani & Bennouar, 2020). The AR-ASAG Dataset is the only publicly available Arabic dataset dedicated to the evaluation of ASAG systems. It was introduced to stimulate research in the Arabic language, as there is a dire need to develop datasets in this language. The AR-ASAG Dataset includes short answer questions with answers extracted from the final tests. Tests were conducted for 170 high school students following the "cybercrimes" course and having a native Arabic language. The dataset consists of 48 questions and 2133 student answers, with 45 answers per question on average. Each question has a reference answer. The questions belonged to 5 types (Define the concept, Explain, What Consequences, Justify, and What is the difference). The responses are independently graded by two human experts, using a scale from 0 (completely incorrect) to 5 (perfect answer) with an Inter Annotator Agreement of $Pearson = 83,84\%$ and $RMSE = 0,8381$.

Mohler et al. (2011)¹⁰ (Mohler et al., 2011). It is widely used to assess work on English ASAGs. The dataset is collected from introductory computer science

⁹ <https://data.mendeley.com/datasets/dj95jh332j/1>

¹⁰ <https://web.eecs.umich.edu/~mihalcea/downloads.html>

Table 8 Proposed system evaluation results on the Arabic and English Datasets (Test Set)

	AR-ASAG Dataset		Mohler English Dataset	
	Pearson↑	RMSE↓	Pearson↑	RMSE↓
Inter-Annotator Agreement (Manual)	83,84	0,8381	64,43	–
Dataset baseline	70,37	1,0454	51,80	0,9780
ASAG-0 (Ours)	77,95	0,8968	66,89	0,8206
ASAG-M (M=10) (Ours)	88,92	0,6955	73,50	0,7790

assignments of a Data Structures course at the University of North Texas. Answers were provided by a class of undergraduate students. The authors used a combination of graph-based alignment and lexical similarity measures to grade short answers. The dataset contains 81 questions with a total of 2273 answers. The dataset was graded by two human judges on a scale of 0–5 with a Total Agreement (giving the same grade) of 57.7%. The dataset is an extended version of the dataset used by Mohler and Mihalcea, (2009) which had a 64,43% inter-annotator agreement per question evaluation.

Evaluation metrics Pearson correlation (r : the higher the better↑) is the most frequently used metric for research in this area. We reported it for all our experiments. We report the Root Mean Squared Error (RMSE, the lower the better↓) and the Pearson coefficient.

Baselines For Arabic, we use the dataset baseline reported in (Ouahrani & Bennouar, 2020). The baseline used the unsupervised K-Means clustering and performed (Pearson = 70,37% & RMSE = 1,0454). For English, we compare to previous work using the Mohler dataset.

Results

The results are presented and discussed from different perspectives: the impact of using multiple reference answers on the ASAG, Comparison with previous works on the Mohler Dataset, analysis of the grading errors and limitations, student achievement using the tool, and the computational complexity of the grader. For evaluation purposes, two variants (ASAG-0 and ASAG-M) are deployed. ASAG-0 indicates ASAG with a single teacher-provided reference answer, and ASAG-M indicates ASAG with the provided reference answer augmented by M multiple generated reference answers corresponding to Beam=M). We chose to deploy ASAG-10 (Beam = 10) presenting the best accuracy on the test set of the ASAG datasets.

The impact of multiple reference answers on the ASAG task The proposed ASAGs results evaluation on the Arabic and English Datasets (Sample Test) are presented respectively in Table 8. Compared to the baseline of the dataset, the proposed models (ASAG-1 and ASAG-M) are significantly better at scoring. Incorporating

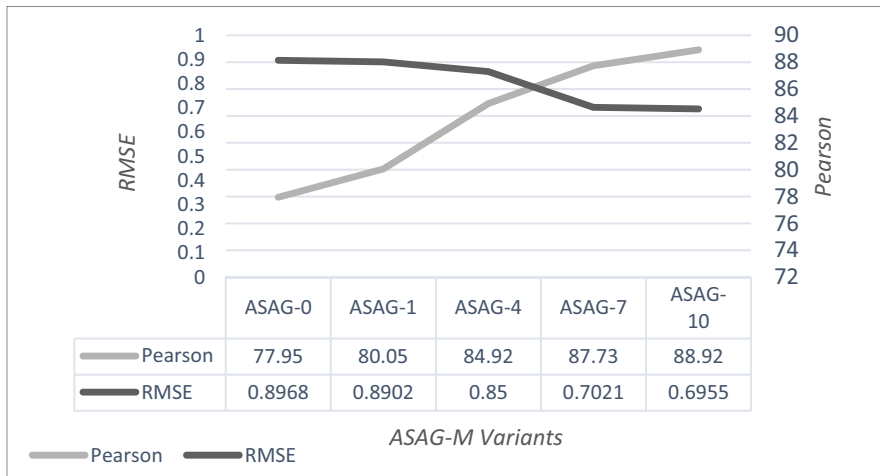


Fig. 5 Ablation study on the multiplicity of reference answers on the Arabic Dataset

paraphrase generation into the grader gave better results. Pearson's correlation improved significantly from 66,89% (without paraphrases) to 73,50% (with paraphrases) for the English dataset. It improved from 77,95% (without paraphrases) to 88,92% (with paraphrases) for the Arabic dataset. The obtained correlation exceeds the Inter-Annotator Agreement on the two datasets. The difference between manual and automatic scores is significantly improved. Indeed, RMSE decreased from 0,8206 (without paraphrases) to 0,7790 (with paraphrases) for the English dataset. It decreased from 0,8968 (without paraphrases) to 0,6955 (with paraphrases) for the Arabic dataset.

Ablation study. While reporting comparative results for the Arabic dataset, we also show experiments with a variant number of generated reference answers to show the impact of the multiplicity of reference answers on grading accuracy. We performed experiments including no paraphrases (ASAG-0). We included then multiple paraphrases (ASAG-1, ASAG-4, ASAG-7, ASAG-10) corresponding to Beam = 1, 4, 7 and 10. As we can see in Fig. 5, not considering paraphrases decreases the correlation. The correlation improves markedly with each increase in the number of paraphrases. Between 7 and 10 paraphrases, improvement is small (Pearson: + 1,19%, RMSE: -0,0066) since with 7 paraphrases the best-paraphrased reference answers are reached often. This observation can be beneficial when deploying the system where it would be sufficient to set the generation of paraphrases to 7. In general, the tendency is linear representing a regular increase in Pearson and a regular decrease in RMSE. The obtained results show that using the generated alternative reference answers combined with the supervised grading model provide a significant improvement over using a single reference answer. This suggests that the proposed Encoder-Decoder yields plausible paraphrases.

Table 9 Comparison with previous work on the Mohler Dataset

ASAG System	Pearson↑	RMSE↓
Pribadi et al. (2018)	46,80	0,8840
Saha et al. (2018)	57,00	0,9000
Ramachandran and Foltz (2015)	61,00	0,8600
Gomaa and Fahmy (2020)	63,00	0,9100
Sultan et al. (2016)	63,00	0,8500
Kumar et al. (2017)	55,00	0,8300
Agarwal et al. (2022)	–	0,7620
Gaddipati et al. (2020)	ELMo WE 48,50	0,9780
	GPT WE 24,80	1,0820
	BERT WE 31,80	1,0570
	GPT-2 WE 31,10	1,0650

It confirms the finding that paraphrase generation significantly helps ASAGs to improve.

Comparison with previous work on the Mohler Dataset As reported in Table 8, our model's correlation using multiple reference answers over all the test data samples is 73,50%. The RMSE is 0,7790. Table 9 shows the correlation and RMSE results of previous reference-based methods as conducted in (Agarwal et al., 2022; Gomaa & Fahmy, 2020; Kumar et al., 2017; Pribadi et al., 2018; Ramachandran & Foltz, 2015; Saha et al., 2018; Sultan et al., 2016) presented in related work (Sect. "ASAG Approaches"). The proposed approach achieves near the state-of-the-art (Agarwal et al., 2022) System (RMSE:-0,017, *Pearson not reported*).

Compared to works using an unsupervised approach, (Ramachandran & Foltz, 2015) and (Pribadi et al., 2018) proposed the generation of alternative reference answers using the Maximum Marginal Relevance (MMR) method (Pribadi et al., 2018) and summarization of the content of top-scoring student responses (Ramachandran & Foltz, 2015). Gomaa and Fahmy (2020) used a Skip-thought vector unsupervised approach to convert reference and student answers into embeddings to measure their similarity. Although they present the advantage of not requiring a training process, the impact of the paraphrase generator is more significant in the proposed approach (Pearson: +26,7%, RMSE: +0,105) compared to the (Pribadi et al., 2018) system, (Pearson: +12,5%, RMSE: +0,081) compared to the (Ramachandran & Foltz, 2015) System and (Pearson: +10,57%, RMSE: +0,131) compared to the (Gomaa & Fahmy, 2020) System. High-precision scoring is still a challenge especially if the ASAG evaluation has high stakes for students.

Compared to works using a supervised approach requiring a training process, the proposed system is similar to what Sultan et al. (2016) used. Sultan et al. (2016) trained a regression model involving semantic similarity, text alignment, question demoting, term weighting, and length ratio features. The use of multiple reference

answers generated by paraphrase generation enabled the proposed approach to outperform Sultan's system by (Pearson + 10,5% and RMSE -0,071).

The proposed approach outperforms (Kumar et al., 2017) and (Saha et al., 2018) systems that used a deeper approach. Kumar et al. (2017) used a Siamese bidirectional LSTMs applied to a reference and a student answer, based on earth-mover distance across all hidden states from both LSTMs and a final regression layer to output grades. Saha et al. (2018) combined Handcrafted features and sentence embedding features to train an end-to-end deep neural network to learn embeddings and a neural network to train a grading classifier. This is challenging, as training embeddings requires large data.

Recent methods for ASAG explored sophisticated feature representations, computed by attention-based and transformer models (Vaswani et al., 2017) to better capture structural and semantic features. Agarwal et al. (2022) applied short text matching using Multi-Relational Graph Transformer representation to incorporate relation-enriched structural information and achieved the state-of-the-art on the Mohler Dataset (RMSE:—0,7620, Pearson not reported).

Transformer-based language models like BERT, showed state-of-the-art results on various NLP tasks such as question answering, sentiment analysis, text summarization and more. They are powerful but they are computationally prohibitive (Huang et al., 2022). With the ASAG Task, they are presenting two challenges: First, these models for ASAG remain difficult in practice because the ASAG's dataset is typically small and cannot provide enough training or fine-tuning data. Gaddipati et al. (2020) used pre-trained embeddings of the transfer learning models ELMo (Peters et al., 2018), BERT (Devlin et al., 2019), GPT (Radford et al., 2018), and GPT-2 (Radford et al., 2019) with the cosine similarity to evaluate their effect on the ASAG task. The performance is poor as we can see in Table 9. When used on a large scale, as is the case in e-learning environments, they present challenges for practitioners. Because saving and loading model parameters require more memory and processing power, they can be computationally expensive during inference, which may limit their applicability in environments with scarce resources. The proposed approach employs a more simplified model augmented by paraphrase generation and achieves near the state-of-the-art (Agarwal et al., 2022) System (RMSE:-0,017) on the Mohler dataset. Our model is simpler to construct, load, and embed into the LMS question engine with low computational complexity. It requires a small dataset, which is easier to obtain from the teaching archive or LMS question Bank.

Analysis of the grading errors and limitations ASAG systems require precise notation and a thorough understanding of response text, making them difficult to implement in real-world settings. When the system fails to achieve reasonable performance, trust issues arise, and errors in automatic grading can have a significant impact on individuals (Azad et al., 2020; Hsu et al., 2021; Schneider et al., 2023). The issue of trust in the tool will remain a major challenge until ASAGs can provide trustworthy human performance. Because of their statistical nature, reliable ASAG systems are critical for trust and practical utility.

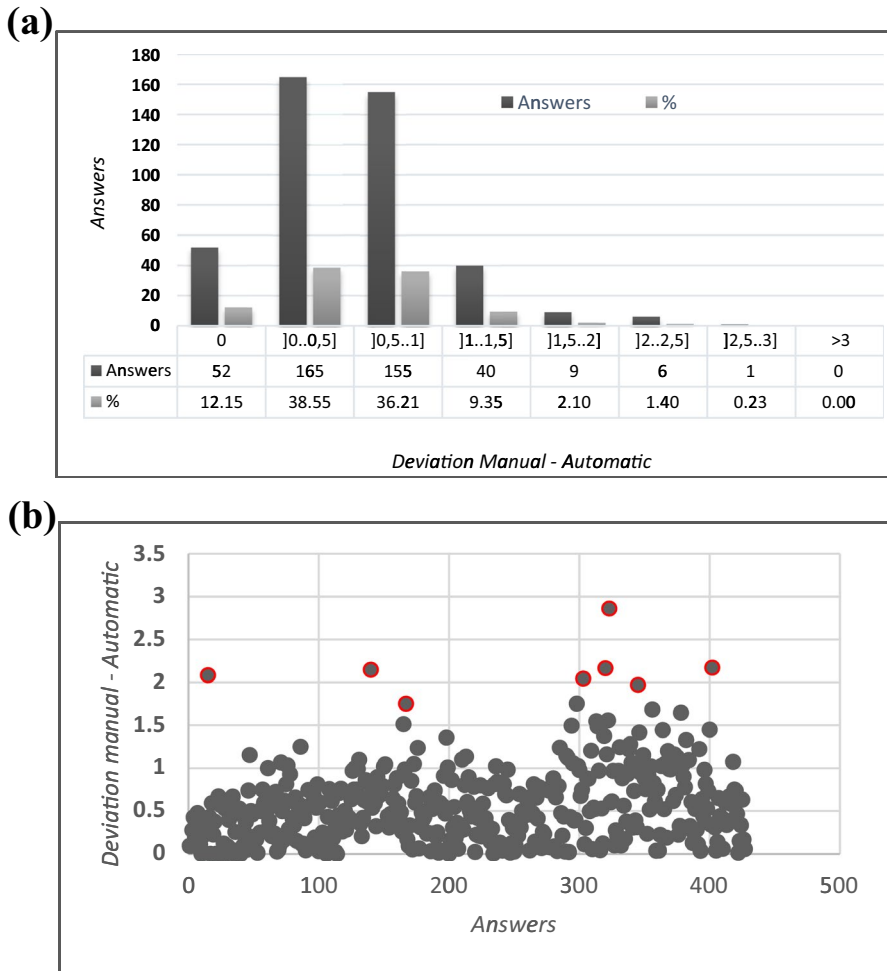


Fig. 6 **a** automatic and Manual Scores Difference on the AR-ASAG Dataset (Set Test). The difference between manual and automatic scores is analyzed on a scale of 5. **b** Automatic and Manual Scores Difference on the AR-ASAG Dataset (Set Test). The points in the interval [1, 1,5] are more likely to be closer to 1 than to 1,5

While errors cannot be completely avoided, they can be managed. For example, it may be preferable to pass a student who should have failed rather than fail a student who should have passed (Schneider et al., 2023). A thorough analysis of automatic versus human grades, as shown in Figs. 6a and b, allows for a more accurate evaluation of performance in order to determine an acceptable margin of error. We investigate the variation between manual and automatic grades on a scale of 5 for the Arabic set test, which contains 428 question-answer pairs. As shown in Fig. 6a, the correlation between the predicted score (by rounding off the scores) and the human score is perfect for 52 responses (12.15%). In 86,91%

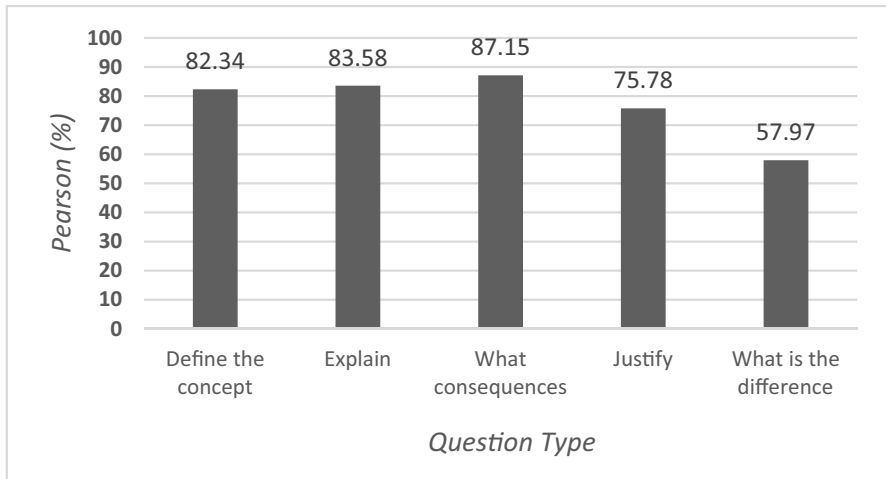


Fig. 7 Distribution of Automatic-Manual grades per question type on the AR-ASAG Dataset Set test. The best correlation was obtained for the "What consequences?" question. Good scores were obtained for the "Explain" and "Define the concept" questions as well. The results were less good for the "What is the difference" and "Justify" questions

of the cases, the difference is between 0 and 1 on a scale of 5. This is reasonable since the correlation exceeds the overall agreement among human annotators (IAA = 83,84% for the entire Arabic dataset). The results give a good indication of the proposed model in consideration of the subjectivity of the evaluation process. In 96,26% the difference is less than or equal to 1,5. The scatter plot of the difference between manual and automatic grades in Fig. 6b shows similarly a strong condensation of the points in the [0, 1] interval and a little less in the interval [1, 1,5]. The points in the interval [1, 1,5] are more likely to be closer to 1 than to 1,5. Beyond that, the difference between the scores becomes problematic, especially if the evaluation has high stakes. The difference exceeds 1,5 in 3.73% of responses (16 answers out of 428). To understand this scoring bias, a deeper analysis by question type is conducted.

An analysis of automatic grades compared to human correlation per type question is presented in Fig. 7 using the Pearson correlation. The questions belonged to five types (Define the concept, Explain, What Consequences, Justify, and What is the difference). The best correlation was obtained for the "What consequences?" question (Pearson = 87,15%) better than the manual correlation between the two human annotators for the whole Arabic Dataset. Good scores were obtained for the "Explain" (83,58%) and "Define the concept" (82,34%) questions as well. The results were less good for the "What is the difference" (57,97%) and "Justify" (75,78%) questions. For example, among the lowest automatically scored answers in the test sample and when evaluating student assignments, there are those relating to the two questions:

- What is the difference between hacking and penetration testing?
ما الفرق بين القرصنة واختبار الاختراق؟ (In Arabic)

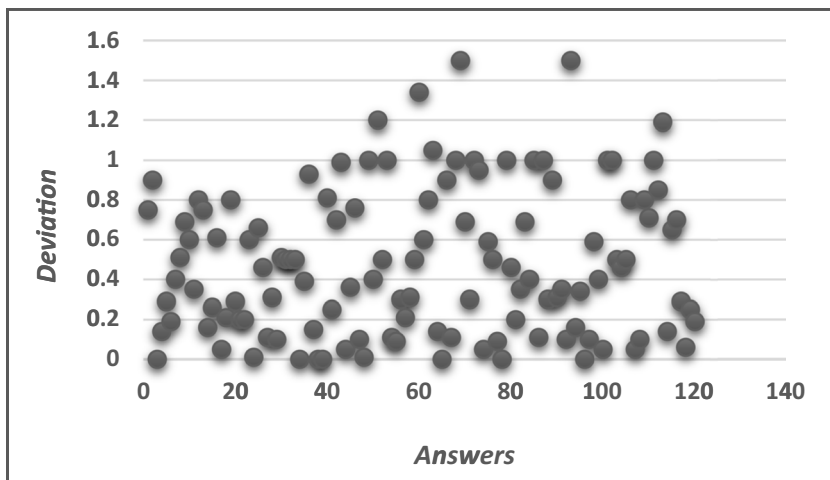


Fig. 8 Manual-automatic grades per-question distribution on final summative assignment

- Justify the truth of this statement: “A false sense of security is more dangerous than a true sense of insecurity”.

(In Arabic) علل صحة العبارة: الإحساس الخاطي بالأمن أخطر من الإحساس الصحيح بعدم الأمن

Some causes of this bias may be observed through analysis of the responses. First, the teacher and students use different comparison criteria, which could indicate that the two answers are different. Second, the topic of short answers to the "justify" question is so broad that it encourages veering off topic quite easily. Third, the length of the student response has a negative effect on the grade. Usually, when they include irrelevant information, students exceed the word limit of the response. In terms of similarity, the grader has poorly assessed a long answer. These findings support the statement that “the issues on which ASAG systems perform poorly are often also the ones on which humans do not agree” (Adams et al., 2016). For automatic short answer questions, we recommend that emphasis be placed on the best way to ask the question. Short answers to questions have a way of veering off-topic quite quickly. This is why it is important to have guidelines in place for students and to be as specific as possible when defining a short-answer question. Learners must be guided to know exactly what is expected of them. A broad topic should not be chosen over specific ideas or concepts. A hybrid assessment approach (manual-automatic) appears to be more effective for questions that are "difficult to assess automatically." The LMS quiz system is configured to allow manual scoring if necessary. All of these observations reflect difficulties and issues with grading in the real world.

Formative and Summative Assessment Using the ASAG We analyze the results obtained by students in the formative and summative tests. The automatic grades of the final test were collected and evaluated per question and per assignment.

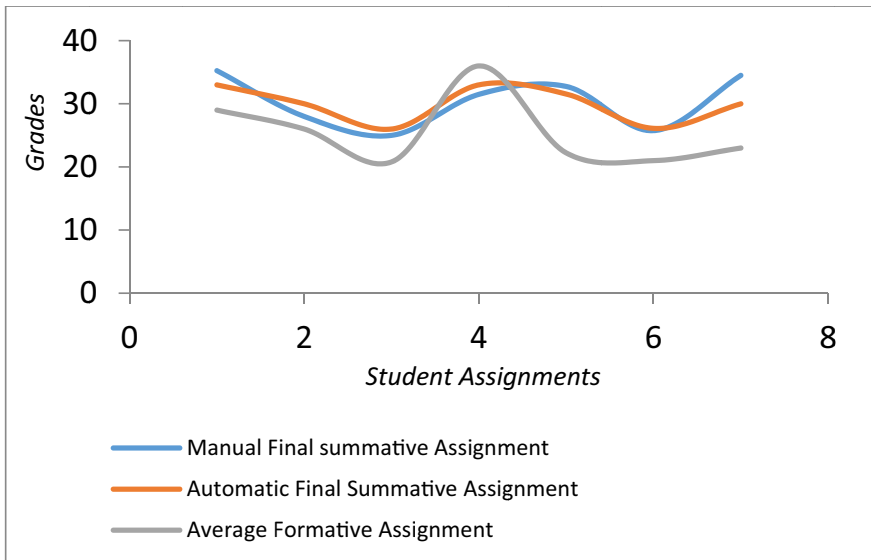


Fig. 9 Per-assignment distribution grades-average formative assignment vs. final summative assignment. The overall response scoring error is reduced as it is compensated for between different questions in the same assignment. The average formative grade distribution curve is mostly below the final summative grade distribution curve

Evaluation per question considers the correlation obtained for all the short answers of all students. It indicates the performance of the tool in a natural context compared to the evaluation using datasets. Evaluation per assignment considers the summative grade of the student in the assignment. In parallel, tutors also made manual grades for the final test to check the performance of the ASAG in a real-world context. The correlation between human scoring and automatic scoring was calculated for only short answers; all other questions were objective (correlation = 1). The tool has been configured so that the student can attempt the same question multiple times without penalty in all the test we have done.

Per-question evaluation. Considering the students' responses to the summative assignment, the automatic scoring gave a good correlation with the human grades, i.e., Pearson = 88,09% & RMSE = 0,6464 (on a five scale). The scatter plot of the difference between manual and automatic grades in Fig. 8 shows a strong condensation of the points in the [0, 1] interval and a little less in the interval [1, 1,5]. Although the tool does not provide complete trust, the results are still acceptable due to the subjective nature of the scoring process of short answers. Compared to the results obtained for the Arabic dataset, the correlation is better. After the formative sessions, the students learned to respond better by emphasizing the target concepts. Therefore, the tool performed better. The analysis of the incorrectly automatically scored responses revealed the negative impact of the length of the student response. In terms of similarity, the grader has poorly assessed a long answer. It will

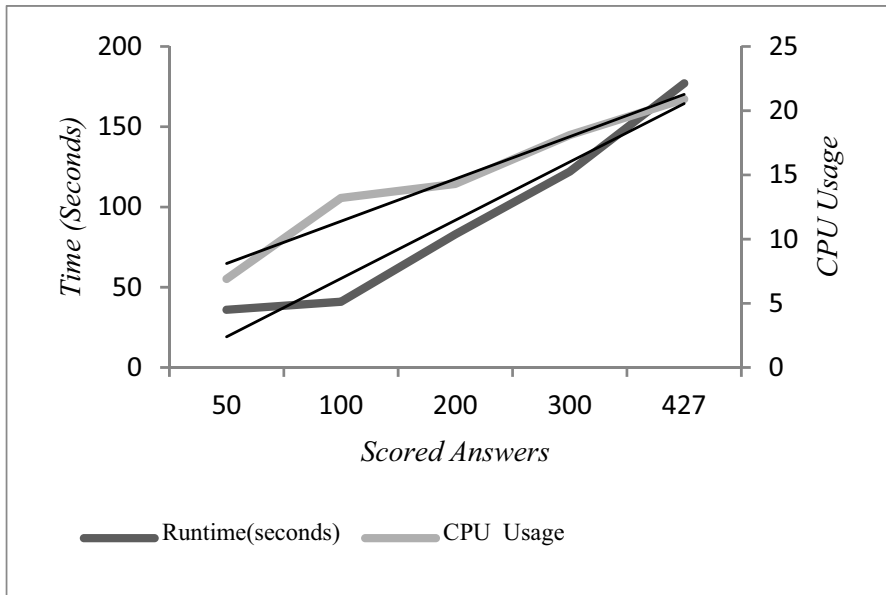


Fig. 10 Runtime and CPU usage

be necessary to impose a maximum length to guide students to target the synthesis aspects and concepts of an open short answer in the formative sessions.

Per assignment evaluation. We consider the summative assessment of students' final scores. The correlation between manual and automatic grades is shown in Fig. 9. The overall response scoring error is reduced as it is compensated for between different questions in the same assignment. As we can see, the manual and automatic student final grades are very close. The correlation with the human score is about 91,81%, which is very encouraging. Regarding the formative evaluation, the average of the assignments is calculated for each student and is compared to the final summative grade. As shown in Fig. 9, the average formative grade distribution curve is mostly below the final summative grade distribution curve. In general, the final mark is better since the tool facilitated better learning through the formative tests available in the course during the learning period.

Computational Complexity When automatic scoring is deployed into e-learning environments, there are some factors other than accuracy that we have to consider, such as server performance and practical efficiency. A light computing complexity is highly needed since the scoring system will be integrated with online learning. We measured computational complexity by the amount of CPU usage and runtime varying the number of pairs of (student answer, reference answer). We conducted the experiments on a machine with Intel(R) Core(TM) i7 CPU @2.50GHz with 8.0 GB of RAM and an Operating System as 64-bit Windows 10 with an internet speed of about 4 Mbps. As illustrated in Fig. 10, the consumption usage has a linear tendency. It took from 6,2% to 20,9%. The runtime took from 36 to 177 s as the number

of pairs of answers to evaluate varied from 50 to 427. This is reasonably correct in an online test session. It indicates that the approach could be adapted alike without much overhaul of LMS existing systems and not need a powerful computer. The tool can be deployed on any Moodle platform where it can provide a uniform assessment system over time for all courses in the LMS.

Overall Discussion and Implications

In practice, a few ASAG tools are implemented and made available in e-learning systems. They still require significant manual supervision. Although our case study was conducted in our academic e-learning environment, the findings are likely to be relevant to a wider educational community.

Until ASAGs can have human performance, automatic grading should be used to supplement, rather than replace, human grading. In this sense, our study aims to provide feedback for improvement.

The major finding of this study is the observation that the use of paraphrase generation significantly helps ASAG improve. It can be useful at a large scale without requiring the availability of hundreds of student responses to train each question. Such an approach can be as effective as approaches using sophisticated models that make the system difficult to achieve and to use at scale. It is easier to build (or find) a corpus that is not necessarily of large size and does not require extensive resources for the implementation of the approach. This fact is interesting for under-resourced languages such as the Arabic language and under resourced online educational environments. The obtained results on the English Mohler dataset indicate that the proposed approach can be reused for other datasets and languages. The model may generalize across domains in higher education where a variety of courses in different domains is taught.

Moodle is the leading open-source virtual learning environment and has proven to be a suitable vehicle for developing our grader engine. In recent years, the increasing adoption at our country's universities has led us to combine expertise and resources to fully embed and thereby enhance the assessment of free-text short answer questions available in Moodle by a supervised learning approach using multiple reference answers. Deploying the proposed ASAG as a plugin would be beneficial to both students and teachers. It would encourage teachers and students to use the LMS in assessments too. LMSs are currently used often in the delivery of course materials and for objective assessments. Teachers would get useful information about individual learners' progress. We are working on the deployment of ASAG for more courses for more improvement. Because the ASAG task is very domain-specific, we need to train our model on a new dataset. The proposed model needs a small dataset to train on the specific domain of the course. We explore the course's history and the LMS question bank for the targeted course.

Consideration of feedback collected, we retain the following implications of integrating the ASAG in the educational e-learning environment. First, the proposed system offers repeat practice opportunities for formative assessment. Assignments are

designed to enable students to complete them in their own time. Having a sequence of assessments is less intimidating to students than a single online final one. This results in a higher participation rate and an improvement in scores. Second, Integrated as a smoothing extension, the ASAG takes advantage of all the strengths of the LMS. Statistics on student performance on each question are provided. For each question, all attempts are saved, even those generated by the first reflection. The individual progress of students through a question can be examined. The teacher can identify the aspects of the questions, which prove problematic; possibly badly formulated by the teacher. He can then adapt the assessment design as well.

Discussing the viability of our approach, a trained model is used to make predictions on new data that has not previously been seen by the model. Our model has been trained to grade short answer questions in the cybercrimes course, and it can be used to grade new short answers in this domain. This means that the teacher can add new questions to the course without having to retrain the model. He should only prepare the question and the reference answer, as is customary in exams. The trained model is loaded into the LMS embedded into a plugin to ensure predictions. To deploy a new ASAG on another course, we need to train our model on a new dataset, as the ASAG task is very domain-specific. The most difficult part is acquiring the dataset. Some supervised approaches require a large dataset (all recent approaches, BERT, GPT, and T5), while others may work with a small dataset to train the model. The proposed model needs a small dataset to train on the specific domain of the course. We explore the course's history and the LMS question bank for the targeted course. Once trained, the model can be loaded and deployed to a new plugin. Open Mark's PMatch system (Jordan & Butcher, 2013) developed at the Open University that is considered the more developed ASAG system in e-learning environments is based on the matching of keywords and their synonyms. The model requires several hundred of student answers to train each question which is challenging. Comparing to Open Mark system, we proposed a general-question model that train for the entire questions in the dataset. Unlike training the model separately for each question, various questions are trained in the same model.

Discussing the limitations of the ASAG model, or rather its challenges, entails determining the margin of precision between human annotators and the ASAG system, determining an acceptable level of error that can be tolerated in a real-world context, and deciding what action to take if necessary. With reasonable accuracy and reliability, automatic grading of short answer questions can be accepted. It may be determined by the assessment's purpose and context, such as whether it is formative or summative, low-stakes or high-stakes. The risk of over-scoring is also problematic. It may be assumed actually by the fact that, in the absence of a performance grading equivalent to human performance (which is currently the case), it is more acceptable to over-score the student than to under-grade him (Schneider et al., 2023). As discussed in the evaluation, the model performed poorly in areas involving broad topic questions. When the topic of short answers is so broad, it encourages straying from it. The grader is influenced by the length of the student's response. A long answer was poorly graded. Furthermore, when defining a short-answer question, it is critical to have guidelines in place for students and to be as specific as

possible. Learners must be guided so that they understand exactly what is expected of them.

Finally, until ASAGs can have human performance, ASAG should help improve and supplement human scoring. In this sense, our study aims to provide feedback for improvement. In low-stakes formative activities while students are studying alone and answering exam preparation questions, the grader may assess their answers. When teachers are unable to provide feedback on students' answers due to time constraints, the ASAG with an acceptable margin of error can be used. The tool may be configured so that the student can attempt the same question multiple times without penalty to encourage motivation. A manual-automatic hybrid assessment approach appears to be more effective for questions that are "difficult to assess automatically" in high-stakes assessments.

Conclusion and Future Work

Two main contributions are proposed in this paper. First, we have presented the first approach that uses paraphrase generation for improving automatic short answer scoring. Our model for generating alternative answers is simple and can generate several paraphrases, for a given reference answer. Evaluation of the proposed model on the Arabic and English datasets demonstrates its effectiveness and improved performance over state-of-the-art approaches by a significant margin, while qualitative human evaluation indicates that the generated paraphrases are well-formed, grammatically correct, and relevant to the input sentences. We successfully adapted a paraphrase generation method to the domain of ASAG to provide a variety of reference answers that can handle the diversity of student responses. The paraphrase generator relieves thus the teacher of the tedious task of manually constructing multiple formulations of the reference answer. Second, generating high-quality paraphrases remains a challenging NLP task. The Arabic language is one of the most complex and rich languages morphologically. The lack of linguistic resources, such as corpora, dictionaries, and lexicons, makes Arabic NLP research more challenging. Our paper responds to this challenging task in Arabic NLP and establishes baselines for future research and a new deep learning-based formulation of the paraphrase generation task in Arabic. We still need to work on improving the quality of the alternative reference answers in terms of readability and relevance. It requires retraining the generator model on rich and contextual datasets. The main obstacle here is the crucial scarcity of such parallel corpora in Arabic. This issue needs to be addressed seriously. To fill this challenge, our scope for future work concerns automatically generating Arabic datasets for paraphrase generation. We approach this end in two ways. First, by exploiting existing parallel bilingual corpora in other languages. Second, by leveraging unstructured web knowledge to generate automatically different datasets of specific domains. An Arabic version AraT5 (Nagoudi et al., 2022) of the mt5 model language (Xue et al., 2021) has very recently become available. It would be interesting to investigate the quality improvement of paraphrased reference responses by fine-tuning the AraT5 model to our Arabic dataset. An important aspect that has not yet received attention in ASAG's work concerns audience

knowledge. When developing short answer questions, the teacher had to develop them with the target audience in mind. When the proposed ASAG is integrated into the LMS, it is difficult to learn about the public's background before developing questions. We plan to explore the use of feedback for predicting student behavior regarding students' comprehension levels, vocabulary, and how much time they have to complete the assessment. All this feedback is available since the ASAG is integrated into the e-learning environment.

Acknowledgements The authors would like to thank Selena LAMARI, Oussama HAMEL, Ahmed Hadersi, and Oussama Benguergoura for their technical help during the experimentation phase.

Author's Contributions All authors worked collaboratively to formulate research questions, conduct the search, select data, and perform analysis, experiments, and discussion. The corresponding author worked on writing the initial draft. All authors reviewed, read, and approved the final manuscript.

Funding The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Data Availability The datasets used during the current study are available in:

- *Al-Raisi Arabic Dataset*: <http://www.cs.cmu.edu/~fraisi/arabic/arparallel/>
- *The Quora English Dataset*: <https://github.com/jakartaresearch/quora-question-pairs>
- *AR-ASAG Dataset (2020)*: <https://data.mendeley.com/datasets/dj95jh332j/1>
- Mohler et al. (2011) *Dataset*: <https://web.eecs.umich.edu/~mihalcea/downloads.html>

Declarations

Competing Interests The authors have no relevant financial or non-financial interests to disclose.

References

- Ab Aziz, M. J., Ahmad, F. D., Ghani, A. A. A., & Mahmud, R. (2009). Automated marking system for short answer examination (AMS-SAE). *Undefined, 1*, 47–51. <https://doi.org/10.1109/ISIEA.2009.5356500>
- Adams, O., Roy, S., & Krishnapuram, R. (2016). Distributed vector representations for unsupervised automatic short answer grading. In *Proceedings of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA2016)* (pp. 20–29). <https://aclanthology.org/W16-4904>. Accessed 22 Feb 2022.
- Agarwal, R., Khurana, V., Grover, K., Mohania, M., & Goyal, V. (2022). Multi-Relational Graph Transformer for Automatic Short Answer Grading. *NAACL 2022 - 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 2001–2012*. <https://doi.org/10.18653/v1/2022.naacl-main.146>
- Alkhatib, M., & Shaalan, K. (2018). Paraphrasing Arabic metaphor with neural machine translation. *Procedia Computer Science, 142*, 308–314. <https://doi.org/10.1016/j.procs.2018.10.493>
- Al-Raisi, F., Bourai, A., & Lin, W. (2018a). Neural symbolic arabic paraphrasing with automatic evaluation. *Computer Science & Information Technology*, 01–13. <https://doi.org/10.5121/CSIT.2018.80601>
- Al-Raisi, F., Lin, W., & Bourai, A. (2018b). A monolingual parallel corpus of Arabic. *Procedia Computer Science, 142*, 334–338. <https://doi.org/10.1016/J.PROCS.2018.10.487>
- Ashton, H. S., Beevers, C. E., Milligan, C. D., Schofield, D. K., Thomas, R. C., & Youngson, M. A. (2005). Moving beyond objective testing in online assessment. In *Online Assessment and Measurement: Case Studies from Higher Education, K-12 and Corporate* (pp. 116–128). IGI Global. <https://doi.org/10.4018/978-1-59140-497-2.ch008>

- Azad, S., Chen, B., Fowler, M., West, M., & Zilles, C. (2020). Strategies for deploying unreliable AI graders in high-transparency high-stakes exams. *Lecture Notes in Computer Science (Including Sub-series Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12163 LNAI, 16–28. https://doi.org/10.1007/978-3-030-52237-7_2
- Babych, B. (2014). Automated MT evaluation metrics and their limitations. *Tradumàtica: Tecnologies de La Traducció*, 12, 464. <https://doi.org/10.5565/rev/tradumatica.70>
- Bahdanau, D., Cho, K. H., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. <https://arxiv.org/abs/1409.0473v7>. Accessed 24 Feb 2022.
- Beckman, K., Apps, T., Bennett, S., Dalgarno, B., Kennedy, G., & Lockyer, L. (2019). Self-regulation in open-ended online assignment tasks: The importance of initial task interpretation and goal setting. *Studies in Higher Education*. <https://doi.org/10.1080/03075079.2019.1654450>
- Bloom, B. S. (1984). Taxonomy of educational objectives book 1: Cognitive domain. In *nancybroz.com*. http://nancybroz.com/nancybroz/Literacy_I_files/BloomIntro.doc. Accessed 31 Aug 2021.
- Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5, 135–146. https://doi.org/10.1162/tacl_a_00051
- Brown, S., & Glasner, A. (Eds.). (1999). *Assessment matters in higher education: Choosing and using diverse approaches*. <https://eric.ed.gov/?id=ED434545>. Accessed 24 Feb 2021.
- Burrows, S., Gurevych, I., & Stein, B. (2015). The eras and trends of automatic short answer grading. In *International Journal of Artificial Intelligence in Education* (Vol. 25, Issue 1, pp. 60–117). Springer New York LLC. <https://doi.org/10.1007/s40593-014-0026-8>
- Cahuantzi, R., Chen, X., & Güttel, S. (2021). *A comparison of LSTM and GRU networks for learning symbolic sequences*. <http://eprints.maths.manchester.ac.uk/>. Accessed 25 May 2023.
- Carbonell, J., & Goldstein, J. (1998). Use of MMR, diversity-based reranking for reordering documents and producing summaries. *SIGIR Forum (ACM Special Interest Group on Information Retrieval)*, 335–336. <https://doi.org/10.1145/290941.291025>
- Carneiro, T., Da Nobrega, R. V. M., Nepomuceno, T., Bian, G. B., De Albuquerque, V. H. C., & Filho, P. P. R. (2018). Performance analysis of google colab as a tool for accelerating deep learning applications. *IEEE Access*, 6, 61677–61685. <https://doi.org/10.1109/ACCESS.2018.2874767>
- Chaganty, A. T., Musmann, S., & Liang, P. (2018). The price of debiasing automatic metrics in natural language evaluation. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 1, 643–653. <https://doi.org/10.48550/arxiv.1807.02202>
- Chen, M., Tang, Q., Wiseman, S., & Gimpel, K. (2020). Controllable paraphrase generation with a syntactic exemplar. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 5972–5984. <https://doi.org/10.18653/v1/p19-1599>
- Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 1724–1734. <https://doi.org/10.3115/v1/d14-1179>
- Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). *Empirical evaluation of gated recurrent neural networks on sequence modeling*. <https://arxiv.org/abs/1412.3555v1>. Accessed 20 Dec 2022.
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference* (vol. 1, pp. 4171–4186). <https://github.com/tensorflow/tensor2tensor>. Accessed 27 Sept 2022.
- Dice, L. R. (1945). Measures of the amount of ecologic association between species. *Ecology*, 26(3), 297–302. <https://doi.org/10.2307/1932409>
- Dzikovska, M., Steinhäuser, N., Farrow, E., Moore, J., & Campbell, G. (2014). BEETLE II: Deep natural language understanding and automatic feedback generation for intelligent tutoring in basic electricity and electronics. *International Journal of Artificial Intelligence in Education*, 24(3), 284–332. <https://doi.org/10.1007/s40593-014-0017-9>
- Gaddipati, S. K., Nair, D., & Plöger, P. G. (2020). *Comparative evaluation of pretrained transfer learning models on automatic short answer grading*. <https://arxiv.org/abs/2009.01303v1>. Accessed 27 May 2023.

- Gomaa, W. H., & Fahmy, A. A. (2020). Ans2vec: A scoring system for short answers. *Advances in Intelligent Systems and Computing*, 921, 586–595. https://doi.org/10.1007/978-3-030-14118-9_59
- Goyal, T., & Durrett, G. (2020). Neural Syntactic Preordering for Controlled Paraphrase Generation. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 238–252. <https://doi.org/10.18653/v1/2020.acl-main.22>
- Gupta, A., Agarwal, A., Singh, P., & Rai, P. (2018). A deep generative framework for paraphrase generation. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 5149–5156. <https://doi.org/10.5555/3504035.3504666>
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/NECO.1997.9.8.1735>
- Hsu, S., Wentin, T., Zhang, Z., & Fowler, M. (2021). Attitudes surrounding an imperfect ai autograder. *Conference on Human Factors in Computing Systems - Proceedings*. <https://doi.org/10.1145/3411764.3445424>
- Huang, S., Wu, Y., Wei, F., & Luan, Z. (2019). Dictionary-guided editing networks for paraphrase generation. *33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*, 6546–6553. <https://doi.org/10.1609/AAAI.V33I01.33016546>
- Huang, X., Bidart, R., Khetan, A., & Karnin, Z. (2022). Pyramid-BERT: Reducing complexity via successive core-set based token selection. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 1, 8798–8817. <https://doi.org/10.18653/v1/2022.acl-long.602>
- Islam, A., & Inkpen, D. (2008). Semantic text similarity using corpus-based word similarity and string similarity. *ACM Transactions on Knowledge Discovery from Data*, 2(2), 1–25. <https://doi.org/10.1145/1376815.1376819>
- Jayashankar, S., & Sridaran, R. (2017). Superlative model using word cloud for short answers evaluation in eLearning. *Education and Information Technologies*, 22(5), 2383–2402. <https://doi.org/10.1007/s10639-016-9547-0>
- Jordan, S. (2013). E-assessment: Past, present and future. *New Directions*, 9(1), 87–106. <https://doi.org/10.11120/ndir.2013.00009>
- Jordan, S., & Butcher, P. (2013). Does the Sun orbit the Earth? Challenges in using short free-text computer-marked questions. In *HEA STEM Annual Learning and Teaching Conference 2013: Where Practice and Pedagogy Meet*. http://www.heacademy.ac.uk/events/detail/2012/17_18_Apr_HEA_STEM_2013_Conf_Bham. Accessed 1 June 2021.
- Kazemnejad, A., Salehi, M., & Soleymani Baghshah, M. (2020). Paraphrase generation by learning how to edit from samples. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, (pp. 6010–6021). <https://doi.org/10.18653/v1/2020.acl-main.535>
- Khan, S., & Khan, R. A. (2019). Online assessments: Exploring perspectives of university students. *Education and Information Technologies*, 24(1), 661–677. <https://doi.org/10.1007/s10639-018-9797-0>
- Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. <https://doi.org/10.48550/arxiv.1412.6980>. Accessed 20 Feb 2022.
- Kingma, D. P., & Welling, M. (2013). Auto-Encoding Variational Bayes. *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*. <https://arxiv.org/abs/1312.6114v10>
- Kreutzer, J., Caswell, I., Wang, L., Wahab, A., Van Esch, D., Ulzii-Orshikh, N., Tapo, A., Subramani, N., Sokolov, A., Sikasote, C., Setyawan, M., Sarin, S., Samb, S., Sagot, B., Rivera, C., Rios, A., Papadimitriou, I., Osei, S., Suarez, P. O., ... Adeyemi, M. (2022). Quality at a glance: An audit of web-crawled multilingual datasets. *Transactions of the Association for Computational Linguistics*, 10, 50–72. https://doi.org/10.1162/tacl_a_00447
- Kumar, S., Chakrabarti, S., & Roy, S. (2017). Earth mover's distance pooling over siamese LSTMs for Automatic short answer grading. *IJCAI International Joint Conference on Artificial Intelligence*, 0, 2046–2052. <https://doi.org/10.24963/ijcai.2017/284>
- Kumar, A., Ahuja, K., Vadapalli, R., & Talukdar, P. (2020). Syntax-guided controlled generation of paraphrases. *Transactions of the Association for Computational Linguistics*, 8, 330–345. https://doi.org/10.1162/tacl_a_00318
- Kumaran, V. S., & Sankar, A. (2015). Towards an automated system for short-answer assessment using ontology mapping. *International Arab Journal of E-Technology*, 4(1), 17–24. <https://dblp.org/db/>

- journals/iajet/iajet4.html%0A, <http://www.iajet.org/Pages/archive-vol-4.aspx%0A>, <http://www.iajet.org/documents/vol.4/no.1/3.pdf>. Accessed 17 Feb 2022.
- Lai, H., Mao, J., Toral, A., & Nissim, M. (2022). Human Judgement as a Compass to Navigate Automatic Metrics for Formality Transfer. *HumEval 2022 - 2nd Workshop on Human Evaluation of NLP Systems, Proceedings of the Workshop*, 102–115. <https://doi.org/10.18653/v1/2022.humeval-1.9>
- Lavie, A. (2010). Evaluating the output of machine translation systems. In *AMTA 2010 - 9th Conference of the Association for Machine Translation in the Americas*. <https://www.cs.cmu.edu/~alavie/Presemtations/MT-Evaluation-MT-Summit-Tutorial-19Sep11.pdf>. Accessed 3 Mar 2022.
- Lavie, A., & Agarwal, A. (2007). METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation, June* (pp. 228–231). <https://aclanthology.org/W07-0734/>. Accessed 20 Feb 2022.
- Leacock, C., & Chodorow, M. (2003). C-rater: Automated scoring of short-answer questions. *Computers and the Humanities*, 37(4), 389–405. <https://doi.org/10.1023/A:1025779619903>
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2020). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7871–7880. <https://doi.org/10.18653/v1/2020.acl-main.703>
- Marvaniya, S., Foltz, P., Saha, S., Sindhgatta, R., Dhamecha, T. I., & Sengupta, B. (2018). Creating scoring rubric from representative student answers for improved short answer grading. *International Conference on Information and Knowledge Management, Proceedings*, 993–1002. <https://doi.org/10.1145/3269206.3271755>
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*. <https://arxiv.org/abs/1310.4546v1>
- Mohler, M., & Mihalcea, R. (2009). Text-to-text semantic similarity for automatic short answer grading. *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics on - EACL '09*, 567–575. <https://doi.org/10.3115/1609067.1609130>
- Mohler, M., Bunesco, R., & Mihalcea, R. (2011). Learning to Grade Short Answer Questions using Semantic Similarity Measures and Dependency Graph Alignments. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 752–762. [http://ejournal.narotama.ac.id/files/Learning to Grade Short Answer Questions using Semantic Similarity Measures and Dependency Graph Alignments..pdf](http://ejournal.narotama.ac.id/files/Learning%20to%20Grade%20Short%20Answer%20Questions%20using%20Semantic%20Similarity%20Measures%20and%20Dependency%20Graph%20Alignments..pdf)
- Moodle. (2011). *Regular expression short-Answer question type*. https://docs.moodle.org/310/en/Regular_Expression_Short-Answer_question_type. Accessed 27 Dec 2020.
- Nagoudi, E. M. B., Elmadany, A., & Abdul-Mageed, M. (2022). AraT5: Text-to-text transformers for arabic language generation. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 628–647. <https://doi.org/10.18653/v1/2022.acl-long.47>
- Napoles, C., Sakaguchi, K., Post, M., & Tetreault, J. (2015). Ground Truth for Grammaticality Correction Metrics. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 588–593. <https://doi.org/10.3115/v1/p15-2097>
- Noorbehhahani, F., & Kardan, A. A. (2011). The automatic assessment of free text answers using a modified BLEU algorithm. *Computers and Education*, 56(2), 337–345. <https://doi.org/10.1016/j.compedu.2010.07.013>
- Omran, A. M. B., & Ab Aziz, M. J. (2013). Automatic essay grading system for short answers in English language. *Journal of Computer Science*, 9(10), 1369–1382. <https://doi.org/10.3844/jcssp.2013.1369.1382>
- Ott, N., Ziai, R., & Meurers, D. (2012). *Creation and analysis of a reading comprehension exercise corpus* (pp. 47–69). John Benjamins Publishing Company. <https://doi.org/10.1075/hsm.14.05ott>
- Ouahrani, L., & Bennouar, D. (2020). AR-ASAG an Arabic dataset for automatic short answer grading evaluation. In *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)* (pp. 2634–2643). <https://aclanthology.org/2020.lrec-1.321>. Accessed 13 Dec 2021.
- Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, (pp. 311–318). <https://doi.org/10.3115/1073083.1073135>
- Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. In *NAACL HLT 2018 - 2018 Conference of the North American*



- Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, (vol. 1, pp. 2227–2237). <https://doi.org/10.18653/v1/n18-1202>
- Prakash, A., Hasan, S. A., Lee, K., Datla, V., Qadir, A., Liu, J., & Farri, O. (2016). Neural paraphrase generation with stacked residual LSTM networks - ACL anthology. In *Proceedings of {COLING} 2016, the 26th International Conference on Computational Linguistics: Technical Papers* (pp. 2923–2934). <https://aclanthology.org/C16-1275/>. Accessed 19 Feb 2022.
- Pribadi, F. S., Permanasari, A. E., & Adji, T. B. (2018). Short answer scoring system using automatic reference answer generation and geometric average normalized-longest common subsequence (GAN-LCS). *Education and Information Technologies*, 23(6), 2855–2866. <https://doi.org/10.1007/S10639-018-9745-Z>
- Qiu, R. G. (2019). A systemic approach to leveraging student engagement in collaborative learning to improve online engineering education. *International Journal of Technology Enhanced Learning*, 11(1), 1–19. <https://dl.acm.org/doi/10.5555/3302810.3302811>. Accessed 19 Feb 2022.
- Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. *Homology, Homotopy and Applications*, 9(1), 399–438. <https://www.bibsonomy.org/bibtex/273ced32c0d4588eb95b6986dc2c8147c/jonaskaiser>. Accessed 30 May 2023.
- Radford, A., Jeffrey, W., Rewon, C., David, L., Dario, A., & Ilya, S. (2019). Language models are unsupervised multitask learners | enhanced reader. *OpenAI Blog*, 1(8), 9. <https://github.com/codelucas/newspaper>. Accessed 30 May 2023.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2020). Language models are unsupervised multitask learners. *OpenAI Blog*, 1(May), 1–7. <https://github.com/codelucas/newspaper>. Accessed 30 May 2023.
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21, 1–67. <https://doi.org/10.48550/arxiv.1910.10683>
- Ramachandran, L., & Foltz, P. (2015). Generating reference texts for short answer scoring using graph-based summarization. *10th Workshop on Innovative Use of NLP for Building Educational Applications, BEA 2015 at the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2015*, 207–212. <https://doi.org/10.3115/v1/w15-0624>
- Ramachandran, L., Cheng, J., & Foltz, P. (2015). Identifying patterns for short answer scoring using graph-based lexico-semantic text matching. *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, 97–106. <https://doi.org/10.3115/v1/W15-0612>
- Real, R., & Vargas, J. M. (1996). The probabilistic basis of Jaccard's index of similarity. In *Systematic Biology* (Vol. 45, Issue 3, pp. 380–385). Taylor and Francis Inc. <https://doi.org/10.1093/sysbio/45.3.380>
- Rocchio, J. (1971). Relevance feedback in information retrieval. In editor Salton, G. (Ed.), *The Smart Retrieval System - Experiments in Automatic Document Processing* (pp. 313–323). Prentice-Hall, Inc. <https://www.bibsonomy.org/bibtex/1c18d843e34fe4f8bd1d2438227857225/bsmyth>
- Saha, S., Dhamecha, T. I., Marvaniya, S., Sindhgatta, R., & Sengupta, B. (2018). Sentence level or token level features for automatic short answer grading?: use both. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10947 LNAI, 503–517. https://doi.org/10.1007/978-3-319-93843-1_37
- Sakaguchi, K., Heilman, M., & Madnani, N. (2015). Effective feature integration for automated short answer scoring. *NAACL HLT 2015 - 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*.
- Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5), 513–523. [https://doi.org/10.1016/0306-4573\(88\)90021-0](https://doi.org/10.1016/0306-4573(88)90021-0)
- Schneider, J., Richner, R., & Riser, M. (2023). Towards trustworthy AutoGrading of short, multi-lingual, multi-type answers. *International Journal of Artificial Intelligence in Education*, 33(1), 88–118. <https://doi.org/10.1007/s40593-022-00289-z>
- Scikit-learn. (2019). *scikit-learn: machine learning in Python — scikit-learn 0.21.0*. <https://scikit-learn.org/stable/>
- Shermis, M. D. (2015). Contrasting state-of-the-art in the machine scoring of short-form constructed responses. *Educational Assessment*, 20(1), 46–65. <https://doi.org/10.1080/10627197.2015.997617>
- Sukkarieh, J. Z., & Blackmore, J. (2009). c-rater: Automatic Content Scoring for Short Constructed Responses. In *Proceedings of the 22nd International FLAIRS Conference. Association for the Advancement of Artificial Intelligence* (pp. 290–295). https://www.ets.org/research/policy_research_reports/publications/chapter/2009/imsb. Accessed 26 Mar 2022

- Sultan, M. A., Salazar, C., & Sumner, T. (2016). Fast and easy short answer grading with high accuracy. *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference*, 1070–1075. <https://doi.org/10.18653/v1/n16-1123>
- Sun, J., Ma, X., & Peng, N. (2021). AESOP: Paraphrase generation with adaptive syntactic control. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 5176–5189. <https://doi.org/10.18653/v1/2021.emnlp-main.420>
- Sychev, O., Anikin, A., & Prokudin, A. (2020). Automatic grading and hinting in open-ended text questions. *Cognitive Systems Research*, 59, 264–272. <https://doi.org/10.1016/j.cogsys.2019.09.025>
- Toutanova, K., Klein, D., Manning, C. D., & Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2003*, 252–259. <https://doi.org/10.3115/1073445.1073478>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 2017-December, 5999–6009.
- Vijayakumar, A. K., Cogswell, M., Selvaraju, R. R., Sun, Q., Lee, S., Crandall, D., D. B. (2018). Diverse beam search: decoding diverse solutions from neural sequence models. *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, 7371–7379.
- Whitelock, D., & Bektik, D. (2018). Progress and Challenges for Automated Scoring and Feedback Systems for Large-Scale Assessments (pp. 1–18). https://doi.org/10.1007/978-3-319-53803-7_39-1
- Wubben, S., van den Bosch, A., & Krahmer, E. (2010). Paraphrase generation as monolingual translation: Data and evaluation. In *Belgian/Netherlands Artificial Intelligence Conference*. <http://ilk.uvt.nl/>. Accessed 22 Feb 2022.
- Xu, P., Kumar, D., Yang, W., Zi, W., Tang, K., Huang, C., Cheung, J.C.K., Prince, S.J.D., Cao, Y., 2021. Optimizing deeper transformers on small datasets, in: ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference. Association for Computational Linguistics (ACL), pp. 2089–2102. <https://doi.org/10.18653/v1/2021.acl-long.163>
- Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., & Raffel, C. (2021). mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. 483–498. <https://doi.org/10.18653/v1/2021.naacl-main.41>
- Yang, Q., Huo, Z., Shen, D., Cheng, Y., Wang, W., Wang, G., & Carin, L. (2020). An end-to-end generative architecture for paraphrase generation. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 3132–3142. <https://doi.org/10.18653/v1/d19-1309>
- Zahran, M. A., Magoooda, A., Mahgoub, A. Y., Raafat, H., Rashwan, M., & Atyia, A. (2015). Word Representations in Vector Space and their Applications for Arabic. In A. Gelbukh (Ed.) (Ed.), *16th international conference, CICLing 2015 Cairo, Egypt, april 14* (Vol. 9041, Issue April, pp. 430–443). Springer International Publishing Switzerland. https://doi.org/10.1007/978-3-319-18111-0_32
- Zeng, D., Zhang, H., Xiang, L., Wang, J., & Ji, G. (2019). User-oriented paraphrase generation with keywords controlled network. *IEEE Access*, 7, 80542–80551. <https://doi.org/10.1109/ACCESS.2019.2923057>
- Zhao, J., Zhu, T., & Lan, M. (2014). ECNU: One Stone Two Birds: Ensemble of Heterogenous Measures for Semantic Relatedness and Textual Entailment. 271–277. <https://doi.org/10.3115/v1/s14-2044>
- Ziai, R., Ott, N., & Meurers, D. (2012). Short Answer Assessment : Establishing Links Between Research Strands. *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP. Association for Computational Linguistics*, 2(2005), 190–200.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Authors and Affiliations

Leila Ouahrani¹  · Djamel Bennouar¹ 

✉ Leila Ouahrani
l.ouahrani@univ-bouira.dz

Djamal Bennouar
djamal.bennouar@univ-bouira.dz

¹ LIM Laboratory, Computer Science Department, Faculty of Applied Sciences, Bouira University, 10000 Bouira, Algeria