Expert Systems With Applications 208 (2022) 118258

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

End-to-End generation of Multiple-Choice questions using Text-to-Text transfer Transformer models

Ricardo Rodriguez-Torrealba, Eva Garcia-Lopez, Antonio Garcia-Cabot *

Departamento de Ciencias de la Computación, Universidad de Alcalá, Alcalá de Henares (Madrid) 28801, Spain

ARTICLE INFO

Keywords:
Multiple-Choice Question Generation
Distractor Generation
Question Answering
Question Generation
Reading Comprehension

ABSTRACT

The increasing worldwide adoption of e-learning tools and widespread increase of online education has brought multiple challenges, including the ability of generating assessments at the scale and speed demanded by this environment. In this sense, recent advances in language models and architectures like the Transformer, provide opportunities to explore how to assist educators in these tasks. This study focuses on using neural language models for the generation of questionnaires composed of multiple-choice questions, based on English Wikipedia articles as input. The problem is addressed using three dimensions: Question Generation (QG), Question Answering (QA), and Distractor Generation (DG). A processing pipeline based on pre-trained T5 language models is designed and a REST API is implemented for its use. The DG task is defined using a Text-To-Text format and a T5 model is fine-tuned on the DG-RACE dataset, showing an improvement to ROUGE-L metric compared to the reference for the dataset. A discussion about the lack of an adequate metric for DG is presented and the cosine similarity using word embeddings is considered as a complement. Questionnaires are evaluated by human experts reporting that questions and options are generally well formed, however, they are more oriented to measuring retention than comprehension.

1. Introduction

The COVID-19 outbreak at the end of 2019 caused the global suspension of school activities, affecting 91 % of the world's student population (UNESCO, 2020), which turned educators' attention to e-learning tools as alternatives to continue the delivery of educational content, assignments, and assessments to the students. However, online education was already on track for a massive adoption by 2025, making it more feasible thanks to the advances and availability of information, technology and communication (Palvia et al., 2018). The increasing worldwide penetration of e-learning and Massive Open Online Courses (MOOC) bring challenges (Olivares Olivares, Hernández, Corolla, Alvarez, & Sánchez-Mendiola, 2021), for example, for measuring effectiveness of learning. This involves everything from discovering relevant key metrics and indicators for this e-environment (Learning Analytics) (UNESCO, 2019), to generating required assessments, grades and exercises for students. Teachers and instructors, who normally invest a lot of time authoring and updating assessments (Heilman, 2011), are now made virtually impossible to adapt these instruments to the scale and speed demanded by the massification of online education, making it even more difficult when considering a personalized environment that maximizes the potential of each student (Holmes, Bialik, & Fadel, 2019).

Under these circumstances, evolution of Artificial Intelligence (AI) tools and techniques provide opportunities to automate and assist educators in routine tasks so they can spend more time with the students, motivating them, and sharing their knowledge (UNESCO, 2019). Thanks to new and larger datasets, such as the Colossal Clean Crawled Corpus (C4) (Raffel et al., 2019); to advances made in neural network architectures for natural language processing (NLP), as Transformer (Vaswani et al., 2017); to transfer learning through vector representation of words (embeddings) (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013; Pennington, Socher, & Manning, 2014) to the use of new neural language models like BERT (Devlin, Chang, Lee, & Toutanova, 2018), GPT-2 (Radford Alec et al., 2019), and to the enormous GPT-3 (Brown et al., 2020) with 175 billion parameters, the NLP community has reached new milestones in topics such as question generation (QG), question answering (QA), and natural language generation (NLG).

This research focuses on using Transformer-based language models for automating the generation of multiple-choice questions (MCQs),

* Corresponding author.
E-mail addresses: ricardo.rodriguezt@edu.uah.es (R. Rodriguez-Torrealba), eva.garcial@uah.es (E. Garcia-Lopez), a.garciac@uah.es (A. Garcia-Cabot).

https://doi.org/10.1016/j.eswa.2022.118258
Received 28 December 2021; Received in revised form 10 June 2022; Accepted 20 July 2022
Available online 22 July 2022

with the purpose of helping or assisting educators in the process of creating reading comprehension (RC) assessments. This is relevant and timely because teachers could invest less time doing routine work and share more time with their students, building engaging experiences for face-to-face classroom interactions. Consequently, this would also reduce one of the students concerns related to the e-learning environment: isolation from teachers and peers (Palvia et al., 2018).

This study addresses the problem of generating questionnaires of MCQs from 3 points of view: QG, QA, and distractor generation (DG). An end-to-end pipeline for generating MCQs is proposed, based on pre-trained T5 language models (Raffel et al., 2019), fine-tuned to perform QG and QA tasks. Also, the T5 model (a small version of 60 million parameters) is fine-tuned for DG's task during this work. The problem of distractor generation is formulated using a Text-To-Text approach, in which the T5 language model is trained on a new task called *generate distractor* that uses the question, correct answer, and context text as input, then is asked to generate the corresponding distractor text as output. A simple API and a demo web application for visualizing generated MCQs were developed, given an English Wikipedia URL as input. MCQs were validated with users, who were shown a selection of generated MCQs, using a 5-paragraph survey, each one accompanied by 2 questions and 2 items to grade the questionnaire.

In summary, main contributions of this research are: (1) A processing pipeline that allows the end-to-end generation of MCQs using fine-tuned T5 models; (2) An alternative approach to DG using a Text-To-Text paradigm; (3) Our approach based on T5 models confirms that Text-To-Text paradigm can be applied to multiple types of tasks and get competitive results; (4) The fine-tuned model for DG, training code and the implementation of the end-to-end generation of MCQs.

The rest of this paper is divided into 5 parts. Section 2 summarizes the state of the art around latest neural language models and their applicability in tasks like QG, QA and DG, analyzing available implementations as well. Section 3, Materials and Methods, describes how the end-to-end pipeline was implemented, by describing the architecture based on T5 models. Next, experimental data are presented in Section 4, and later we reflect on them and discuss our approach in Section 5. Finally, Section 6 summarizes the contributions and proposes directions for future work.

2. Background

In recent years, we have seen how tasks related to NLP and natural language understanding (NLU) have evolved from rule-based implementations, state machines, and hidden Markov models (Jurafsky & Martin, 2009), to an ecosystem dominated by neural language models. Since the formulation of the Transformer (Vaswani et al., 2017), the community of NLP researchers has turned to its use given its proven effectiveness (Devlin et al., 2018; Radford Alec et al., 2019). This has allowed training larger models with greater efficiency (Vaswani et al., 2017) that lead to establishing new performance parameters and surpassing humans in some cases, for example, in RC benchmarks such as SQuAD (Raffel et al., 2019; Rajpurkar, Zhang, Lopyrev, & Liang, 2016). The Transformer is becoming the most common architecture used for developing and training language models (Raffel et al., 2019). GPT-2, BERT, ALBERT (Lan et al., 2019), RoBERTa (Liu et al., 2019) and T5 are some of these language models that are also available as pre-trained versions in different variants, each of them with outstanding performance in tasks such as QA.

The GPT-3 model has reached new limits on performance and language model dimensions (175B parameters) although its use requires an external API. However, its smaller "brother" GPT-2 is available in different pre-trained variants, and it is characterized by being a self-regressive and one-way language model. Since GPT-2 is able to generate text in a "creative" way, given a prompt as reference, its use in DG tasks (with and without fine-tuning) has been subject of research (Von Davier, 2019). BERT model is characterized by being bidirectional and is considered to be a masked language model. Its design has inspired the development of other models, but also the analysis of the entire ecosystem of NLP around the Transformer and transfer learning (Raffel et al., 2019). The T5 model emerged from it and was trained on the C4 dataset, consisting of 750 GB of text. It is a multitask model with a unified Text-To-Text approach for specifying NLP problems. T5 model has been successfully fine-tuned using a multitasking approach for QG and QA simultaneously, also, the source code[1] and different pre-trained versions are available through the Python Transformers library (Wolf et al., 2019). Given the QA task is one of the main topics of research in NLP and it is one of the key performance metrics (i.e. SQuAD EM and F1 scores), most of the models mentioned so far are usually trained on SQuAD and perform very well on QA, but we need to consider that QG and DG are not mainstream.

This research focuses on the use of neural language models, specifically those based on the Transformer architecture, however, it is important to mention that rule-based solutions for QG have been developed (Das, Ray, Mondal, & Das, 2016) using frameworks for syntax and feature recognition, also the source code for some of them is public[2,3] On the implementation side using neural language models for QG, several researchers have published their proposed solutions and codebase, such as DAANet (Dual Ask-Answer Network) (Xiao, Wang, Yan, & Zheng, 2018), QG-Net which is trained on the SQuAD dataset to generate questions for educational content (Wang et al., 2018) and the research to evaluate the reward scheme in QG models (Hosking & Riedel, 2019) that implements a QG neural model based on the Text-To-Text paradigm (Yuan et al., 2017). QG can be seen as a reverse task to QA (Zhou et al., 2018) and similar to the QA ecosystem. The creation of the SQuAD dataset has contributed to the development of this area (Du, Shao, & Cardie, 2017). Several studies have proposed unified architectures to simultaneously train a model capable of performing both tasks (Tang, Duan, Qin, Yan, & Zhou, 2017; Xiao et al., 2018), or a joint model based on GPT-2 and BERT (T. Klein & Nabi, 2019) that uses the generative capabilities of GPT-2 for QG and a BERT model fine-tuned for QA.

Distractors are a key component of MCQs and they represent one of the tasks that require more effort in the development of such assessment tools given the number of elements to be generated (Shin, Guo, & Gierl, 2019). In addition to this, various strategies are involved in the process, such as the identification by experts of common reasoning errors, thinking and solving problems associated with a particular area (Haladyna & Rodriguez, 2013), and in the end, this is not always effective in the formulation of such distractors. In this sense, when generating distractors for MCQs, different factors such as grammar correspondence with the context and consistency between distractors and the correct answer must be considered (Liang et al., 2018). Traditional automatic DG implementations revolve around similarity calculations between the distractor, the correct response and/or the context, using approaches based on embeddings (Jiang & Lee, 2018), lexical databases such as WordNet (Miller, 1995), ontologies, thesauruses, among others, which subsequently lead to a selection of candidate elements based on some ranking strategy (Liang et al., 2018). However, studies using neural language models have recently been published: using the QA capabilities of a generative model such as GPT-2 (Von Davier, 2019) and optionally applying fine-tuning for a specific context.

Another study proposes a seq2seq model that incorporates static and dynamic attentional structures (Gao, Bing, Li, King, & Lyu, 2019), which is trained using GloVe embeddings and a dataset adapted from RACE dataset (Lai, Xie, Liu, Yang, & Hovy, 2017) (DG-RACE), where irrelevant distractors are removed by applying POS-tagging and frequency analysis of the words found in the distractor versus their context article. The

[1] https://github.com/patil-suraj/question_generation.
[2] https://github.com/dipta1010/Automatic-Question-Generator.
[3] https://github.com/hemantpugaliya/Automatic-Question-Answer-Generation.

source code (but not the pre-trained model) has been published as well as the generated dataset.[4] Model implementation is Python-based using OpenNMT-py (G. Klein et al., 2017). As mentioned before, right now, DG is not a hot topic for research when compared to QA, so the existence of pre-trained neural language models is limited, as well as datasets oriented to this specific task.

This study proposes an end-to-end pipeline for generating MCQs based on T5 language models, exploring their Text-to-Text approach for generating distractors in addition to the already known applications in QG and QA.

3. Materials and methods

The Transformer architecture has been gaining considerable adoption in the world of NLP. Unlike sequence-to-sequence architectures with attention mechanisms, where the computing capacity required to train them increases given the sequential nature of the setup, the Transformer is an encoder-decoder architecture based mainly on attentional structures, allowing to remove such recurring elements (Vaswani et al., 2017). Thus, parallelization of training can be achieved, allowing to generate much larger models with lower cost and shorter execution time thanks to reduced computational complexity.

In this architecture, both the encoder and the decoder are, in fact, multiple layers of encoders and decoders. Encoders are quite simple in terms of being made up of only one attention layer and one feed-forward network (which is the same on all encoders). On the other hand, decoders are similar to encoders, except that they include an extra layer of attention that aims to help the decoder to observe the elements coming from the input that are relevant to it (Vaswani et al., 2017). This mechanism of attention is called "self-attention" (Vaswani et al., 2017), and is what allows the encoder, for example, to learn how to associate in the sentence "The dog didn't catch the ball because it was distracted", the pronoun "it" refers to the dog and not to the ball. Some language models based on this architecture use only parts of it, for example GPT-2, uses only decoder blocks from the Transformer architecture (Radford Alec et al., 2019), however, BERT is based only on encoders (Devlin et al., 2018).

One of the problems in machine learning is the need to fully retrain models when new data is presented, because it is assumed as a good practice that training and test datasets must have the same distribution and use the same feature extraction process, but under real circumstances, this would be extremely costly to implement (Pan & Yang, 2010). Under ideal circumstances, this should not be necessary, if knowledge learned by a model about a domain could be reused (transferred) to another model to perform other tasks (not necessarily in the same domain). In this sense, Transfer Learning is a set of techniques that allow to transfer knowledge from a source domain to a target domain, relaxing the hypothesis that the training and test data set should be independent and have a similar distribution (Tan et al., 2018). In the case of Transfer Learning applied to NLP tasks, some reusable learned representations for word embedding exist, such as GloVe (Pennington et al., 2014) and Word2Vec (Mikolov, Chen, Corrado, & Dean, 2013). Similarly, models such as BERT or T5, which are trained using a very large corpus, constitute networks of millions of parameters, which can be used as a basis for tuning (fine-tune) another task-oriented model (Dodge et al., 2020).

Finding new ways to represent these words in a vector space and better encode meaning and context is an active area of research, as it is not only a key component in today's NLU models, but it has been shown that a good representation of these word vectors has led to the best results in tasks such as QA and semantic analysis (Peters et al., 2018). Methods like GloVe are widely used, not only because of their good results, but because you can easily access a set of pre-trained vectors with different characteristics (such as dimensionality), ready to be used in different problem domains. While word vectors are an important component in NLP-oriented neural network architectures, they can also be used outside that context, such as to compare text sentences by computing the similarity of words using cosine similarity of word vectors (Pennington et al., 2014).

As mentioned before, T5 model uses the Transformer architecture including all its elements (encoder and decoder) and has a "Text-to-Text" interface. For the purposes of this study, T5 model characteristics are interesting. First, it has been pre-trained in multiple tasks, including QA, and second, it can be easily trained to perform new tasks, such as QG and DG, using the same "Text-to-Text" interface. This allows to simplify the training and fine-tuning process, but also takes advantage of transfer learning of language modeling and representation. In this sense, the pipeline proposed for generating end-to-end MCQs relies on T5 models.

Following subsections present the pipeline for MCQs generation and explain the implementation of each step.

3.1. Processing pipeline

The proposal is a sequential processing based on T5 language models for generating MCQs from paragraphs extracted from Wikipedia articles. This pipeline consists of five processes or steps and considers the three dimensions previously described: QG, QA and DG.

Fig. 1 shows these numbered steps. In step 1 (Parsing), the system parses a Wikipedia article and generates a list of paragraphs. In step 2 (Generate QA Pairs), it takes a paragraph and generates pairs of question/correct answer using the QAPModel. Step 3 (Generate Distractors) uses the paragraph as context and takes a question/correct answer pair for generating distractors using the DGModel. In step 4 (Compute Similarity), the system computes the similarity of each distractor using word vectors based on the correct answer as a reference. Finally, in step 5 (Prepare Questionnaire), MCQ is prepared and formated, then steps 3 to 5 are repeated.

3.2. Parsing

The first process in the pipeline is called "Parsing", and it is responsible for generating a list of paragraphs when given a reference to a Wikipedia article. Each of the obtained paragraphs can be selected as an input to the next step for generating a questionnaire of MCQs.

3.3. QA pairs generation

The next element in the processing sequence is responsible for generating pairs of questions with its correct answer using one of the paragraphs obtained in the previous step. The output of this process may contain several of these described pairs, as multiple questions can be derived from a paragraph of information. In order to generate these pairs, a model that has been called QAPModel (Question/Answer Pairs Model) is used, taking advantage of the duality of QG and QA (Klein & Nabi, 2019; Tang et al., 2017; Xiao et al., 2018). The implementation consists of a set of pre-trained T5 models on QG and QA tasks in conjunction with a processing pipeline (Fig. 2), based on the open-source development for QG published on GitHub (Patil, 2020), by using the Transformers Python library(Wolf et al., 2019). This pipeline represents an answer aware question generation method, in which the model uses a context and annotated answers (Fig. 3) in that context to generate the corresponding questions (Patil, 2020) (Chan & Fan, 2019).

3.4. Distractors generation

The third process called "Generate Distractors", progressively takes each of the questions and correct answer pairs from the previous step and along with the paragraph (information context), generates a list of incorrect answers (distractors). For this task, a model called DGModel is

[4] https://github.com/Yifan-Gao/Distractor-Generation-RACE.

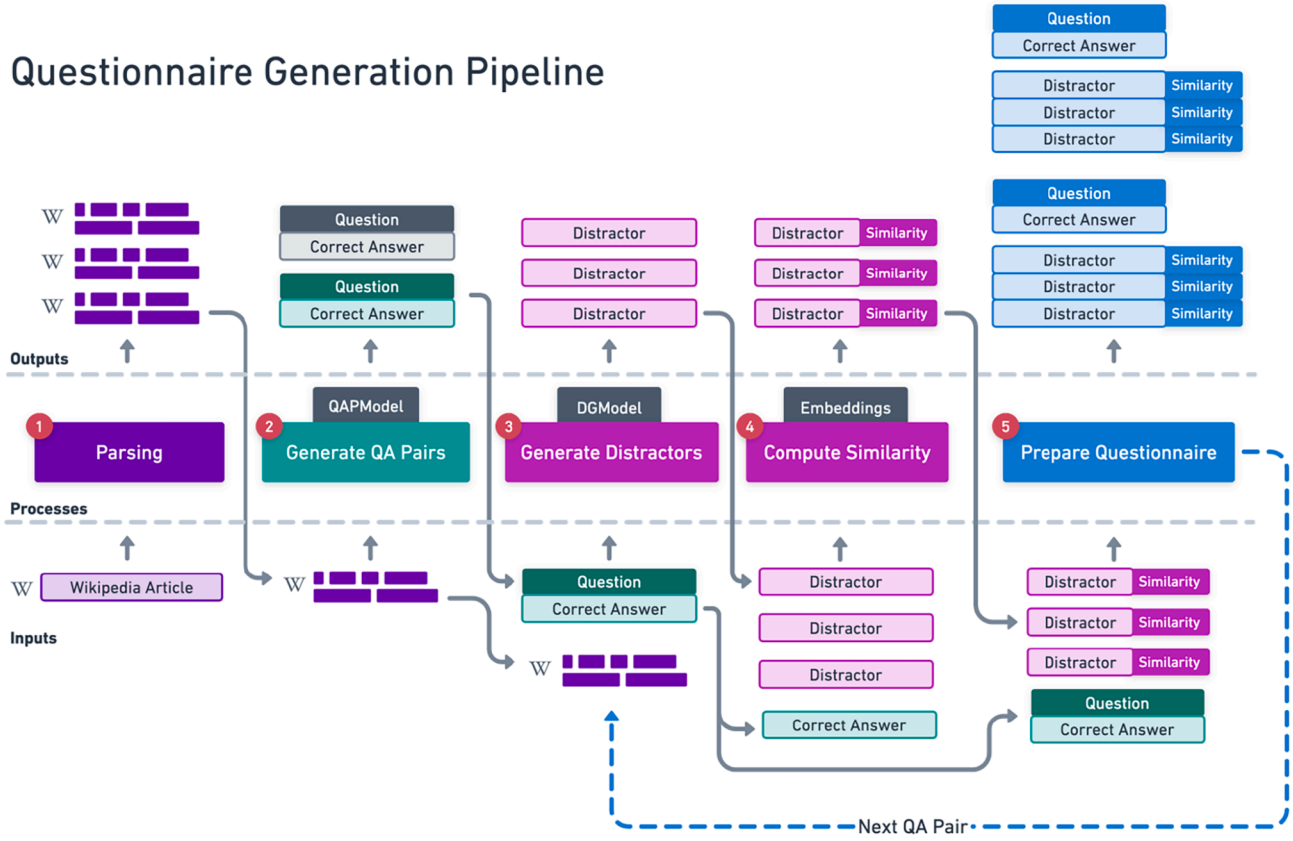

Fig. 1. Pipeline for generating questionnaires composed of MCQs from Wikipedia paragraphs. Multiple language models are used for QG, QA and DG tasks.

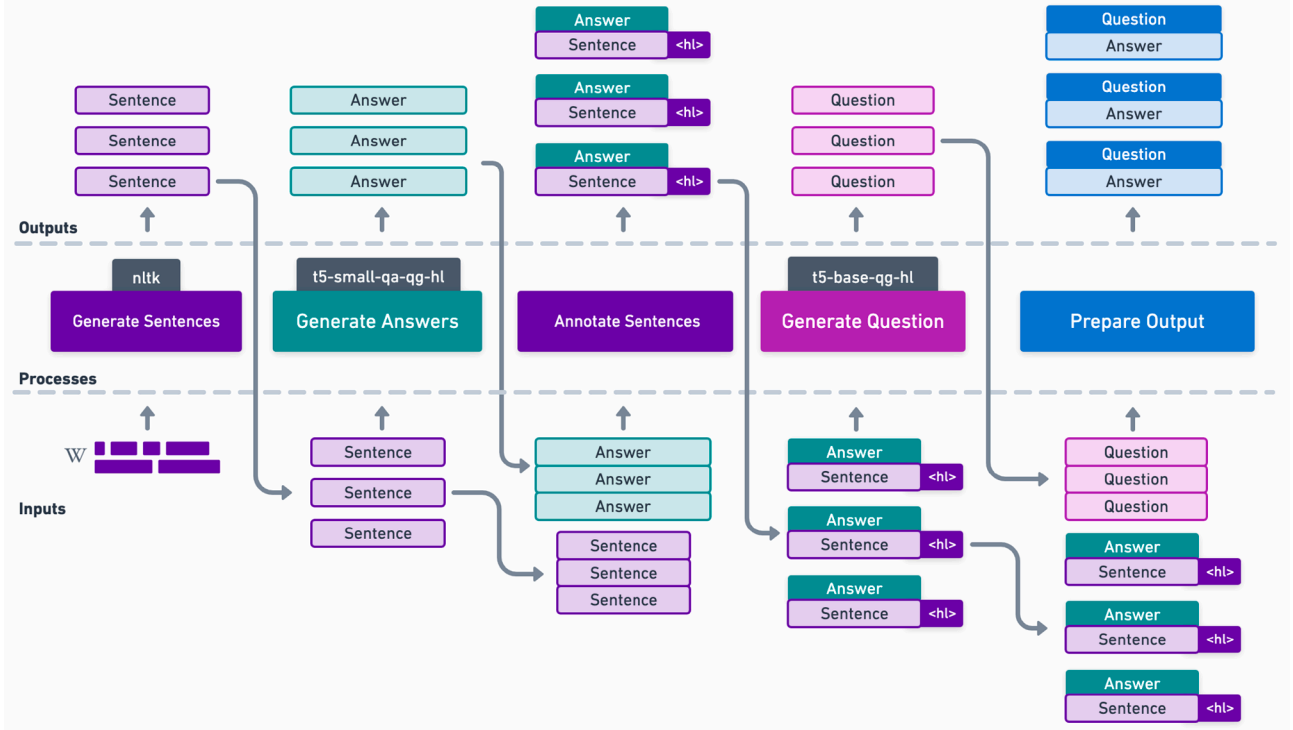

Fig. 2. QAPModel pipeline, based on (Patil, 2020). First, phrases are extracted from the input paragraph, then a fine-tuned T5 model extracts answer spans used for annotating the phrases, next, a fine-tuned T5 on QG produces questions based on annotated phrases and finally the output is structured using a JSON format. Pretrained T5 Models in use: t5-small-qa-qg-hl (60 M parameters), t5-base-qh-hl (220 M parameters).

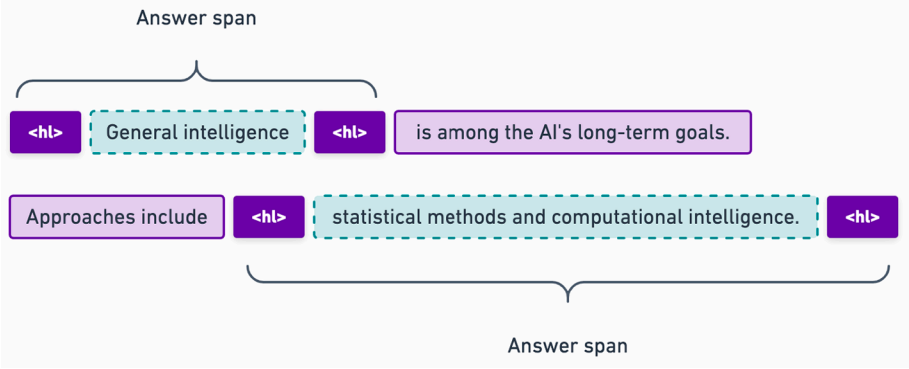

Fig. 3. Example of annotated sentences. Answers are wrapped with a highlight tag.

proposed, which is based on a single T5 language model, fine-tuned to transform the correct answer into an incorrect one using the associated question and context. Unlike the QAPModel, which depends on previously fine-tuned and published models (Patil, 2020), the T5-small model (60 M parameters) is fine-tuned for DG as part of this research. One of the challenges of training a distractor generation model is the dataset to use, for this purpose, we use DG-RACE (Gao et al., 2019). This dataset consists of 120,874 observations, for a total of 25,207 articles (context) divided into training (80 %), test (10 %) and dev (10 %) sets (Gao et al., 2019). Fig. 4 shows a training example from the dataset.

In order to fine-tune the T5 model in the distractor generation task, the input and output must be formulated in a Text-To-Text format, so the input text acts as a context or conditioning element and the model must try to match the expected output (Raffel et al., 2019).

Each observation is formatted as shown in Fig. 5, the prefix *generate distractor* has been added to identify the task to perform, which precedes the *question*, *answer*, and *context* labeling, ending with the </s>delimiter.

3.5. Distractors score

Once the distractors have been obtained, the process "Calculate Similarity" is performed. This step has the goal of obtaining a score that can be used as a reference of the distractor quality. The calculation is

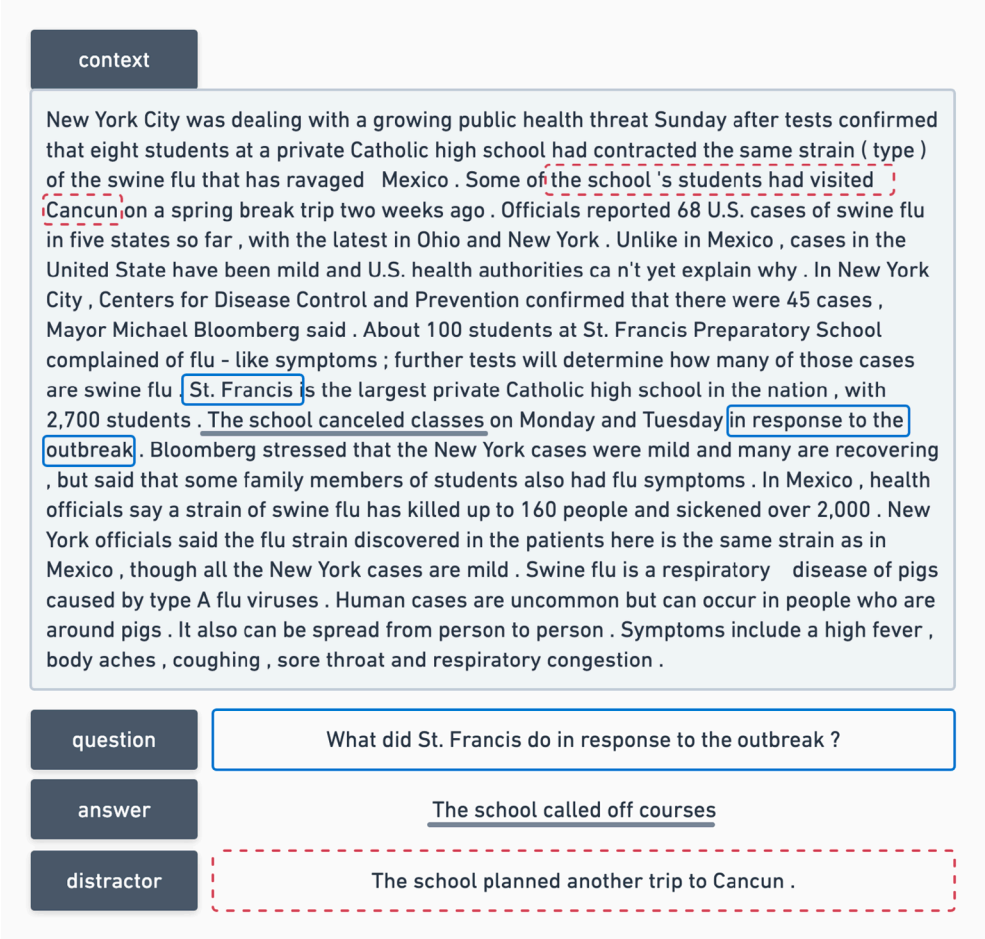

Fig. 4. Training example from DG dataset. The key *context* is the article used as a context for the *question* and the correct *answer*, while *distractor* is an incorrect answer that may or may not be extracted from the context. In the image, text spans corresponding to each key are highlighted (using the same color of the key).

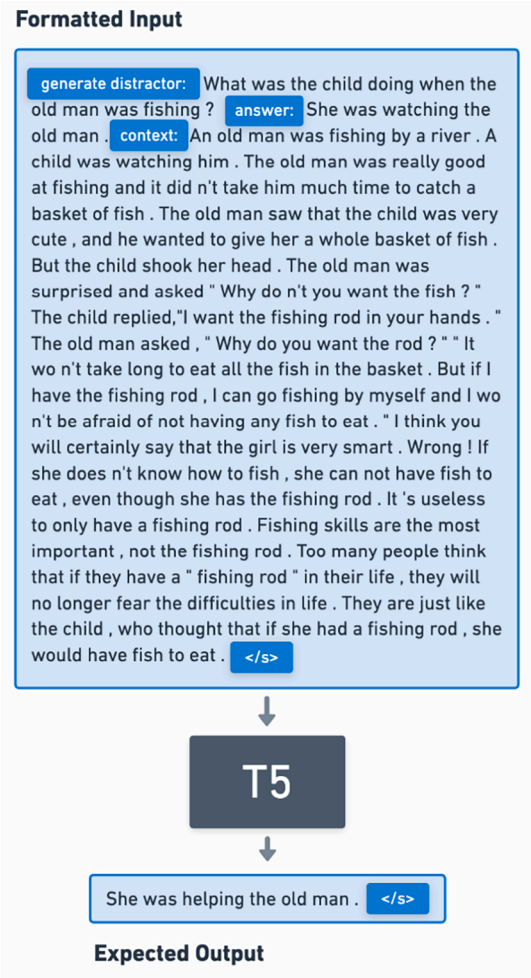

Fig. 5. Text-To-Text formulation for fine-tuning the T5 language model on QG. Note that the correct answer and context are separated by "answer:" and "context:" tokens respectively, also, the prefix *generate distractor* identifies the task.

based on the cosine similarity between the word vectors of the distractor (which is an incorrect answer) and the correct answer, with stop words removed. The output of this step is a collection of distractors and their scores.

3.6. Preparing the output

Finally, the process "Prepare Questionnaire" is responsible for coordinating and collecting the output of each step, following the QA pairs obtained from step 2, then takes care of formatting the questionnaire of MCQs and generating the final output in a structured JSON which includes a list of questions for a given paragraph, each question accompanied by a collection of answers identified as the correct or not and including the similarity score (Fig. 6).

4. Results

The DGModel was implemented using the Transfromers Python library, which facilitates the access to pre-trained T5 models and fine-tuning methods. The full implementation of the DG model (as described in the Material and Methods section) has been published on Github.[5] Next, the full proposed MCQs pipeline was built[6][7] on top of the models and a demo app was developed[8] for making easier the exploration of questionnaires (Fig. 7).

4.1. DG implementation details

Implementation was based on the T5ForConditionalGeneration model included in the Transformer Python library, which includes the language modeling head allowing the use of text-to-text generation tasks and the tokenization is the same as the pretrained T5 model.

4.1.1. DGModel params

Fine-tune process used the same base parameters of the small T5 model, which has the number of encoder and decoder blocks set to 6, each block having 512 layers (d_{model}), the feed-forward network dimensionality set to 2048 (d_{ff}) and 8 heads per attention mechanism, with the key-value matrices size set to 64 (d_{kv}) (Raffel et al., 2019). Similarly, regularization params are kept, having epsilon set to 1e-06 and dropout probability set to 0.1.

Regarding text generation, for evaluating the model, beam search (Vijayakumar et al., 2016) with a configuration of 3 beams is used, avoiding repetitions 2-words sequences (no_repeat_ngram_size = 2) and early stopping enabled (finish the search beam hypothesis after finding an end-of-sequence token). For inference time, however, a more creative model is used, relying on top-k and top-p sampling (Fan, Lewis, & Dauphin, 2018) (Holtzman, Buys, Du, Forbes, & Choi, 2019), with k = 120 and p = 0.98.

4.1.2. Training params

Adam optimizer (Kingma & Ba, 2015) without weight decay is used for fine-tuning the model during 2 epochs, with a learning rate of 3e-4 and epsilon set to 1e-8. The training batch size is 6 with gradient accumulation steps set to 16.

4.2. Validation

Previously, it was discussed that the problem of automatic questionnaire generation can be addressed from 3 dimensions, QG, QA and DG, this also leads to the automatic evaluation of each dimension independently to understand the performance of each component within the proposed pipeline (Fig. 1). However, through this perspective, there are no insights of the full performance of a questionnaire, specifically, how these 3 components successfully connect to each other to generate a useful MCQ that can be used as a source for educators when creating assessments. Then, in this regard, human judgment is ideal.

4.2.1. Automatic validation

The automatic evaluation in this work is focused on the DG task because for generating the question and correct answer, pre-trained and fine-tuned models are used, so their metrics are already known (Patil, 2020). Taking into consideration that we are measuring the performance of a text-generation system, BLEU (from 1 to 4n-grams) (Papineni, Roukos, Ward, & Zhu, 2002) and ROGUE-L (based on the longest common sequence present on the text) (Lin, 2004) are used as metrics. These metrics are usually used in Machine Translation (MT) tasks for measuring the quality of the outputs from a scale ranging between 0 and 1, being 1 an exact match. Cosine similarity based on GloVe embeddings is also used for having a reference of the semantic distance of the generated distractors to the reference (Pennington et al., 2014).

The DG model is evaluated using the DG-RACE dataset, whose test

[5] https://github.com/rrodrigu3z/t5-distractors.
[6] https://github.com/rrodrigu3z/questionnaire-generator-models.
[7] https://github.com/rrodrigu3z/questionnaire-generator-api.
[8] https://github.com/rrodrigu3z/questionnaire-generator-demo.

```
{
  "data": [
    {
      "question": "What is one of the long-term goals of AI",
      "answers": [
        {
          "answer": "Digital intelligence",
          "correct": false,
          "similarity": 0.743
        },
        {
          "answer": "Social analysis",
          "correct": false,
          "similarity": 0.633
        },
        {
          "answer": "Cognitive intelligence",
          "correct": false,
          "similarity": 0.791
        },
        {
          "answer": "General intelligence",
          "correct": true
        }
      ]
    },
    {
      "question": "What approaches are used in AI research?",
      "answers": [
        {
          "answer": "statistical methods, computational intelligence, and traditional symbolic AI",
          "correct": true
        },
        {
          "answer": "general intelligence, artificial neural networks, and traditional symbolic AI",
          "correct": false,
          "similarity": 0.908
        },
        {
          "answer": "search and mathematical optimization, artificial neural networks, and traditional symbolic AI",
          "correct": false,
          "similarity": 0.902
        },
        {
          "answer": "natural language processing, perception and the ability to move and manipulate objects",
          "correct": false,
          "similarity": 0.77
        }
      ]
    }
  ]
}
```


Fig. 6. Example of the output of "Prepare Questionnaire" process, formatted as JSON.

set is composed of 12,284 observations, also, we use their proposed model as the baseline. With the help of the nlg-eval tool (Sharma, Asri, Schulz, & Zumer, 2017), the ground truth is compared against three sets of distractors generated by the fine-tuned T5 model. Table 1 shows a summary of the results.

4.2.2. Human evaluation

The approach was focused on obtaining the opinion of evaluators on a set of MCQs, which were automatically generated by the implemented pipeline. The main objective was to get feedback about how easy or difficult was to answer the questions presented and assess they were well formed.

For these purposes, a survey was designed, consisting of five paragraphs of text (presented one at a time), each one accompanied by two MCQs related to the text and two items for assessing the MCQs. The first one required the user to answer was "*How easy or difficult was to answer these questions?*" by using a 5-point Likert scale representing difficulty levels: 1-*Very Easy, 2-Easy, 3-Moderate, 4-Difficult, and 5-Very Difficult*. The second item the user had to respond was "*How well formed do you consider these questions and answers to be? Consider aspects such as spelling, syntax, clarity and meaning*". This question had to be answered, again, by using a 5-point Likert scale representing quality levels, which in this case were: 1-*Very Poor, 2-Poor, 3-Moderate, 4-Good, and 5-Very Good* (Fig. 8).

Paragraphs included in the survey were extracted from English Wikipedia articles, and MCQs were generated using the demo app. Text

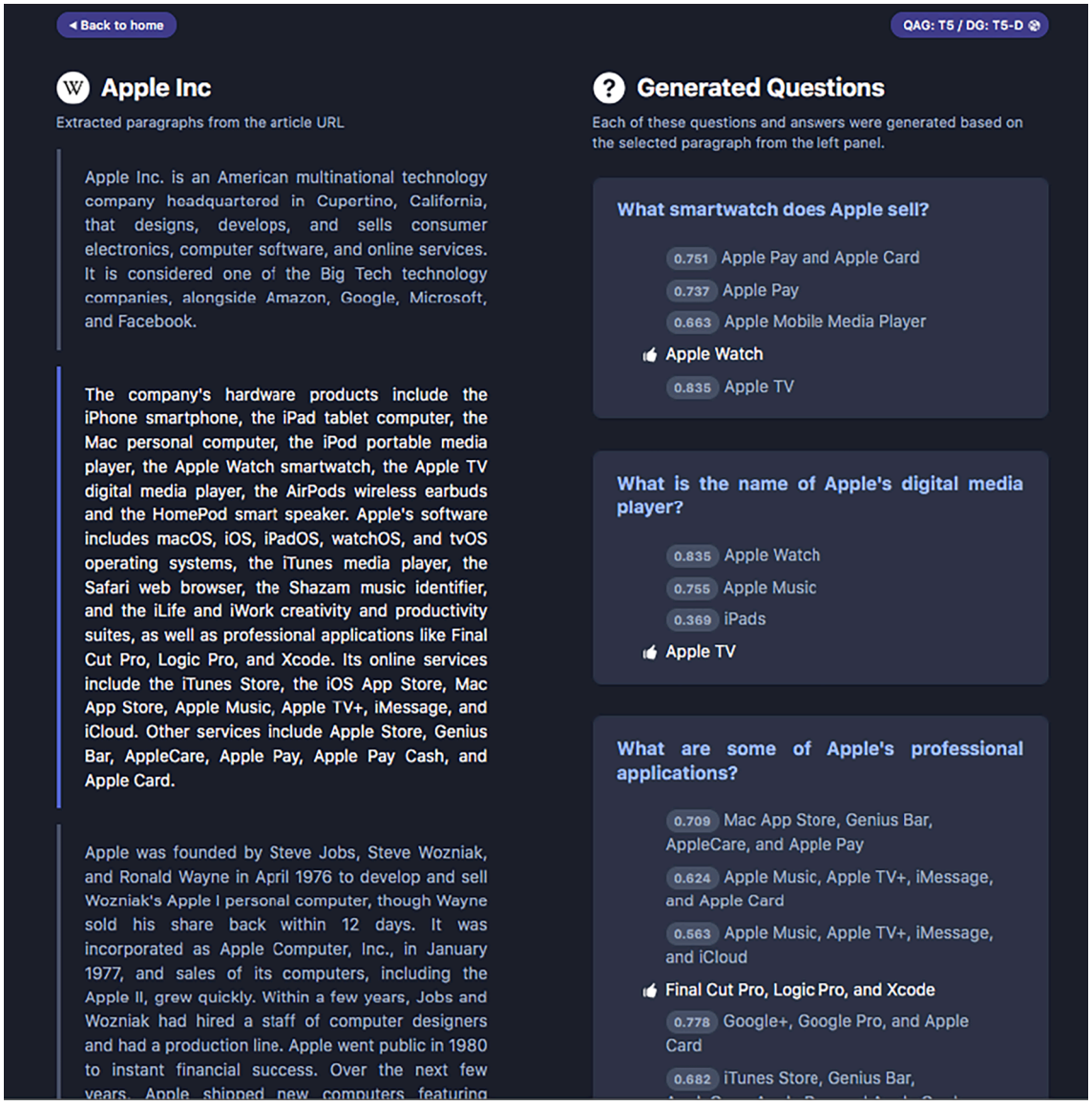

Fig. 7. Demo app developed for exploring the questionnaires. Paragraphs from the Wikipedia page are listed on the left pane, after selecting one of the paragraphs, the generated MCQs are generated and displayed on the right pane.

Table 1
Automatic evaluation results of the T5 model for DG. Higher values are better. * Cosine similarity of the distractor based on GloVe embeddings.

Description	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	CS*
Distractor 1	14.91	7.14	3.81	2.19	15.14	0.73
Distractor 2	15.02	7.18	3.80	2.21	15.31	0.72
Distractor 3	14.47	6.87	3.65	2.09	14.27	0.69
Avg. Distractor	14.80	7.06	3.75	2.16	**14.91**	0.71
Baseline dg-race	26.93	13.57	8.0	5.21	14.54	–

of generated questions, correct answer and distractors were not altered, however, the order of the correct answer and distractors from the pipeline output were randomized. Also, the most convenient distractors were selected from the list displayed by the demo application, given it could generate more than the three distractors needed for the MCQs.

This survey was taken by 17 professionals with university degree. All of them, with English language proficiency: 53 % of them have formal English language studies (degrees or language certifications) or they are native English speakers. The completion rate obtained was 82 % (14 people), with a typical time spent of 13 min with 40 s. Table 2 summarizes the results regarding the perceived difficulty form the evaluators for answering the presented MCQs and Table 3 shows the results for the perceived quality.

In addition, post-survey interviews with a subset of the participants were conducted. The idea was to get more details about their appreciation in an open conversation. They expressed that, questions were considered easy because the answers appear in the paragraph in a literal way so, after answering a couple of questions, the person starts looking for the answers directly in the text, so in general no reasoning or deduction is required to respond to them. However, this is a particularity of the survey layout since the paragraph is available above the questions. In a real evaluation scenario, it may work fine since the respondent would need to depend on their ability to retain and understand the text paragraph.

5. Discussion

From Table 1 we can interpret that, while generated distractors have little overlap at word level compared with the ground truth, which is reflected by the low score for BLEU, the performance obtained for the

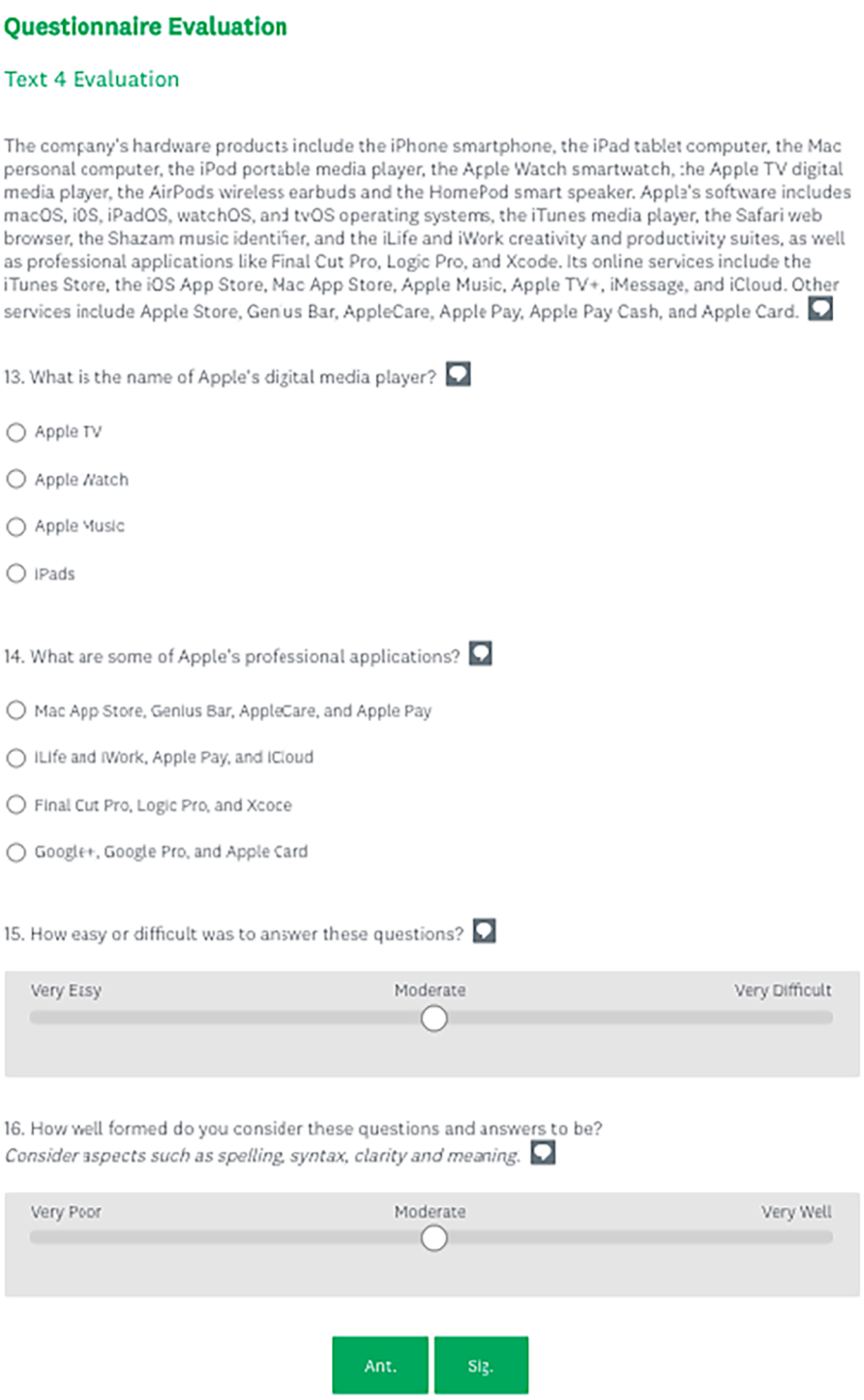

Fig. 8. Example of question included in the survey for human evaluation.

ROGUE-L metric is better (average of 14.91), in fact, it exceeds the results obtained by the model used in the research that proposed DG-RACE, where the model averaged in the same task and dataset 14.54 for ROGUE-L (Gao et al., 2019). In terms of DG, there is a question regarding metrics based on word-overlapping like BLEU: it is accepted that these metrics have high correlation with the judgment of human evaluators in tasks such as MT, however, when used in other tasks that require a broader domain, they tend to produce very low results because they do not capture the semantic relationships of generated texts (Papineni et al., 2002). This limitation has also been mentioned for QG tasks (T. Klein & Nabi, 2019), where metrics like ROGUE or BLEU cannot capture if a generated question is semantically correct. In this sense, BLEU and ROUGE do not take into consideration the meaning of the sentences or the syntax, because their computation is strictly based on word overlapping using n-grams, so the reward (or penalty) is the same for common words and for words that give meaning or enrich the sentence. Therefore, these metrics are unable to effectively capture the presence of synonyms and paraphrases (Sulem, Abend, & Rappoport, 2018) nor to measure the inclusion of topics in the meaning of the evaluated statements (Ganesan, 2018). Thus, in this study, the similarity based on embeddings is considered as a complementary metric for the DG task, helping to capture the semantic relationship between the

Table 2
Results of human evaluation regarding the level of difficulty for answering the generated MCQs.

Wikipedia page	Questions for selected paragraph	How easy or difficult was to answer these questions?					
		Min	Max	Median	Mean	Std. Dev	Result
Artificial_intelligence	What are the traditional goals of AI research? What is one of the long-term goals of AI?	1	5	2	2.24	1.21	2.24/5 (Easy)
Darth_Vader	Who was the visual effects artist for Dark Forces? In what movie is Vader seen boarding his shuttle?	1	4	3	2.24	1	2.24/5 (Easy)
World_War_II	When is the start of the war in Europe generally held to be? When did the Second Sino-Japanese War begin?	1	3	2	1.73	0.77	1.73/5 (Easy)
Apple_Inc.	What is the name of Apple's digital media player? What are some of Apple's professional applications?	1	4	2	2.29	1.16	2.29/5 (Easy)
The_Hitchhiker%27s_Guide_to_the_Galaxy	Who created the Earth? What was the answer to the Ultimate Question of Life, the Universe, and Everything?	1	4	2.5	2.36	0.89	2.36/5 (Easy)
					Total	**1.00**	**2.17/5 (Easy)**

Table 3
Results of human evaluation about quality and how well structured were the MCQs generated.

Wikipedia page	Questions for selected paragraph	How well formed do you consider these questions and answers to be?					
		Min	Max	Median	Mean	Std. Dev	Result
Artificial_intelligence	What are the traditional goals of AI research? What is one of the long-term goals of AI?	2	5	5	4.24	0.94	4.24/5 (Good)
Darth_Vader	Who was the visual effects artist for Dark Forces? In what movie is Vader seen boarding his shuttle?	1	5	5	4.18	1.20	4.18/5 (Good)
World_War_II	When is the start of the war in Europe generally held to be? When did the Second Sino-Japanese War begin?	3	5	5	4.73	0.57	4.18/5 (Good)
Apple_Inc.	What is the name of Apple's digital media player? What are some of Apple's professional applications?	3	5	5	4.5	0.82	4.5/5 (Very Good)
The_Hitchhiker%27s_Guide_to_the_Galaxy	Who created the Earth? What was the answer to the Ultimate Question of Life, the Universe, and Everything?	3	5	5	4.29	0.88	4.29/5 (Good)
					Total	**0.88**	**4.28/5 (Good)**

reference and the generated distractor. Cosine similarity has been used in this case, however, other methods can be explored and compared in future works, including word mover's distance (Kusner, Sun, Kolkin, & Weinberger, 2015), Euclidean distance, Smooth Inverse Frequency (Arora, Liang, & Ma, 2017) or more complex neurofuzzy approaches (Martinez-Gil, Mokadem, Küng, & Hameurlain, 2021). Also, it is possible to experiment with other vector representations like Word2vec or the Universal Sentence Encoder (Cer et al., 2018).

The presented results indicate that generated distractors are not very precise (in terms of the words used), based on word-overlapping metrics, however, their semantic context remains relatively close. From a DG point of view, this can be favorable, because the trained model has a good ability to generate text within a context, with considerable variability in the words included in the output. This may be ideal for DG, since it is necessary to be able to obtain multiple and different outputs for the same input. Fig. 9 shows an example of the difference of applying a metric such as BLEU in contrast to cosine similarity based on embeddings, where the generated distractor retains semantic connection and context, but the resulting BLEU score is zero while the similarity value is much higher.

Regarding human evaluation of the MCQs, evaluators consider the questionnaires to be well formed, and relatively easy to answer because there is little reasoning or inference involved in the process of selecting the right option. This assessment is aligned with how the QA model works in the proposed pipeline, because it extracts spans of text considered as candidate answers and then, based on those spans, it generates the questions (answer-aware question generation). Therefore, generated questions seem to be more oriented to measuring retention and memory than comprehension. However, this characteristic is also found on human generated MCQs, raising concerns about MCQs being frequently structured for knowledge recall instead of higher order thinking (Brown & Abdulnabi, 2017).

Human evaluation of MCQs is challenging and costly. In this study, we only focused on validating a small sample of automatically generated items (10 questions), based on 5 paragraphs of text covering different topics. However, this took on average more than 13 min to the

Fig. 9. Example of BLEU score and Cosine Similarity using embeddings for a generated distractor.

participant to complete the survey, so it would be expected that a very large sample would increase the time and make it more difficult and time consuming for them. Another approach would be needed to increase the number of participants and the size of the sample. For instance, crowdsourcing platforms like Amazon Mechanical Turk can be used for launching jobs with the objective of answering and evaluating a set of machine-generated MCQs. These jobs would be performed by a distributed workforce, allowing to have multiple evaluators. This approach has been used for generating datasets like SQuAD, generate high-quality MCQs for specific domains (Welbl, Liu, & Gardner, 2017) and generate distractors (Scheponik et al., 2020). Even so, other challenges are expected, such as controlling the selection of participants or ensuring that subjects are conscientiously answering the survey (Scheponik et al., 2020). A hybrid approach is also possible, using crowdsourcing for providing feedback for the MCQs and relying on a model trained for answering them, serving as a classifier for identifying well-formed MCQs. This allows to validate the full composition of the questionnaries as the model is able to properly choose the right option in the presence of the context, question, correct answer and distractors. For instance, UNIFIEDQA (Khashabi et al., 2020) is a model trained for this task and supports MCQs. Its performance is on par with specific-dataset models and shows good generalization for unseen datasets.

6. Conclusion

In this work, we addressed the problem of generating questionnaires composed of MCQs using three dimensions: QG, QA and DG. A processing pipeline for this approach is designed and an implementation based on pre-trained T5 language models is presented.

To implement the DG task, a definition using a Text-To-Text format is proposed and a T5 model is fine-tuned on the DG-RACE dataset, showing 14.91 for ROUGE-L and improving the performance reference for this dataset in that metric. A discussion about the lack of an adequate metric for DG is presented and the cosine similarity using word embeddings is considered as a complementary metric. The trained model achieved an average similarity of 0.71 for the DG-RACE dataset.

Questionnaires are evaluated by human experts reporting that questions and options are generally well-structured regarding spelling, syntax, meaning and clarity, with a mean score of 4.28 out of 5 in this topic. However, generated MCQs tend to be oriented towards measuring retention instead of comprehension, because questions and answers are directly extracted from text spans of context paragraphs.

Future lines of research can explore the use of larger and more powerful models, such as the GPT-3 language model (Brown et al., 2020). On the other hand, there is a line of research that explores QA in scenarios where a certain degree of inference between multiple sentences is required to be able to answer the questions, rather than approaches in which the answer is based on contiguous sections of the text (spans). Datasets such as HybridQA (Chen et al., 2020) and MetaQA (Dhingra et al., 2020) can be used for this kind of QA task called multi-hop QA. This also leads to a similar line of work but for QG, since the problem can be seen as the opposite of QA.

Another line of work points to building larger and specific datasets for DG. This can boost the number of research works about DG (Gao et al., 2019), helping to train models with better performance. On the other hand, with an appropriate dataset, the problem can be addressed differently, for example, a model could be trained so that the distractor generation task produces multiple options in the same output, rather than having to make multiple predictions to generate the options. Additionally, future work can explore the formulation of a better metric for measuring the QG performance. This would help to have a better reference of the state-of-the-art in this topic.

CRediT authorship contribution statement

Ricardo Rodriguez-Torrealba: Conceptualization, Investigation, Methodology, Software, Visualization, Writing – review & editing. **Eva Garcia-Lopez:** Conceptualization, Investigation, Methodology, Supervision, Writing – review & editing. **Antonio Garcia-Cabot:** Funding acquisition, Investigation, Methodology, Supervision, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partially co-funded by the Comunidad de Madrid (Grant number: CM/JIN/2021-034) and University of Alcala (Grant number: PIUAH21/IA-010).

References

Arora, S., Liang, Y., & Ma, T. (2017). A simple but tough-to-beat baseline for sentence embeddings. *5Th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*.

Brown, G. T. L., & Abdulnabi, H. H. A. (2017). Evaluating the quality of higher education instructor-constructed multiple-choice tests: Impact on student grades. *Frontiers in Education, 2*. https://doi.org/10.3389/feduc.2017.00024

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., … Openai, D. A. (2020). *Language Models are Few-Shot Learners*.

Cer, D., Yang, Y., Kong, S., Hua, N., Limtiaco, N., John, R., … st., Kurzweil, R.. (2018). Universal Sentence Encoder. *EMNLP 2018 - Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Proceedings*.

Chan, Y.-H., & Fan, Y.-C. (2019). *A Recurrent BERT-based Model for Question Generation*. 10.18653/v1/d19-5821.

Chen, W., Zha, H., Chen, Z., Xiong, W., Wang, H., & Wang, W. (2020). *HybridQA: A Dataset of Multi-Hop Question Answering over Tabular and Textual Data*.

Das, R., Ray, A., Mondal, S., & Das, D. (2016). A rule based question generation framework to deal with simple and complex sentences. *2016 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2016*. 10.1109/ICACCI.2016.7732102.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. (Mlm). Retrieved from http://arxiv.org/abs/1810.04805.

Dhingra, B., Zaheer, M., Balachandran, V., Neubig, G., Salakhutdinov, R., & Cohen, W. W. (2020). *Differentiable Reasoning over a Virtual Knowledge Base*. 1–16.

Dodge, J., Ilharco, G., Schwartz, R., Farhadi, A., Hajishirzi, H., & Smith, N. (2020). *Fine-Tuning Pretrained Language Models: Weight Initializations, Data Orders, and Early Stopping*. Retrieved from http://arxiv.org/abs/2002.06305.

Du, X., Shao, J., & Cardie, C. (2017). *Learning to Ask: Neural Question Generation for Reading Comprehension*.

Fan, A., Lewis, M., & Dauphin, Y. (2018). Hierarchical neural story generation. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers), 1*. 10.18653/v1/p18-1082.

Ganesan, K. (2018). *ROUGE 2.0: Updated and Improved Measures for Evaluation of Summarization Tasks*. 10.48550/arxiv.1803.01937.

Gao, Y., Bing, L., Li, P., King, I., & Lyu, M. R. (2019). Generating Distractors for Reading Comprehension Questions from Real Examinations. In *Proceedings of the AAAI Conference on Artificial Intelligence*. https://doi.org/10.1609/aaai.v33i01.33016423

Haladyna, T. M., & Rodriguez, M. C. (2013). Developing and validating test items. In *Developing and Validating Test Items*. 10.4324/9780203850381.

Heilman, M. (2011). Automatic Factual Question Generation from Text. *Dissertation*.

Holmes, W., Bialik, M., & Fadel, C. (2019). *Artificial intelligence in education: Promises and implications for teaching and learning*. Boston, MA: Center for Curriculum Redesign.

Holtzman, A., Buys, J., Du, L., Forbes, M., & Choi, Y. (2019). The curious case of neural text degeneration. *CEUR Workshop Proceedings*.

Hosking, T., & Riedel, S. (2019). Evaluating rewards for question generation models. NAACL HLT 2019–2019 Conference of the North American Chapter of the Association for Computational Linguistics. *Human Language Technologies - Proceedings of the Conference, 1*, 2278–2283. https://doi.org/10.18653/v1/n19-1237

Jiang, S., & Lee, J. (2018). *Distractor Generation for Chinese Fill-in-the-blank Items*. 10.18653/v1/w17-5015.

Jurafsky, D., & Martin, J. H. (2009). *Speech and language processing* (2nd Edition). Upper Saddle River, NJ, USA: Prentice-Hall Inc.

Khashabi, D., Min, S., Khot, T., Sabharwal, A., Tafjord, O., Clark, P., & Hajishirzi, H. (2020). *UNIFIEDQA: Crossing Format Boundaries with a Single QA System*. 10.18653/v1/2020.findings-emnlp.171.

Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. *3Rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.

Klein, G., Kim, Y., Deng, Y., Crego, J., Senellart, J., & Rush, A. M. (2017). OpenNMT: Open-source toolkit for neural machine translation. *20th Annual Conference of the European Association for Machine Translation, EAMT 2017*.

Klein, T., & Nabi, M. (2019). *Learning to Answer by Learning to Ask: Getting the Best of GPT-2 and BERT Worlds*.

Kusner, M. J., Sun, Y., Kolkin, N. I., & Weinberger, K. Q. (2015). From word embeddings to document distances. *32nd International Conference on Machine Learning*.

Lai, G., Xie, Q., Liu, H., Yang, Y., & Hovy, E. (2017). RACE: Large-scale ReAding comprehension dataset from examinations. *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*. 10.18653/v1/d17-1082.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). *ALBERT: A Lite BERT for Self-supervised Learning of Language Representations*.

Liang, C., Yang, X., Dave, N., Wham, D., Pursel, B., & Giles, C. L. (2018). *Distractor Generation for Multiple Choice Questions Using Learning to Rank*. 10.18653/v1/w18-0533.

Lin, C. Y. (2004). Rouge: A package for automatic evaluation of summaries. *Proceedings of the Workshop on Text Summarization Branches out (WAS 2004)*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., … Allen, P. G. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*.

Martinez-Gil, J., Mokadem, R., Küng, J., & Hameurlain, A. (2021). A Novel Neurofuzzy Approach for Semantic Similarity Measurement. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 12925 LNCS*. https://doi.org/10.1007/978-3-030-86534-4_18

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *1St International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations ofwords and phrases and their compositionality. *Advances in Neural Information Processing Systems*.

Miller, G. A. (1995). WordNet: A lexical database for English. *Communications of the ACM, 10*(1145/219717), Article 219748.

Olivares Olivares, S. L., Hernández, R. I. E., Corolla, M. L. T., Alvarez, J. P. N., & Sánchez-Mendiola, M. (2021). MOOC learning assessment in clinical settings: Analysis from quality dimensions. *Medical Science Educator*. https://doi.org/10.1007/s40670-020-01178-7

Palvia, S., Aeron, P., Gupta, P., Mahapatra, D., Parida, R., Rosner, R., & Sindhi, S. (2018). Online education: Worldwide status, challenges, trends, and implications. *Journal of Global Information Technology Management, 21*, 233–241. https://doi.org/10.1080/1097198X.2018.1542262

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*. https://doi.org/10.1109/TKDE.2009.191

Papineni, K., Roukos, S., Ward, T., & Zhu, W. (2002). BLEU: A Method for automatic evaluation of machine translation. *Computational Linguistics*.

Patil, S. (2020). Neural question generation using transformers. Retrieved September 6, 2020, from https://github.com/patil-suraj/question_generation.

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. Stroudsburg, PA, USA: Association for Computational Linguistics. 10.3115/v1/D14-1162.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*. 10.18653/v1/n18-1202.

Radford Alec, Wu Jeffrey, Child Rewon, Luan David, Amodei Dario, & Sutskever Ilya. (2019). Language Models are Unsupervised Multitask Learners. *OpenAI Blog*.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., … Liu, P. J. (2019). *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. 21*, 1–67. Retrieved from http://arxiv.org/abs/1910.10683.

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). SQuad: 100,000+ questions for machine comprehension of text. *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*.

Scheponik, T., Golaszewski, E., Herman, G., Offenberger, S., Oliva, L., Peterson, P. A. H., & Sherman, A. T. (2020). Investigating crowdsourcing to generate distractors for multiple-choice assessments. *Advances in Intelligent Systems and Computing, 1055*. https://doi.org/10.1007/978-3-030-31239-8_15

Sharma, S., Asri, L. El, Schulz, H., & Zumer, J. (2017). *Relevance of Unsupervised Metrics in Task-Oriented Dialogue for Evaluating Natural Language Generation*.

Shin, J., Guo, Q., & Gierl, M. J. (2019). Multiple-choice item distractor development using topic modeling approaches. *Frontiers in Psychology*. https://doi.org/10.3389/fpsyg.2019.00825

Sulem, E., Abend, O., & Rappoport, A. (2018). BLEU is not suitable for the evaluation of text simplification. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., & Liu, C. (2018). A survey on deep transfer learning. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/978-3-030-01424-7_27

Tang, D., Duan, N., Qin, T., Yan, Z., & Zhou, M. (2017). Question answering and question generation as dual tasks. *ArXiv Preprint*. ArXiv:1706.02027.

Unesco. (2019). Artificial intelligence in education: Challenges and opportunities for sustainable development. *Working Papers on Education Policy, 7*, 46.

UNESCO. (2020). School closures caused by Coronavirus (Covid-19). Retrieved April 19, 2020, from https://en.unesco.org/covid19/educationresponse.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 2017-Decem*(Nips), 5999–6009.

Vijayakumar, A. K., Cogswell, M., Selvaraju, R. R., Sun, Q., Lee, S., Crandall, D., & Batra, D. (2016). *Diverse Beam Search: Decoding Diverse Solutions from Neural Sequence Models*.

Von Davier, M. (2019). *Training Optimus Prime, M.D.: Generating Medical Certification Items by Fine-Tuning OpenAI's gpt2 Transformer Model*.

Wang, Z., Lan, A. S., Nie, W., Waters, A. E., Grimaldi, P. J., & Baraniuk, R. G. (2018). QG-Net: A Data-Driven question generation model for educational content. *Proceedings of the 5th Annual ACM Conference on Learning at Scale, L at S 2018*. 10.1145/3231644.3231654.

Welbl, J., Liu, N. F., & Gardner, M. (2017). Crowdsourcing Multiple Choice Science Questions. *3rd Workshop on Noisy User-Generated Text, W-NUT 2017 - Proceedings of the Workshop*. 10.18653/v1/w17-4413.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., … Rush, A. M. (2019). *Transformers: State-of-the-Art Natural Language Processing*.

Xiao, H., Wang, F., Yan, J., & Zheng, J. (2018). *Dual Ask-Answer Network for Machine Reading Comprehension*.

Yuan, X., Wang, T., Gulcehre, C., Sordoni, A., Bachman, P., Zhang, S., … Trischler, A. (2017). *Machine Comprehension by Text-to-Text Neural Question Generation*. 10.18653/v1/w17-2603.

Zhou, Q., Yang, N., Wei, F., Tan, C., Bao, H., & Zhou, M. (2018). Neural question generation from text: A preliminary study. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/978-3-319-73618-1_56