Computers and Education: Artificial Intelligence 4 (2023) 100110

Contents lists available at ScienceDirect

Computers and Education: Artificial Intelligence

journal homepage: www.sciencedirect.com/journal/computers-and-education-artificial-intelligence

AI-based learning content generation and learning pathway augmentation to increase learner engagement

Chaitali Diwan [a,*], Srinath Srinivasa [a], Gandharv Suri [a], Saksham Agarwal [a], Prasad Ram [b]

[a] IIIT Bangalore, 26/C, Electronics City, Hosur Road, Bengaluru, 560100, India
[b] Gooru Inc., Redwood City, CA, United States

ARTICLE INFO

Keywords:
Learner engagement
Open educational resources
Curating learning pathways
Educational content generation
Language models
Definition generation
Automatic question generation
Multiple choice question

ABSTRACT

Retaining learner engagement is a major challenge in online learning environments, which is even more intensified with learning spaces increasingly built by combining resources from multiple independent sources. Narrative-centric learning experience has been found to improve learner engagement by several researchers. Towards this end, we propose an AI-based approach that generates auxiliary learning content called *narrative fragments* which are interspersed into the learning pathways to create interactive learning narratives. The proposed approach consists of the automatic generation of two types of narrative fragments– overviews of the learning pathway segments and reflection quizzes or formative assessments from learning resources in any format including open educational resources. The pipeline for the generation of the narrative fragments consists of various components based on different semantic models and a natural language generation (NLG) component based on a pre-trained language model GPT-2 (Generative Pre-trained Transformer 2). Automation enables the generation of narrative fragments on the fly whenever there are changes in the learning pathway due to the need for reiteration of concepts, pre-requisite knowledge acquisition, etc., enabling adaptability in the learning pathways. The proposed approach is domain agnostic which makes it easily adaptable to different domains. The NLG model is evaluated using ROUGE scores against several baselines. Automatically generated narrative fragments are evaluated by human evaluators. We obtained encouraging results in both cases.

1. Introduction

Learners and educators have been increasingly relying on the Internet for education. The prevalence of the COVID-19 pandemic has increased this reliance even more. While ubiquitous online learning platforms such as MOOCs (Massive Open Online Courses) have massively benefited the student community at large (Khalil & Ebner, 2014), online learning also suffers from issues such as high dropout rates (Siemens, 2013), dearth of learner motivation, and lack of engagement (Aldowah et al., 2020; Borrella et al., 2019; Hartnett et al., 2011; Paas et al., 2005; Simpson, 2013; Wang et al., 2019).

While MOOCs generalise on the familiar experience of classroom lectures in the online space, there are several attempts to rethink and reinvent learning experiences for online learning. Technologies like ITS (Intelligent Tutoring System) recognise the potential for enhancing learner engagement by curating personalised learning pathways. However, a main source of disengagement comes from managing learner transitions across disparate resources (Brusilovsky & Henze, 2007; Diwan et al., 2018). This becomes even more acute when learning resources are sourced from multiple sources, with each of them having different exposition styles (Brusilovsky & Henze, 2007; Diwan et al., 2018). Moreover, these learning environments are course specific and require significant human effort to build them. This makes it difficult to use them across different subjects without incurring the high cost of re-building.

Formulating the learning experience in the form of an interactive "narrative" or "story" narrated by the learning environment to the learner is found to improve learner engagement by several researchers (Laurillard et al., 2000; Marsh et al., 2011; Plowman et al., 1999). Students discern a narrative more easily when the narrative conforms to the generic narrative expectations of a beginning, a middle and an end (Plowman et al., 1999). Interactive narratives are between two extreme ends of learning– that of *narrative guidance* which is teacher driven, and *narrative construction* which is student driven (Plowman et al., 1999).

Keeping that in view, we propose to address the issue of learner engagement by:

* Corresponding author.
E-mail address: chaitali.diwan@iiitb.ac.in (C. Diwan).

https://doi.org/10.1016/j.caeai.2022.100110
Received 18 February 2022; Received in revised form 14 November 2022; Accepted 17 November 2022
Available online 7 December 2022

1. Making the learning pathways interactive.
2. Formulating the learning pathways as narratives conforming to a generic narrative structure.

Towards this end, we propose a suite of AI-based approaches to automatically generate learning content and add the auto-generated learning content to the learning pathways at appropriate positions. In other words, the interactive narrative sequences that conform to generic narrative expectations are constructed by interspersing learning pathways with proposed auto-generated learning content called *narrative fragments*. The proposed approach builds such interactive narratives in a domain agnostic way. It can generate narrative fragments for learning resources in any format, unlike many AI-based learning environments which require learning resources in a particular format. This opens up the possibility of utilising a large pool of available open learning resources.

Learning pathways are a static sequence of learning resources which can either be curated manually or automatically using the approaches outlined in Diwan et al. (2019); Shmelev et al. (2015); Chi (2009). The learning pathways are the baseline macro-structures which are first segmented to form micro-structures or sub-goals. Each sub-goal starts with an outline or a sketch of the segments in the form of a proposed narrative fragment called *Overview* (beginning of the narrative), followed by consuming the learning resources required to reach the sub-goal (middle of the narrative) and closes with reflecting on the same through the proposed narrative fragment called *Reflection quiz*.

Overviews provide insights about what to expect in the next segment of the learning pathway and help the learner understand the larger picture about the topic which is later clarified by the learning resources in the pathway. Since Overviews provide information and meta-information about the pathways, they also help a learner make an informed choice about continuing on the pathway, thus reducing the chances of dropping out at a later stage. *Reflection quizzes* are formative assessments that help in gauging the learners' knowledge. While Reflection quizzes are useful to assess recall of specific information or the learners' knowledge of the information, they also help in deciding if the learner can continue further on the pathway or take a different learning pathway.

While it is possible to create the proposed narrative fragments manually, it takes an enormous amount of time and effort to create such content manually. It also requires a domain expert. The automatic generation of narrative fragments not only reduces the expenses associated with manual construction but also satisfies the need for a continuous supply of new learning content.

Apart from this, the AI-based approach for the generation of learning content enables adaptive learning by generating content dynamically according to the learner driven changes. For example, if a learner changes the pathway (in case a pre-requisite needs to be learnt, or when the depth of knowledge is desired, etc.), or explores a different learning resource around the same subject, the Overviews and Reflection quizzes are automatically generated on the fly for the new adapted learning pathway.

Automatic generation of Reflection quizzes also makes it possible to generate a large pool of quizzes that can be used in rotation, enabling variety in assessing recall of different knowledge points in the learning pathways.

Our proposed approach of Narrative fragments generation is based on NLP (Natural Language Processing) and NLG (Natural Language Generation). The pipeline for the generation of both the narrative fragments consists of various components based on different semantic models and a Natural Language Generation component based on a pre-trained language model GPT-2 (Generative Pre-trained Transformer 2) (Radford et al., 2019). GPT-2 is known to generate natural language text that appears realistic and is a coherent continuation of the provided input. It achieves state-of-the-art performance on many language modelling benchmarks, hence we chose GPT-2 to build our NLG model for creating the narrative fragments.

Fig. 1 shows a depiction of a learning pathway with different segments and the addition of narrative fragments at the segment boundaries.

The proposed approach can be applied across different corpora of learning resources and learning pathways. The adaptability to different domains is very helpful in domains that do not have well defined learning content or learning pathways, such as educational resources in a niche area of study, or for a field that is emerging.

The proposed suite of AI-based approaches that generate narrative fragments is evaluated using automatic metrics and human evaluators. The GPT-2 based NLG model (Definition generator) is evaluated using the ROUGE metrics (Lin, 2004) which measures n-grams overlap between the generated natural language text and the ground truth. The learning pathway segments are evaluated by comparing the human generated segments with automatically generated segments of the learning pathways. Narrative fragments are evaluated by human evaluators. We obtained encouraging results in all the cases.

2. Related work

In this section, we discuss related work in the area of engagement in large scale learning environments and automatic learning content generation.

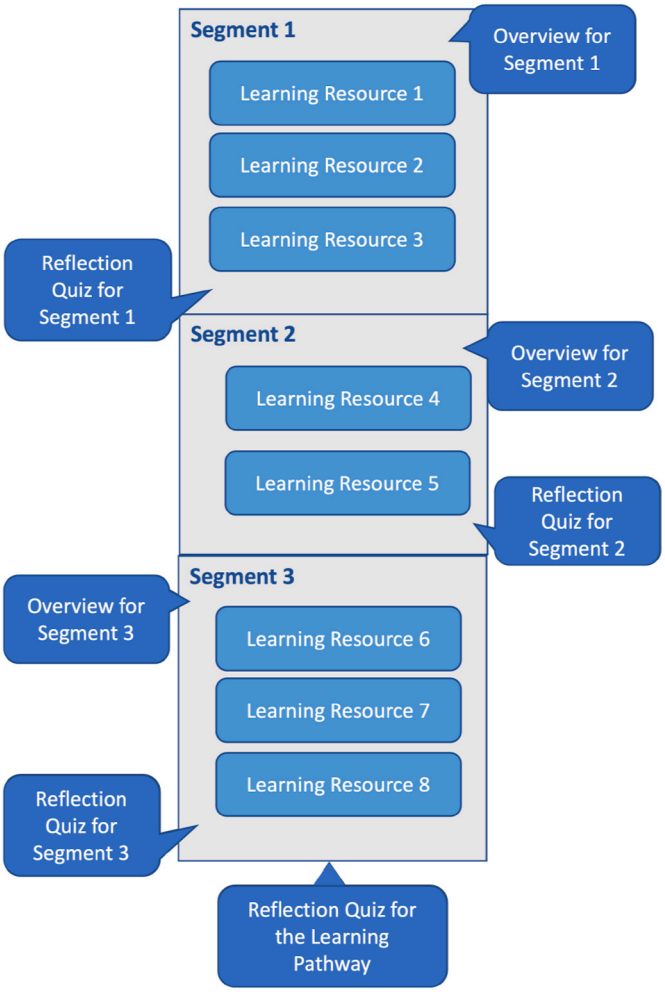

Fig. 1. Learning Pathway showing different segments and addition of narrative fragments.

2.1. Engagement in learning environments

Various approaches are proposed to increase learner engagement by adding external interventions to the learning pathways based on the learners' activities (Anderson et al., 2014; Borrella et al., 2019; Ortega-Arranz et al., 2018). Borrella et al. (2019) propose an approach to identify learners at risk of dropping out from a course and send out tailored encouragement emails to at-risk learners. Anderson et al. (2014) study different ways of characterising students' engagement with MOOCs and propose the design and deployment of badges to produce incentives for activity and contribution. Ortega-Arranz et al. (2018) present a proposal of a system supporting the design, implementation and enactment of redeemable rewards.

While these approaches provide interventions to increase learner engagement, they are independent of the learning content. Whereas our proposed suite of AI-based approaches generates learning content based on the text of the learning resources and adds them to the learning pathways, thereby creating interactive learning narratives which help in increasing student engagement (Laurillard et al., 2000; Plowman et al., 1999).

Recently, there have been studies in the direction of integrating learning analytics based student models of ITS into MOOCs to increase learner engagement (Aleven et al., 2015; Baneres et al., 2016). However, constructing pedagogically effective tutoring systems has been widely recognised as a challenging problem (Murray, 2003; Sottilare et al., 2012). The complexity increases manifold when the large scale of MOOCs is considered.

2.2. Automatic learning content generation

In this work, we propose the automatic generation of two types of learning content: Overviews and Reflection Quizzes. We discuss related work around both these areas in the remaining part of the section.

Our proposed approach of Overview generation might resemble with that of summarisation (El-Kassas et al., 2021; Gupta & Gupta, 2019; Lewis et al., 2019; Paulus et al., 2017; Radford et al., 2019; Raffel et al., 2020) or multi-document summarisation (Haghighi & Vanderwende, 2009; Kumar & Salim, 2012; Lebanoff et al., 2018; Liu & Lapata, 2019), since both Overview generation and summarisation shorten and abstract out information from a substantial sized input source. However, Overviews have a different purpose. They provide an outline of what the learner can expect to learn in the learning pathway. Unlike summaries, Overviews need not be complete in their coverage but should be able to provide important bits of information or glimpses of the upcoming learning pathway.

As far as we know, the Overview generation of the learning pathways or a sequence of learning resources is a unique concept and is unexplored.

Most of the works on learning content generation discuss automatic question generation. Automatic question generation is itself a well-studied topic, however, it is still a challenge to automatically generate good quality questions from a given text, more so, when the text is non-standardised or noisy as could be the case with the open educational resources.

We refer to comprehensive reviews on automatic question generation (Ch & Saha, 2018; Kurdi et al., 2020; Pan et al., 2019). Kurdi et al. (2020) present a detailed review of various works on automatic question generation in an educational setting. Pan et al. (2019) discuss the neural network based automatic question generation in different domains including education. Ch and Saha (2018) present a survey on automatic Multiple Choice Question (MCQ) generation from the text. Even though many different methods for automatic question generation are studied in these reviews, such as rule-based, template based and neural networks based, most of the works mentioned in these reviews are based on extracting a single sentence, multiple lines or a text span from the input paragraph and converting such a "question worthy" sentence to a question. Other approaches use all the sentences in a learning resource to generate questions and rank the generated questions.

Approaches to automatic question generation that are neural network based (Chan & Fan, 2019; Liu et al., 2020; Pan et al., 2019; Steuer et al., 2020) and graph based (Chen et al., 2019; Lukovnikov et al., 2019), use various methods such as pre-trained transformers over knowledge graphs (Lukovnikov et al., 2019), Reinforcement Learning and Graph Neural Networks (Chen et al., 2019), answer identification techniques (Steuer et al., 2020) and GPT-2 (Liu et al., 2020) for sentence or span selection. The sequence to sequence model using BERT proposed in (Chan & Fan, 2019) implicitly identifies question-worthy spans within the passage using the attention mechanism (Pan et al., 2019). The selected span or sentences are then converted to questions.

The disadvantage of sentence or span extraction from the learning resources is that whenever there is no "question appropriate sentence" in the learning resource, there is no question generation. Steuer et al. (2020) claim that only a fraction of such statements can generate a question. This is an even bigger problem in open learning resources such as ours, which may have text with a lot of noise and in a non-standard format. Another advantage of our method of generating definitions for question formation is that the generated definitions may have a different vocabulary and a different way of defining the same concept as compared to the text in the learning resources, which is an important aspect in generating assessments.

Other methods proposed in Kurdi et al. (2020) include sentence simplification rules such as triples extraction and generation of the required sentence using parse rules, and sentence classification into categories, selecting and re-framing them into question appropriate sentences. While the sentences generated by these approaches are semantically relevant, the appropriateness of syntax or sentence formation is not so satisfactory.

In our work, we propose a Natural Language Generation model to generate a question worthy sentence instead of relying on selecting a sentence that can be transformed into a question. As far as we know, this is a unique method of question generation and has not been studied in the literature. The advances made in NLG with language models such as GPT-2 have opened up the possibility of generating text in the desired format, thus enabling content generation for any given educational resource.

In our proposed approach of narrative fragment generation, the learning resources can be in any format that can be converted to text such as videos, PDFs, presentations, notes, blogs, interactive web etc. They need not be in a particular/standard format or need any additional meta-data, unlike many AI-based learning environments which require learning resources in a particular format.

Automatic question generation for a specific domain is more prevalent than generating questions in a domain-agnostic manner. The aforementioned neural network models are trained and evaluated on the SQuAD dataset (Stanford Question Answering Dataset) (Rajpurkar et al., 2016) which is a reading comprehension dataset with question-answer pairs. Hence, the limitation here is that the questions are generated for the reading comprehension domain, unlike our model which can generate questions for any domain.

There are approaches for automatic question generation using pre-trained sequence to sequence model (Tamang et al., 2022) and graph based models (Chung & Hsiao, 2022; Ghosh et al., 2022) in the domain of programming languages. While these models have promising results, they are specific to programming languages. Parasa et al. (2022) discuss a template based method for riddle generation based on KDTrees and binary search algorithm, but the approach can be used for a limited number of domains which have concepts with clearly defined attributes.

Closest to our model of question generation are the methods proposed by Lopez et al. (2021) and Bhat et al. (2022) which use pre-trained language models GPT-2 and T5 (Raffel et al., 2020) for question generation. Lopez et al. (2021) propose question generation using GPT-2 on a repurposed SQuAD database, however, most of the generated

questions seem to be simply pulled from the given context, since the SQuAD database contains 88.26% identification type questions in the training set. Bhat et al. (2022) propose a method to automatically generate questions using T5, however, the questions mostly lead to descriptive answers, unlike the MCQ type of questions proposed by our model which has different challenges and applications.

Approaches based on ontology address automatic question generation across domains (Bühmann et al., 2015; Vega-Gorgojo, 2019; Vinu et al., 2015). Vega-Gorgojo (2019) proposes a semi-automatic template based tool called Clover Quiz for automatic MCQ generation using SPARQL queries for data extraction from DBPedia. However, this approach requires a human editor to guide the data extraction process, sometimes requiring the writing of SPARQL queries which requires technical knowledge. Another semi-automatic approach based on Linked Open Data is proposed by Bühmann et al. (2015) which uses entity summarisation techniques for content selection and RDF verbalization for question generation. Vinu et al. (2015) propose a method that uses terminological axioms in the ontology such as existential, universal and cardinality restrictions on concepts and roles for MCQ generation.

Apart from the fact that the aforementioned ontology based approaches are semi-automatic, one more disadvantage of using ontologies for automatic question generation is that creating ontologies requires significant human efforts for the domains which are not defined in the available open ontologies such as DBPedia.

Another contribution of our work in automatic question generation is our unique approach to the automatic generation of "distractors"– that are incorrect but tantalising answer choices, meant to help in testing the learner's comprehension. Our approach describes the generation of two types of distractors, one being semantically closer and the other being syntactically closer to the correct answer, while not being too close to the correct answer and maintaining the clarity of the correct options.

3. Proposed approach

In this section, we discuss the proposed approach for learning pathway segmentation, the proposed NLG model for *Definition Generation* and the proposed approach for automatic generation of narrative fragments - *Overviews* and *Refection quizzes* which use the *Definition Generation* model in their generation pipeline. The implementation details are discussed in the next section. In this work, we specifically consider learning pathways that are curated from open learning resources. (The solution can easily be applied to the learning pathways created by a single platform; however, it has lesser challenges than the open source resources.)

3.1. Learning pathway segmentation into regions

To be able to segment the learning pathways semantically, it is necessary to find the breakpoints that indicate a semantic shift in topical focus. These would be the places where the different narrative fragments can be meaningfully integrated.

Our approach to automatic segmentation of the learning pathways uses sentence embeddings where the sentences are embedded in a shared multi-dimensional semantic vector space. First, the canonical text-based transcripts are generated for all the learning resources of all the learning pathways in the corpus. The transcripts are then summarised into a few sentences using an extractive graph-based text summariser.

In a learning pathway consisting of a sequence of learning resources, all the learning resources (summaries) are then encoded using Universal Sentence Encoder (USE) (Cer et al., 2018). Cosine similarity scores are then computed between the embeddings of the consecutive learning resources in the learning pathways. Whenever the cosine similarity between two consecutive resources dips below a certain threshold, the learning pathway is segmented at that point and the next learning resource in the learning pathway is considered as the beginning of a new segment.

We choose to use USE among the various sentence embedding techniques since it is the most promising sentence embedding technique as studied by Perone et al. (2018). The threshold for segmentation is obtained by computing cosine similarities between consecutive resource pairs of learning resources for all the pathways in the corpus. Fig. 2 shows the graph obtained by plotting the cosine similarities between consecutive resource pairs for all the learning pathways in our science corpus. Here, the threshold is considered near the peak of the graph.

3.2. Definition Generator

In this sub-section, we discuss our proposed NLG model called *Definition Generator* which generates natural language text in the form of definitions for any given keyword or concept. The definitions generated by this model are important for creating both the proposed narrative fragments.

Definition Generator model is built by fine-tuning the pre-trained language model GPT-2 (Generative Pre-trained Transformer 2) (Radford et al., 2019). GPT-2 is a large transformer-based language model trained on 40 GB of Internet text, with a simple objective of predicting the next word given all the previous words within that sequence. Transformer is a Neural Network architecture that aims to solve sequence-to-sequence tasks while handling long-range dependencies with ease (Vaswani et al., 2017).

Fine-tuning of the model is done by training the model in a supervised way on our dataset that contains the input and output pairs of keywords and their definitions. By fine-tuning, the model will learn the format of the current task of definition generation and the context that is currently important, while keeping its long-term memory, i.e., its sense of grammar, structure, style, etc., that it has learned from the unsupervised training.

Formally, each data-point for training is represented as $d_i(x_i, y_i)$, of the dataset $D(X, Y)$ with 'n' data-points, where $x_i \in X$ represents the keyword and $y_i \in Y$ represents the definition. The input and output are separated using a separator # and concatenated by $e =< |endoftext| >$ at the end, such that data-point is now $d_i = x_i \# y_i e$.

For simplification of the objective representation, we substitute each token of the data-point d_i with the notation u_i, such that $d_i = u_1, u_2, \ldots, u_{L_t}$ where L_t is the length of the data-point. We maximise the objective as defined in Eq. (1).

$$P(D) = \prod_{i=1}^{n} \prod_{j=1}^{L_t} P\left(u_j | u_1, u_2, .., u_{j-1}; \theta\right) \tag{1}$$

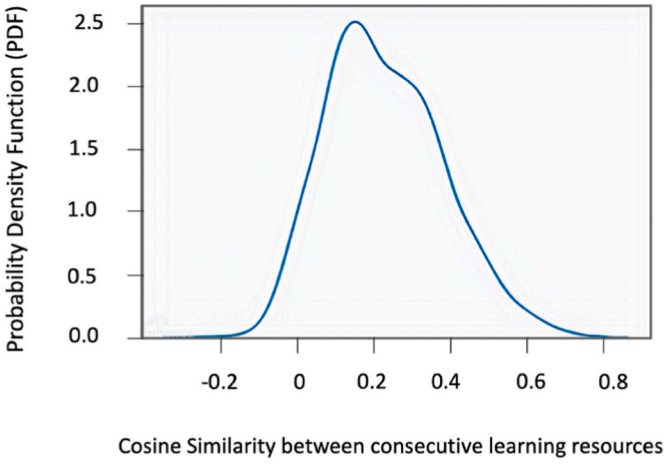

Fig. 2. Cosine similarity between consecutive resource pairs in the learning pathways.

Table 1
Examples of generated definitions.

Key-phrase	*Definition*
Atmosphere	Atmosphere is the layer of gasses around a given planet.
Aquaculture	Aquaculture is the branch of science that involves breeding, rearing of fish, shellfish, algae, and other organisms.
Fossils	Fossils are the remains of organisms that have been preserved through time and space.

Here, $P(u_j|u_1 \ldots u_{j-1}; \theta)$ is the conditional probability of obtaining u_j given the previous tokens $u_1 \ldots u_{j-1}$, and θ represents the model parameters.

The model is optimised on the cross-entropy loss as defined in Eq. (2).

$$Loss = -\sum_i p_i \log_2 p_i^{'} \tag{2}$$

where $p_i^{'}$ is the probability generated by our model for the word i and p_i is the ground truth probability of the word i.

For a given keyword, the *Definition Generator* model starts generating the words in the definition one word at a time till the end of the text or max length is reached. Every previously generated word is fed back to the model which adds to the context for the generation of the next word. At every timestep, the *Definition Generator* model outputs a soft-max probability distribution over a vocabulary. From this distribution, a word must be sampled which would be the next word in the definition. For this, we use temperature sampling (Ackley et al., 1985), which reshapes the probability distribution of the word with a temperature factor t, where t is between 0 and 1. Lower temperature results in a less random choice of the next word (0 temperature is equivalent to argmax or max likelihood) and higher temperature allows increased randomness in choosing the next word. We experimented with different temperatures to see the suitability for our application (Details are in Section. 4.1.1).

Table .1 shows a few examples of the definitions generated by our *Definition Generator* model for some key-phrases.

3.3. Overview Generation

Overview provides insights about what to expect in the next segment of the learning pathway. *Overview* consists of the topical anchor (Rachakonda & Srinivasa, 2009) of each learning resource in the learning pathway segment, the overlaying theme of the segment along with a short description of it and meta-data elements like the number of resources in the segment, order of the learning resources etc.

Fig. 3 shows the pipeline of *Overview* Generation. In the first step, key-phrases and the topical anchors are extracted for all the resources in the learning pathway segment. Next, the main theme encapsulating all the learning resources is found. Then a simple description for the main theme is generated using the GPT-2 based *Definition Generator* which is passed to the *Definition Selector* to select the best fit definition. This information and the computed meta-data are passed on to the *Overview Generator* which selects an appropriate template and generates the *Overview*.

3.3.1. Key-phrase extraction

Selecting the correct key-phrases that represent a given learning resource is very crucial for generating the *Overview*. Hence, we evaluate various key-phrase extraction methods such as *EmbedRank* (Bennani-Smires et al., 2018), *TextRank* (Mihalcea & Tarau, 2004), *TF-IDF*, *Noun keyphrase extraction* and *BERT based Key-phrase Extraction* on our corpus. *BERT based Key-phrase Extraction* method performed well on our corpus. (Details of the comparison of the different key-phrase extraction techniques are discussed in Section.4.2.). Among the keywords extracted, the top ranked key-phrase represents the *topical anchor* for the learning resource. Top ranked 'n' key-phrases for a learning resource are used to find the main theme for the learning pathway segment as described in the next section.

3.3.2. Finding the main theme

The *main theme* for a learning pathway segment is the concept that collectively represents the concepts in each of the learning resources. To identify the *main theme* for a set of learning resources present in the segment, first the semantic neighbourhood for the learning resources is obtained. The semantic neighbourhood consists of all the key-phrases representing the learning resource along with the set of terms that are in the vicinity of the learning resource in the semantic space. The reason for obtaining the neighbourhood is that the key-phrases are based only on the text of the learning resource, however, there could be terms that the learning resource is about but they do not appear in the learning resource. For example, refer to example 1 of Fig. 4; here all the learning resources discuss about renewable energy but the term renewable itself does not appear in any of the learning resources (summaries), however, it appears in the semantic neighbourhood of the learning resource.

To get the semantic neighbourhood, first, a *Dictionary* is created which consists of all the unigrams, bigrams and phrases obtained from the learning pathway. Then all the key-phrases representing the learning resources and the terms in the learning pathways forming the Dictionary are embedded within the shared multi-dimensional vector space. The embeddings capture semantic relatedness via the distances or the cosine similarity between the corresponding vector representations. For each keyword in the learning resource, the top-n tokens of the dictionary which have the highest cosine similarity with the keyword are obtained, and the unique tokens among them are added to the neighbourhood set. Refer to *getNeighbourhood()* function in Algorithm. 1.

Once the neighbourhood is obtained, the main theme is determined by taking the intersection of the neighbourhoods of all the learning resources. We again use Maximal Marginal Relevance (MMR) (Carbonell & Goldstein, 1998) method to select the main theme. Maximal Marginal Relevance maximises the relevance and novelty in the finally retrieved

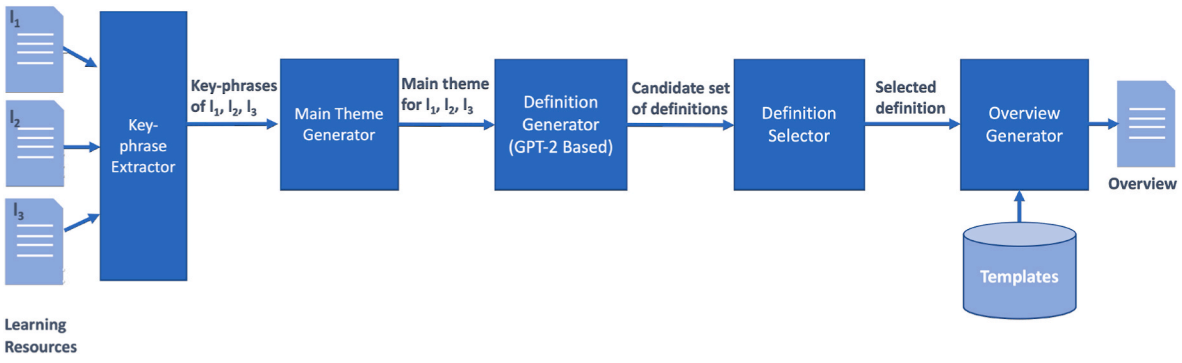

Fig. 3. Pipeline of *Overview Generation*.

top-ranked themes. Here, the new candidates which are similar to the key-phrase, but not similar to the already selected phrases are iteratively selected. Finally, the root-keyphrase which represents the main theme of the learning pathway segment is the key-phrase with maximum average cosine similarity with all the learning resources in the segment. Refer to *getAvgSemanticSimilarity()* function in Algorithm. 1. In case of the intersection set being null, the key-phrases that occur in the highest number of learning resources are considered and among them, the key-phrase that has the highest average cosine similarity with all the learning resources in the segment is chosen. Algorithm. 1 outlines the steps to find the main theme.

Algorithm 1. Pseudo code for finding the Main Theme.

Input: Set of all learning resources (summary) in the segment $LR\{\}$
Output: Main theme
Require: Set of key-phrases $K = \{K_1\{\}, K_2\{\}, ...\}$ for all learning resources, Dictionary D of all tokens in a learning pathway
Set each Neighbourhood set $N_1\{\}, N_2\{\}, N_3\{\}...$ to $\emptyset$;
foreach *keyword set* $K_i\{\} \in K$ **do**
  $N_i \leftarrow getNeighbourhood(K_i\{\})$;
$intersection-set\{\} \leftarrow N_1 \cap N_2 \cap N_3 \cap ...N_n$;
if $intersection-set\{\}$ *is not null* **then**
  foreach *term* is_k *in* $intersection-set\{\}$ **do**
    $ranking[] \leftarrow getAvgSemanticSimilarity(is_i, LR\{\})$;
  return is_k *with* $max(ranking[])$
else
  foreach *term* t_i *in* $N_1 \cup N_2 \cup N_3 \cup ...N_n$ **do**
    $common(t_i) \leftarrow$ number of sets in N having t_i;
  $max-count-keywords\{\} \leftarrow max(common((t_i))$;
  if $count(max-count-keyword\{\}s) > 1$ **then**
    foreach *term* t_k *in* $max-count-keywords\{\}$ **do**
      $ranking[] \leftarrow getAvgSemanticSimilarity(t_k, LR\{\})$;
    return t_k *with* $max(ranking[])$
Function *getNeighbourhood*$(K\{\})$
  Input: Dictionary $D \leftarrow$ learning pathway tokens
  vectorize all tokens in Dictionary D ;
  /* i^{th} token represented as v_{i_i} */
  foreach *key-phrase* k_i *in K{}* **do**
    $v_{k_i} = vectorize(k_i)$;
    foreach *token* t_i *in D* **do**
      cs[] $\leftarrow$ cosine-similarity(v_{k_i}, v_{l_i});
      sort cs[] by descending order;
      selected-tokens $\leftarrow$ choose top 'n' tokens;
  foreach *token* st_i *in selected-tokens* **do**
    if st_i *is not in* N **then**
      N $\leftarrow st_i$;
  return N
Function *getAvgSemanticSimilarity*$(c_i, LR\{\})$
  $v_{c_i} = vectorize$(candidate token c_i);
  foreach $lr_i \in LR\{\}$ **do**
    $v_{lr_i} = vectorize$(learning resource lr);
    $simAvg+ = cosine-similarity(v_{c_i}, v_{lr_i})$;
  return $simAvg/size(lr\{\})$

3.3.3. Definition Selector

For the *main theme* of the segment identified in the previous step, a definition or a simple description is generated using the *Definition Generator* (Section. 3.2) model. However, *Definition Generator* is probabilistic, i.e., it generates the next word in the text based on the probability over a set of tokens. Hence, the generated definition may not be the same for every inference or call to the model. Also, some generated definitions could be better than others. Hence in our proposed model, we call *Definition Generator* 'n' number of times and all the generated definitions form a candidate set. A relevant definition is chosen from this set.

Algorithm 2 outlines the pseudo code for this. To start with, GPT-2 based *Definition Generator* is called to generate a definition for a given key-phrase. The *Definition Generator* is called five times to obtain five definitions for the key-phrase. And as discussed above, one of the definitions is selected among these.

We do not have any definitive answer for the choice of generating *'five'* definitions for each keyword. This is because the generative model chooses the next word based on the probability over a set of words and the number of higher probability words may be more or less depending on the occurrences of the key-phrases in different contexts. Hence for some keywords, we may get just one or two unique definitions and for some, a lot more. However, given the fact that inferencing a model many times strains the system resources, we restrict to calling the *Definition Generator* at the most 5 times. From this candidate set, the following two types of generated definitions are eliminated.

1. The definitions that have the key-phrases repeated in them in the original form or in one of their morphological forms. For example, consider the following sentence generated by our *Definition Generator- "Acceleration is accelerating an object to a speed that is greater than the object's normal speed."*. Here the definition consists of the repetition of key-phrase in its morphological form - *accelerating*. We speculate that these types of definitions could be generated for the

Example 1

Learning Resource 1: Geothermal power plants use steam to produce electricity. The steam comes from reservoirs of hot water found a few miles or more below the earth's surface. The steam rotates a turbine that activates a generator, which produces electricity.

Learning Resource 2: A solar farm is a large collection of photovoltaic (PV) solar panels that absorb energy from the sun, convert it into electricity and send that electricity to the power grid for distribution and consumption by customers like you. Solar farms — which you'll sometimes see being called solar parks or photovoltaic power stations — are usually mounted to the ground instead of rooftops and come in all shapes and sizes.

Learning Resource 3: Tidal power is one of the major renewable energy sources, but also one of the most infantile. What are some tidal energy advantages and disadvantages to consider when looking to invest in this relatively green energy source? Using the power of the tides, energy is produced from the gravitational pull from both the moon and the sun, which pulls water upwards, while the Earth's rotational and gravitational power pulls water down, thus creating high and low tides.

Overview: In this section, you will learn about Renewables. The Renewables are about the energy production from the natural energy source. There are 3 learning resources in this segment. Specifically, you will learn about geothermal energy, renewable energy from solar farm and tidal energy.

Example 2

Learning Resource 1: Article on religion in Mesopotamia. Utu - The god of the sun as well as justice and the law, Utu is drawn holding a saw like instrument. Her primary city was Uruk, but she was also prominent in the city of Babylon. The Babylonian version of Utu Ea - Same as Enki Marduk - god of Babylon by Unknown Assyrian Gods Ashur (Assur) - The primary god of the Assyrians. He was also the god of war and married to the goddess Ishtar.

Learning Resource 2: Reference article on ancient Egyptian legal systems - They were responsible for court cases involving small claims and minor disputes. Egyptian women were also allowed to seek justice, and like men could have their day in court. The head of the legal system was officially the pharaoh, who was responsible for enacting laws, delivering justice, and maintaining law and order, a concept the ancient Egyptians referred to as Ma'at. Beginning in the New Kingdom, oracles played a major role in the legal system, dispensing justice in both civil and criminal cases.

Learning Resource 3: In both Sumer and Babylon, there was an unusual form of government that came pretty close to an early form of democracy. There was a king and nobles who made the laws and declared war and decided how to honor the gods. Then there was an assembly of wise men, elected by the people, who could overrule the king and say, this is not a good law, get rid of it; or the assembly might say we don't want to go to war, so stop it. # Sumerian Laws: The Sumerians did not, to our knowledge, write down their laws. The king passed a law, and everyone was expected to learn it and obey it.

Overview: This section discusses about Mesopotamian civilization. Mesopotamian civilization existed between the 5th century and the 10th century. There are 3 learning resources in this segment. In the first learning resource, you would learn about Mesopotamia religion, then, you will learn about ancient Egypt legal system, and finally, you will learn about Sumerian Government. Let's get started!

Example 3

Learning Resource 1: Wavelength is the distance between two consecutive and equivalent points on a wave. Watch the animation to see examples of wavelength. Natural waves come in many different wavelengths, covering a vast range relative to human senses of scale. Gamma rays which are a form of electromagnetic radiation have wavelengths as short as one trillionth of a meter.Tsunami waves can have wavelengths greater than 100 miles (161 km).

Learning Resource 2: When the energy of a wave passes through the medium, particles of the medium move. Wave amplitude of a longitudinal wave is the distance between particles of the medium where it is compressed by the wave. Wave amplitude is the maximum distance the particles of the medium move from their resting positions when a wave passes through. Review Define wave amplitude. What is the amplitude of the transverse wave modelled in the Figure below if the height of a crest is 3 cm above the resting position of the medium?

Learning Resource 3: Practice Wavelength and Amplitude - This handout walks you through step by step how to find the wavelength or amplitude from a graph and provides several practice questions at the end.

Learning Resource 4: The SI unit for wave frequency is the hertz (Hz), where 1 hertz equals 1 wave passing a fixed point in 1 second.Wave Frequency and Energy The frequency of a wave is the same as the frequency of the vibrations that caused the wave. You can see examples of different frequencies in the Figure below (Amplitude is the distance that particles of the medium move when the energy of a wave passes through them.)[Figure3] Summary Wave frequency is the number of waves that pass a fixed point in a given amount of time.Explain how wave frequency is related to the energy of a wave.

Overview: In this segment, you would be learning about the waves. Waves are a type of electromagnetic oscillation. This segment has 4 learning resources. The different concepts covered in this segment are wavelength, wave amplitude, practice wavelength and amplitude, wave frequency. Let's get started!

Fig. 4. Examples of generated Overviews for some learning pathway segments.

key-phrases that have a smaller number of co-occurring terms in the training data.

To identify such definitions and eliminate them, a set K consisting of key-phrases occurring in all forms is built. For example, if the keyword is acceleration, its morphological forms like accelerate, accelerating, accelerates, accelerated etc., are included in this set. This is done by stripping and appending the keyword and checking for its validity as a word using a language dictionary. This step is the opposite of stemming and lemmatizing in NLP. This refers to the function *KeyphraseInAllForms ()* in Algorithm. 2.

2. The definitions in which the generated definitions do not match the given/input key-phrase are also eliminated. We see such discrepancies mostly for bi-grams, where sometimes the definition gets generated only for the first part of the key-phrase. For example, if the key-phrase is *Nash Equilibrium* and the definition generated by our model is for *Equilibrium*, it is not accepted.

Once we filter out the above two types of unacceptable definitions from the candidate set, the next step is to filter out the definitions that may not be relevant in the context. This may happen with homograph words. The definitions may be generated for different contexts. To eliminate such cases, we define a semantic similarity threshold and choose the definitions which are above the threshold. To choose the threshold γ for the definition acceptance, we conducted experiments. Refer to Section. 4.1.2 for details. The semantic similarity is the average of the semantic similarity of the generated definition and the learning resources in the segment. Refer *getAvgSemanticSimilarity()* function in Algorithm. 1. If this average is above the threshold, such a definition is an "acceptable definition" and if it is below the threshold, that definition is dropped from the candidate set.

Definitions are then ranked in descending order of their semantic similarity to the learning resources considered. Whenever there is a semi-automatic deployment, the teacher/instructor can choose the definition among the ranked ones. In the case of a fully automated deployment, the one with the highest semantic similarity is chosen.

Algorithm 2. Definition Selector Pseudo Code

Input: learning resource set - lr{ }, key-phrase - k
Output: Selected Definition
Threshold γ is set;
set of candidate definitions {C} $\leftarrow \emptyset$;
for *i=1 to i=5* **do**
  {C} $\leftarrow DefinitionGenerator()$;
set $SemanticSim[] \leftarrow emptydictionary$;
keyword set $K \leftarrow \emptyset$;
$K \leftarrow KeyphraseInAllForms(k)$;
foreach $c_i \in C$ **do**
  if *candidate definition c_i has repetition of keywords in any form as in set {K}* **then**
    drop the definition from {C}
  if *candidate definition c_i is not generated for exact key-phrase* **then**
    drop the definition from {C}
  $SemSim[c_i] \leftarrow getSemanticSimilarity(k_i, lr\{\})$;
  if $SemSim[c_i] < \gamma$ **then**
    drop the definition;
return $max(SemanticSim[])$

3.3.4. Overview Generator

The final step is to use all the information generated in the above steps and generate the *Overview*. This is done using templates. There are three categories of templates for generating Overviews. In the first category, there is an availability of complete information required for the generation of the overviews such as identified topical anchors for all the learning resources, the main theme of the segment, acceptable definition of the main theme, and meta-data of the learning pathways. In this category, there are different templates with differences in text and information arrangement adding to the variety of the generated Overviews. When all the required information is available, one template is chosen randomly among the templates of the first category. The second category of templates is for the segments where the learning resources have the same or similar topics. In this case, meta-information if available is selected, for example, the type of learning resource such as video, presentation, picture etc., is used. If that is not available, the next best keyword that differentiates the learning resource is chosen. The third category of the template is for the learning resources which do not have generated descriptions/definitions for the main theme, i.e., none of the generated definitions for the main theme are acceptable (above threshold), and the candidate set of definitions is null. Such cases are generally considered outliers, nevertheless, they must be handled in an automatic/semi-automatic system.

Fig. 4 shows examples of generated Overviews for the learning resources for a few learning pathway segments. The summary for the learning resources is presented along with the *Overview* that is generated for the same.

3.4. Reflection quiz generation

Here, we discuss the proposed methodology for the automatic generation of Reflection quizzes in the form of *Multiple Choice Questions (MCQs)*.

The first step in generating questions is to identify domain-specific terms which serve as *'anchors'* of the learning resources. The definitions are then generated for these *'anchors'* using the GPT-2 based *Definition Generator* model (Section 3.2). To keep the test item comprehensible and avoid additional complexity, the questions are generated from these definitions using simple transformational rules which, in turn, result in only minimal change of the original wordings.

Fig. 5 shows the component diagram of Multiple Choice Question generation. First, the key-phrases are extracted from the learning resources of the learning pathway region. For each of these key-phrases, a set of candidate definitions are obtained using *Definition Generator*. Among these candidate definitions, the *Definition Selector* filters out the unaccepted definitions and then ranks the definitions. The selected definition is then passed to the *Question Generator* module. Finally, the *Options Generator* module is called to generate options for the generated questions.

The components *Key-phrase Extractor*, *Definition Generator* are the

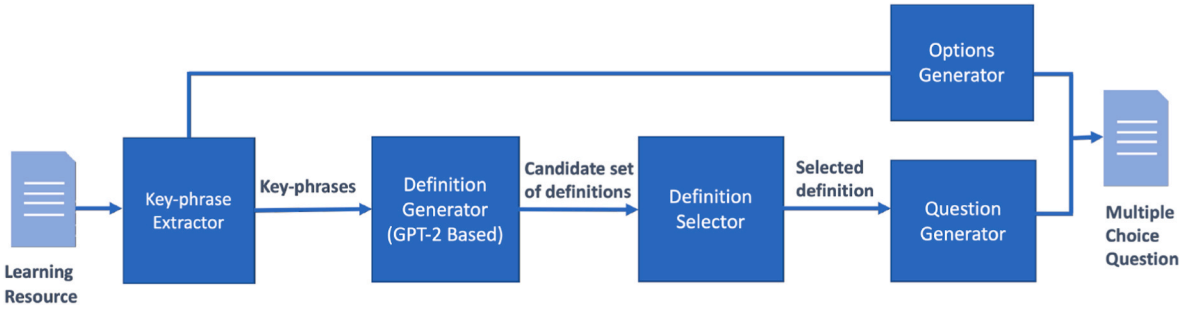

Fig. 5. Pipeline of multiple choice question generation.

same as in the Overview Generation ((Refer Section. 3.3.1 and 3.2). Definition Selector is also similar to the component by the same name in Overview generation (Refer Section. 3.3.3), except in the computation of semantic similarity used in the selection and ranking of the definitions. Semantic similarity in the Reflection quiz is computed by taking a cosine similarity between the sentence embeddings of the generated definition and the learning resource summary, whereas, for Overview generation, it is computed by taking an average of cosine similarities between the sentence embeddings of the generated definition and all the learning resource summaries. The computation of semantic similarity for the Reflection quiz is outlined in Algorithm. 3. Please note, whenever we mention semantic similarity in this paper, we refer to Algorithm. 3, unless a different specific algorithm is mentioned.

Algorithm 3. Semantic Similarity Pseudo Code.

FUNCTION getSemanticSimilarity(c_i, **lr**)
$v_{c_i} = vectorize$(candidate definition c_i);
$v_{lr} = vectorize$(learning resource lr);
return $cosine - similarity(v_{lr}, v_{c_i})$

The definitions are ranked in descending order of their semantic similarity to the learning resource. Whenever there is a semi-automatic deployment, the instructor can choose the definition that will be converted to a question, among the ranked definitions. In the case of a fully automated deployment, the definition with the highest semantic similarity to the learning resource is chosen for question generation.

3.4.1. Question Generator

As stated before, our methodology does not need elaborate transformational rules for question generation because the definitions that are generated by our model are closer to our training data which represents simple declarative sentences for a given *anchor*.

Most of the definitions that are generated by our *Definition Generator* can be used as is, to convert to questions. However, there are two types of definitions that need modifications. These are explained below:

1. If there are appositives in the definition such as short forms, full forms or aliases or "also known as" in the definition. Then these appositives are removed from the definitions.
2. If the definitions have domain information before the definition, for example: "In Science, <keyword> is ….". Here, the domain information part is removed.

A sentence is converted to a question by replacing the key-phrase with "What" and by adding a question mark at the end of the sentence by replacing the ".". Finally, the grammar correction package is called to correct the questions or flag grammatical errors if any.

3.4.2. Options Generator

We generate two types of options or distractors – a distractor that is semantically closer to the anchor and a distractor that is syntactically similar to the anchor. The anchor itself is the correct answer. All the options are randomly shuffled to create the MCQ.

To identify a distractor that is semantically similar to the key-phrase or the anchor in consideration, word embeddings are used to obtain the semantic neighbourhood (as described in Overview Generation pipeline Section. 3.3.2) and top 'n' semantically closest words are picked from the semantic neighbourhood.

The distractor that is syntactically similar to the anchor is identified using Levenshtein distance (Levenshtein et al., 1966). The Levenshtein distance is a metric to measure how apart are two sequences of words. In other words, it measures the minimum number of edits that you need to do to change a one-word sequence into the other. Levenshtein distance is used to create a candidate set of 'n' syntactically similar tokens and one token is chosen randomly from this set as a distractor. If there is an overlap in the candidate set of semantically similar distractors, the overlapped distractor is removed from the syntactically similar distractors candidate set. Also, if a partial part of the distractor is in the anchor word, this distractor is dropped from the candidate set as well.

The tokens that are considered for neighbourhood formation can either be from only the learning pathway or the entire domain corpus (Tokens that are used to compare with key-phrase or the anchor.). Usually, if the distractors are generated on an unseen corpus (where the training corpus is different from the corpus used for narrative fragments generation), it is preferred to use the tokens from the learning pathway to get context specific distractors.

Fig. 6 shows a few examples of Multiple Choice Questions generated by our model. For each example, a summary of a learning resource is shown, followed by the question and the options generated by our approach. An important point to note here is, even though the quality of the scrapped data and consequently the summarises may not be very good, the quality of questions generated by our module is not affected since we do not extract but generate the "question worthy sentences".

3.4.3. Fill in the blanks or Match the following

The definitions generated by our model are also used to create *Fill in the blanks* and *Match the following* type of Reflection quizzes. For *Fill in the blanks*, the *"anchor"* in the selected definition is replaced by a blank. Similarly, *Match the followings* are created by randomly arranging the 'n' definitions for a learning pathway segment by separating the *anchors*

Example 1

Learning Resource: Mutation, an alteration in the genetic material (the genome) of a cell of a living organism or a virus that is more or less permanent and that can be transmitted to the cell's or the virus's descendants. (The genomes of organisms are all composed of DNA, whereas viral genomes can be of DNA or RNA; see heredity: The physical basis of heredity.) Mutation in the DNA of a body cell of a multicellular organism (somatic mutation) may be transmitted to descendant cells by DNA replication and hence result in a sector or patch of cells having abnormal function

Question: What is the term for the genetic changes of an organism that are passed down from generation to generation?

Correct Option: mutation

Semantic Option 1: deleterious

Semantic Option 2: gene expression

Syntactic Option: computation

Example 2

Learning Resource: An ecosystem includes all of the living things (plants, animals and organisms) in a given area, interacting with each other, and also with their non-living environments (weather, earth, sun, soil, climate, atmosphere).These may depend on non-living things like water, sunlight, turbulence in the puddle, temperature, atmospheric pressure and even nutrients in the water for life.(Click here to see the five basic needs of living things) This very complex, wonderful interaction of living things and their environment, has been the foundations of energy flow and recycle of carbon and nitrogen.Anytime a stranger (living thing(s) or external factor such as rise in temperature) is introduced to an ecosystem, it can be disastrous to that ecosystem.This is because the new organism (or factor) can distort the natural balance of the interaction and potentially harm or destroy the ecosystem.

Question: What is the term for a collection of organisms that interact with other organisms and the environment?

Correct Option: ecosystem

Semantic Option 1: biodiversity

Semantic Option 2: habitat

Syntactic Option: system

Example 3

Learning Resource: This physics video explains the concept of nuclear fission reaction by illustrating an example of nuclear fission of Uranium 235 atom. Nuclear fission is nuclear reaction process in which nucleus, when bombarded with a neutron, splits into smaller parts, often producing free neutrons, and releasing a very large amount of energy. One of the most important applications of nuclear fission reactions in creating chain reactions. When a free neutron hits the nucleus of a fissile atom like uranium-235 (235U), the uranium splits into two smaller atoms called fission fragments, plus more neutrons. Fission can be self-sustaining because it produces more neutrons with the speed required to cause new fissions. This creates the chain reaction.

Question: What is a form of self sustaining nuclear reaction that occurs under conditions of an increase in the rate of a reaction?

Correct Option: nuclear fission

Semantic Option 1: oxidation

Semantic Option 2: decay

Syntactic Option: nuclear explosion

Example 4

Learning Resource: The smallest particle of an element or compound that can exist in the free state and still retain the characteristics of the element or compound: the molecules of elements consist of one atom or two or more similar atoms; those of compounds consist of two or more different atoms. Most molecules are far too small to be seen with the naked eye, although molecules of many polymers can reach macroscopic sizes, including biopolymers such as DNA. Molecules commonly used as building blocks for organic synthesis have a dimension of a few angstroms to several dozen, or around one billionth of a meter. Single molecules cannot usually be observed by light (as noted above), but small molecules and even the outlines of individual atoms may be traced in some circumstances by use of an atomic force microscope. Some of the largest molecules are macromolecules or super molecules.

Question: What is the smallest part of something that retains the chemical properties of the whole?

Correct Option: molecule

Semantic Option 1: atom

Semantic Option 2: electron

Syntactic Option: granule

Fig. 6. Examples of generated MCQs for some learning resources.

Table 2
Average ROUGE scores for various models on test set.

Model	R-1	R-2	R-L
Seq2Seq With Attention	0.1946	0.2076	0.1911
Zero shot GPT-2	0.1976	0.0947	0.1783
Our model	0.2311	**0.2325**	**0.2526**
Our model- input with context	**0.2565**	**0.2311**	**0.2501**

and their *definitions*.

Since our method has definitions that are easy to convert to any format, we are able to create three different formats of Reflection quizzes, thus adding variety.

4. Implementation and experiments

4.1. Definition Generator - implementation, experiments and results

In this section, we discuss the dataset, implementation details and evaluation of the *Definition Generator* model.

4.1.1. Dataset

We use the Wikipedia dataset for our experiments since Wikipedia has an impressive breadth and depth of topical coverage and Wikipedia's content is extensively used in education.[1]

We use the pre-processed Wikipedia dataset (Scheepers, 2017) [2] developed for NLP and ML research. The dataset contains the topics and their definitional summaries for Wikipedia pages, with the first sentences being the definitions of the topics. The dataset contains 5,315,384 data-points. We choose a subset of it that belonged to subject *Science* which consisted of 200000 keyword definition pairs. We split the dataset into train set (80%) and test set (20%) and sampled 80,000 data-points from the train set for training and 10,000 data-points from the test set for testing. We did not use all the data-points for training since there was no change in the model performance with a larger dataset. The same sampled test data-points were used for testing our model and the baseline models.

4.1.2. Implementation

GPT-2 is available in 3 sizes – small, medium and large with 124M, 355M and 774M parameters respectively. We use the large size GPT-2 model with 774 Million parameters for building our *Definition Generator* model. We use a Python package gpt-2-simple[3] which contains a wrapper and generation scripts over the work of Radford et al. (2019). Following are the important hyper-parameters used by our model: Learning rate - 0.0001, epochs - 4 and batch size - 1000. All the necessary hyper-parameters mentioned were selected based on the best performing parameters on the validation set. The model was trained on a free instance of Google Colab[4] and it took roughly 2 hours on a single server.

The training data used to fine-tune the model consisted of data-points in the format of topic/keyword followed by "#" followed by the definition, followed by $e =< |endoftext| >$. i.e., "<keyword>#<definition>e". During inference, the keyword is concatenated with "#" as a prefix, such that the input is "<keyword>#". The definition is generated till "$< |endoftext| >$" or till the length of 40, whichever is earlier.

We explored another variant of our model in which there is an addition of context information to the input to the *Definition Generator* model. For this experiment, we used the same fine-tuned model as described above, however, during inference, we added the context of the learning resource before the keyword. The input thus becomes "<context><keyword>#". The results of the same are reported ahead with the label *Our model-input with context*.

Table 3
Examples for the sentence pair types.

Sentence Pair Type	Sentence 1	Sentence 2
Neutral	The young boys are playing outdoors and the man is smiling nearby	A group of kids is playing in a yard and an old man is standing in the background
Contradiction	The young boys are playing outdoors and the man is smiling nearby	There is no boy playing outdoors and there is no man smiling
Entailment	The young boys are playing outdoors and the man is smiling nearby	The kids are playing outdoors near a man with a smile

4.1.3. Baselines

We use the following baselines to compare with our *Definition Generator* model.

1. **Sequence to Sequence Network with Attention** Sequence to Sequence Networks with Attention has been established as state of the art approaches in sequence modelling and transduction problems such as language modelling and machine translation tasks (Cho et al., 2015; Hermann et al., 2015; Vaswani et al., 2017). Hence, we consider this model as one of the baselines. The model consists of the encoder containing the stacked LSTM (Long Short Term Memory) layers. The Decoder is a single LSTM layer. To predict the definition of the i^{th} token of the output, the Decoder's hidden state representing the $(i-1)^{th}$ token in the predicted definition, goes through the multiplicative attention layer along with the hidden states from the Encoder. The multiplicative attention layer generates a score for each hidden state. The hidden state with the highest score along with the Decoder hidden states are passed through a fully connected layer to predict the i^{th} token. We build upon the code implemented by Arunachalam and Thomas.[5] This model was trained using the same train set which was used for our *Definition Generator* model.
2. **Zero-shot evaluation with GPT-2** In this baseline, we use GPT-2 pre-trained model as it is, without fine-tuning it to our dataset. While the GPT-2 model which is fine-tuned for a specific task and domain achieves superior results, the pre-trained GPT-2 model is also capable of zero shot task transfer. GPT-2 zero-shot achieved state of the art performance on 7 out of 8 tested language modelling datasets (Radford et al., 2019). Hence, we use this as another baseline. We use the large sized GPT-2 pre-trained language model for evaluation.

4.1.4. Results

We evaluate our model and its variant by comparing it to the baseline algorithms as described above. We use ROUGE (Lin, 2004) metrics for comparison since it is the standard evaluation metric for the NLP tasks such as ours. The metric essentially depends on n-gram overlaps between the generated text and the actual text with ROUGE measuring recall. ROUGE-1, ROUGE-2 and ROUGE-L compute the overlap of unigrams, bigrams and Longest Common Subsequences (LCS) respectively between the ground-truth text and the generated text.

The results on the test set are reported in Table 2. We can see from Table 2 that our model and its variant, both have higher ROUGE scores as compared to the baseline models. ROUGE-1 scores are slightly higher for the context based input variant compared to the original model. However, ROUGE -2 and ROUGE-L scores are the same (up to two digits

[1] https://en.wikipedia.org/wiki/Wikipedia#Cultural_impact.
[2] https://github.com/tscheepers/Wikipedia-Summary-Dataset.
[3] https://github.com/minimaxir/gpt-2-simple.
[4] https://colab.research.google.com.
[5] https://github.com/jananiarunachalam/Research-Paper-Summarization.

Table 4
Finding threshold.

Sentence Pair Type	BERT Based Sentence Transformer		USE	
	SNLI	SICK	SNLI	SICK
Neutral	0.56636818	0.57651554	0.490319760	0.57167034
Entailment	0.66832686	0.90320616	0.57890686	0.87279555
Contradiction	0.35880584	**0.44881147**	0.35300264	0.51279218

decimal part) for our model when input is with or without context.

We speculate that our model had a comparable performance with or without the context in input on the test set, since the model was trained on the same domain, albeit the test set was unseen. However, the addition of context would make a positive difference when inferencing on a new unseen domain. (Refer Section.6.)

4.1.5. Choosing temperature for definition generation

We experimented with different temperatures for sampling the next word in definition generation. We used two metrics for comparison– average semantic similarity between the generated text and ground-truth using sentence embeddings (Reimers et al., 2019; Sanh et al., 2019) and percentage of accepted definitions (Section. 3.3.3). The temperature search space is between 0 and 1 where for efficiency and tractability we quantise it at 0.1. For our dataset, a temperature of 0.4 had higher scores for both the metrics with an average semantic similarity score of 0.2177 and a percentage of accepted definitions at 22.2%. The reason for this could be that, for our task of generating short definitions, we do not need a lot of diversity in text and a most probable path of high quality is needed.

4.1.6. Determining threshold for definition acceptance

Here, we describe the methodology we used to obtain the threshold of semantic similarity for a pair of sentences above which a sentence pair can be considered to be semantically similar to each other.

For this, we consider two popular NLP datasets SNLI (Stanford Natural Language Inference) (Bowman et al., 2015) and SICK (Sentences Involving Compositional Knowledge) (Marelli et al., 2014). The SNLI dataset and SICK dataset consists of human annotated labels that define the type of sentence pairs. The labels are Entailment, Neutral and Contradiction. They have 57,000 and 10,000 sentence pairs respectively. Table 3 shows an example for each of the three types of sentence pairs. (Example is from the SICK dataset). As you can see in the example, there is an overlap in words in all the sentence pairs, while there is extra negation in *Contradiction* pairs. It is very important for us to compute a threshold which is able to filter out the Contradictory statements from the Neutral or Entailment sentence pairs.

To determine this, we compute semantic similarity between the sentence pairs. We use both *USE* (Cer et al., 2018) and *BERT based sentence Transformers* (Reimers et al., 2019; Sanh et al., 2019) techniques.

Table 4 shows the average semantic similarity scores obtained using two sentence embedding techniques– *USE* and *BERT based sentence Transformers* for the human labelled sentence pairs. We can see that both the embeddings were able to clearly distinguish between the sentence types, however, *BERT based sentence Transformers* had a clearer distinction, more so, on the SICK dataset. Hence we decided to use *BERT based sentence Transformers* for our Definition selection. We chose a threshold such that the semantic similarity score is above the average score of Contraction and closer to Neutral or higher. This corresponds to the contradiction column of the SICK dataset (marked in BOLD) in the table. Rounding off this number (to 0.5) gives us the threshold for definition selection.

Table 5
Comparison of Keyword Extractor techniques.

Method	Max Semantic Similarity	Avg Semantic Similarity
TF-IDF	0.803	0.597
TextRank	0.728	0.555
EmbedRank	0.826	0.616
Noun Phrase Extraction	0.830	0.576
BERT based Key-phrase Extraction	**0.838**	**0.6236**

4.2. Key-phrase extraction

Selecting the correct key-phrases that represent a given learning resource is very crucial for all our algorithms. Hence to check the suitability of the model to our corpus, we tried the following key-phrase extraction methods:

1. **TF-IDF** stands for term frequency-inverse document frequency, a formula that measures how important a word is to a document in a collection of documents.
2. **TextRank** is an unsupervised graph based method to perform keyword and sentence extraction. We used the implementation of TextRank[6] based on the work of Mihalcea and Tarau (2004).
3. **EmbedRank** is an unsupervised embedding based method for key-phrase extraction. By selecting phrases whose semantic embeddings are close to the embeddings of the whole document, the best candidate phrases can be separated from the rest. We used the implementation of EmbedRank based on the work of Bennani-Smires et al. (2018) .[7]
4. **Noun Phrase Extraction** is one of the widely used approaches for keyword extraction. Here the keywords are extracted based on the regular expressions consisting of Noun phrases Adjectives and prepositions.

   We use nltk[8] python library to perform the above operations.

5. **BERT based Key-phrase Extraction** is also embedding based key-phrase extraction method. Here the document and the candidate key-phrases are vectorised using DistilBERT (Devlin et al., 2018; Reimers et al., 2019; Sanh et al., 2019). Candidate key-phrases are a list of all the unigrams and bigrams, excluding the stop words. Top 'n' candidates that are most semantically similar to the document are selected as key-phrases. For implementation, sentence-transformers model[9] of DistilBERT is used.

For the comparison of various keyword generation techniques, we use the Wikipedia keyword-description dataset. (Same one that we used to fine-tune the GPT-2 model, however for fine-tuning we consider only the first sentence which is the definition, here we consider the entire paragraph.) The original keywords are taken as the ground truth and compared with the extracted keywords. We considered 10,000 datapoints for our test. We use semantic similarity metrics for the comparison of all the keyword extraction techniques. Average semantic similarity and maximum semantic similarity between the top 5 key-phrases and the ground truth key-phrase are computed.

Findings are reported in Table 5. While most of the keyword techniques have comparable performance on our test set. We can see that *BERT based Key-phrase Extraction* gives the maximum scores for both the

[6] https://gist.github.com/BrambleXu/3d47bbdbd1ee4e6fc695b0ddb88cbf99.
[7] https://github.com/swisscom/ai-research-keyphrase-extraction.
[8] https://www.nltk.org.
[9] https://huggingface.co/sentence-transformers.

metrics, and hence we use this method in our work.

4.3. Overviews and Reflection quiz implementation details

All the algorithms for the generation of Overviews and Reflection quizzes are written using Python. Standard Python libraries are used wherever needed. Once the Overviews and questions are generated, we use the grammar correction package Gingerit[10] to correct the questions or generated text in Overviews or flag grammatical errors if any.

For generating **semantically closer option**, we used Gensim's Word2Vec model[11] based on the work of Mikolov et al. (2013). Gensim is one of the most mature and fastest ML libraries and is being used in many academic works. Hence, we used this for our implementation. We trained Gensim's Word2Vec model on our corpus instead of using a pre-trained model so as to include the domain specific vocabulary and increase the model accuracy.

The function *most-similar* defined in its library gets the top 'n' most similar words for the given keyword. This method first converts the word into the vectors (word embeddings), then computes cosine similarity between a simple mean of the projection weight vectors of the given words and the vectors for each word in the model.

Although this method returns semantically closer words for most of the key-phrases, there is a possibility that the key-phrase itself may not have an embedding. In such scenarios, we used DistilBERT model (bert-base-nli-mean-tokens).[12] We could use DistilBERT because BERT is context based and if the exact word is not present, it uses the context to get a word closer to the given word, hence we will always be able to find the keywords and its neighbours with this option. However, we did not use this method as a first choice because Word2Vec model returns a neighbourhood in an efficient manner since the model is trained to keep information of its neighbours for the words present in its dictionary (Mikolov et al., 2013).

The same method is used in obtaining semantic neighbourhood in *Overview* generation as well.

For generating **syntactically closer option**, the fuzzy matching algorithm based on Levenshtein distance was used. We used SeatGeek's fuzzywuzzy python library[13] which is based on Levenshtein distance for our implementation. For candidate set selection, we chose the words whose ratio of similarity was in a certain range, i.e., above 80% and below 95%. This was done to avoid having morphological forms of the anchor word in the candidate set.

5. Evaluation and results

In this section, we discuss about the open learning resources dataset that we used. We also discuss the evaluation and the results of learning pathway segmentation, Overviews and Reflection quizzes generated by our approach.

Human evaluations are typically viewed as the most important form of evaluation for NLG systems when they are open ended. If ground truth is available, automated metrics such as n-grams overlap is considered for evaluation. We used n-gram automated metrics for evaluation of our NLG component of Definition Generator (Refer Section 4.1). For the evaluation of auto-generated narrative fragments, we perform human evaluation.

For most NLG tasks, there is little consensus on how human evaluations should be conducted (van der Lee et al., 2021). A detailed study of 304 recent NLP papers presented by van der Lee et al. (2021) observed a median of 3 human evaluators even though the range was between 1 and 670 evaluators, and a median of 100 items with a range between 2 and 5400 items. For many NLG evaluation tasks, no specific expertise is required of the evaluators other than proficiency in the language of the generated text, especially when the fluency-related aspects are the focus of the evaluation (Celikyilmaz et al., 2020). In other cases, along with language proficiency, domain knowledge of the generated text is necessary to perform evaluations.

Considering this and the nature of our evaluation which aims to evaluate the generated text in terms of relevance, grammar and semantic accuracy, we designed human evaluation for the narrative fragments. Human volunteers who participated in the evaluations are from our institute and are well versed in the domain of our dataset as well as in NLP/NLG. The evaluators include Graduates, Post Graduates, and PhD students, some of whom are current and past teachers.

5.1. Dataset and pre-processing

We work with the dataset of open learning resources aggregated by an open educational platform *Gooru*.[14] The dataset comprises of about 4.2 million learning resources and hand-curated learning pathways which are maintained and updated by the educators. For our experiments, we consider a subset of these learning resources for K-12 science subject.

Gooru platform only indexes and aggregates the resources, actual content resides on the original open websites or on the cloud. We build a simple algorithm that selects appropriate methods to generate canonical text-based transcripts for disparate types of resources. Beautiful Soup[15] is used for web-pages, pdfminer,[16] pytube and pydub[17] are used for URL's that have embedded content in the form of PDFs, videos and audios respectively.

The transcripts are then summarised to a few sentences using an extractive graph-based text summarisation library sumy.[18] We use sumy and not any other advanced neural network based summarisers like T5 (Raffel et al., 2020) and BART (Lewis et al., 2019), because such models require a large amount of training data in a standard form. Even if we were to use the zero shot versions of these models (pre-trained model without fine-tuning), they would still require data in a standard form and often of shorter length due to the model limitations, making them unsuitable for summarising open learning resources such as ours.

5.2. Learning pathway segmentation

For learning pathway segmentation evaluation, we used 25 human segmented learning pathways. Total number of segments for these pathways was 72. We compared this with the segments created by our model. The accuracy of identifying the segment boundaries for these learning pathways by our model was 73.61% when compared to the ground-truth of human segmented learning pathways.

5.3. Overview evaluation

35 human evaluators participated in the evaluation study. 20 samples of generated Overviews were evaluated. Each evaluator was presented with 10 samples and for each sample, a sequence of learning resources (only the summaries) that make up a learning pathway segment was presented along with the generated Overview. This was followed by questions regarding the generated Overviews.

Following questions were asked for each of the generated Overviews:

Q1. Does the Overview correctly represent the learning resources?

[10] https://pypi.org/project/gingerit/.
[11] https://radimrehurek.com/gensim.
[12] https://huggingface.co/sentence-transformers.
[13] https://github.com/seatgeek/thefuzz.
[14] https://gooru.org.
[15] https://www.crummy.com/software/BeautifulSoup/.
[16] https://pypi.org/project/pdfminer/.
[17] https://github.com/topics.
[18] https://github.com/miso-belica/sumy.

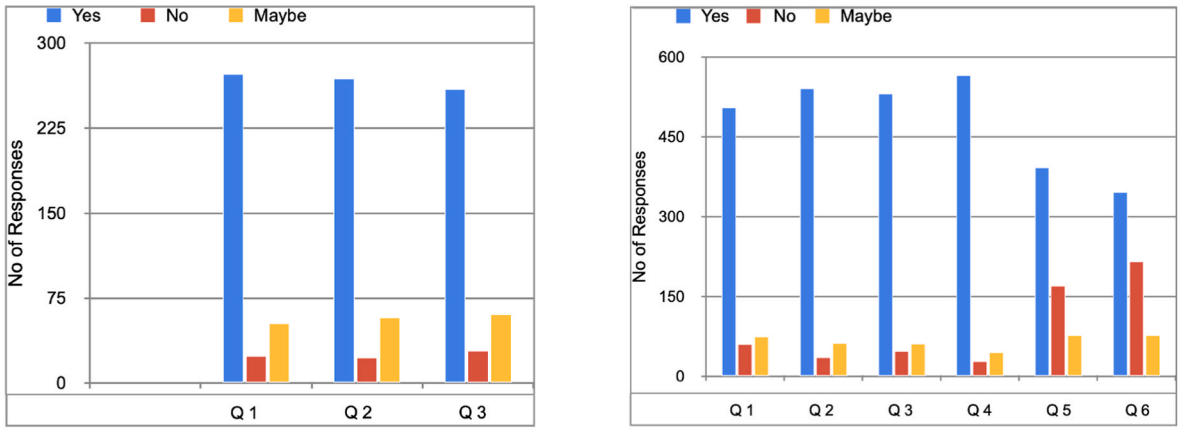

Fig. 7. Evaluation results (a) Overview evaluation and (b) MCQ evaluation.

Q2. Is the Overview meaningful?
Q3. Is the Overview grammatically correct?

Fig. 7a shows the bar graph of the results of the evaluation. The evaluators' agreement is measured using a 3 point Likert scale of *Yes*, *Maybe* and *No*. We can see from the graph that, for question Q1, 78% of the evaluators chose *Yes*, 15.14% of the evaluators chose *Maybe* and 6.85% of the evaluators chose *No*. Similarly, for questions, Q2 and Q3, 76.85% and 74.28% of the evaluators chose *Yes* and 16.57% and 17.42% of the evaluators chose *Maybe* respectively. Less than 10% of the evaluators chose *No* on all three questions. The results suggest that the Overviews generated by our approach are meaningful, grammatically correct and represent the learning resources in the learning pathway segment correctly.

5.4. MCQ evaluation

For the evaluation of our Multiple Choice Questions, 32 human evaluators participated in the study. The evaluation was done to validate the quality of the generated content in terms of the relevance, grammatical and semantic accuracy of the questions, and the clarity as well as the closeness of the options with respect to the correct answer.

Each evaluator was presented with 20 samples of the generated MCQs containing a question followed by four options. (The first option being the correct answer). A short auto-generated summary of a learning resource was also presented to the user. This was followed by a set of questions related to the generated MCQs which had to be answered by the evaluators based on the provided data.

Following questions were asked to the evaluators for each of the generated MCQs:

Q1. Is the question relevant to the learning resource?
Q2. Is the question semantically correct?
Q3. Is the question grammatically correct?
Q4. Is there good clarity of the correct option?
Q5. Do you think semantic options are close enough to the correct option?

Table 6
Average ROUGE scores on the validation set of NSW Corpus.

Method	R-1	R-2	R-L
Zero-shot GPT-2	0.2034	0.0876	0.1723
Our model-Original Corpus	0.2565	0.2311	0.2501
Our model-NSW Corpus	0.2403	0.2068	0.2170

Table 7
Examples of generated definitions for NSW Corpus.

Key-phrase	*Definition*
Market equilibrium price	Market equilibrium price is the price at which a stock price is stable over time.
Optimistic agents	Optimistic agents are agents who perform a task that is likely to be highly successful.
Availability bias	Availability bias refers to the tendency of a person to believe something because it is available to them.

Q6. Do you think the syntactic option is close enough to the correct option?

Fig. 7b shows the bar graph of the results of the evaluation. The evaluators' agreement is measured using a 3 point Likert scale of *Yes*, *Maybe* and *No*. We can see that the percentage of answer choice *Yes* is 79.06%, 84.68%, and 83.12% for the questions Q1 to Q3 respectively. Whereas, the answer choice of Maybe is 11.56%, 9.68% and 9.53%, respectively, with the negative choice being less than 10% at 9.37%,5.62% and 7.34% respectively. Overall, the evaluators felt the questions generated were relevant, and were semantically and syntactically accurate.

Regarding the evaluation of the options, question Q4 which evaluates if there is good clarity in choosing the right answer among the given options, received 88.6% *Yes* and just 7% *No*. However, questions Q5 and Q6 related to semantic and syntactic option generation received about 61.4% and 54.2% *Yes*, 12.03% and 12.03% *Maybe* and 26.6% and 33.75% of *No* respectively. We further investigated responses for Q5 and Q6 and observed that a few MCQs had more than 50% negatives for these options. For example, refer to example 1 in Fig. 6. Here the semantic option was "deleterious" which means "causing harm or damage" and is similar in meaning to the correct option "mutation", but we feel that since the word did not appear to be in the same context, evaluators may not have found it semantically closer to the correct option. Consider the syntactic option "computation" which had the last 6 letters the same as the correct option "mutation", however, we guess the reason it was voted negative by the evaluators is because they would have implicitly thought about the semantics behind it. We observed a similar pattern for the MCQs having majority negatives for Q5 and Q6, thus pulling down the overall scores for these questions.

6. Adaptability of Definition Generator Model

Our aim is to develop interactive narratives that can adapt effortlessly to different domains. For this, we mainly wanted to test if our

Definition Generator Model has the ability to adapt to a new, previously unseen domain. This is because the Reflection quiz and the Overview models are built on top of the *Definition Generator* model.

Hence, we tested the *Definition Generator* model on a new unseen domain of subject *Network Science for Web (NSW)*. We collected the learning resources from our institute's LMS (Learning Management System) for the subject. There were in all 1575 keywords for which we generated definitions. We report the ROUGE scores comparing the generated definitions with the summaries of the learning resources. We compared it with the Zero-shot GPT-2 model. The scores are reported in Table 6. We can see from Table 6 that the ROUGE scores of our model on the new unseen corpus of *NSW* have higher scores than the Zero-shot GPT-2. We also report the results of our model tested on our Wikipedia corpus (Copied from 4^{th} row of Table 2 in Section.3.2). Even though test sets here are incomparable, we can compare the range of ROUGE scores that are generated for the respective domains. We can see that R-1, R-2 and R-L scores for *Our model on NSW corpus* are comparable to the scores for *Our model on Wikipedia corpus*, albeit slightly lower (as expected for an unseen corpus). We also compared the number of definitions that were above the acceptance threshold, and it was around 17.46% for 0.4 temperature for the NSW domain as compared to 22.2% for the Wikipedia Science domain.

Table 7 shows some examples of generated definitions for NSW domain.

7. Discussion

In this work, we proposed a suite of semantic models that use NLP, NLG and ML techniques to enhance learning pathways to make them interactive learning narratives. It consists of methodologies to automatically generate auxiliary learning content and a method to intersperse the generated content at appropriate places in the learning pathways.

Auxiliary learning content generation, especially automatic question generation is a very active research area in Educational Technology (Kurdi et al., 2020; Pan et al., 2019). However, most often the auxiliary learning content such as reflection quizzes are added manually to the learning pathways, or in case of automatic addition, they are typically inserted after every learning resource. There are a number of studies that discuss where to add interventions in the learning pathways to make them interactive (Verma et al., 2022; Whitehill et al., 2014). However, content-based methodologies to identify strategic breakpoints for augmentation of learning pathways and creating sub-narratives, are not much studied in the literature.

Interactive Learning Environments (ILE) (Stranieri & Yearwood, 2008) and Game based narratives (Lester et al., 2013; Marsh et al., 2011) are also very popular in the Educational Technology field, where fictitious stories with different characters are interspersed in the learning pathways. However, most of these systems require manual creation of stories, decision trees, etc. As far as we know, the whole concept of bringing together the methodologies of automatic learning content generation and automatic learning pathway augmentation is not explored much.

Moving on to discussions on the automatic generation of Reflection quizzes and Overviews. In recent years, several studies have been proposed to apply AI techniques to automate the generation of questions. Typically, these studies are built upon AI techniques driven by deep neural networks (Chan & Fan, 2019; Liu et al., 2020; Pan et al., 2019; Steuer et al., 2020). The success of such approaches often relies on the availability of large-scale and relevant datasets used to train those deep neural network models. In these studies, the question-worthy sentences in an article are identified and converted to questions. This is unlike our approach which uses Natural Language Generation model to generate a question worthy sentence. There are many advantages of our proposed approach. First, since the words/sentences are not directly extracted from the learning resource, the questions generated could have variety in the vocabulary and definition, which is an important aspect in generating Reflection questions. Also, if we need questions in a different format, we can simply fine-tune the language model again with just a few thousand data-points in the needed format. Since a learning resource may contain many significant key-phrases, numerous questions could be developed for each of them.

Our automatic question generation approach is generalisable to different domains. While ontology based methods (Bühmann et al., 2015; Vega-Gorgojo, 2019; Vinu et al., 2015) can also generate questions for any domain, building ontologies takes enormous effort.

Overviews are another type of learning content whose automatic generation is proposed in this work, which as far as we are aware, is a unique concept and is unexplored. Here, the sketch of what to expect in the learning pathway is presented to the learner along with the metadata of the learning pathway segments. Overviews are intended to assist learners in forming realistic expectations for the learning pathway and in helping them make an informed choice about starting or continuing on the learning pathway.

For the generation of both types of auxiliary learning content, we propose a Natural Language Generation model based on GPT-2, which generates a definition given a keyword. Usage of sophisticated text generation models such as GPT-n (Brown et al., 2020; Radford et al., 2019), T5 (Raffel et al., 2020), and CTRL (Keskar et al., 2019), which produce coherent and human-like text is relatively new in the NLP domain. Only a few recent studies in Educational Technologies have explored it. Methodologies proposed by Liu et al. (2020); Chan and Fan (2019); Steuer et al. (2020) use GPT-2 and other pre-trained language models in their pipeline of Automatic Question Generation, however, they use it for tasks such as sentence extraction, classification, etc., unlike our model which generates the definitions that are converted to questions. Wang et al. (2022) investigate using effective prompts as inputs to pre-trained Language models GPT-2 and GPT-3 to generate questions. Lopez et al. (2021); Bhat et al. (2022) use GPT-2 and T5 respectively to generate questions, however, these methods have different challenges and limitations (as discussed in detail in 2.2).

As mentioned earlier, GPT-2 based Definition Generation model brings in variety and interesting information which we may not bring out as humans, for example, some other less known but important facts about concepts. However, this also brings in an important task of selecting the suitable generated text. Hence, we also propose a unique method of first generating a candidate set of definitions and then choosing the best fit definition from it.

7.1. Discussion on evaluation results

From the empirical results, we can confirm that our NLG approach to auxiliary learning content generation has a significant contribution in the area of automatic content generation. Human evaluation of both types of auxiliary learning contents generated by our approach showed promising results (Section 5.3 5.4). They were able to produce learning content for open learning resources obtained from transcripts, summarisers etc., with some of them containing noisy data. The automatically generated learning content were semantically and syntactically accurate, and were relevant to the given learning resources.

This lays the foundation for a shift from extractive methodologies to natural language generation methodologies, for supplementary learning content generation.

The proposed approach of automatic segmentation of the learning pathways into coherent sub-topics which form strategic break-points to augment the learning pathways was empirically evaluated to be good and it matched with the segments generated by humans (Section 5.2).

The evaluation of the Definition Generator model was performed by comparing it to the state of the art text generation models. Our model performed significantly better than the baseline models on the test set (Section 4.1). This model was then evaluated on an unseen domain of Network Science for the Web. Promising results were obtained on this

dataset, hence, we can say that the proposed model is generalisable (Section 6).

The natural language generation model generates one word at a time, and there are different ways to sample the next word from the probability distribution over words given the current text (generated and the input). We hence experimented with different temperatures that shape this probability distribution and chose the temperature that suited our application (Section 4.1.1).

While the Definition Generation model performed well, it was important to filter out the definitions that were relevant. For this, we conducted experiments to choose a threshold of semantic similarity for a pair of sentences, above which a sentence pair can be considered semantically similar to each other (Section 4.1.2).

Key-phrase extraction is also very critical for our methodologies of auxiliary learning content generation since the text is generated based on it. Hence, we conducted experiments to choose the most suitable key-phrase extraction technique. BERT based key-phrase extraction gave the most robust performance when evaluated on our dataset (Section 4.2).

While our approach to automatic MCQ generation and automatic Overview generation produced relevant and good-quality text, following are some of the limitations of our proposed methodologies.

Since an important aspect of using AI systems in education is ensuring trust in the AI system, we had to make sure that the generated content was highly relevant to the learning resources. For this, in our definition selection module 3.3.3, we set a high threshold for the selection of the generated text. This was an attempt to be more cautious and eliminate any suspicious definitions. However, this decreased the accepted definitions rate and created considerable false negatives. i.e., many good and relevant definitions were dropped from the candidate set. On average, the percentage of the accepted definitions was at 22.2% on our test dataset. Another issue with lower acceptance rates is that some keywords may not have any acceptable definitions resulting in fewer questions for the learning resources.

We believe that training a classification algorithm such as SVM (Support Vector Machine) that decides the acceptance and non-acceptance of the definition, would be a good option to explore in the future. The data for training such an algorithm could be obtained from a deployed system which has the data of the generated and the accepted definitions (marked during the verification phase).

We would also like to point out a limitation in the methodology of automatic Overview generation. As, we discussed earlier, to generate an Overview, we proposed a semantic algorithm that selects the *Main Theme* for a group of learning resources (Section 3.3.2). While in most cases our model was able to find an appropriate *Main Theme*, there were a few cases where our model could have come up with a better *Main Theme*. One such example is example 3 in Fig. 4. Here, we can see that our model recognised the common root word as *Waves*, but the segment is actually about *The Properties of the Waves*. In future, we could enhance the algorithm to hierarchically derive such relations and generalisations.

There is also a lot of scope for enhancing the auxiliary learning content generation. For example, the proposed model of automatic question generation addressed the automatic generation of “What?” type of questions. In future, we would like to add questions of the type “Why?” and “How?”. Also, the textual Overviews can be further enhanced with pictures, enhanced visual layouts, voice-overs etc. The Reflection quizzes can be made interactive by adding hints based on user inputs.

7.2. Future applications and pedagogical implications

Our proposed solution of learning pathway augmentation can be implemented as a stand alone learning system or it can be plugged into any existing learning system. Any algorithm for automatic learning pathway generation from open learning resources could be used (Chi, 2009; Diwan et al., 2019; Shmelev et al., 2015) to generate the learning pathways. These learning pathways could then be augmented by our proposed suite of algorithms, to create interactive learning narratives. Since the proposed approach is domain-agnostic and does not require a large dataset to train, it is a natural choice for niche domains, where only a bunch of learning resources are available.

The augmented learning pathways can also serve as a bootstrap or baseline for personalised or adaptable online learning platforms. Whenever there are changes in the pathways, the auxiliary learning content can be generated dynamically or on the fly. Similar application is in the Educational Recommender Systems, where the auxiliary learning content can be generated for any recommended learning resources.

Our automated approach to auxiliary learning content generation from available open learning resources is extremely beneficial in emerging Smart Learning Environments (SLEs) (Ruiz-Calleja et al., 2019). SLEs support ubiquitous and adaptive learning by exploiting mobile technologies, which creates the need for using open learning resources to generate learning content. For example, if you are a history student and visit a monument, you would receive a notification on your smartphone with information about that place in the form of an Overview or a Quiz.

In emergency online education, like the one we experienced during the COVID lockdown worldwide, instructors found themselves having to generate large question banks to accommodate this new learning format. Our approach of automatically generating useable assessment questions based on learning materials would have been invaluable, and it will be in the future if such a situation arises.

While there is no need for manual intervention to generate the learning content or augment the learning pathways, given the complexity of learning and accuracy expected in the learning environments, it would be recommended to have a human-in-the-loop design, where a human expert verifies the auto-generated narrative fragments before deployment. This is a one-time activity which can be done before the narrative fragments are released. Once the narrative fragments and the deployment are verified for a domain, the auto-generated learning content which gets generated dynamically can be deployed without a need for human verification.

8. Conclusion and future work

We proposed a suite of semantic models for the automatic generation of auxiliary learning content and automatic augmentation of the learning pathways, such that the learning pathways are interactive and conform to a generic narrative expectation.

Evaluation of the proposed NLG model, which generates a definition given a keyword, produced promising results when compared to the state of the art baseline models. It was able to generalise well on a new domain. Human evaluation of the automatically generated learning content of Multiple Choice Questions and Overviews generated by our model showed encouraging results.

In this work, automatically generated learning content was evaluated in terms of the quality of the generated text and the relevance of the fragments to the learning resources in the pathways. However, they were not evaluated in a real-time or classroom setup. In future, we plan to evaluate the efficacy of the automatically generated learning content in the real world such as a classroom setup. We also want to measure the increase in engagement of the proposed augmented learning pathways as compared to the original learning pathways.

While our prototype may not be perfect, we see that such systems have great potential as content generators for creating an engaging learning experience. They also pave the way for building learning environments that utilise a large pool of available open learning resources, in an inexpensive way.

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.

Acknowledgements

The authors would like to thank Mirambika Sikadar, Sonal Garg, Sumanth Yenikapati and Harshvardhan Kumar for their contribution. The authors would also like to thank all the evaluators of the research outputs.

References

Ackley, D. H., Hinton, G. E., & Sejnowski, T. J. (1985). A learning algorithm for Boltzmann machines. *Cognitive Science, 9*, 147–169.
Aldowah, H., Al-Samarraie, H., Alzahrani, A. I., & Alalwan, N. (2020). Factors affecting student dropout in moocs: A cause and effect decision-making model. *Journal of Computing in Higher Education, 32*, 429–454.
Aleven, V., Sewall, J., Popescu, O., Xhakaj, F., Chand, D., Baker, R., Wang, Y., Siemens, G., Rosé, C., & Gasevic, D. (2015). The beginning of a beautiful friendship? Intelligent tutoring systems and moocs. In *International conference on artificial intelligence in education* (pp. 525–528). Springer.
Anderson, A., Huttenlocher, D., Kleinberg, J., & Leskovec, J. (2014). Engaging with massive online courses. In *Proceedings of the 23rd international conference on World wide web* (pp. 687–698).
Baneres, D., Caballé, S., & Clarisó, R. (2016). Towards a learning analytics support for intelligent tutoring systems on mooc platforms. In *2016 10th international conference on complex, intelligent, and software intensive systems (cisis)* (pp. 103–110). IEEE.
Bennani-Smires, K., Musat, C., Hossmann, A., Baeriswyl, M., & Jaggi, M. (2018). Simple unsupervised keyphrase extraction using sentence embeddings. In *Proceedings of the 22nd conference on computational Natural Language learning* (pp. 221–229).
Bhat, S., Nguyen, H. A., Moore, S., Stamper, J., Sakr, M., & Nyberg, E. (2022). *Towards automated generation and evaluation of questions in educational domains*.
Borrella, I., Caballero-Caballero, S., & Ponce-Cueto, E. (2019). Predict and intervene: Addressing the dropout problem in a mooc-based program. In *Proceedings of the sixth* (pp. 1–9). ACM Conference on Learning@ Scale, 2019.
Bowman, S. R., Angeli, G., Potts, C., & Manning, C. D. (2015). *A large annotated corpus for learning natural language inference*. arXiv preprint arXiv:1508.05326.
Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems, 33*, 1877–1901.
Brusilovsky, P., & Henze, N. (2007). Open corpus adaptive educational hypermedia. In *The adaptive web* (pp. 671–696). Springer.
Bühmann, L., Usbeck, R., & Ngonga Ngomo, A. C. (2015). Assess—automatic self-assessment using linked data. In *International semantic web conference* (pp. 76–89). Springer.
Carbonell, J., & Goldstein, J. (1998). The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 335–336).
Celikyilmaz, A., Clark, E., & Gao, J. (2020). *Evaluation of text generation: A survey*. arXiv preprint arXiv:2006.14799.
Cer, D., Yang, Y., Kong, S.y., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., et al. (2018). *Universal sentence encoder*. arXiv preprint arXiv:1803.11175.
Chan, Y. H., & Fan, Y. C. (2019). Bert for question generation. In *Proceedings of the 12th international conference on Natural Language Generation* (pp. 173–177).
Chen, Y., Wu, L., & Zaki, M. J. (2019). *Reinforcement learning based graph-to-sequence model for natural question generation*. arXiv preprint arXiv:1908.04942.
Chi, Y. L. (2009). Ontology-based curriculum content sequencing system with semantic rules. *Expert Systems with Applications, 36*, 7838–7847.
Cho, K., Courville, A., & Bengio, Y. (2015). Describing multimedia content using attention-based encoder-decoder networks. *IEEE Transactions on Multimedia, 17*, 1875–1886.
Ch, D. R., & Saha, S. K. (2018). Automatic multiple choice question generation from text: A survey. *IEEE Transactions on Learning Technologies, 13*, 14–25.
Chung, C. Y., & Hsiao, I. H. (2022). Programming question generation by a semantic network: A preliminary user study with experienced instructors. In *International conference on artificial intelligence in education* (pp. 463–466). Springer.
Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). *Bert: Pre-training of deep bidirectional transformers for language understanding*. arXiv preprint arXiv: 1810.04805.
Diwan, C., Srinivasa, S., & Ram, P. (2018). Computing exposition coherence across learning resources. In *OTM confederated international conferences" on the move to meaningful Internet systems"* (pp. 423–440). Springer.
Diwan, C., Srinivasa, S., & Ram, P. (2019). Automatic generation of coherent learning pathways for open educational resources. In *European conference on Technology enhanced learning* (pp. 321–334). Springer.
El-Kassas, W. S., Salama, C. R., Rafea, A. A., & Mohamed, H. K. (2021). Automatic text summarization: A comprehensive survey. *Expert Systems with Applications, 165*, Article 113679.
Ghosh, A., Tschiatschek, S., Devlin, S., & Singla, A. (2022). *Adaptive scaffolding in block-based programming via synthesizing new tasks as pop quizzes*. AIED.
Gupta, S., & Gupta, S. (2019). Abstractive summarization: An overview of the state of the art. *Expert Systems with Applications, 121*, 49–65.
Haghighi, A., & Vanderwende, L. (2009). Exploring content models for multi-document summarization. In *Proceedings of human language technologies: The 2009 annual conference of the north American chapter of the association for computational linguistics* (pp. 362–370).
Hartnett, M., St George, A., & Dron, J. (2011). Examining motivation in online distance learning environments: Complex, multifaceted, and situation-dependent. *International Review of Research in Open and Distance Learning, 12*, 20–38.
Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., & Blunsom, P. (2015). Teaching machines to read and comprehend. *Advances in Neural Information Processing Systems, 28*.
Keskar, N. S., McCann, B., Varshney, L. R., Xiong, C., & Socher, R. (2019). *Ctrl: A conditional transformer language model for controllable generation*. arXiv preprint arXiv: 1909.05858.
Khalil, H., & Ebner, M. (2014). Moocs completion rates and possible methods to improve retention-a literature review. In *EdMedia: World conference on educational media and Technology, association for the advancement of computing in education (AACE)* (pp. 1305–1313).
Kumar, Y. J., & Salim, N. (2012). Automatic multi document summarization approaches. In *KS Gayathri, eceived BE degree in CSE from madras university in 2001 and ME degree from anna university, Chennai. She is doing Ph. D. In the area of reasoning in Smart* (Citeseer).
Kurdi, G., Leo, J., Parsia, B., Sattler, U., & Al-Emari, S. (2020). A systematic review of automatic question generation for educational purposes. *International Journal of Artificial Intelligence in Education, 30*, 121–204.
Laurillard, D., Stratfold, M., Luckin, R., Plowman, L., & Taylor, J. (2000). Affordances for learning in a non-linear narrative medium. *Journal of Interactive Media in Education, 2* (2), 2.
Lebanoff, L., Song, K., & Liu, F. (2018). *Adapting the neural encoder-decoder framework from single to multi-document summarization*. arXiv preprint arXiv:1808.06218.
van der Lee, C., Gatt, A., van Miltenburg, E., & Krahmer, E. (2021). Human evaluation of automatically generated text: Current trends and best practice guidelines. *Computer Speech & Language, 67*, Article 101151.
Lester, J. C., Rowe, J. P., & Mott, B. W. (2013). Narrative-centered learning environments: A story-centric approach to educational games. In *Emerging technologies for the classroom* (pp. 223–237). Springer.
Levenshtein, V. I., et al. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady* (pp. 707–710). Soviet Union.
Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2019). *Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension*. arXiv preprint arXiv:1910.13461.
Lin, C. Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text summarization branches out* (pp. 74–81). Barcelona, Spain: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/W04-1013.
Liu, Y., & Lapata, M. (2019). *Hierarchical transformers for multi-document summarization*. arXiv preprint arXiv:1905.13164.
Liu, B., Wei, H., Niu, D., Chen, H., & He, Y. (2020). Asking questions the human way: Scalable question-answer generation from text corpus. In *Proceedings of the web conference 2020* (pp. 2032–2043).
Lopez, L. E., Cruz, D. K., Cruz, J. C. B., & Cheng, C. (2021). Simplifying paragraph-level question generation via transformer language models. In *Pacific rim international conference on artificial intelligence* (pp. 323–334). Springer.
Lukovnikov, D., Fischer, A., & Lehmann, J. (2019). Pretrained transformers for simple question answering over knowledge graphs. In *International semantic web conference* (pp. 470–486). Springer.
Marelli, M., Menini, S., Baroni, M., Bentivogli, L., Bernardi, R., Zamparelli, R., et al. (2014). A sick cure for the evaluation of compositional distributional semantic models. In *Lrec, reykjavik* (pp. 216–223).
Marsh, T., Xuejin, C., Nickole, L. Z., Osterweil, S., Klopfer, E., & Haas, J. (2011). Fun and learning: The power of narrative. In *Proceedings of the 6th international conference on foundations of digital games* (pp. 23–29).
Mihalcea, R., & Tarau, P. (2004). Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing* (pp. 404–411).
Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).
Murray, T. (2003). An overview of intelligent tutoring system authoring tools: Updated analysis of the state of the art. *Authoring tools for advanced technology learning environments*, 491–544.
Ortega-Arranz, A., Kalz, M., & Martínez-Monés, A. (2018). Creating engaging experiences in moocs through in-course redeemable rewards. In *2018 IEEE global engineering education conference (EDUCON)* (pp. 1875–1882). IEEE.
Paas, F., Tuovinen, J. E., Van Merrienboer, J. J., & Darabi, A. A. (2005). A motivational perspective on the relation between mental effort and performance: Optimizing learner involvement in instruction. *Educational Technology Research & Development, 53*, 25–34.
Pan, L., Lei, W., Chua, T. S., & Kan, M. Y. (2019). *Recent advances in neural question generation*. arXiv preprint arXiv:1905.08949.
Parasa, N. S., Diwan, C., & Srinivasa, S. (2022). Automatic riddle generation for learning resources. In *International conference on artificial intelligence in education* (pp. 343–347). Springer.
Paulus, R., Xiong, C., & Socher, R. (2017). *A deep reinforced model for abstractive summarization*. arXiv preprint arXiv:1705.04304.
Perone, C. S., Silveira, R., & Paula, T. S. (2018). *Evaluation of sentence embeddings in downstream and linguistic probing tasks*. arXiv preprint arXiv:1806.06259.

Plowman, L., Luckin, R., Laurillard, D., Stratfold, M., & Taylor, J. (1999). Designing multimedia for learning: Narrative guidance and narrative construction. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 310–317).

Rachakonda, A. R., & Srinivasa, S. (2009). Finding the topical anchors of a context using lexical cooccurrence data. In *Proceedings of the 18th ACM conference on Information and knowledge management* (pp. 1741–1744).

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog, 1*, 9.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P. J., et al. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research, 21*, 1–67.

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). *Squad: 100,000+ questions for machine comprehension of text*. arXiv preprint arXiv:1606.05250.

Reimers, N., Gurevych, I., Reimers, N., Gurevych, I., Thakur, N., Reimers, N., Daxenberger, J., Gurevych, I., Reimers, N., Gurevych, I., et al. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in Natural Language processing*. Association for Computational Linguistics.

Ruiz-Calleja, A., Bote-Lorenzo, M. L., Vega-Gorgojo, G., Serrano-Iglesias, S., Asensio-Pérez, J. I., Dimitriadis, Y., & Gómez-Sánchez, E. (2019). The potential of open data to automatically create learning resources for smart learning environments. In *Multidisciplinary digital* (p. 61). Publishing Institute Proceedings.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). *Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter*. arXiv preprint arXiv:1910.01108.

Scheepers, T. (2017). *Improving the compositionality of word embeddings*. Amsterdam, Netherlands: Universiteit van Amsterdam. Science Park 904. Master's thesis.

Shmelev, V., Karpova, M., & Dukhanov, A. (2015). An approach of learning path sequencing based on revised bloom's taxonomy and domain ontologies with the use of genetic algorithms. *Procedia Computer Science, 66*, 711–719.

Siemens, G. (2013). *Massive open online courses: Innovation in education open educational resources: Innovation, research and practice*.

Simpson, O. (2013). Student retention in distance education: Are we failing our students? *Open Learning: The Journal of Open, Distance and e-Learning, 28*, 105–119.

Sottilare, R. A., Brawner, K. W., Goldberg, B. S., & Holden, H. K. (2012). *The generalized intelligent framework for tutoring (gift)*. Orlando, FL: US Army Research Laboratory–Human Research & Engineering Directorate (ARL-HRED).

Steuer, T., Filighera, A., & Rensing, C. (2020). Remember the facts? Investigating answer-aware neural question generation for text comprehension. In *International conference on artificial intelligence in education* (pp. 512–523). Springer.

Stranieri, A., & Yearwood, J. (2008). Enhancing learning outcomes with an interactive knowledge-based learning environment providing narrative feedback. *Interactive Learning Environments, 16*, 265–281.

Tamang, L. J., Banjade, R., Chapagain, J., & Rus, V. (2022). Automatic question generation for scaffolding self-explanations for code comprehension. In *International conference on artificial intelligence in education* (pp. 743–748). Springer.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).

Vega-Gorgojo, G. (2019). Clover quiz: A trivia game powered by dbpedia. *Semantic Web, 10*, 779–793.

Verma, M., Nakashima, Y., Takemura, N., & Nagahara, H. (2022). Multi-label disengagement and behavior prediction in online learning. In *International conference on artificial intelligence in education* (pp. 633–639). Springer.

Vinu, E. V., et al. (2015). A novel approach to generate mcqs from domain ontology: Considering dl semantics and open-world assumption. *Journal of Web Semantics, 34*, 40–54.

Wang, W., Guo, L., He, L., & Wu, Y. J. (2019). Effects of social-interactive engagement on the dropout ratio in online learning: Insights from mooc. *Behaviour & Information Technology, 38*, 621–636.

Wang, Z., Valdez, J., Basu Mallick, D., & Baraniuk, R. G. (2022). Towards human-like educational question generation with large language models. In *International conference on artificial intelligence in education* (pp. 153–166). Springer.

Whitehill, J., Serpell, Z., Lin, Y. C., Foster, A., & Movellan, J. R. (2014). The faces of engagement: Automatic recognition of student engagementfrom facial expressions. *IEEE Transactions on Affective Computing, 5*, 86–98.