Learning to Reuse Distractors to support Multiple Choice Question Generation in Education

Semere Kiros Bitew, Amir Hadifar, Lucas Sterckx, Johannes Deleu,
Chris Develder *Senior Member, IEEE,* and Thomas Demeester

***Abstract*—Multiple choice questions (MCQs) are widely used in digital learning systems, as they allow for automating the assessment process. However, due to the increased digital literacy of students and the advent of social media platforms, MCQ tests are widely shared online, and teachers are continuously challenged to create new questions, which is an expensive and time-consuming task. A particularly sensitive aspect of MCQ creation is to devise relevant distractors, i.e., wrong answers that are not easily identifiable as being wrong. This paper studies how a large existing set of manually created answers and distractors for questions over a variety of domains, subjects, and languages can be leveraged to help teachers in creating new MCQs, by the smart reuse of existing distractors. We built several data-driven models based on context-aware question and distractor representations, and compared them with static feature-based models. The proposed models are evaluated with automated metrics and in a realistic user test with teachers. Both automatic and human evaluations indicate that context-aware models consistently outperform a static feature-based approach. For our best-performing context-aware model, on average 3 distractors out of the 10 shown to teachers were rated as high-quality distractors. We create a performance benchmark, and make it public, to enable comparison between different approaches and to introduce a more standardized evaluation of the task. The benchmark contains a test of 298 educational questions covering multiple subjects & languages and a 77k multilingual pool of distractor vocabulary for future research.**

***Index Terms*—Distractor generation, multiple choice question, natural language processing, online learning, transformers.**

I. Introduction

ONLINE learning has become an indispensable part of educational institutions. It has emerged as a necessary resource for students and schools all over the globe. The recent COVID-19 pandemic has made the transition to online learning even more pressing. One very important aspect of online learning is the need to generate homework, test, and exam exercises to aid and evaluate the learning progress of students [1]. Multiple choice questions (MCQs) are the most common form of exercises [2] in online education as they can easily be scored automatically. However, the construction of MCQs is time consuming [3] and there is a need to continuously generate new (variants of) questions, especially for testing, since students tend to share questions and correct answers from MCQs online (e.g., through social media).

The authors are with the Text-to-Knowledge (T2K) team at IDLab, Ghent University-imec, Belgium. (e-mail: semerekiros.bitew, firstname.lastname@ugent.be), except for *Lucas Sterckx* (lucassterckx@gmail.com), who is currently at LynxCare. He contributed to this work while working in the T2K team.

The rapid digitization of educational resources opens up opportunities to adopt artificial intelligence (AI) to automate the process of MCQ construction. A substantial number of questions already exist in a digital format, thus providing the required data as a first step toward building AI systems. The automation of MCQ construction could support both teachers and learners. Teachers could benefit from an increased efficiency in creating questions, in their already high workload. Students' learning experience could improve due to increased practice opportunities based on automatically generated exercises, and if these systems are sufficiently accurate, they could power personalized learning [4].

A crucial step in MCQ creation is the generation of distractors [5]. Distractors are incorrect options that are related to the answer to some degree. The quality of an MCQ heavily depends on the quality of distractors [3]. If the distractors do not sufficiently challenge learners, picking the correct answer becomes easy, ultimately degrading the discriminative power of the question. The automatic suggestion of distractors will be the focus of this paper.

Several works have already proposed distractor generation techniques for automatic MCQ creation, mostly based on selecting distractors according to their similarity to the correct answer. In general, two approaches are used to measure the similarity between distractors and an answer: graph-based and corpus-based methods. *Graph-based* approaches use the semantic distance between concepts in the graph as a similarity measure. In language learning applications, typically WordNet [6], [7] is used to generate distractors, while for factoid questions domain-specific (ontologies) are used to generate distractors [8]–[11]. In *corpus-based methods*, similarity between distractors and answers has been defined as having similar frequency count [12], belonging to the same POS class [13], having a high co-occurrence likelihood [14], having similar phonetic and morphological features [7], and being nearby in embedding spaces [15]–[17]. Other works such as [5], [18]–[20] use machine learning models to generate distractors by using a combination of the previous features and other types of information such as tf-idf scores.

While the current state-of-the-art in MCQ creation is promising, we see a number of limitations. First of all, existing models are often *domain specific*. Indeed, the proposed techniques are tailored to the application and distractor types. In language learning, such as vocabulary, grammar or tense usage exercises, typically similarity based on basic syntactic and statistical information works well: frequency, POS information, etc. In other domains, such as science,

health, history, geography, etc., distractors should be selected on deeper understanding of context and semantics, and the current methods fail to capture such information.

The second limitation, *language dependency*, is especially applicable to factoids. Models should be agnostic to language because facts do not change with languages. Moreover, building a new model for each language could be daunting task as it would require enough training data for each language.

In this work, we study how the automatic retrieval of distractors can facilitate the efficient construction of MCQs. We use a high-quality large dataset of question, answer, distractor triples that are diverse in terms of language, domain, and type of questions. Our dataset was made available by a commercial organization active in the field of e-assessment (see Section III-B), and is therefore representative for the educational domain, with a total of 62k MCQ, none of them identical, encompassing only 92k different answers and distractors. Despite an average of 2.4 distractors per question, there is a large reuse of distractors over different questions. This motivates our premise to retrieve and *reuse* distractors for new questions. We make use of the latest data-driven Natural Language Processing (NLP) techniques to retrieve candidate distractors. We propose *context-aware multilingual models* that are based on deep neural network models that select distractors by taking into account the context of the question. They are also able to handle variety of distractors in terms of length and type. We compare our proposed models to a competitive *feature-based* baseline that is based on classical machine learning methods trained on several handcrafted features.

The methods are evaluated for distractor quality using automated metrics and a real-world user test with teachers. Both the automatic evaluation and the user study with teachers indicate that the proposed context-aware methods outperform the feature-based baseline. Our contribution can be summarized as follows:

- We built three multilingual Transformer-based distractor retrieval models that suggest distractors to teachers for multiple subjects in different languages. The first model (Section III-D3) requires similar distractors to have similar semantic representations, while the second (Section III-D2) learns similar representations for similar questions, and the last combines the complementary advantages of of these two models (Section III-D3).
- We performed a user study with teachers to evaluate the quality of distractors proposed by the models, based on a four-level annotation scheme designed for that purpose.
- The evaluation of our best model on in-distribution held-out data reveals an average increase of 20.4% in terms of recall at 10, compared to our baseline model adapted from [19]. The teacher-based annotations on language learning exercises show an increase by 4.3% in the fraction of good distractors among the top 10 results, compared to teacher annotations for the same baseline. For factoid questions, the fraction of quality distractors more than doubles w.r.t. the baseline, with an improvement of 15.3%.
- We released[1] a test-set of educational questions of 6 subjects with 50 MCQs per subject and annotated distractors, and 77k size distractor vocabulary as benchmark to stimulate further research. The dataset, which is made by experts, contains multilingual and multi-domain distractors.

The remainder of the paper is organized as follows: Section II describes the relevant work in MCQs in general and distractor generation in particular. Section III introduces the dataset, explains the details of the proposed methods and the evaluation setup of the user study with teachers. In Section V, the results of both the user study and automated evaluations is reported. And finally, in Section VI, we present the conclusion, lines for future work, and limitations of our proposed models.

II. Related work

A. MCQ*s in Education*

Multiple choice questions (MCQs) are widely used forms of exercises that require students to select the best possible answer from a set of given options. They are used in the context of learning, and assessing learners' knowledge and skills. MCQs are categorized as objective types of questions because they primarily deal with the facts or knowledge embedded in a text rather than subjective opinions [21]. It has been shown that recalling information in response to a multiple-choice test question bolsters memorizing capability, which leads to better retention of that information over time. It can also change the way information is represented in memory, potentially resulting in deeper understanding [22] of concepts.

An MCQ item consists of three elements:

- *stem*: is the question, statement, or lead-in to the question.
- *key*: the correct answer.
- *distractors*: alternative answers meant to challenge students' understanding of the topic.

For example, consider the MCQ in the first row of Table III: the stem of the MCQ is *"Which inhabitants are not happy with Ethiopia's plans of the Nile?"*. Four potential answers are given with the question. Among these, the correct answer is *"Egyptians"*, which is the key. The alternatives are the distractors.

MCQs are used in several teaching domains such as information technology [23], health [24], [25], historical knowledge [26], etc. They are also commonly used in standardized tests such as GRE and TOEFL. MCQs are preferred to other question formats because they are easy to score, and students can also answer them relatively quickly since typing responses is not required. Moreover, MCQs enable a high level of test validity if they are drawn from a representative sample of the content areas that make up the pre-determined learning outcomes [25]. The most time-consuming and non-trivial task in constructing MCQ is distractor generation [3], [19]. Distractors should be plausible enough to force learners to put some thought before selecting the correct answer.

[1]https://dx.doi.org/10.21227/gnpy-d910 or https://github.com/semerekiros/dist-retrieval

Preparing good multiple-choice questions is a skill that requires formal training [27], [28]. Moreover, several MCQ item writing guidelines are used by content specialists when they prepare educational tests. These guidelines also include recommendations for developing and using distractors [29]–[31]. Despite these guidelines, inexperienced teachers may still construct poor MCQs due to lack of training and limited time [32].

Besides reducing teachers' workloads, the automation of the distractor generation could potentially correct some minor mistakes made by teachers. For example, one of the rules suggested by [29] says: "the length of distractors and the key should be about the same". Such property could be easily integrated in the automation process.

MCQs also have drawbacks; they are typically used to measure lower-order levels of knowledge, and guesswork can be a factor in answering a question with a limited number of alternatives. Furthermore, because of a few missing details, learners' partial understanding of a topic may not be sufficient to correctly answer a question, resulting in partial knowledge not being credited by MCQs [22]. Nonetheless, MCQs are still extensively utilized in large-scale tests since they are efficient to administer and easy to score objectively [2].

B. Distractor Generation

Many strategies have been developed for generating distractors for a given question. The most common approach is to select a distractor based on its similarity to the key for a given question. Many researchers approximate the similarity between distractor and key according to WordNet [33]–[35]. WordNet [36] is a lexical database that groups words into sets of synonyms, and concepts semantically close to the key are used as distractors. The usage of such lexical databases is sound for language or vocabulary learning but not for factoid type questions. We instead provide a more general approach that could be used for both tasks, and instead of only using the key as the source of information while suggesting distractors, we also make use of the stem.

For learning factual knowledge, several works rely on the use of specific domain ontology as a proxy for similarity. Papasalouros *et al.* [8] employ several ontology-based strategies to generate distractors for MCQ questionnaires. For example, they generate "Brussels is a mountain" as a good distractor for an answer "Everest is a mountain" because both concept *City* and concept *Mountain* share the parent concept *Location*. Another very similar work by Lopetegui *et al.* [37] selects distractors that are declared siblings of the answer in a domain-specific ontology. The work by Leo *et al.* [10] improves upon the previous works by generating multi-word distractors from an ontology in the medical domain. Other works that rely on knowledge bases apply query relaxation methods, where the queries used to generate the keys were slightly relaxed to generate distractors that share similar features with the key [9], [38], [39]. While the methods in these works are dependent on their respective ontologies, we provide an approach that is ontology-agnostic and instead uses contextual similarity between distractors and questions.

Another line of works for distractor generation uses machine-learning models. Liu *et al.* [5] use a regression model based on characteristics such as character glyph, phonological, and semantic similarity for generating distractors in Chinese. Liang *et al.* [19] use two methods to rank distractors in the domain of school sciences. The first method adopts machine learning classifiers on manually engineered features (i.e., edit distance, POS similarity, etc.) to rank distractors. The second uses generative adversarial networks to rank distractors. Our baseline method is inspired by their first approach but was made to account for the multilingual nature of our dataset by extending the feature set.

There have also been a number of works on generating distractors in the context of machine comprehension [40]. Distractor generation strategies that fall in this category assume access to a contextual resource such as a book chapter, an article or a wikipedia page where the MCQ was produced from. The aim is then to generate a distractor that takes into account the reading comprehension text, and a pair composed of the question and its correct answer that originated from the text [41]–[43]. This line of work is incomparable to our work because we do not have access to an external contextual resource the questions were prepared from.

In this paper, we focus on building one model that is able to suggest candidate distractors for teachers both in the context of language and factual knowledge learning. Unlike previous methods, we tackle distractor generation with a multilingual dataset. Our distractors are diverse both in terms of domain and language. Moreover, the distractors are not limited to single words only.

III. Methodology

In this section, we formally define distractor generation as a ranking problem; describe our datasets; describe in detail the feature-based baseline and proposed context-aware models including their training strategies & prediction mechanisms.

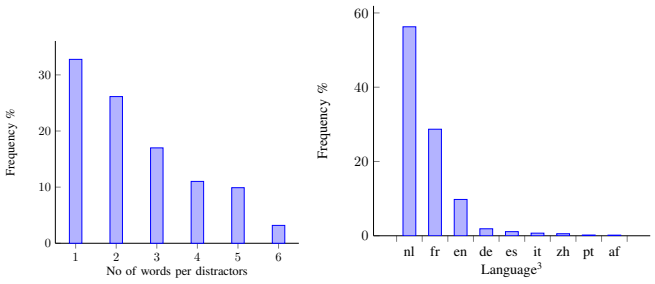
A. Task Definition: Distractor Retrieval

We assume access to a distractor candidate set $\mathcal{D}$ and a training MCQ dataset $\mathcal{M}$. Note that $\mathcal{D}$ can be obtained by pooling all answers (keys and distractors) from $\mathcal{M}$ (as in our experimental setting), but could also be augmented, for example, with keywords extracted from particular source texts. We formally write $\mathcal{M} = \{(s_i, k_i, \mathcal{D}_i) | i = 1, \ldots, N\}$. where for each item i among all N available MCQs, s_i refers to the question stem, k_i is the correct answer key, and $\mathcal{D}_i = \left\{d_i^{(1)}, ..., d_i^{(m_i)}\right\} \subseteq \mathcal{D}$ are the distractors in the MCQ linked to s_i and k_i. The aim of the distractor retrieval task is to learn a point-wise ranking score $r_i(d) : (s_i, k_i, d) \rightarrow [0, 1]$ for all $d \in \mathcal{D}$, such that distractors in $\mathcal{D}_i$ are ranked higher than those in $\mathcal{D} \setminus \mathcal{D}_i$, when sorted according to the decreasing score $r_i(d)$.

This task definition resembles the metric learning [44] problem in information retrieval. To learn the ranking function, we propose two types of models: feature-based models and context-aware neural networks.

TABLE I
THE STATISTICS OF OUR DATASET

	Train	Validation	Test
# Questions	61758	600	500
# Distractors per question	2.4	2.3	2.3
Avg question length	27.8 tokens	28.1 tokens	27.6 tokens
Avg distractor length	2.2 tokens	2.3 tokens	2.1 tokens
Avg answer length	2.2 tokens	2.3 tokens	2.2 tokens
Total # distractors	94,205	-	-
Total # distractors $\leq$ 6 tokens	77,505	-	-



(a) (b)

Fig. 1. (a) distractor length in number of tokens and (b) language distribution for the Televic dataset.

B. Data

In this section, we describe our datasets, namely: (i) *Televic dataset*, a big dataset that we used to train our models. (ii) *Wezooz dataset*, a small-scale external test set used for evaluation.

1) Televic dataset: this data is gathered through Televic Education's platform assessmentQ.[2] The tool is a comprehensive online platform for interactive workforce learning and high-stakes exams. It allows teachers to compose their questions and answers for practice and assessment. As a result, the dataset is made up of a large and high-quality set of questions, answers and distractors, manually created by experts in their respective fields. It encompasses a wide range of domains, subjects, and languages, without however any metadata on the particular course subjects that apply to the individual items.

We randomly divide our dataset into train/validation/test splits. We discard distractors with more than 6 tokens as they are very rare and unlikely to be reused in different contexts. We keep questions with at least one distractor. Table I summarizes the statistics of our dataset. The dataset contains around 62k MCQs in total. The size of the dataset is relatively large when compared to previously reported educational MCQ datasets such as SCiQ [45], and MCQL [19] which contain 13.7K and 7.1K MCQs respectively. On average, a question has more than 2 distractors, and contains exactly one answer. We use all the answer keys and distractors in the preprocessed dataset as the pool of candidate distractors (i.e., list of 77,505 filtered distractors) for proposing distractors for any new question.

The distractors in the dataset are not limited to single word distractors. More than 65% of the distractors contain two or more words as can be seen in Fig. 1a.

The data stems from multiple languages. Figure 1b shows the language distribution as detected by an off-the-shelf language classifier.[4] Given that Televic is a Belgian company, more than 50% of the questions are in Dutch, while French and English are the next most common languages in the dataset.

Another dimension of the dataset is its domain diversity. It comprises questions about language/vocabulary learning (e.g., French and English) and factoids covering subjects such as Math, Health, History, Geography, and Sciences. Besides material from secondary school education, it covers materials from assessment tasks for professionals such as training in hospitals or manufacturing firms. The data is anonymized and contains no customer information.

Depending on the question type we observe different types of distractors. (1) Factoid distractors: names of people, locations, organizations, concepts, dates. (2) Distractors with numerical elements, such as multiples, factors, rounding errors, etc. (3) Language distractors: spelling, grammatical, tense, etc. However, the proposed models are agnostic of the type and origin of the data, and the automated evaluation on the Televic test set contains a random sample covering the different question types and origins (see Section V-A). Note that although our dataset is a real-world commercial dataset, it only contains single-answer MCQs. However, the models we will put forward, could be readily extended towards multiple-answer MCQs, if such data were available.

2) WeZooz dataset: this data is a small-scale test set of questions gathered from WeZooz Academy,[5] which is a Flanders-based company providing an online platform with digital teaching materials for secondary school students and teachers. We selected four subjects; Natural sciences, Geography, Biology and History. Each subject was made to contain a fixed list of 50 questions that were randomly selected, and augmented with distractor annotations by teachers for these respective subjects (see Section IV). Note that this is an *external* test set, in the sense that the data distribution in the training set is not necessarily representative for this test set. This serves as a proof-of-concept for the general validity of our proposed method and models to specific use cases.

C. Feature-based Distractor Scoring

We built a strong feature-based model as our baseline. Feature-based models are a class of machine learning models that require a pre-specified set of handcrafted features as input. We designed 20 types of features capturing similarity between questions, answers, and the collection of candidate distractors. Formally, given a triplet (s, k, d) of question stem, key and distractor, our feature-based model first maps the input into a 20-dimensional feature vector $\phi(s, k, d) \in \mathbb{R}^{20}$, after which a classifier is trained to score the triplets according to compatibility of the question-answer-distractor combination. Our set of features can be segmented into four categories which are described below. A more detailed explanation of each feature can be found in Appendix B.

[2]https://www.televic-education.com/en/assessmentq
[3]We used ISO 639-1:2002 standard for names of languages.
[4]We used the *langid* language classifier: https://github.com/saffsd/langid.py
[5]https://www.wezoozacademy.be/

(i) *Morphological Features*: this category contains features that are related to the form and shape of words that occur in our (s, k, d) triplets. This includes features such as edit distance, difference in token length, longest common suffix between k & d, etc.

(ii) *Static embedding based features*: We trained a Word2Vec model [46] on our dataset to learn static embeddings for the distractors. We treat distractors and answers attached to the same question as chunks sharing similar context. The objective is to learn a vector space in which their representations will also be closer. We leverage the embedding representations to extract several numerical features. For example, we calculate the cosine similarity and word mover's distance [47] between the embeddings of d & k.

(iii) *Language Prior*: since our data is multilingual we also calculate the prior probability of the candidate distractor matching with the language of the question, and attach it to each feature vector.

(iv) *Corpus-based Features*: this category contains features that are derived from the statistics of words in the corpus. It includes features that such as the frequency of a distractor in the dataset and the inverse document frequency of distractors.

As classifier, we apply a *logistic regression* model to distinguish feature representations of actual question-answer-distractor triplets, present in the training, from triplets for which the distractor components belong to different question-answer combinations, sampled randomly. During training, the model's parameters are set to output high scores for actual triplets while the model is penalized for predicting high scores for others.

D. Context-aware Neural Distractor Scoring

Advanced context-aware neural models, unlike traditional feature-based models, do not require manual feature engineering. They have the ability to represent words depending on their semantic role and context in the considered text. In this work, we primarily focus on such context-aware models called *transformers* [48], which provide rich representations, and proved to achieve state-of-the-art results for many tasks in NLP such as question answering [49], machine translation [50], and text summarization [51]. A transformer is a deep neural network that uses a self-attention mechanism to assign importance weights to every part of the input sequence in how they are related to all other parts of the input. Transformers can scale to very large numbers of trainable parameters, stacked into very deep networks, which can still be trained very efficiently on parallel GPU hardware and thus learn from very large amounts of data. In NLP, such models are often trained on large unlabeled corpora to learn the inherent word and sentence level correlations (i.e., to model language structure) between varying contexts. This process is called *pretraining*, and downstream NLP tasks usually rely on such a pretrained generic model to be *finetuned* to their more specific needs instead of training a new model from scratch. Leveraging the knowledge gained during a generic pretraining process to improve prediction effectiveness for a specific supervised learning task, is a form of *transfer learning* [52], [53]. A common language task often used for pretraining transformer models called *masked language modelling* (MLM) requires masking a portion of the input text and then training a model to predict the masked tokens — in other words, to reconstruct the original non-masked input. BERT (Bidirectional Encoder Representations from transformers) [54] is the most popular pretrained masked language model and has been widely used in many downstream tasks such as question answering & generation, machine reading comprehension, and machine translation, by fine-tuning it using a labelled dataset that provides supervision signal.

In this work, we present models to rank and retrieve distractors, based on such a pretrained transformer text encoder, which we finetuned by requiring similar distractors to have similar representations, and similar questions also to have similar representation. In the following paragraphs, we provide a detailed description of these models, visualized in Fig. 2, followed by a description of the training procedure and the inference mechanism.

*1) Distractor similarity based model (*D-SIM*):* We hypothesize that distractors co-occurring within the same MCQ item are semantically related through their link with the corresponding question stem and answer key. Following that hypothesis, the D-SIM model is designed (and trained) to yield a similar vector representation for a given (stem, key) pair (s_i, k_i), as each of the corresponding distractors d_i. Following the same logic, all candidate distractors $d \in \mathcal{D}$ can then be scored in terms of their similarity (in representation space) with a given new (stem, key) pair, after which the top candidates are returned by the model as likely valid distractors. We use the pretrained multilingual BERT (mBERT) encoder [54], followed by a fully connected linear layer (i.e., dense layer) to obtain initial representations for a (stem, key) pair, as well as for the distractors. We designed our model in a bi-encoder setting inspired by [55], and schematically shown on the left-hand side of Fig. 2. The distractor d is fed into the mBERT encoder, and the output representation of the $[CLS]$ token[6] is used as an input to the dense layer. The output from the dense layer is taken as the corresponding representation h_d. The considered stem and key are concatenated into a single sequence of tokens[7] as "s_i $[SEP]$ k_i", which is fed into the *same* mBERT encoder (i.e., with parameter reuse, as indicated by the double arrow in Fig. 2). Similar to the distractor embedding, we take the [CLS] token representation and feed it to the dense layer (i.e., *different* dense layer with no parameter sharing), and take its output as the vector representation of the key-aware stem h_{sk}. Finally, the similarity score between (s_i, k_i) and d is obtained as the dot product between their respective representations:

$$r_i^{\text{D-SIM}}(d) = h_i^{(sk)} \cdot h^{(d)}$$

[6][CLS] is a special token that is prepended to the input, and its corresponding output representation is pretrained to represent the entire sequence that is used for classification tasks.

[7]The often used [SEP] token is a special token known by the model, that separates input sentences.

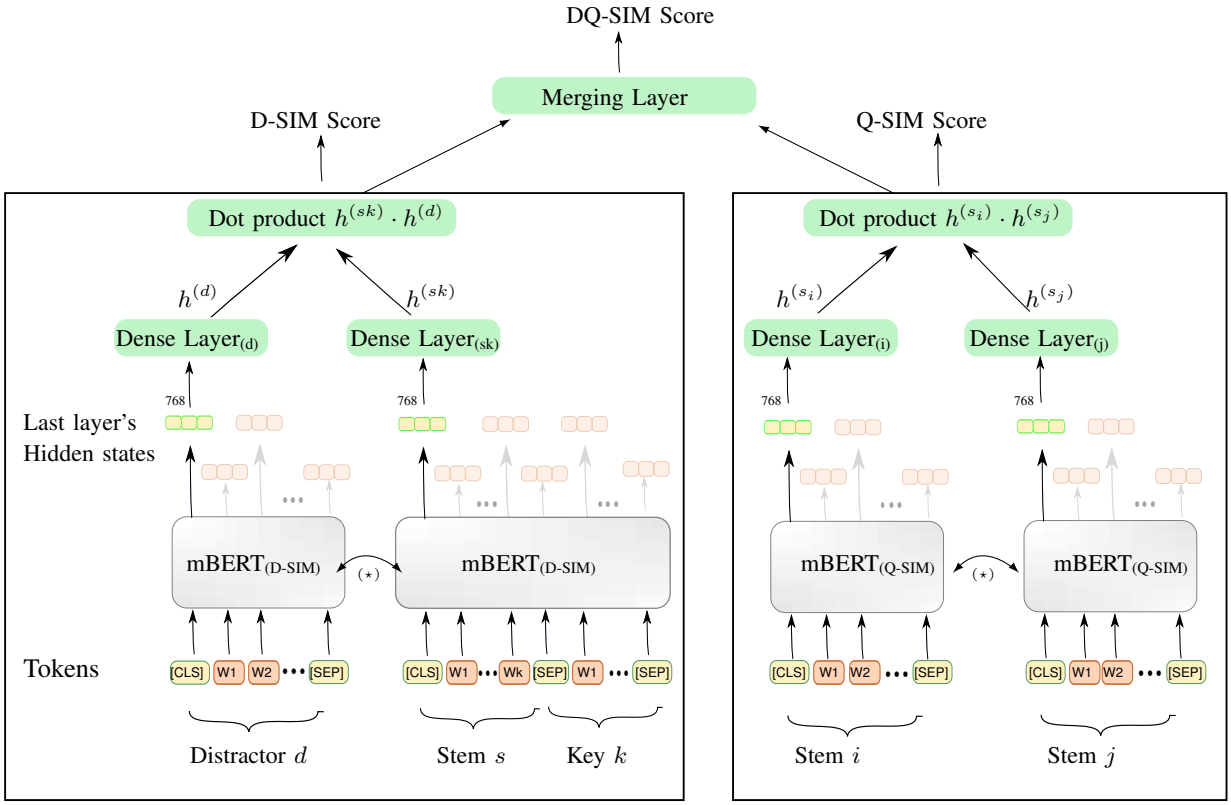


Fig. 2. Our proposed context-aware distractor retrieval systems. For the D-SIM model (i.e., left), distractor d and concatenation of the stem s & key k separated by [SEP] are fed into the same mBERT$_{(D\text{-}SIM)}$ encoder, and then their respective vector representations at [CLS] are used as inputs to two different dense layers that do not share parameters. The outputs of these dense layers, $h^{(d)}$ & $h^{(sk)}$ are used to calculate the similarity between d & $s[sep]k$ using the dot product. Similarly for Q-SIM (i.e., right), two question stems i & j are encoded separately using the same mBERT$_{(Q\text{-}SIM)}$, and their respective [CLS]output vectors are fed into two different dense layers (i.e., dense layer$_{(i)}$ & dense layer$_{(j)}$) to produce their corresponding representations $h^{(s_i)}$ & $h^{(s_j)}$. These are used to calculate their similarity between the two stems using dot product. The DQ-SIM model (i.e., top) linearly combines the two models using a merging layer with an α parameter. $(\star)$ denotes parameter reuse by the encoders.

During training, the encoder is fine-tuned to achieve higher scores for compatible stem/key and distractor combinations, and lower scores for incompatible ones (as described in Section III-E in more detail).

*2) Question similarity based model (*Q-SIM*):* This model is based on the assumption that different questions that share one or more distractors or answer keys are likely semantically related, such that their associated distractors could be used as good candidate distractors for one another. To accomplish this, we first rearrange the training data in such a way that these questions, sharing at least one distractor or key, are clustered together (see Table II for an example). Then, we train our Q-SIM model to produce similar representation for question stem pairs (s_i, s_j) that are in the same cluster. The right-hand side of Fig. 2 depicts the Q-SIM model, again based on a bi-encoder architecture. The stem representation $h_i^{(s)}$ for a question MCQ$_i$ is again obtained through an mBERT encoder, followed by a fully connected linear layer, similarly to $h_i^{(sk)}$ but ignoring the question key. The Q-SIM scoring function is defined as

$$r_i^{\text{Q-SIM}}(d_j) = h_i^{(s)} \cdot h_j^{(s)}$$

and can be interpreted as follows. For a given question MCQ$_i$, its stem representation $h_i^{(s)}$ is compared through dot product similarity with the representation of any candidate distractor d_j originating from a question MCQ$_j$. The particular representation of d_j assumed in Q-SIM is in fact MCQ$_j$'s stem representation $h_j^{(s)}$. Note that Q-SIM does not allow making a distinction in terms of score between different distractors from the same MCQ. Candidate distractors with the same score are considered equally likely according to Q-SIM, and ranked in an arbitrary order. Based on the intuition outlined above, more complex formulations for Q-SIM can be designed, for example with a feature characterizing the nature of the pairwise comparison (i.e., the actual answers of the considered questions, two of their respective distractors, or the answer for the one and a distractor for the other). However, given the already significant improvement of the presented basic Q-SIM formulation (see Section V-A), we chose to include only that model in our evaluation. In fact, its simple intuitive formulation makes it straightforward to explain to teachers, which is an important aspect in their trust in the model [56].

*3) Distractor and Question similarity model (*DQ-SIM*):* this model combines the previous two models using a merging

TABLE II
Q-SIM TRAINING DATA EXAMPLES.

Distractor/Answer	Associated Questions	Description
koolhydraten en vetten	1. Welke groepen voedingsstoffen leveren vooral energie? 2. Welke voedselcomponenten kunnen stoffen leveren die zowel bij assimilatie als bij dissimilatie in cellen worden gebruikt?	Factoid questions with multi-word distractor in Dutch.
surrounded	1. The guest house is … on the countryside. 2. The valley was … by forests.	A fill in the gap question for English language learning.
Marokko	1. Welk land is in 2011 gesplitst door het langdurig conflict in Darfur ? 2. Rabat is de hoofdstad van …	A combination of fill-in the gap and normal questions

layer (visualized on top of Fig. 2), based on the intuition that a well-chosen combined model may benefit from the complementary advantages of both individual models. This merging layer combines the outputs from D-SIM and Q-SIM using a merging parameter α, to control the contribution of the individual models. We investigated empirical score-based and rank-based merging strategies. The score-based model assumes a linear combination of both respective *scores* $r_i^{\text{D-SIM}}$ and $r_i^{\text{Q-SIM}}$ from D-SIM and Q-SIM, in which their individual contribution is controlled by the hyperparameter α:

$$r_i^{\text{DQ-SIM-score}}(d) = \alpha\, r_i^{\text{D-SIM}}(d) + (1-\alpha)\, r_i^{\text{Q-SIM}}(d)$$

The rank-based model combines the distractor *ranks* $\rho_i^{\text{D-SIM}}$ and $\rho_i^{\text{Q-SIM}} \in \{1, 2, 3, ..., N\}$ from D-SIM and Q-SIM into the score

$$r_i^{\text{DQ-SIM-rank}}(d) = \frac{\alpha}{\log\left(\rho_i^{\text{D-SIM}}(d)+1\right)} + \frac{1-\alpha}{\log\left(\rho_i^{\text{Q-SIM}}(d)+1\right)}$$

This scoring function is based on weighted combination of inverse distractor rankings, such that high-ranked distractors have more weight. We use logarithmic smoothing to avoid the potential contribution of low-ranked distractors from vanishing too rapidly.

E. Training

We use *contrastive learning* as our training strategy [57]. Contrastive learning [46], [58], [59] is a machine learning technique that aims to learn representations of data by contrasting similar and dissimilar examples. It aims to bring similar instances closer together in the representation space by maximizing the similarity between their embeddings, while pushing dissimilar samples further apart by minimizing their similarity.

In a contrastive learning setting, it is often the case that similar example pairs (i.e., also called positive examples) are available explicitly in training datasets, whereas dissimilar or negative examples need to be sampled from an extremely large pool of instances. For the Q-SIM model, a positive pair consists of two questions sharing at least one distractor, whereas for the D-SIM model, we require similar representations for a given (stem, key) item and a distractor corresponding to the same MCQ.

As a negative sampling strategy, we use in-batch negatives [60] while training our models. For D-SIM, the in-batch negatives are gold-standard positive distractors for the other instances in the same batch. While for Q-SIM, the in-batch negatives are the positive questions that come from the other instances in the same batch. Reusing gold standard distractors or questions from the same batch as negatives makes training more efficient, compared to randomly sampling negatives for each positive pair in the batch.

With the notation $r_i(d)$ (common in both D-SIM and Q-SIM) for scoring MCQ_i against distractor d, and by introducing the sigmoid function $\sigma(r) = 1/(1+e^{-r})$, we can write the contrastive loss [61] $\mathcal{L}_i$ to be minimized for MCQ_i with matching distractors d^+ as follows:

$$\mathcal{L}_i = -\sum_{d^+} \log \sigma\big(r_i(d^+)\big) - \sum_{d^-} \log \sigma\big(-r_i(d^-)\big)$$

in which $r_i(d^+)$ denotes the score of a positive distractor for the considered question, and $r_i(d^-)$ the scores for the in-batch negatives (summed over the considered batch of training instances). If the quantity $\sigma\big(r_i(d)\big)$ is interpreted as the probability that distractor d is compatible with MCQ_i (in the sense of model D-SIM or Q-SIM), then minimizing the above loss term can be understood as maximizing the joint estimated probability of d^+ being compatible distractors for MCQ_i, and the in-batch negatives d^- to be incompatible ones.

F. Using the models for predictions

This section describes the inference mechanism for our models. Inference refers to using a trained model to make predictions about new data. For each of the models, the goal is inducing an ordering of all candidate distractors in response to a given question stem and answer key, such that the top ranked ones can be proposed as fitting distractors.

For the D-SIM model, since the considered (stem, key) pair and the distractor to be scored against it are independently fed to the network, the embeddings of the pool of distractors can be computed offline. The vector representation $h^{(sk)}$ of a given stem and its answer key is calculated, compared through the dot product with each of the pre-calculated distractor representations $h^{(d)}$, and these are then ordered according to decreasing score.

Similarly, for the Q-SIM model, the pool of questions' embeddings is calculated offline and stored. At run time, for a given question stem s, we compute its embedding $h^{(s)}$, score it against all pre-calculated stem representations for the MCQs

in the corpus, and rank the candidate distractors according to the decreasing score of their corresponding question stem. Note that we assign that same score to each of the distractors of a given stem (for use in DQ-SIM-score). We then rank all distractors according to decreasing scores (randomly ordering those with identical scores).

Finally, once the scores for D-SIM and Q-SIM are calculated for each candidate distractor, the DQ-SIM model can be evaluated directly, by ranking them according to the decreasing score $r^{\text{DQ-SIM-score}}$ or $r^{\text{DQ-SIM-rank}}$.

IV. Experimental Design

This section describes the evaluation methodology and the metrics we used to measure the quality of the generated distractors using the different methods described in Section III. Section IV-A introduces our hypotheses and the experiments we designed to test them. The automatic evaluation metrics we used are explained in Section IV-B.

A. Evaluation Setup

In order to validate our models' theoretical effectiveness and practical applicability, we formulate the following three key hypotheses, which we will test through experiments based on both automatic and human annotator evaluation:

- **Hypothesis 1**: *Context-aware models, based on generic pre-trained language models, lead to more effective distractor selection models than shallow prediction models based on manually engineered features.*
- **Hypothesis 2**: *Manual distractor quality scores are correlated with machine-generated distractor candidate rankings.*
- **Hypothesis 3**: *Top-ranked machine-proposed distractor candidates are comparable in quality to expert-generated distractors, for a given question stem and answer key.*

For Hypothesis 1, we first of all set up a large-scale automatic evaluation experiment with the Televic dataset (see Table I). In addition, a focused small-scale automatic evaluation of context-aware and feature-based models was carried out on the WeZooz external data (see Section III-B2 for details) that contains several subjects.

We complemented that automatic evaluation with human evaluation, since hard comparison of ground-truth distractors with machine-generated distractors may not give the whole picture of accuracy. Indeed, both for language learning and factual knowledge learning, MCQs can have a potentially large set of viable distractors that are not included by the gold standard distractor set. Thus, automated metrics could flag a correctly proposed candidate distractor as wrong because of the scarcity of the gold standard dataset. To avert this problem, many previous works asked human experts to judge the quality of the distractors that were generated by their systems [62], [63]. Hence, we also invited teachers to provide their expert opinion, each focusing solely on a set of questions on their own subject of expertise. In the following paragraphs, we explain the procedure we followed to set up that expert evaluation, which we will use in assessing all aforementioned Hypotheses 1–3.

First, we prepared a small sample of test questions for language and factual knowledge learning. For language learning, we used French and English. These questions were randomly drawn from the held-out test split of the Televic dataset introduced in Section III-B1. For the factoid type questions, we use the WeZooz dataset introduced in Section III-B2. Each of the subjects contains a fixed list of 50 questions. Second, we applied the different trained models to rank distractors according to their relevance for each question in the test set. We then kept the top-10 ranked candidate distractors for each of the models. Finally, teachers were shown distractor predictions unified over all models (i.e., duplicates were removed) as well as the provided gold-truth distractors for each test question (see the illustration provided in Fig. 4 in Appendix C). Note that the order of the unified list of distractors was randomized, to avoid introducing order bias.

The teacher participants were explicitly instructed to rate each candidate distractor based on how much they thought it would help them if they were given the task of preparing distractors for that specific question. Specifically, we asked them to annotate each distractor independently of the other distractors in the list, based on a four-level annotation scheme that we designed to measure the quality of distractors. Our scale is closely related to the three-point evaluation scale proposed by [63] (Table III shows examples of each category):

- *True Answer*: specifies that the distractor partially or completely overlaps with the answer key.
- *Good distractor*: specifies that the distractor is viable and could be used in an MCQ as is.
- *Poor distractor*: specifies that the distractor is on topic but could easily be ruled out by students. This could happen due to one or both of the following reasons.
  - *Poor meaning*: the distractor has poor meaning. For example, it is too general, although not completely off-topic.
  - *Poor format*: the distractor's format is different from the format of the answer key and does not fit with the stem.
- *Nonsense distractor*: specifies that the proposed distractor is completely out of context.

Although the third category (i.e., poor distractor) implies that the proposed distractor is ineffective as is, a minor tweak may result in a useful distractor. Furthermore, even if a significant change is required, it may inspire teachers to create new effective distractors.

Using the annotations we gathered from the teachers, we tested Hypotheses 2 and 3. For *Hypothesis 2*, we evaluated whether the higher ranked distractors also have a higher perceived usefulness. This was done by comparing the human scoring of distractor candidates in the top-5 to that of those ranked 5–10: for a good distractor generation model, the top-5 should on average contain significantly more 'good' ones. We designed a statistical analysis to test the null hypothesis that the rating distribution is not related to whether candidate distractors were ranked top-5 or 5–10. We used Fisher's exact

test[8] to test this hypothesis.

For *Hypothesis 3*, we evaluated the extent to which the teachers perceived the system-generated distractor candidates as the ground-truth distractors. Again, we use Fisher's exact test to test the null hypothesis that the distribution of quality of distractors is not related to whether the distractors are human-generated or system-generated.

B. Automated Metrics

We use two groups of information retrieval metrics to automatically evaluate our systems: (1) Order-unaware metrics: Recall@k and Precision@k, which measure the fraction of gold-standard distractors that are in the top-k distractors and the fraction of relevant distractors in the top-k retrieved distractors, respectively. (2) Order aware metrics: mean reciprocal rank (MRR) and mean average precision (MAP), which respectively reflect how high the most relevant item is ranked in the list, and how high all relevant ones are ranked on average.

V. Results and Discussion

In this section, we provide evidence of the effectiveness of our context-aware models by reporting the experimental results and discussing the insights gained. Section V-A compares the baseline with our proposed context-aware models using reproducible automated metrics (Hypothesis 1). Section V-B discusses the user study results with experts (Hypotheses 1–3). Note that all the numerical results reported in this section are in percentage points.

A. Automatic Evaluation

When considering the results of our automated evaluation based on the recovery of ground-truth distractors, it is essential to note that information about ground-truth distractors for a given item was never used during the model's training. Table IV shows the large-scale evaluation of the systems on the Televic test set. We report our results as the mean and standard deviation of five different runs of our models using five random seeds as shown in Table IV. All three context-aware models consistently outperform our feature based model (denoted 'baseline') on all metrics. DQ-SIM performs the best according to most metrics, confirming that Q-SIM and D-SIM have their own (complementary) merits. Q-SIM is better than D-SIM at recovering ground truth distractors (i.e., Recall@10 of 82.3 compared to 76.0), but inferior at ranking the best relevant distractor at the top in the list, which we conclude from the lower Precision@1 (40.4 vs. 44.9) and MRR (55.6 vs. 60.7) scores. This is related to the nature of the Q-SIM model. The candidate distractors belonging to its best matching question would be put at the top of the returned distractors in a random order. Our results show that D-SIM is better at estimating the most likely distractor than Q-SIM is in finding a relevant question *and* arriving with the relevant distractor on top after random ordering. However, the Precision@4 results show that Q-SIM has more success in identifying a question with good distractors, than D-SIM has in detecting good distractors among its top 4 results. The other reported metrics (Recall@10, Precision@4, MAP) indicate the overall higher effectiveness of Q-SIM when looking further than only the top result. In our MCQ generation setting, recall within the top 10 results is the more important metric, since the presence of high quality distractors in the automatically generated list is more important than their correct ranking.

[8]We also conducted a chi-square test and reached the same conclusions.

TABLE IV
Automatic ranking evaluation Full-ranking

Models	R@10	P@1	P@4	MAP	MRR
Baseline	71.3$\pm$1.2	21.1$\pm$1.8	23.7$\pm$0.5	33.5$\pm$1.0	43.9$\pm$1.9
D-SIM	76.0$\pm$0.7	**44.9$\pm$0.5**	24.4$\pm$0.8	44.9$\pm$0.6	60.7$\pm$1.3
Q-SIM	82.3$\pm$0.5	40.4$\pm$1.5	35.9$\pm$0.9	54.9$\pm$0.9	55.6$\pm$1.1
DQ-SIM	**91.7$\pm$0.6**	41.9$\pm$0.8	**38.2$\pm$0.7**	**57.3$\pm$0.5**	**62.8$\pm$0.4**

R:recall, P: precision, MAP: mean avg. precision,
MRR: mean reciprocal rank; evaluation on Televic test set.

Figure 3 depicts the performance of DQ-SIM for the two merging strategies, in terms of Recall@10 on the validation set described in Section III-D3. The linear combination of the scores outperforms the rank-based merging strategy. The score-based strategy achieves the best performance at $\alpha = 0.8$, giving more weight to the Q-SIM model. This is reasonable given that the Q-SIM model outperforms the D-SIM model on the recall metric.

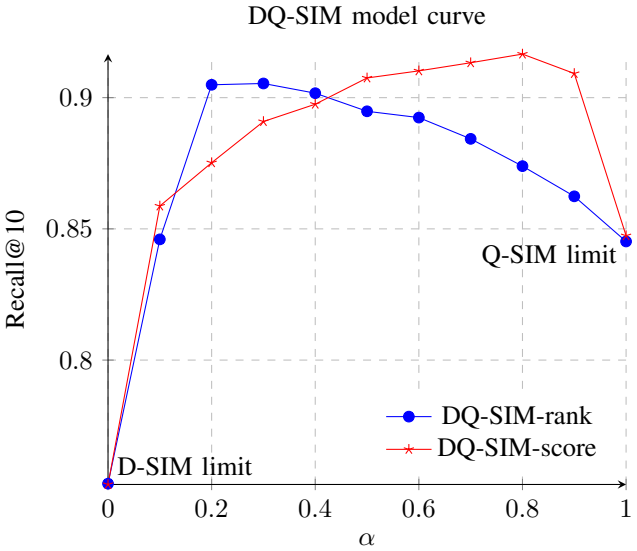


Fig. 3. Different α values for combining Q-SIM and D-SIM models using rank and raw scores on the validation set

Table V compares the baseline with DQ-SIM (i.e., the best context-aware model according to the evaluation on the Televic dataset) in a small-scale setting for all four Wezooz dataset subjects as well as English and French from the Televic test set. Since we want to compare models in terms of their ability to rank relevant distractors higher in the list, we added the ground-truth distractors from all the subjects to the existing distractors pool. Ideally, the best model would rank all the

TABLE III
ANNOTATION SCHEME EXAMPLES

Question	Answer	Distractors	Category	Moderation
Which inhabitants are not happy with Ethiopia's plans of the Nile?	Egyptians	1. Itali 2. Kenyans 3. gypsies	Poor format Good Poor meaning	because of wrong spelling. - because
My mum brought the washing in .... it was raining.	because	1. until 2. since 3. investigate	Good True Answer Nonsense	- out of context
How old was Beethoven when he died?	56 years	1. 1.5v 2. 60 years 3. 180 years	Nonsense Good Poor meaning	out of context - humans cannot live 180 years.

ground-truth distractors high in the list. Similar to the large-scale evaluation, DQ-SIM consistently outperforms the baseline for all subjects on both metrics. Recall@10 and MAP are higher for the language category than for factoid questions because the test questions for the former come from the same distribution (i.e., Televic test questions) as the data the models were trained on (i.e., Televic train set). On the other hand, the test data for the factoids come from a different distribution (i.e., WeZooz dataset) than the training datasuch that the evaluation for these subjects additionally measures the robustness of the model to a data distribution shift (i.e., its domain transfer abilities). The DQ-SIM model is far more robust than the baseline.

TABLE V
SMALL SCALE AUTOMATIC RANKING EVALUATION

Models	Baseline		DQ-SIM	
	R@10	MAP	R@10	MAP
English*	60.1	33.6	**98.3**	**85.8**
French*	46.6	17.7	**81.1**	**61.1**
Nat. Sciences	24.3	7.7	**74.3**	**37.3**
History	14.3	3.4	**62.2**	**35.7**
Biology	30.6	7.6	**72.0**	**41.8**
Geography	32.3	12.1	**61.5**	**34.4**

R: recall, MAP: mean avg. precision; * denotes subject is drawn from the Televic test set, while the rest are from WeZooz.

B. Expert Evaluation

Following the procedure introduced in Section IV, a total of 12,723 ratings for distractor quality were gathered from the annotations by teachers (see Table IX for details of rating statistics). These ratings come from the top-10 ranked distractors for each of the four models, and the ground-truth distractors (i.e., all simultaneously presented and randomly shuffled). We retained the gold standard distractors in the lists to be annotated, because we wanted to investigate the agreement among teachers in creating distractors. In the following subsections, we study teachers' (dis)agreement on the quality of distractors, compare the various models using the evaluation from experts, and revisit Hypotheses 1–3 in light of these results.

TABLE VI
INTER-ANNOTATION AGREEMENT OF GROUND-TRUTH DISTRACTORS (%)

	True Ans.	Good	Poor	Nonsense
Languages	5	70	14	11
Factoids	2	83	9	6
Overall	3	79	11	7

1) Inter-annotator agreement: We adopt 2 strategies to assess inter-annotator agreement. First, we analyze how teachers rated the ground-truth distractors, which were made by other teachers who prepared the questions. As can be seen from Table VI, in general, we find that 79% of the actual distractors were deemed good, 11% poor, followed by 7% nonsense and 3% true answers. There is greater agreement between teachers in what is considered a good distractor on the factoids than for language learning exercises (83% vs. 70%).

Second, we study the agreement of teachers by asking them to rate the same set of distractors using our four-level scale annotation scheme. We selected the subjects English, from the languages category, and History, from factoids, for annotations by at least two teachers. Table VII shows the inter-annotator agreement of teachers using the Jaccard similarity coefficient. The Jaccard similarity measures similarity between two sets of data by calculating what fraction of the union of those datasets is covered by their intersection. In our case, it is calculated as the number of times the teachers agreed on a distractor category label (i.e., one of the four quality labels) divided by the total number of distractors that were annotated (by either annotator) with that label. In general, we note a higher agreement on what is considered a good distractor and a nonsense distractor. Particularly, the overall agreement between the History teachers is higher than the English teachers. This is in line with the higher agreement for factoid type questions discussed in the previous paragraph. The Jaccard similarity is sensitive to small sample sizes. For example, a total of only two distractors were rated 'true answer' by the history teachers which yielded no similarity (i.e., a '0' in the first column in Table VII).

Calculating the inter-annotator agreement with the commonly used Cohen's kappa [64] value, we confirm afore-

TABLE VII
INTER-ANNOTATION AGREEMENT OF EXPERTS IN TERMS OF JACCARD SIMILARITY COEFFICENT (%)

Subjects	True	Good	Poor	Nonsense	Overall
English	25.8	42.9	12.8	40.0	47.9
History	0.0	43.6	24.3	59.7	57.7

TABLE VIII
EXPERT EVALUATION OF DISTRACTORS (%)

Models	Language learning		Factoid learning	
	GDR@10 ↑	NDR@10 ↓	GDR@10 ↑	NDR@10 ↓
Baseline	23.6	45.4	13.6	66.0
D-SIM	25.9	45.2	15.0	64.8
Q-SIM	26.3	45.3	19.0	61.6
DQ-SIM	**27.9**	**44.6**	**28.9**	**50.1**

GDR: good distractor rate, NDR: nonsense distractor rate;
↑: higher is better, ↓: lower is better; evaluation on WeZooz test set

mentioned higher agreement for factoid questions than for language: Cohen's kappa is 29.3 among English teachers, which represents "fair agreement", and 40.5 among History teachers, indicating "moderate agreement".

As a final metric to assess potential ambiguity in scoring distractors, we calculate conditional probabilities $P(X|Y)$ of having a second annotator assigning label X given that a first one said Y. For example, unsurprisingly, the probability of rating a distractor 'good' given that it was rated 'nonsense' by another teacher and vice-versa was 6% for English and 5% for History. This implies that the confusion in differentiating good distractors from nonsense distractors was minimal. Details are presented in Table XII in Appendix D.

2) Evaluation of models by experts: Table VIII shows the expert evaluation of distractors in terms of *good distractor rate* (GDR@10) and *nonsense distractor rate* (NDR@10). GDR@10 is calculated as the percentage of distractors that were rated 'good' among the top 10 ranked distractors for each model. Similarly, NDR@10 is calculated as the percentage of distractors that were rated 'nonsense' among the top 10 ranked distractors for each model. We are interested in reporting the NDR metric because (i) it could be used to distinguish between good and bad systems, and (ii) in a real-world scenario discarding a system with high NDR score could be helpful since the frequent occurrence of nonsense distractors may scare away users by eroding their trust in the model. The reported metrics are averages of all the subjects in each category. ↑ indicates larger values are better and ↓ indicates smaller values are better. In general, context-aware models were rated better in proposing plausible distractors than the baseline model. They also produced fewer nonsense distractors. The DQ-SIM outperformed all the other models. On average, 3 out of its top 10 proposed distractors were rated good distractors. Moreover, on average 5.5 distractors for languages and 5 for factoids were generally found on-topic (i.e., distractors rated as either good or poor distractors) for DQ-SIM.

The NDR@10 is lower for all models for language subjects than for factoid questions. We hypothesize this is because the test data for the language category comes from the same distribution the models were trained on.

3) Discussion of key hypotheses: We now discuss to what extent our experimental results confirm our aforementioned key Hypotheses 1–3.

Hypothesis 1 states that the context-aware models generate better quality distractors than the feature-based models. As discussed in Section V-A, the automated evaluation shows that the context-aware models consistently outperform the feature-based model on the Televic and WeZooz datasets. The human evaluation in Section V-B2 further confirms this by demonstrating that distractors generated by context-aware models were rated higher in quality than those generated by feature-based models.

Hypothesis 2 states that human distractor quality ratings are correlated with the automated candidate distractor rankings. To test this hypothesis, we collapsed the four ratings into two categories: *plausible* (i.e., rated as good distractors) and *less plausible* (i.e., rated as true answer, bad and nonsense distractors). Table X in Appendix D shows the contingency table for Fisher's exact test for our best model, i.e., DQ-SIM. The fraction of top-5 ranked distractors that received 'good distractor' ratings (i.e., 30.3%) is higher than that for the ones ranked 5–10 (i.e., 21.6%). We found that this difference is statistically significant. Indeed, the null hypothesis that the automatic ranking of distractors is unrelated to how teachers rated them is strongly rejected (p =1.7e-8).

Hypothesis 3 asserts that the quality of top-ranked machine-generated distractors is comparable with human-made distractors. To test this, we compare the distribution of ratings of the ground-truth distractors (i.e., expert-generated distractors) with the distribution of ratings for the DQ-SIM model (i.e., system-generated distractors). As for Hypothesis 2, we collapse the ratings into *plausible* and *less plausible* classes. Table XI in Appendix D shows the contingency table for Fisher's exact test, to compare the quality between system-generated and human-generated distractors. The null hypothesis that the source of the distractor (i.e., human-generated or system-generated) is unrelated to the quality label assigned by the teachers, is strongly rejected (p <1.e-10). Indeed, the quality of the human-generated distractors was found to be better than the system-proposed distractors. Still, we believe system-generated distractors have value: given that they can be generated quickly and automatically, presenting them as suggestions — rather than relying on a fully automated system — seems a practically meaningful way of working, which could save teachers a significant amount of time (compared to purely creating a list of distractors without any assistance).

VI. CONCLUSION AND FUTURE WORK

This paper introduced and evaluated multilingual context-aware distractor retrieval models for reusing distractor candidates that can facilitate the task of MCQ creation. Partic-

ularly, we proposed three models: (1) The D-SIM model that learns similar contextual representations for similar distractors, (2) The Q-SIM model that requires similar questions to have similar representations, and (3) The DQ-SIM model that linearly combines the previous two models benefiting from their respective strengths. Importantly, the DQ-SIM model showed a considerably reduced nonsense distractor rate, which we consider a useful asset in terms of trust in the model by teachers. We also asked teachers to evaluate the quality of distractors using a four-level annotation scheme that we introduced. As the result, teachers considered 3 out of 10 suggested distractors as high-quality, to be readily used. Additionally, they found two more distractors to be within topic, albeit of lower quality, and useful as inspiration for teachers to come up with their own good distractors. Finally, we released a test consisting of 298 educational MCQs with annotated distractors covering six subjects and a 77K distractor vocabulary to promote further research.

In future work, we foresee three directions. First, it is worth reiterating that the current work assumes access to a substantial pool of distractors. Even though with such large item pools, it is expected that many options are available for an incoming newly written question, the current work is unable to generate a brand new distractor. A possible solution could be to employ pure generative models that can freely generate distractors. Moreover, generative models could correct the 'poor format' errors. However, it has to be noted that such models require access to a context where the distractors and questions come from, such as a chapter of a book, Wikipedia article, etc. A second research direction is to extend the current work to a multimodal system that considers other sources of information, e.g., images that accompany MCQs in digital learning tools. Finally, an area that we are currently investigating is how to make sure the complete list of distractors in a single MCQ is sufficiently diverse: note that in the present study, we were only interested in retrieving a list of plausible distractors independent of each other. However, typical MCQ distractors should not only be plausible but also sufficiently diverse.

APPENDIX A
TRAINING AND IMPLEMENTATION DETAILS

- **Feature-based models**: for feature extraction and model training we use components from the scikit-learn package for python [65]. As negative training examples, we sample a total of 100 non-distractors for each MCQ.
- **Context-aware models**: our transformer based model is implemented using Pytorch [66] and Huggingface [67]. We initialize our encoder with `bert-base-multilingual-uncased`. We fine-tune the last two layers and leave the other layers frozen. The most important hyper-parameters are the learning rate, batch size, the duration of training, and the output width of our dense layer. To avoid extensive hyper-parameter tuning, we made the following choices. First, we choose the output dimension of the dense layer to be $d_{out} = 700$ because we empirically found that it yielded good results. For the learning rate, we kept the choice of 10^{-5} from Karpukhin *et al.* [60] in combination with the robust Adam optimizer [68]. Also in line with [60], we know increasing batch size may lead to slightly improved results, and thus decided the batch size to be 64, the highest value that would fit our V100 memory. We train each model for 25 epochs at which point performance on the development begins to plateau due to overfitting.

APPENDIX B
FEATURE VECTOR DESCRIPTION

We describe each feature we used to build our feature-based classifiers in below.

1) `tfidf_word_match_share` : a word overlap metric between both *k* & *d* and *s* & *d* which weighs overlapping words according to their inverse document frequency value.
2) `word_match_share`: fraction of word tokens that are shared between both *k* & *d* and *s* & *d*.
3) `equal_num` : boolean feature that checks whether *k* & *d* have equal numbers of digits.
4) `longest_substring` : fraction of longest matching sub-string between *k* and *d*.
5) `token_len_sim` : boolean feature that checks if the amount of tokens in *k* is equal with *d*.
6) `token_len_diff` : difference in amount of tokens in *k* and *d*.
7) `char_len_sim` : boolean feature that checks if the amount of characters in *k* is equal with *d*.
8) `char_len_diff` : difference in amount of tokens in *k* and *d*.
9) `is_caps` : boolean feature that checks if both *k* and *d* are capitalized.
10) `count_caps` : boolean feature that checks if both *k* and *d* have the same number of upper cased characters.
11) `has_num` : boolean feature that checks if the strings *k* and *d* have numbers.
12) `get_count` : absolute number of occurrences of *d* in our dataset.
13) `first_char_match` : boolean feature that checks if both *k* and *d* start with the same 5-gram characters.
14) `last_char_match` : boolean feature that checks if both *k* and *d* end with the same 5-gram characters.
15) `w2v_ad_sim` : a numeric feature that calculates the cosine similarity between the answer key and distractor using their word2vec representations.
16) `wmd_w2v_qd` : word mover's distance between the question and distractor using their word2vec vector representations.
17) `wmd_w2v_ad` : word mover's distance between the answer and distractor using their word2vec vector representations.
18) `glove_ad_sim` : the cosine similarity between the answer and distractor using their averaged glove embeddings.
19) `wmd_glove_ad` : the word mover's distance between the answer and distractor using their averaged glove embeddings.
20) `lang_prior` : the prior distribution of the source language of the question.

APPENDIX C
ANNOTATION PLATFORM

Figure 4 shows the annotation tool that we built for the quality annotation task by teachers. Each page presents a question, its actual answers, and a randomly shuffled list of candidate distractors that are proposed by the different models. Teachers assign quality labels to each of these proposed distractors by selecting one of the four radio-button options. If the teacher selects *poor distractor* as a label for a distractor, then a drop-down menu with two more options (i.e., *poor format* and *poor meaning*) is shown. Finally, the annotator/teacher can go to the following question by pressing the 'Next' button displayed at the left bottom of the screenshot.

TABLE IX
RATINGS DATA DESCRIPTION

Subjects	Item count	Dist. count		Ratings count		No of Raters
		Original dist	Proposed dist	Gold dist	Proposed dist	
English	48	130	723	260	1882	2
French	50	92	1148	92	1650	1
Geography	50	145	966	290	1960	1
History	50	130	1335	260	2420	2
Biology	50	88	1266	88	1761	1
Nat. Sciences	50	100	1407	100	1960	1
Total	298	685	6845	1090	11633	8

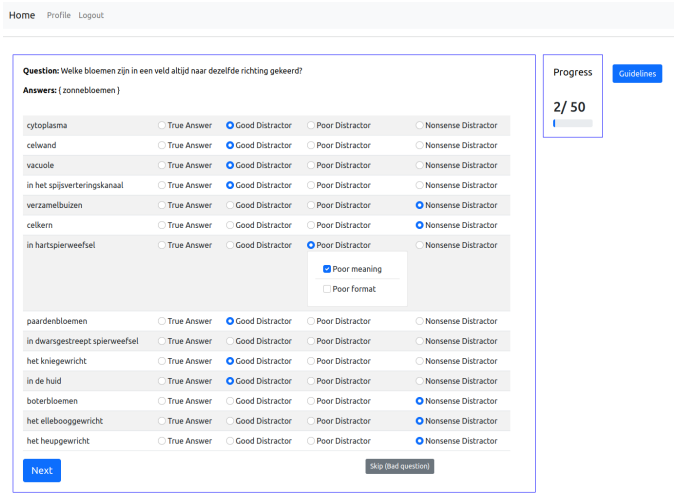


Fig. 4. Screenshot of the distractor annotation tool. The teacher is shown a question, an answer, and a shuffled list of ground-truth distractors & candidate distractor suggestions by all the models.

TABLE X
CONTINGENCY TABLE FOR AUTOMATIC RANKING & HUMAN RATING CORRELATION USING DQ-SIM

	Plausible	Less plausible
Ranked top 5	425	977
Ranked 5–10	303	1097

APPENDIX D
USER STUDY DETAILS

This section contains the user study details. Table IX describes the data gathered from the annotations provided by the teachers. Every subject has 50 questions except English, which had two duplicates that were later removed, leaving only 48 questions. On average, there are 2 distractors for each question item. We collected 1090 annotations for the original ground-truth distractor, and 11,633 annotations for the proposed candidate distractors (i.e., top ten ranked distractors by each of the four models). A total of 8 teachers participated in the study. English (i.e., from languages) and History (i.e., from factoids) were annotated twice by two different teachers for the purposes of calculating interannotator agreement.

TABLE XI
CONTINGENCY TABLE FOR COMPARING HUMAN & SYSTEM GENERATED DISTRACTORS

	Plausible	Less plausible
Human-generated	511	156
System-generated	255	412

TABLE XII
CONDITIONAL PROBABILITIES BETWEEN RATERS (AVERAGE OF BOTH DIRECTIONS)

Sub.	$gd \mid tr$	$gd \mid pf$	$gd \mid pm$	$gd \mid ns$	$pf \mid ns$	$pm \mid ns$
Eng.	35%	19%	44%	6%	11%	14%
His.	50%	22%	34%	5%	12%	6%

Table X shows the contingency table for hypothesis 2 that tests the correlation between automated distractor rankings (i.e., using our best model DQ-SIM) and human ratings using Fisher's exact test. The *plausible* column contains the count of distractors that were rated 'good' and the *less plausible* column the count of all distractors that were rated otherwise (i.e., 'poor', 'true answer' and 'nonsense' distractors). The rows indicate the count of top-5 ranked distractors and the 5–10 ranked distractors.

Table XI shows the contingency table for testing hypothesis 3 that compares the quality of human-generated with system-generated distractors. We use Fisher's exact test to test the hypothesis. The table shows counts of ratings in each category. For the human-generated row, we keep track of how each ground-truth distractor was rated, and update the counts depending on whether the distractors were rated 'good' (i.e., *plausible*) or the other labels (i.e., *less plausible*). Similarly, for the system-generated row, we count the ratings of top-k proposed distractors and update the counts in each column accordingly, where k is determined by the number of ground-truth distractors for that specific question.

Table XII illustrates the confusion observed between teachers in choosing which label to assign to a distractor. We show the confusion using conditional probabilities computed over both directions of the raters, where gd, tr, pf, pm, and ns denote good, true answer, poor format, poor meaning and

nonsense distractors respectively. For example, the first column (i.e., $P(gd \mid tr)$) shows the probability of rating a distractor 'good' given that it was rated 'true answer' by the other rater.

ACKNOWLEDGMENT

This work was funded by VLAIO ('Flanders Innovation & Entrepreneurship') in Flanders, Belgium, through the *imec-icon* project AIDA ('AI-Driven e-Assessment'). This research also received funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme. We would like to thank the AIDA partners Televic Education and WeZooz Academy for contributing data and use cases, as well as the ZAVO ('Zorgzaam Authentiek Vooruitstrevend Onderwijs') secondary school teachers for participating in the study.

REFERENCES

[1] J. Dunlosky, K. A. Rawson, E. J. Marsh, M. J. Nathan, and D. T. Willingham, "Improving students' learning with effective learning techniques: Promising directions from cognitive and educational psychology," *Psychological Science in the Public interest*, vol. 14, no. 1, pp. 4–58, 2013.

[2] M. J. Gierl, O. Bulut, Q. Guo, and X. Zhang, "Developing, analyzing, and using distractors for multiple-choice tests in education: A comprehensive review," *Review of Educational Research*, vol. 87, no. 6, pp. 1082–1116, 2017.

[3] B. G. Davis, *Tools for teaching*. John Wiley & Sons, 2009.

[4] W. Ma, O. O. Adesope, J. C. Nesbit, and Q. Liu, "Intelligent tutoring systems and learning outcomes: A meta-analysis." *Journal of educational psychology*, vol. 106, no. 4, p. 901, 2014.

[5] M. Liu, V. Rus, and L. Liu, "Automatic chinese multiple choice question generation using mixed similarity strategy," *IEEE Transactions on Learning Technologies*, vol. 11, no. 2, pp. 193–202, 2017.

[6] R. Mitkov, A. Varga, L. Rello *et al.*, "Semantic similarity of distractors in multiple-choice tests: extrinsic evaluation," in *Proceedings of the workshop on geometrical models of natural language semantics*, 2009, pp. 49–56.

[7] J. Pino, M. Heilman, and M. Eskenazi, "A selection strategy to improve cloze question quality," in *Proceedings of the Workshop on Intelligent Tutoring Systems for Ill-Defined Domains. 9th International Conference on Intelligent Tutoring Systems, Montreal, Canada*. Citeseer, 2008, pp. 22–32.

[8] A. Papasalouros, K. Kanaris, and K. Kotis, "Automatic generation of multiple choice questions from domain ontologies." *e-Learning*, vol. 1, pp. 427–434, 2008.

[9] A. Faizan and S. Lohmann, "Automatic generation of multiple choice questions from slide content using linked data," in *Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics*, 2018, pp. 1–8.

[10] J. Leo, G. Kurdi, N. Matentzoglu, B. Parsia, U. Sattler, S. Forge, G. Donato, and W. Dowling, "Ontology-based generation of medical, multi-term mcqs," *International Journal of Artificial Intelligence in Education*, vol. 29, no. 2, pp. 145–188, 2019.

[11] T. Alsubait, B. Parsia, and U. Sattler, "Generating multiple questions from ontologies: How far can we go?" in *Proceedings from the First International Workshop on Educational Knowledge Management (EKM 2014), Linköping, November 24, 2014*, no. 104. Linköping University Electronic Press, 2014, pp. 19–30.

[12] D. Coniam, "A preliminary inquiry into using corpus word frequency data in the automatic generation of english language cloze tests," *Calico Journal*, pp. 15–33, 1997.

[13] T. Goto, T. Kojiri, T. Watanabe, T. Iwata, and T. Yamada, "Automatic generation system of multiple-choice cloze questions and its evaluation," *Knowledge Management & E-Learning: An International Journal*, vol. 2, no. 3, pp. 210–224, 2010.

[14] J. Hill and R. Simha, "Automatic generation of context-based fill-in-the-blank exercises using co-occurrence likelihoods and google n-grams," in *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, 2016, pp. 23–30.

[15] G. Kumar, R. E. Banchs, and L. F. D'Haro, "Automatic fill-the-blank question generator for student self-assessment," in *2015 IEEE Frontiers in Education Conference (FIE)*. IEEE, 2015, pp. 1–3.

[16] Q. Guo, C. Kulkarni, A. Kittur, J. P. Bigham, and E. Brunskill, "Questimator: Generating knowledge assessments for arbitrary topics," in *IJCAI-16: Proceedings of the AAAI Twenty-Fifth International Joint Conference on Artificial Intelligence*, 2016.

[17] S. Jiang and J. S. Lee, "Distractor generation for chinese fill-in-the-blank items," in *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, 2017, pp. 143–148.

[18] C. Liang, X. Yang, D. Wham, B. Pursel, R. Passonneaur, and C. L. Giles, "Distractor generation with generative adversarial nets for automatically creating fill-in-the-blank questions," in *Proceedings of the Knowledge Capture Conference*, 2017, pp. 1–4.

[19] C. Liang, X. Yang, N. Dave, D. Wham, B. Pursel, and C. L. Giles, "Distractor generation for multiple choice questions using learning to rank," in *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*, 2018, pp. 284–290.

[20] M. Liu, V. Rus, and L. Liu, "Automatic chinese factual question generation," *IEEE Transactions on Learning Technologies*, vol. 10, no. 2, pp. 194–204, 2016.

[21] D. R. Ch and S. K. Saha, "Automatic multiple choice question generation from text: A survey," *IEEE Transactions on Learning Technologies*, vol. 13, no. 1, pp. 14–25, 2018.

[22] A. C. Butler, "Multiple-choice testing in education: Are the best practices for assessment also good for learning?" *Journal of Applied Research in Memory and Cognition*, vol. 7, no. 3, pp. 323–331, 2018.

[23] K. Woodford and P. Bancroft, "Using multiple choice questions effectively in information technology education," in *Ascilite*, vol. 4, 2004, pp. 948–955.

[24] A.-M. Brady, "Assessment of learning with multiple-choice questions," *Nurse Education in Practice*, vol. 5, no. 4, pp. 238–242, 2005.

[25] J. Collins, "Education techniques for lifelong learning: writing multiple-choice questions for continuing medical education activities and self-assessment modules." *Radiographics: a review publication of the Radiological Society of North America, Inc*, vol. 26, no. 2, pp. 543–551, 2006.

[26] R. Blackey, "So many choices, so little time: Strategies for understanding and taking multiple-choice exams in history," *The History Teacher*, vol. 43, no. 1, pp. 53–66, 2009. [Online]. Available: http://www.jstor.org/stable/40543353

[27] H. M. Abdulghani, F. Ahmad, M. Irshad, M. S. Khalil, G. K. Al-Shaikh, S. Syed, A. A. Aldrees, N. Alrowais, and S. Haque, "Faculty development programs improve the quality of multiple choice questions items' writing," *Scientific reports*, vol. 5, no. 1, pp. 1–7, 2015.

[28] N. Naeem, C. van der Vleuten, and E. A. Alfaris, "Faculty development on item writing substantially improves item quality," *Advances in health sciences education*, vol. 17, no. 3, pp. 369–376, 2012.

[29] T. M. Haladyna and S. M. Downing, "A taxonomy of multiple-choice item-writing rules," *Applied measurement in education*, vol. 2, no. 1, pp. 37–50, 1989.

[30] T. M. Haladyna and M. C. Rodriguez, *Developing and validating test items*. Routledge, 2013.

[31] R. Moreno, R. J. Martínez, and J. Muñiz, "Guidelines based on validity criteria for the development of multiple choice items," *Psicothema*, vol. 27, no. 4, pp. 388–394, 2015.

[32] R. Vyas and A. Supe, "Multiple choice questions: a literature review on the optimal number of options," *Natl Med J India*, vol. 21, no. 3, pp. 130–3, 2008.

[33] R. Mitkov, H. Le An, and N. Karamanis, "A computer-aided environment for generating multiple-choice test items," *Natural language engineering*, vol. 12, no. 2, pp. 177–194, 2006.

[34] R. Mitkov *et al.*, "Computer-aided generation of multiple-choice tests," in *Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing*, 2003, pp. 17–22.

[35] Y.-C. Lin, L.-C. Sung, and M. C. Chen, "An automatic multiple-choice question generation scheme for english adjective understanding," in *Workshop on Modeling, Management and Generation of Problems/Questions in eLearning, the 15th International Conference on Computers in Education (ICCE 2007)*, 2007, pp. 137–142.

[36] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, "Introduction to wordnet: An on-line lexical database," *International journal of lexicography*, vol. 3, no. 4, pp. 235–244, 1990.

[37] M. A. Lopetegui, B. A. Lara, P.-Y. Yen, Ü. V. Çatalyürek, and P. R. Payne, "A novel multiple choice question generation strategy: alternative uses for controlled vocabulary thesauri in biomedical-sciences education," in *AMIA Annual Symposium Proceedings*, vol. 2015. American Medical Informatics Association, 2015, p. 861.

[38] D. Seyler, M. Yahya, and K. Berberich, "Knowledge questions from knowledge graphs," in *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*, 2017, pp. 11–18.
[39] K. Stasaski and M. A. Hearst, "Multiple choice question generation utilizing an ontology," in *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, 2017, pp. 303–312.
[40] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy, "Race: Large-scale reading comprehension dataset from examinations," *arXiv preprint arXiv:1704.04683*, 2017.
[41] Y. Gao, L. Bing, P. Li, I. King, and M. R. Lyu, "Generating distractors for reading comprehension questions from real examinations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 6423–6430.
[42] H. Zhu, F. Wei, B. Qin, and T. Liu, "Hierarchical attention flow for multiple-choice reading comprehension," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.
[43] H.-L. Chung, Y.-H. Chan, and Y.-C. Fan, "A bert-based distractor generation scheme with multi-tasking and negative answer training strategies," *arXiv preprint arXiv:2010.05384*, 2020.
[44] B. Kulis *et al.*, "Metric learning: A survey," *Foundations and Trends® in Machine Learning*, vol. 5, no. 4, pp. 287–364, 2013.
[45] J. Welbl, N. F. Liu, and M. Gardner, "Crowdsourcing multiple choice science questions," *arXiv preprint arXiv:1707.06209*, 2017.
[46] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.
[47] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, "From word embeddings to document distances," in *International conference on machine learning*. PMLR, 2015, pp. 957–966.
[48] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
[49] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training," 2018.
[50] S. Edunov, M. Ott, M. Auli, and D. Grangier, "Understanding back-translation at scale," *arXiv preprint arXiv:1808.09381*, 2018.
[51] M. Zhong, P. Liu, Y. Chen, D. Wang, X. Qiu, and X. Huang, "Extractive summarization as text matching," *arXiv preprint arXiv:2004.08795*, 2020.
[52] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.
[53] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
[54] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.
[55] M. Guo, Y. Yang, D. Cer, Q. Shen, and N. Constant, "Multireqa: A cross-domain evaluation for retrieval question answering models," *arXiv preprint arXiv:2005.02507*, 2020.
[56] H. Khosravi, S. B. Shum, G. Chen, C. Conati, Y.-S. Tsai, J. Kay, S. Knight, R. Martinez-Maldonado, S. Sadiq, and D. Gašević, "Explainable artificial intelligence in education," *Computers and Education: Artificial Intelligence*, vol. 3, p. 100074, 2022.
[57] K. Sun, T. Yao, S. Chen, S. Ding, J. Li, and R. Ji, "Dual contrastive learning for general face forgery detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 2, 2022, pp. 2316–2324.
[58] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1. IEEE, 2005, pp. 539–546.
[59] N. A. Smith and J. Eisner, "Contrastive estimation: Training log-linear models on unlabeled data," in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, 2005, pp. 354–362.
[60] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, "Dense passage retrieval for open-domain question answering," *arXiv preprint arXiv:2004.04906*, 2020.
[61] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," *Advances in neural information processing systems*, vol. 29, 2016.
[62] A. Singh Bhatia, M. Kirti, and S. K. Saha, "Automatic generation of multiple choice questions using wikipedia," in *International conference on pattern recognition and machine intelligence*. Springer, 2013, pp. 733–738.
[63] J. Araki, D. Rajagopal, S. Sankaranarayanan, S. Holm, Y. Yamakawa, and T. Mitamura, "Generating questions and multiple-choice answers using semantic analysis of texts," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2016, pp. 1125–1136.
[64] M. L. McHugh, "Interrater reliability: the kappa statistic," *Biochemia medica*, vol. 22, no. 3, pp. 276–282, 2012.
[65] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.
[66] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
[67] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz *et al.*, "Huggingface's transformers: State-of-the-art natural language processing," *arXiv preprint arXiv:1910.03771*, 2019.
[68] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

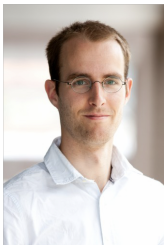
Semere Kiros Bitew is a PhD student at the Internet Technology and Data Science Lab (IDLab) at the Ghent University. He is part of the Text-to-Knowledge (T2K) Group. His supervisors are Prof. Chris Develder and Prof. Thomas Demeester. He received his master's degree in Data Science & Smart services from University of Twente, The Netherlands in 2018. His research interest includes AI in education, controlled text generation and psycholinguistics.

Amir Hadifar is a PhD student at the Internet Technology and Data Science Lab (IDLab) at the Ghent University. He is part of the Text-to-Knowledge (T2K) Group. His supervisors are Prof. Chris Develder and Prof. Thomas Demeester. He received his master's degree in Computer Engineering from University of Amirkabir Tehran, in 2018. His research interest includes AI in education and conversational systems.

Lucas Sterckx is a former member of the Text-to-Knowledge (T2K) Group. He obtained his Ph.D. in machine learning for natural language processing from Ghent University, Belgium, in 2018. His research focused on information extraction and weak supervision for natural language processing. He has participated in multiple research projects, developing automatic content enrichment systems for the media sector, tools clinical information extraction and more recently the medical sector.

Johannes Deleu received the Master of Science degree in computer science engineering from Ghent University, Belgium, in 2005. He is currently a Senior Research Engineer within the IDLab, Department of Information Technology, Ghent University-imec. His research concentrates on information extraction, machine learning, and in particular deep learning applied to natural language processing (NLP). He has participated in multiple research projects, developing automatic content enrichment systems for the media sector and more recently the education sector.

Chris Develder is associate professor with the research group IDLab in the Dept. of Information Technology (INTEC) at Ghent Universityimec, Ghent, Belgium. He received the MSc degree in computer science engineering and a PhD in electrical engineering from Ghent University (Ghent, Belgium), in Jul. 1999 and Dec. 2003 respectively (as a fellow of the Research Foundation, FWO). He has stayed as a research visitor at UC Davis, CA, USA (Jul.-Oct. 2007) and at Columbia University, NY, USA (Jan. 2013 - Jun. 2015). He was and is involved in various national and European research projects (e.g., FP7 Increase, FP7 C-DAX, H2020 CPN, H2020 Bright, H2020 BIGG, H2020 RENergetic, H2020 BD4NRG). Chris currently (co-)leads two research teams within IDLab, (i) UGent-T2K on converting text to knowledge (i.e., NLP, mostly information extraction using machine learning), and (ii) UGent-AI4E on artificial intelligence for energy applications (e.g., smart grid). He has co-authored over 200 refereed publications in international conferences and journals. He is Senior Member of IEEE, Senior Member of ACM, and Member of ACL.



Thomas Demeester is assistant professor at IDLab, at the Department of Information Technology, Ghent University-imec in Belgium. After his master's degree in electrical engineering (2005), he obtained his Ph.D. in computational electromagnetics, with a grant from the Research Foundation, Flanders (FWO-Vlaanderen) in 2009. His research interests then shifted to information retrieval (with a research stay at the University of Twente in The Netherlands, 2011), natural language processing (NLP) and machine learning (with a stay at University College London in the UK, 2016), and more recently to Neuro-Symbolic AI. He has been involved in a series of national and international projects in the area of NLP, and co-authored around one hundred peer-reviewed contributions in international journals and conferences. He is a member of AAAI and ELLIS.