Teacher in the Loop: Customizing Educational Games Using Natural Language

Nacir Bouali[1,3][a], Violetta Cavalli-Sforza[2][b] and Markku Tukiainen[3][c]

[1]Data Management and Biometrics, University of Twente, 7522 NB Enschede, The Netherlands
[2]School of Science and Engineering, Al Akhawayn University in Ifrane, 53000 Ifrane, Morocco
[3]School of Computing, University of Eastern Finland, 80101 Joensuu, Finland
fi

Keywords: Serious Games, Virtual Reality, Natural Language Processing.

Abstract: Despite significant advances in educational technology and design methodologies, current educational games demonstrate a fundamental limitation: educators are unable to modify content after the games are deployed, limiting curriculum alignment and pedagogical customization. This paper introduces Imikathen-VR, a solution built upon a text-to-animation system, supporting K-1 and K-2 teachers to create minigames for their students to practice basic writing skills. Our implementation extends an existing animation pipeline by integrating a fine-tuned T5 model for sentence simplification, achieving 95% F1 BERT score and 76% ROUGE-L score in maintaining semantic and lexical fidelity. We improve visual reasoning by transforming the task of identifying missing visual details into a Masked Language Modeling problem. Preliminary results demonstrate the system's effectiveness in generating curriculum-aligned VR exercises, though comprehensive classroom testing remains pending. This work advances the integration of customizable VR technology in early education, providing teachers with enhanced control over educational content.

1 INTRODUCTION

Educational games provide an appealing context for children to learn. Their success depends however on how well the educational content is integrated within the gameplay experience (Fisch, 2005). The promise of educational games in improving the learning experience within or outside the classroom for different subjects, such as in maths (Devlin, 2011), chemistry (Smaldone et al., 2017) or language learning (Birova, 2013; Derakhshan and khatir, 2015; Miftakhova and Yapparova, 2019) has been supported via various research works. Despite their potential in improving student engagement and achievement in language learning for example, the integration of such educational games within the classroom is still a challenging area as it is difficult for the teachers to find relevant games that align with the curriculum (Koh et al., 2012; Kirriemuir and McFarlane, 2004; Rice, 2007). For a teacher to invest into adapting existing games to fit the specific needs of their curriculum or student population, they may require a lot of time and effort, or lack the proper training for such a task. However, despite these challenges, research suggests that well-designed and well-adapted games can significantly enhance language learning.

Approaches such a participatory design and co-design have been adopted in the educational games field to improve or ensure, among other things, the alignment of the games with the curriculum (Walsh, 2012; Ismail et al., 2019). The involvement of teachers in these co-creation activities is however challenged by problems such as time constraints as the primary mission of teachers is related to teaching and administrative tasks, and as such, engaging them in the design process with its various meetings, feedback sessions and testing can overload them. Issues in teachers involvement extend also to the nature of their expertise mainly related to educational content and pedagogy, while they may lack the technical expertise required for game design. This gap can lead to challenges in communicating needs and understanding the technical possibilities and limitations within the design process (Munoz et al., 2016; Padilla-Zea et al., 2018).

In the context of this research, we aim to provide K-

[a] https://orcid.org/0000-0001-7465-9543
[b] https://orcid.org/0000-0002-9877-0008
[c] https://orcid.org/0000-0002-8630-5248

Bouali, N., Cavalli-Sforza, V. and Tukiainen, M.
Teacher in the Loop: Customizing Educational Games Using Natural Language.
DOI: 10.5220/0013480500003932

In *Proceedings of the 17th International Conference on Computer Supported Education (CSEDU 2025) - Volume 2*, pages 446-457
ISBN: 978-989-758-746-7; ISSN: 2184-5026

1 and K-2 teachers with a tool that allows them to customize educational games using natural language, which not only mitigates time investment by allowing design from their teaching or personal spaces, but also eliminates the need for extensive knowledge of game mechanics, enabling teachers to focus solely on the game's content.

In this paper, we present Imikathen-VR, an educational game built on top of Imikathen (Bouali et al., 2024), a text-to-animation system that converts natural language stories into animations. Imikathen-VR was specifically developed to support language learning for K-1 and K-2 students through in-class writing activities. However, the system has limitations that make its direct use challenging, particularly regarding output accuracy. To address these limitations, we modified the system's format to give teachers more control over the educational content. This new approach allows teachers to first design and verify the stories and resulting animations, and then create writing exercises based on the approved content. This ensures that outputs align with curriculum requirements and that students receive accurate material tailored to practice specific vocabulary and language skills.

Imikathen's current pipeline, shown in Figure 1, processes input stories through several key steps. First, it performs TimeML-based event detection to identify animatable sentences (Saurí et al., 2006), it then uses imagery scores for nouns and adjectives to determine renderable content (Coltheart, 1981). The system then simplifies multi-event sentences using a rule-based algorithm and processes dialogue through either direct speech rendering or by leveraging a fine-tuned BART model for indirect-to-direct speech conversion. Finally, a visual semantic role labeler extracts roles from the simplified sentences, creates visual semantic frames for each event, and converts them into an object-oriented animation language. These animations are finally rendered in either 3D or VR using a Unity-based graphics engine. Prior to discussing how we aim to adapt Imikathen to in-class use, and turn it into a customizable game that can be tailored to accommodate the different curricula teachers might be using, we will first invest in improving some of its subprocesses, namely the sentence simplification and visual reasoning modules.

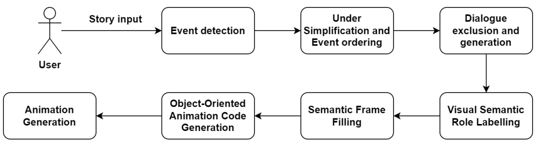

User
Story input
Event detection
Under Simplification and Event ordering
Dialogue exclusion and generation
Visual Semantic Role Labelling
Semantic Frame Filling
Object-Oriented Animation Code Generation
Animation Generation

Figure 1: Pipeline for the NLU of Imikathen (Bouali et al., 2024).

The rest of this paper is organized as follows: In Section 2, we review the VR games used in language learning. Imikathen's architecture, alongside the improvements on sentence simplification and visual reasoning are presented in Sections 3, 4 and 5, respectively. Section 6 presents the system prototype, followed by a discussion of its potential and limitations in Section 7. We conclude with future work and closing remarks in Section 8.

2 RELATED WORK

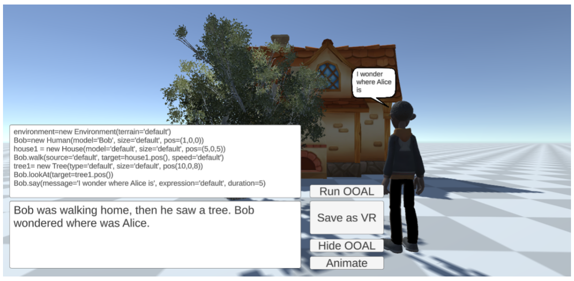
Virtual reality (VR) language learning games have been explored for various tasks, languages, and learning contexts. Cheng et al. adapted the Crystallize game for Japanese language learning using an Oculus Rift, showing that VR enhances cultural immersion (Cheng et al., 2017). Khatoony used VR games to improve vowel pronunciation in low-intermediate Iranian learners of English through immersive, interactive methods (Khatoony, 2019). Amoia et al. introduced I-FLEG, a serious game for learning French, which leverages AI-driven personalization in a 3D virtual environment to enhance language acquisition (Amoia et al., 2012). Tazouti et al. developed ”ImALeG VR,” a multi-platform serious game for Tamazight vocabulary learning and self-assessment (Tazouti, 2020). Chen and Hsu studied the impact of VR-based English learning applications, finding that interactive game features enhance engagement and motivation (Chen and Hsu, 2020).

Alfadil examined VR's role in vocabulary acquisition, concluding that encountering words in context (e.g., ordering food in a virtual restaurant) improves retention, though effectiveness depends on content quality and alignment with learning objectives (Alfadil, 2020).

In Imikathen-VR, we extend our text-to-VR system to support a language learning game where teachers can align VR content with their course requirements.

3 CURRENT ARCHITECTURE AND ADAPTATION AS A GAME

Imikathen-VR builds upon our existing text-to-VR system, Imikathen (Bouali et al., 2024). This section outlines the current architecture and its adaptation into a VR game, where teachers control content delivery and timing.



```
environment=new Environment(terrain='default')
Bob=new Human(model='Bob', size='default', pos=(1,0,0))
house1 = new House(model='default', size='default', pos=(5,0,5))
Bob.walk(source='default', target=house1.pos(), speed='default')
tree1= new Tree(type='default', size='default', pos(10,0,8))
Bob.lookAt(target=tree1.pos())
Bob.say(message='I wonder where Alice is', expression='default', duration=5)
```

Bob was walking home, then he saw a tree. Bob wondered where was Alice.

Figure 2: Imikathen current interface (Bouali et al., 2024).

3.1 Imikathen: Text-to-VR System

According to (Bouali and Cavalli-Sforza, 2023), developing a text-to-animation system presents 15 different challenges, spanning Natural Language Understanding (NLU), temporal reasoning, and visual reasoning. NLU challenges include identifying animatable sentences, handling modal verbs, and visualizing abstract concepts and underspecified language. Temporal reasoning struggles with verb entailment, non-linear narratives, tense interpretation, and simultaneous actions. Visual reasoning complexities involve common-sense reasoning, object quantification, and animation-specific elements like lighting, camera work, and character interactions. Overcoming these hurdles requires significant advancements across NLU, knowledge representation, and animation techniques. Imikathen attempts to solve these challenges by employing a modular architecture that consists of a client-side interface and a server-side processing pipeline (Bouali et al., 2024). The key components of this architecture include:

- **Client Module**
  - Accepts natural language input (stories) and renders the final animation output, providing a simpler user experience.
- **Server Module**
  - **Event Detection:** Uses TimeML event classes and MRC psycholinguistic scores to accurately detect events and evaluate object animatability, addressing challenges in natural language understanding.
  - **Dialogue Detection:** Employs a fine-tuned BART-based generative model to identify and process direct and indirect speech, improving dialogue transformation.
  - **Sentence Simplification:** Applies rule-based algorithms with dependency parsing to decompose complex sentences, enhancing the system's ability to handle multi-event sentences and refine the temporal order of events, effectively addressing the challenges associated with temporal reasoning.
  - **Semantic Role Labeling:** Integrates RoBERTa with dependency parse trees to improve visual role tagging, such as identifying actors, actions, objects and their related modifiers.
  - **Visual Semantic Frames:** Leverages ConceptNet to enrich scene details and enable visual reasoning, tackling common-sense reasoning and object quantification challenges.
  - **Object-Oriented Animation Language Generation:** Converts semantic frames into an Object-Oriented Animation Language (OOAL), facilitating the creation of animation outputs.

Imikathen demonstrates significant advancements in NLU, knowledge representation, and animation techniques. However, despite its capabilities, the system is not without limitations. Event detection inaccuracies, inconsistent dialogue transformation outputs, and limited coverage of sentence simplification rules can impact the quality and reliability of the final animation output, highlighting areas for further improvement and refinement.

We developed OOAL to allow users to manually adjust animations when the system's linguistic analysis produces inaccuracies. However, given its complexity, OOAL proved to be beyond the technical abilities of young children. To explore its educational potential, we previously implemented OOAL in a VR-based game and evaluated it as a teaching tool for object-oriented programming concepts at a university level, where student feedback was positive (Bouali et al., 2019; Sunday et al., 2023).

When we later consulted K-1 and K-2 educators about integrating the text-to-animation system into early education, they raised a key pedagogical concern: animation failures could mislead students about the correctness of their language usage, potentially negatively impacting their learning process. This feedback underscored the need for a more reliable approach before deploying the system in classroom settings.

3.2 A Teacher-in-the-Loop Approach

To address the limitations above, we enhanced our system by incorporating teacher involvement in the content creation process. This allows educators to define stories that align with their curriculum objectives throughout the academic year.

We extended the architecture shown in Figure 3 by adding a teacher interface (highlighted in blue). This

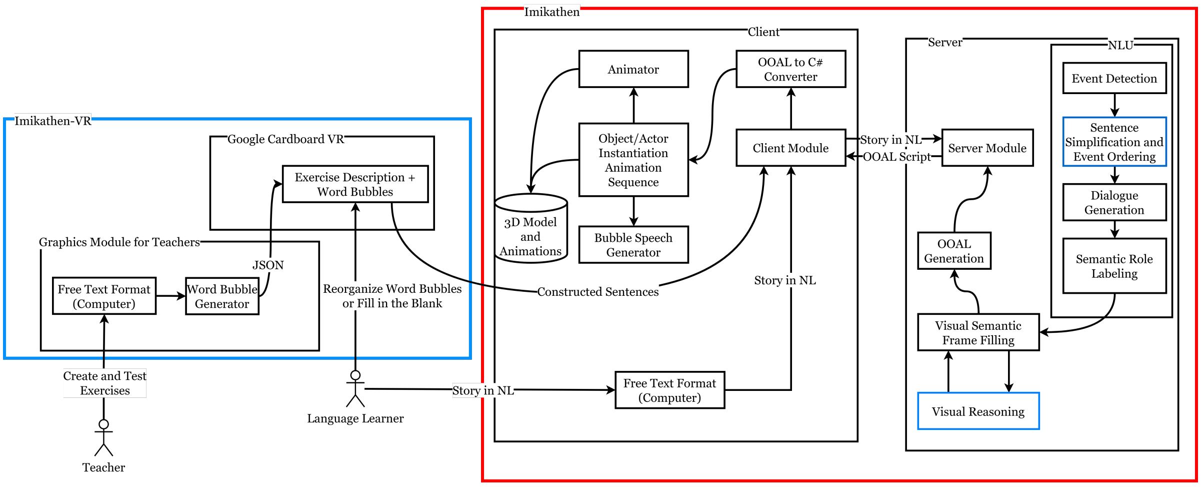


Figure 3: System architecture.

interface enables teachers to create and validate exercises before assigning them to students. Teachers can first test their stories using Imikathen to ensure proper output, then convert these validated stories into exercises.

Students can access these exercises through a VR game, which they can play using a Google Cardboard headset with their parents' smartphones. The game content is controlled by teachers, who can assign specific exercises for different sessions or weeks, ensuring alignment with their teaching objectives.

Before detailing the new teacher interface and the customizable VR game, we focused on enhancing the animation pipeline, specifically improving sentence simplification and visual reasoning capabilities (highlighted in blue in Figure 3). It's important to note that the VR game uses the same animation pipeline for creating virtual worlds and animations, rather than implementing a new one.

4 SENTENCE SIMPLIFICATION

As highlighted in the architecture, after the initial event detection and dialogue identification stages in our pipeline, non-dialogue sentences require simplification to facilitate subsequent processing tasks and enable proper temporal event ordering. This simplification process is crucial as it transforms complex, multi-event sentences into manageable units that can be properly sequenced along a time axis and that are easier to tag for the semantic role labeler.

To process the complex sentences while maintaining input fidelity, we implement a two-stage approach. First, we decompose complex sentences (those containing multiple events) into simple ones (each containing exactly one event) using a data-driven syntactic simplification approach. This preserves the original meaning while avoiding the ambiguity often introduced by vocabulary-based simplification methods (Praveen Kumar et al., 2022). Second, we employ dependency parsing to analyze and reorder these simplified sentences according to their temporal relationships.

This syntactic simplification strategy serves two crucial purposes: it ensures faithful preservation of the input text's meaning, critical for preventing underspecification in the animation output, and it simplifies the task of visual semantic parsing.

4.1 A Data-Driven Approach to Sentence Simplification

Our goal is to decompose complex sentences into multiple simple ones, each capturing a single event while preserving all semantic arguments essential for visual interpretation. We refer to this as a *lossless simplification*, which, combined with temporal event reordering, ensures no information is lost, a crucial factor in addressing the underspecification challenge inherent in text-to-animation tasks (Bouali and Cavalli-Sforza, 2023).

We begin by defining a dataset suitable for the task; That is a dataset that covers the vocabulary expected from an early-stage language learner. We thus manually annotate 1800 input sentences, derived from Children Picture Books of Project Gutenberg [1], describing complex events linked with a temporal expression or simply a conjunction, as shown in Table 1.

[1] https://www.gutenberg.org/ebooks/bookshelf/22

Table 1: Complex vs. simplified sentences.

Complex	Simplified
Once upon a time in the middle of winter, when the flakes of snow were falling like feathers from the clouds, a Queen sat at her palace window, which had an ebony black frame, stitching her husband's shirts.	It was the middle of winter. The flakes of snow were falling like feathers from the clouds. A Queen sat at her palace window. The window had an ebony black frame. The Queen was stitching her husband's shirts.
While she was thus engaged and looking out at the snow she pricked her finger, and three drops of blood fell upon the snow.	She was thus engaged. She was looking out at the snow. She pricked her finger. Three drops of blood fell upon the snow.
Soon afterwards a little daughter came to her, who was as white as snow, and with cheeks as red as blood, and with hair as black as ebony, and from this she was named "Snow-White."	A little daughter came to her. The daughter was as white as snow. The daughter had cheeks as red as blood. The daughter had hair as black as ebony.

In this work, we address sentence simplification as a monolingual translation task, following established approaches (Wubben et al., 2012; Wang et al., 2016; Narayan and Gardent, 2014). While traditional sequence-to-sequence (Seq2Seq) neural networks have shown promise for this task, their effectiveness is heavily dependent on large training datasets. Our initial experiments using a Long Short-Term Memory (LSTM) architecture yielded suboptimal results, likely due to our limited dataset size. To overcome this limitation, we explore transfer learning as an alternative approach, leveraging pre-trained generative models for sentence simplification.

We investigate three small-sized generative models: GPT-2 (Radford et al., 2019), T5 (Raffel et al., 2020), and BART (Lewis, 2019). Our methodology involves fine-tuning the smallest versions of these models on data consisting of complex-simple sentence pairs, enabling them to generate simplified versions of input text.

We used *GPT-2 Small* with 124 million parameters, *T5 Small* with 60 million parameters, and *BART Base* with 140 million parameters. The dataset comprised 1,888 examples, divided into 65% training (1,227 examples), 15% validation (283 examples), and 20% test (378 examples) sets. Training and validation spanned 10 epochs, with per-epoch durations of 92–101 seconds for T5, 167 seconds for BART, and 161–177 seconds for GPT-2. The models' varying sizes and architectures influenced their performance as we report below.

4.2 Evaluating a Lossless Sentence Simplification

The sentence simplification approach aims to decompose complex sentences into simpler ones, with each output sentence describing a single event. This process is evaluated using a combination of metrics that ensure both structural integrity and lexical and semantic fidelity. We created a *Structural Score* to evaluate the model's ability to identify and separate individual events from a complex sentence. We calculate it as:

$$\text{Structural Score} = \frac{\min(n,m)}{\max(n,m)} \tag{1}$$

where n is the number of sentences in the ground truth, and m is the number of sentences in the prediction. A higher score indicates better event identification.

Consider the complex input sentence: "While she was thus engaged and looking out at the snow she pricked her finger, and three drops of blood fell upon the snow." The ground truth decomposition identifies four distinct events. If the model outputs only 3 sentences, the Structural Score would be:

$$\text{Structural Score} = \frac{3}{4} \approx 0.75 \tag{2}$$

To evaluate the quality of each predicted simplification, we need to assess how well it matches its corresponding ground truth sentence. The first step is examining the lexical alignment - checking whether both sentences share the same vocabulary. For this purpose, we use the *ROUGE-L Score*, which quantifies the lexical similarity by identifying the Longest Common Subsequence (LCS) between the predicted and reference sentences (Lin, 2004).

We configure all three models with temperature *0.0* to enforce deterministic generation. To account for lexical variations that maintain semantic equivalence, we implement semantic similarity evaluation using *BERT Score* (Zhang et al., 2019). This metric relies on contextual embeddings from pre-trained BERT model to quantify the semantic alignment between generated outputs and ground truth reference

For this particular task, T5 demonstrated superior performance on the test set compared to both BART and GPT-2. While BART showed stronger performance in the validation set for ROUGE-L and BERT

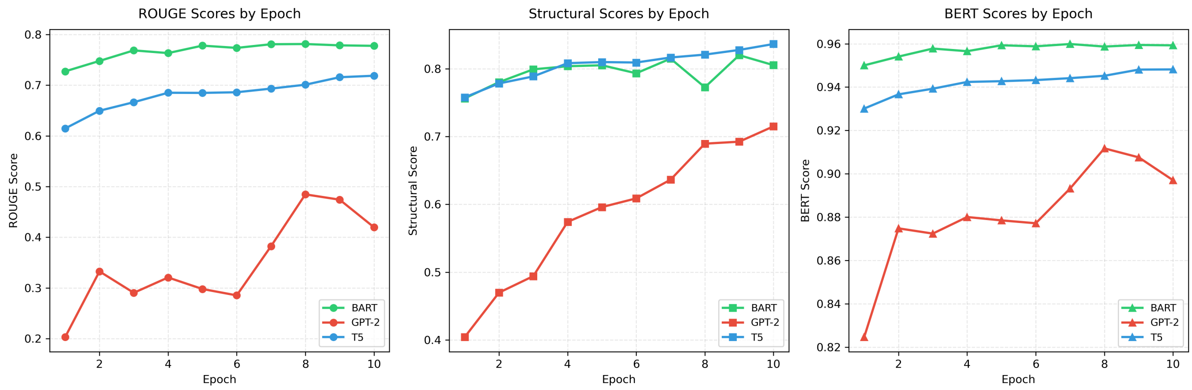


Figure 4: Performance comparison of GPT-2, T5, and BART on the validation set.

scores, as illustrated in Figure 4 , T5 achieved better structural scores during validation. GPT-2 notably struggled to effectively learn the simplification task throughout both validation and testing phases, which can be attributed to its decoder-only architecture (Radford et al., 2019). Unlike GPT-2, both T5 and BART rely on encoder-decoder architectures that allow for bidirectional understanding of the input text, making them better suited for tasks requiring comprehensive sentence understanding and restructuring (Raffel et al., 2020; Lewis, 2019). When evaluating our model on 20% of the dataset (test set), we achieved good results across all metrics: a ROUGE-L score of 0.76, a BERT Score of 0.95, and a Structural Score of 0.84. These high scores indicate strong event identification and separation capabilities (demonstrated by the high Structural Score of 0.84), reliable lexical fidelity maintenance (shown by the ROUGE-L score of 0.76), and excellent semantic content preservation (demonstrated by the BERT Score of 0.95). This comprehensive evaluation framework confirms that our simplification model performs well in two aspects: accurately breaking down complex sentences and maintaining the original meaning and content of each event.

4.3 Reordering the Events

To order events in a storyline chronologically, we develop a rule-based approach based on dependency parsing. Our method processes both the original complex sentences and their simplified counterparts produced by the T5-based sentence simplifier.

To identify which primitives to use in the analysis of the events of a storyline, we look at the sight words from K-1 and K-2 levels according to Dolch (Dolch, 1936)[2]. Children in these grades learn essential temporal expressions such as *first, then, once, when, finally, before, after,* and *at*. We use these expressions as indicators for establishing the chronological order of events.

In dependency parsing, *advcl* or adverbial clause modifier is a type of clause that adds more information to a verb, adjective, or other predicate in a sentence. However, it does this in a modifying role, not as a core part of the sentence. Examples of adverbial clause modifiers include clauses that describe when something happens (temporal clause), what results from it (consequence), what conditions are required for it (conditional clause), or why it happens (purpose clause). To be classified as an adverbial clause modifier, the clause must be dependent, meaning it can't stand alone as a complete sentence. Also, the main action or state described in the clause (the predicate) is what the adverbial clause modifier is providing additional information about. If the modifier is not a clause, it would be classified as an adverbial modifier (advmod) instead. With this understanding of how temporal clues are expressed in dependency structures, we can now turn to applying these insights to our simplified sentences.

Based on the dependency parse in Figure 5, we can reorder the events the following way:

1. We identify the main verbs ”shining” and ”started” in the complex sentence, both marked with *VERB*.
2. Through the dependency parse, we determine their relationship, *advcl*, marked with the red arrow.
3. We then identify the link between the modified event *”started”* to the temporal expression that modifies it (green arrow).
4. We locate the simplified sentences containing these verbs.

[2] https://sightwords.com/sight-words/dolch/#lists

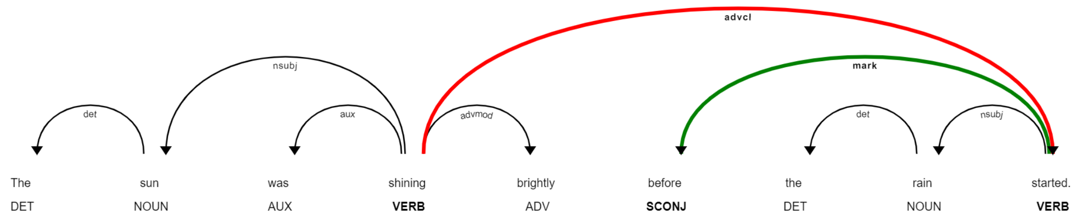


Figure 5: Parse tree for a sentence with a temporal clause.

5. If needed, we reorder them to match the original temporal sequence: ”The sun was shining brightly. The rain started”.

This approach ensures that the temporal relationships expressed in the original complex sentence are preserved in the simplified output, even when the T5 model produces sentences in a different order.

5 VISUAL AND COMMON SENSE REASONING

In the sentence ”The girl painted the wall,” we encounter a common linguistic phenomenon where the text, while syntactically correct, omits crucial visual details such as what tools the girl used to paint. This is an example of underspecification, where natural language leaves out details that humans can easily infer through their understanding of the world, but which are essential for creating an accurate visual representation.

In the current version of Imikathen, we address this underspecification using ConceptNet’s semantic network (Speer et al., 2017). When processing ”paint wall,” ConceptNet helps infer related elements through multiple relationship types: ”HasPrerequisite” relationships reveal necessary tools like ”get a paintbrush and paint,” while ”HasSubevent” relationships suggest actions like ”use a paintbrush” and ”open the paint can.” However, ConceptNet also suggests many elements that are not relevant to the immediate scene (such as ”run out of red” or ”keep your sanity” as relationships), making it challenging to automatically select the proper elements for scene completion.

To improve our approach, we propose using Masked Language Models (MLMs), specifically BERT and RoBERTa (Devlin et al., 2018; Liu et al., 2019), which offer superior contextual understanding compared to predefined relationships. These transformer-based models process each word in relation to all other words in a sentence, making them particularly effective at predicting missing elements based on broader context. We prefer BERT and RoBERTa over GPT-based models because they consider context both before and after the missing word, with RoBERTa being particularly effective due to its more extensive training corpus (Liu et al., 2019). This approach would provide more natural and contextually appropriate completions for underspecified scenes while maintaining flexibility across different visualization scenarios.

5.1 Masked Language Models for Underspecification

To leverage RoBERTa for identifying missing visual elements, we analyze our input sentences using Imikathen’s seven predefined visual semantic frames. These frames define different types of events visually - such as motion, communication, or posture change - and serve as templates for what visual information is needed to animate a given event. Our semantic role labeling module processes each sentence to detect key visual components based on these frames, which then guide our queries to the knowledge base. For instance, when analyzing a motion event like ”He ran north towards the forest through the meadow,” the system identifies the actor (”He”), the path (”meadow”), the destination (”forest”), and the default location (where the actor starts, inferred from context).

$$\textbf{MOTION VERBS}\left\{\begin{array}{rl} Actor: & he \\ Action: & ran \\ Source: & default \\ Destination: & forest \\ Pace: & default \\ Manner: & default \\ Emotion: & default \\ Path: & meadow \\ Direction: & north \end{array}\right.$$

Example: He (actor) ran (pace) north (direction) towards the forest (destination) through the meadow (path).

We associate the default value with the pace element, meaning that this will use the predefined speed encoded in the running animation file. A verb modifier (adverb, such as quickly or slowly) would have been used to modify the pace and the manner otherwise. The input is not considered as underspecified as the running motion can be animated properly given the textual description.

Other cases of underspecification might result in unnatural animations. In a sentence like ”Bob wrote a letter to his friend”.

$$\textbf{COMMUNICATION}\begin{cases} Actor: & Bob \\ Action: & wrote \\ Message: & letter \\ Recipient: & friend \\ Instrument: & NONE \end{cases}$$

Example: Bob (actor) wrote (action) a letter (message) to his friend (recipient)

From the above, a realistic animation cannot be generated. The extracted visual elements do not cover the minimum details required for the writing animation as the instrument (pen, pencil..etc) is missing. We can now leverage RoBERTa as a Masked Language Model (MLM), and use the semantic frame as a template for the input with the masks. We formulate the query to the model as:

Bob wrote a ***<mask_message>*** *to a* ***<mask_recipient>*** *using* ***<mask_instrument>***.

Since in our example, the message and the recipient are explicitly stated, these are part of the context and the model has only the instrument to infer. The query is thus formulated as:

Previous sentences. Bob wrote a letter to a friend using a ***<mask>***. *Following sentences ...*

The model is able to generate the following distribution for the top 10 probable tokens:

Token	⟶	**Prob**
pen	⟶	0.3589
pseudonym	⟶	0.2809
computer	⟶	0.0495
calculator	⟶	0.0229
pencil	⟶	0.0143
mouse	⟶	0.0103
laptop	⟶	0.0097
keyboard	⟶	0.0071
photograph	⟶	0.0059
robot	⟶	0.0055

While LLMs, RoBERTa or others, are good in generating contextually-relevant text, in this case, one token (word) that should serve as the writing instrument, there’s no way to actually force it to limit the predictions to instruments only, the second most probable prediction for the masked token was ”pseudonym”, which despite being contextually correct is not an element we can visualize or that will help in the naturalness of the visualization. To force the MLM to focus only on the terms relevant for our query, we reinclude ConceptNet in our visual reasoning process. The approach for fetching only visually-relevant arguments using ConceptNet involves querying specific relations that ConceptNet defines. For the action ”write,” the program queries ConceptNet for relations such as ”UsedFor,” ”CapableOf,” ”UsedBy,” and ”CreatedBy.” These relations help in identifying objects and tools commonly associated with the action. For example, ”UsedFor” might return ”pen,” ”paper,” and ”book” as objects that writing is used for, while ”CapableOf” might return tools like ”computer” and ”typewriter”. Initially, RoBERTa predicts the top candidate tokens for a masked position in the sentence, providing both the tokens and their probabilities. These predictions are then filtered by using ConceptNet to verify if each token is a suitable instrument for the specified action, focusing on the ’UsedFor’ relationship. For the example above, when we check the overlap of ConceptNet’s output for writing instruments and RoBERTa’s predictions, we can converge towards the following tokens and their associated normalized probabilities:

Token	⟶	**Prob**
pen	⟶	0.9439
pencil	⟶	0.0376
keyboard	⟶	0.0185

The above then indicates that our approach limits the candidate tokens to those relevant for the visualization task and our visualization pipeline can proceed with the generation. It remains, however, hard to test our approach to verify the degree of its effectiveness.

6 SYSTEM PROTOTYPE

This section details the development of a system designed to enable educators to create exercises that facilitate students’ practice of vocabulary and grammar relevant to the weekly classroom lessons. The system architecture has been modified, as illustrated in Figure 3, to incorporate a layer where educators can specify the vocabulary to be practiced each week.

6.1 Designing for Google Cardboard

Our decision to use Google Cardboard was driven by its cost-effectiveness, which extends the accessibility of VR technology to children in developing countries. In a prior project, we developed a cardboard VR programming game, Imikode (Bouali et al., 2019), which was tested in Nigeria and wherein the learners indicated their satisfaction with the learning experience (Sunday et al., 2023). Despite its affordability, the interaction capabilities of Cardboard are constrained to head movements and a single button or gaze-based controls. Users can navigate by looking in the desired direction and then looking down. Given these limitations, Cardboard is most suitable for brief VR experiences due to comfort and hardware constraints.

With these considerations, we designed short games aimed at helping children practice weekly vocabulary, allowing educators to create new exercises aligned with their curriculum.

6.2 Setting up the Exercises

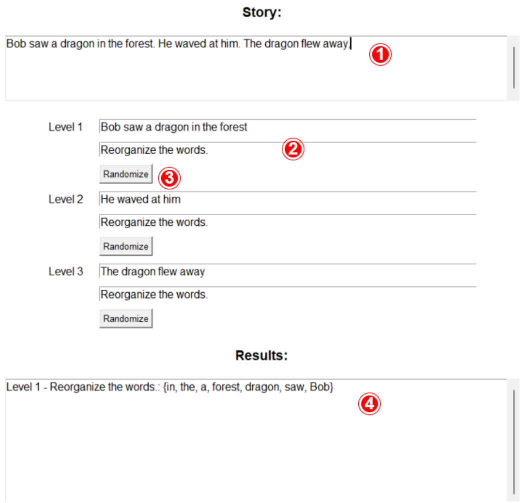
The system includes two primary types of exercises. The first type addresses word reordering, typically used for grammar exercises. The second type includes fill-in-the-blank exercises, which are highly adaptable. Educators can ask students to use correct punctuation, temporal expressions, adverbs, or verb tenses, adapting to a diverse range of questions.

6.2.1 Type 1: Word Reordering

As depicted in Figure 6, to set up a word reordering exercise, educators input a story with multiple sentences into area 1, this is a story that was already tested in Imikathen. The system then segments the story into individual sentences, generating exercises when the user clicks on ”Tokenize”. Each sentence becomes a candidate question. In area 2, educators specify the instructions for students. By clicking on ”randomize” (marked with 3), the exercise is created in the results area (marked with 4).

6.2.2 Type 2: Fill in the Blank

To create a fill-in-the-blank exercise, the steps shown in Figure 7 are followed. A story is input into area 1, which the system then tokenizes into sentences, each becoming an exercise. For example, level 2 keeps the default instruction ”fill in the blank,” while level 3 is edited to a conjugation exercise (marked with 3). Educators select the word to hide in area 4 and specify the candidate words in area 5. The exercise is added to the results section (marked with 6). This process generates a JSON file with the teacher-defined exercises that the VR game can use in its game levels.



Story:

Bob saw a dragon in the forest. He waved at him. The dragon flew away

Level 1 — Bob saw a dragon in the forest — Reorganize the words.

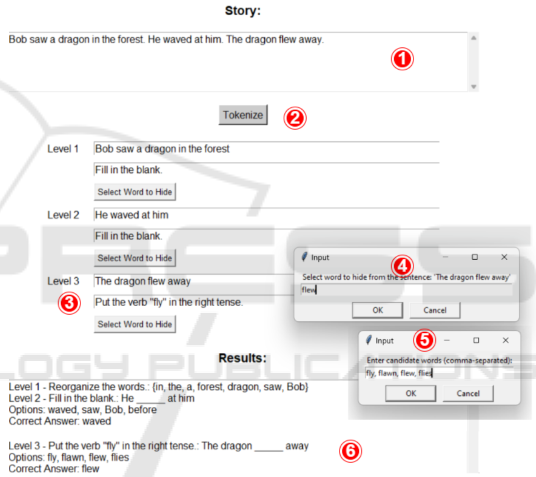
Level 2 — He waved at him — Reorganize the words.

Level 3 — The dragon flew away — Reorganize the words.

Results:

Level 1 - Reorganize the words.: {in, the, a, forest, dragon, saw, Bob}

Figure 6: Setup for reordering exercises.



Story:

Bob saw a dragon in the forest. He waved at him. The dragon flew away.

Level 1 — Bob saw a dragon in the forest — Fill in the blank.

Level 2 — He waved at him — Fill in the blank.

Level 3 — The dragon flew away — Put the verb "fly" in the right tense.

Input: Select word to hide from the sentence: 'The dragon flew away' — flew

Input: Enter candidate words (comma-separated): fly, flawn, flew, flies

Results:

Level 1 - Reorganize the words.: {in, the, a, forest, dragon, saw, Bob}
Level 2 - Fill in the blank.: He _____ at him
Options: waved, saw, Bob, before
Correct Answer: waved

Level 3 - Put the verb "fly" in the right tense.: The dragon _____ away
Options: fly, flawn, flew, flies
Correct Answer: flew

Figure 7: Setup for fill-in-the-blank exercises.

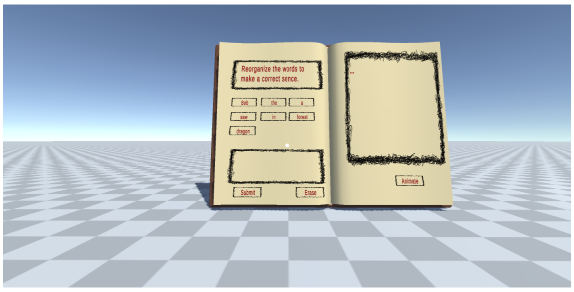
6.3 Imikathen VR

Upon configuring the exercises, the system reads the JSON file created by the educator through the exercise creation menu.

The system subsequently loads an empty environment with a storybook presenting the instructions for the first question, the candidate words, and an area for concatenating the selected words.

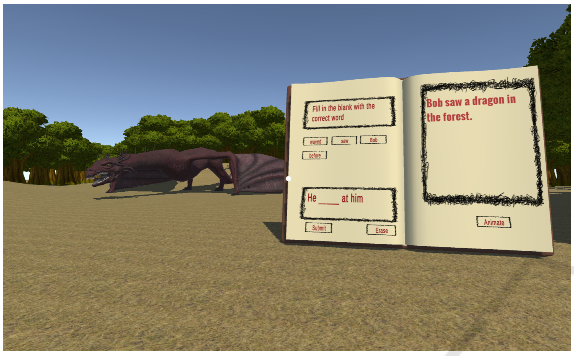
Using the cardboard headset, the child gazes at a word for 5 seconds to select it, adding it to the answer box as shown in Figure 8. Upon selecting the correct words in the correct order, the child submits their answer and clicks ”animate.”

The surrounding environment transforms to re-



Reorganize the words to make a correct sence.

Figure 8: Game screenshot – first exercise loaded.

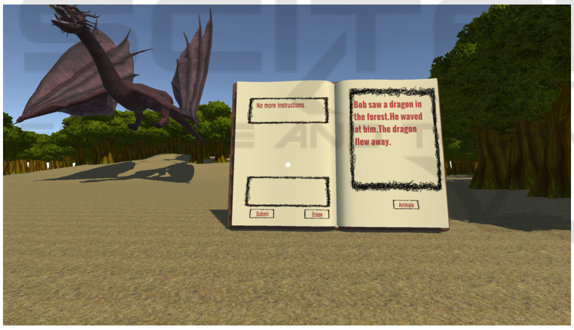


Fill in the blank with the correct word

He ____ at him

Bob saw a dragon in the forest.

Figure 9: Game screenshot – first exercise executed and second exercise loaded.



No more instructions.

Bob saw a dragon in the forest.He waved at him.The dragon flew away.

Figure 10: Game screenshot – third exercise loaded and game shows no further instructions.

flect the constructed sentence, as illustrated in Figures 9 and 10. The subsequent exercise loads with new instructions, allowing the child to proceed, continuously modifying the environment around them based on the educator's design.

This tool enables educators to update exercises dynamically to reflect new vocabulary and grammar skills that students need to practice, releasing the teachers from the dependency on the rigid predefined game narratives in typical educational games.

7 DISCUSSION AND SYSTEM LIMITATIONS

The implementation of Imikathen-VR demonstrates a promising step towards integrating teacher-centered customization in educational gaming. By enabling teachers to design specific vocabulary and grammar exercises, this system addresses the significant challenge of aligning educational games with classroom curriculum, a gap often noted in the literature (Koh et al., 2012; Kirriemuir and McFarlane, 2004; Rice, 2007). However, several limitations and areas for further improvement were identified during the development and initial testing phases.

7.1 Game Design Advantages and Limitations

The primary advantage of Imikathen-VR lies in its flexibility for teachers. They can tailor content to meet their specific educational goals, which is a significant improvement over static, pre-defined game content. Despite this flexibility, the current implementation restricts teachers to two types of exercises: fill-in-the-blank and word reordering. This limitation could be expanded by incorporating additional exercise types, such as sentence construction or interactive storytelling, to offer a more comprehensive learning experience.

Additionally, while the current setup provides a basic framework for language practice, incorporating dynamic feedback mechanisms could significantly enhance the learning experience. For instance, a virtual assistant could provide hints or corrections, guiding students through their mistakes, thereby fostering a more interactive and supportive learning environment.

7.2 Challenges of Google Cardboard

Google Cardboard provides affordable VR access, especially in developing regions, but has a few limitations affecting user experience:

1. Limited Interaction: Head movement and single-button control restrict the complexity of the activities we can design.
2. Poor Comfort: Cardboard is designed for brief use which causes discomfort in longer sessions, impacting learning.
3. Technical Constraints: Lower resolution and processing power reduce immersion and visual quality.

7.3 System Usability

Imikathen-VR, though a promising language learning game, has yet to be tested properly by educators and children to verify its effectiveness. While the tool was developed to give the teachers the chance to decide what their pupils learn, up to now the teacher's input on the design of the system has been very limited. We, thus, recognize, that before we start classroom testing, a validation round and possibly some degree of redesign might be needed to make the most out of the system's potential.

8 CONCLUSION AND FUTURE WORK

Imikathen-VR advances educational technology by putting customizable content creation directly in educators' hands, enabling them to align educational games with their classroom objectives. While this represents significant progress, several key development areas remain. The system needs expanded exercise variety, including sentence construction, metaverse-like group activities, and context-based questions to broaden language practice options. Interaction quality could be enhanced through voice recognition and advanced gesture controls, while AI-driven feedback systems would enable more personalized learning experiences. Additionally, exploring advanced VR platforms while maintaining Cardboard compatibility would balance innovation with accessibility.

Our choice of Google Cardboard enables planned testing with teachers in Morocco and Nigeria, though we acknowledge the current lack of educator and student feedback. Our immediate focus is improving the graphics library and capabilities before proceeding with classroom testing. Through continued development in these areas, Imikathen-VR can evolve into a more effective educational tool, improving language learning outcomes across diverse educational contexts.

REFERENCES

Alfadil, M. (2020). Effectiveness of virtual reality game in foreign language vocabulary acquisition. *Comput. Educ.*, 153:103893.

Amoia, M., Brétaudière, T., Denis, A., Gardent, C., and Perez-Beltrachini, L. (2012). A serious game for second language acquisition in a virtual environment. *Journal of Systemics, Cybernetics and Informatics*, 10(1):24–34.

Birova, I. (2013). Game as a main strategy in language education. *American Journal of Educational Research*, 1:6–10.

Bouali, N. and Cavalli-Sforza, V. (2023). A review of text-to-animation systems. *IEEE Access*, 11:86071 – 86087.

Bouali, N., Cavalli-Sforza, V., and Tukainen, M. (2024). Imikathen: A text-to-vr tool for language learning. Manuscript submitted for publication.

Bouali, N., Nygren, E., Oyelere, S. S., Suhonen, J., and Cavalli-Sforza, V. (2019). Imikode: A vr game to introduce oop concepts. In *Proceedings of the 19th Koli Calling International Conference on Computing Education Research*, Koli Calling '19, New York, NY, USA. Association for Computing Machinery.

Chen, Y.-L. and Hsu, C.-C. (2020). Self-regulated mobile game-based english learning in a virtual reality environment. *Comput. Educ.*, 154:103910.

Cheng, A. Y., Yang, L., and Andersen, E. (2017). Teaching language and culture with a virtual reality game. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*.

Coltheart, M. (1981). The mrc psycholinguistic database. *The Quarterly Journal of Experimental Psychology Section A*, 33(4):497–505.

Derakhshan, A. and khatir, E. (2015). The effects of using games on english vocabulary learning. *Journal of Applied Linguistics and Language Research*, 2:39–47.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Devlin, K. (2011). *Mathematics education for a new era: Video games as a medium for learning*. CRC Press.

Dolch, E. W. (1936). A basic sight vocabulary. *The Elementary School Journal*, 36(6):456–460.

Fisch, S. M. (2005). Making educational computer games" educational". In *Proceedings of the 2005 conference on Interaction design and children*, pages 56–61.

Ismail, R., Ibrahim, R., and Ya'acob, S. (2019). Participatory design method to unfold educational game design issues: A systematic review of trends and outcome. *2019 5th International Conference on Information Management (ICIM)*, pages 134–138.

Khatoony, S. (2019). An innovative teaching with serious games through virtual reality assisted language learning. *2019 International Serious Games Symposium (ISGS)*, pages 100–108.

Kirriemuir, J. and McFarlane, A. (2004). Literature review in games and learning. *Futurelab, A Graduate School of Education, University of Bristol*.

Koh, E., Kin, Y. G., Wadhwa, B., and Lim, J. (2012). Teacher perceptions of games in singapore schools. *Simulation & gaming*, 43(1):51–66.

Lewis, M. (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, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Miftakhova, A. and Yapparova, V. (2019). The game as a necessary component of e-learning materials for children. *EDULEARN19 Proceedings*.

Munoz, H. T., Baldiris, S., and Fabregat, R. (2016). Co design of augmented reality game-based learning games with teachers using co-creaargbl method. *2016 IEEE 16th International Conference on Advanced Learning Technologies (ICALT)*, pages 120–122.

Narayan, S. and Gardent, C. (2014). Hybrid simplification using deep semantics and machine translation. In *The 52nd annual meeting of the association for computational linguistics*, pages 435–445.

Padilla-Zea, N., Vela, F. G., Medina-Medina, N., and González, C. (2018). Involving teachers in the educational video games design process. pages 152–157.

Praveen Kumar, A., Nayak, A., Shenoy K., M., Manoj, R. J., and Priyadarshi, A. (2022). Pattern-based syntactic simplification of compound and complex sentences. *IEEE Access*, 10:53290–53306.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Rice, J. W. (2007). New media resistance: Barriers to implementation of computer video games in the classroom. *Journal of Educational Multimedia and Hypermedia*, 16(3):249–261.

Saurí, R., Littman, J., Knippen, B., Gaizauskas, R., Setzer, A., and Pustejovsky, J. (2006). Timeml annotation guidelines. *Version*, 1(1):31.

Smaldone, R. A., Thompson, C., Evans, M. J., and Voit, W. (2017). Teaching science through video games. *Nature chemistry*, 9 2:97–102.

Speer, R., Chin, J., and Havasi, C. (2017). Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Sunday, K., Oyelere, S., Agbo, F., Aliyu, M., Balogun, O., and Bouali, N. (2023). Usability evaluation of imikode virtual reality game to facilitate learning of object-oriented programming. *Technology, Knowledge and Learning*, 28:1871–1902. Publisher Copyright: © 2022, The Author(s).

Tazouti, Y. (2020). A virtual reality serious game for language learning. *International Journal of Advanced Trends in Computer Science and Engineering*.

Walsh, G. (2012). Employing co-design in the video game design process. In *Handbook of Research on Serious Games as Educational, Business and Research Tools*, pages 1048–1063. IGI Global.

Wang, T., Chen, P., Rochford, J., and Qiang, J. (2016). Text simplification using neural machine translation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1).

Wubben, S., van den Bosch, A., and Krahmer, E. (2012). Sentence simplification by monolingual machine translation. In Li, H., Lin, C.-Y., Osborne, M., Lee, G. G., and Park, J. C., editors, *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1015–1024, Jeju Island, Korea. Association for Computational Linguistics.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2019). Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.