# Promptiverse: Scalable Generation of Scaffolding Prompts Through Human-AI Hybrid Knowledge Graph Annotation

Yoonjoo Lee
School of Computing, KAIST
Daejeon, Republic of Korea
yoonjoo.lee@kaist.ac.kr

John Joon Young Chung
Computer Science & Engineering,
University of Michigan
Ann Arbor, USA
jjyc@umich.edu

Tae Soo Kim
School of Computing, KAIST
Daejeon, Republic of Korea
taesoo.kim@kaist.ac.kr

Jean Y. Song
Information and Communication
Engineering, DGIST
Daegu, Republic of Korea
jeansong@dgist.ac.kr

Juho Kim
School of Computing, KAIST
Daejeon, Republic of Korea
juhokim@kaist.ac.kr

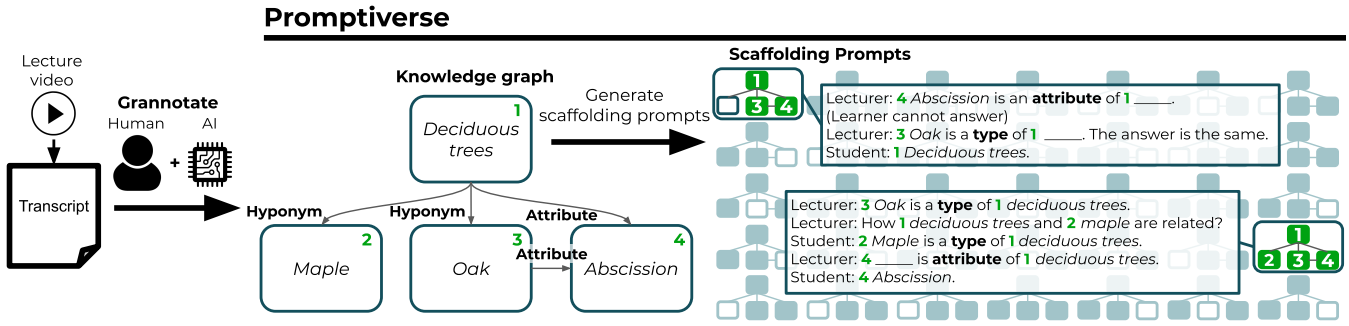



**Figure 1: With Promptiverse, lecture designers can create a large number of diverse scaffolding prompts by extracting rich knowledge graphs from a video script. These knowledge graphs can be efficiently extracted with the help of Grannotate, a human-AI hybrid graph annotation tool. Promptiverse generates scaffolding prompts by traversing knowledge graphs in a way that is guided by the learning patterns of Ausubel's theory [5]. Prompts in each round are generated from a triplet (e.g., *Deciduous trees*-Attribute-*Abscission*) with a variation over knowledge elements that can be asked or provided. Example conversations under Scaffolding Prompts show how elements in the knowledge graph can be traversed in different ways to create diverse multi-turn prompts.**

## ABSTRACT

Online learners are hugely diverse with varying prior knowledge, but most instructional videos online are created to be one-size-fits-all. Thus, learners may struggle to understand the content by only watching the videos. Providing scaffolding prompts can help learners overcome these struggles through questions and hints that relate different concepts in the videos and elicit meaningful learning. However, serving diverse learners would require a spectrum of scaffolding prompts, which incurs high authoring effort. In this work, we introduce Promptiverse, an approach for generating diverse, multi-turn scaffolding prompts at scale, powered by numerous traversal paths over knowledge graphs. To facilitate the construction of the knowledge graphs, we propose a hybrid human-AI annotation tool, Grannotate. In our study (N=24), participants produced 40 times more on-par quality prompts with higher diversity, through Promptiverse and Grannotate, compared to hand-designed prompts. Promptiverse presents a model for creating diverse and adaptive learning experiences online.


*CHI '22, April 29-May 5, 2022, New Orleans, LA, USA*

ACM ISBN 978-1-4503-9157-3/22/04…$15.00
https://doi.org/10.1145/3491102.3502087

## CCS CONCEPTS

• **Human-centered computing** → **Human computer interaction (HCI)**; • **Applied computing** → **Education**; • **Computing methodologies** → **Natural language generation**.

## KEYWORDS

Scaffolding prompt, knowledge graph, human-AI hybrid annotation

**ACM Reference Format:**
Yoonjoo Lee, John Joon Young Chung, Tae Soo Kim, Jean Y. Song, and Juho Kim. 2022. Promptiverse: Scalable Generation of Scaffolding Prompts Through

Human-AI Hybrid Knowledge Graph Annotation. In *CHI Conference on Human Factors in Computing Systems (CHI '22), April 29-May 5, 2022, New Orleans, LA, USA.* ACM, New York, NY, USA, 18 pages. https://doi.org/10.1145/3491102.3502087

# 1 INTRODUCTION

While video learning online has become a widely adopted method for a huge variety of learners, most online video learning environments provide the same lecture video for all learners. With the one-size-fits-all design, learners may struggle with different parts of the lecture content due to their varying prior knowledge. Finding another video that better explains the learner's struggle point might be a solution, but this would require the learner to search for such a video, incurring additional effort. Moreover, as learners are novices, they might not have enough knowledge to discern which video can help overcome their struggles.

One way to alleviate learners' diverse pain points that arise from one-size-fits-all video designs is to utilize diverse scaffolding prompts, which provide hints and ask learners about learning content. For example, if the video's explanation of a concept was not detailed enough for a learner, then the learner would struggle with understanding the content. With diverse prompts, if learners ask the online learning system for help about the concept, then the system can provide suitable scaffolding prompts that give learners a chance to facilitate their understanding about the concept. The prompt would let learners check their understanding, maintain their engagement, and allow them to relate the concept with other concepts they already know [11, 12]. Ultimately, the understanding of the concept would broaden learners' scheme of organizing and perceiving new information, allowing learners to comprehensively understand the lecture. For such support to be successful, a wide spectrum of scaffolding prompts must be prepared to deal with the diverse pain points of learners. However, due to the high cost of authoring, manually creating diverse scaffolding prompts can be impractical and, without this diversity, prompts will often fail to comprehensively address the various concepts dealt within a lecture.

In this work, we introduce Promptiverse, a scaffolding prompt generation approach that uses knowledge graphs to create diverse prompts at scale with low authoring cost. With knowledge graphs on the lecture content, Promptiverse traverses through knowledge entities and relations while considering the meaningful learning patterns from Ausubel's theory [4, 6], which gives insight into designing pedagogically effective prompts. Promptiverse not only generates a large number of prompts out of the traversed paths, but also diversifies prompts' content by varying which knowledge elements are provided as hints and which are elicited from learners. For example, in Figure 1, Promptiverse generates two different scaffolding prompts (green boxes) by varying the traversed paths. Prompts of these various contents allow alternative explanations to be provided to learners who may struggle with understanding the explanation of the video.

Though Promptiverse holds promise for the scalable creation of prompts, constructing the underlying knowledge graph requires lecture designers' manual effort. Therefore, we designed Grannotate, a human-machine hybrid system that assists lecture designers in annotating knowledge entities and relations on lecture transcripts and building hierarchical knowledge graphs. We adopted a mixed-initiative approach that combines human effort and machine recommendations to reduce the human load when constructing knowledge graphs. Based on the lecture designer's first few annotations, the machine recommends candidate knowledge entities, their hierarchical relations, and relation classes, which can then be refined by the lecture designers. Grannotate then allows the lecture designer to inspect how their annotations would impact the generated prompts by showing a preview of the type of prompts that would be generated.

To evaluate Promptiverse and Grannotate, we recruited experts with domain knowledge and teaching experience to create prompts using four different approaches. The four approaches were (1) manually designing prompts from scratch, and generating prompts with knowledge graphs that were constructed using (2) manual, (3) human-AI (using Grannotate), or (4) fully-automatic methods. We compared the approaches in terms of the quantity, quality, and diversity of prompts, and the self-reported cognitive load of experts. Results show that experts using Promptiverse generated 40 times more prompts with more diversity than those who manually designed the prompts. Between the manual, human-AI, and fully automatic graph construction methods, only graphs made with human-AI method using Grannotate generated prompts that were of comparable quality to the hand-designed prompts. We expect that our approach can diversify the authoring of learning content and lead online learning environments to provide eclectic and adaptive learning experiences to different learners.

The contributions of this work are as follows:

- Promptiverse, a novel framework that generates diverse prompts in a scalable manner by traversing knowledge graphs with effective learning patterns.
- Grannotate, a system that allows lecture designers to annotate hierarchical knowledge structures for prompt generation, with the help of AI recommendations on entities and relations.
- Experimental results showing that Promptiverse and Grannotate could produce a higher number of diverse prompts while maintaining a similar level of quality to hand-designed prompts.

# 2 RELATED WORK AND BACKGROUND

As our work introduces a novel framework for generating diverse scaffolding prompts in a large scale with the help of a human-AI hybrid annotation tool, we review research on 1) pedagogical effects of prompting, 2) Ausubel's meaningful learning theory, 3) automatic prompt generation, 4) knowledge representation in learning, and 5) knowledge graph annotation.

## 2.1 Scaffolding Prompts

In online video learning, one-size-fits-all design of videos provide limited support to learners with varying prior knowledge, and scaffolding prompts can be one of the solutions to facilitate learning. Instructors or learning systems can employ scaffolding prompts to elicit learners' knowledge through questions and explanations [12, 24]. These prompts have generally been found to

enhance learning [19, 29, 56]. They also stimulate learners' retrieval practice and can help them realize what they did not understand from the lecture [15, 28]. Shin et al. [49] categorized video prompts according to two dimensions: 1) comprehension-experience orientation, whether the prompt is asking learners' comprehension or asking about the learning experience, and 2) the level of specificity, whether the prompts refer to a specific part of lecture content or not. In Shin et al.'s taxonomy, prompts that focus on experience would not fall in our scaffolding prompts, and we specifically target prompts on specific content. To provide the benefits of scaffolding prompts to learners with diverse prior knowledge, we introduce an approach to diversify the creation of such prompts. These prompts can be used for various learning materials such as in-video quizzes [28], interactive tutoring instruction in online learning [11], and educational conversational agents [59].

## 2.2 Ausubel's Meaningful Learning Theory

To create diverse scaffolding prompts, Promptiverse adopts Ausubel's theory [3–5] on learning patterns. The overarching idea in Ausubel's theory is that knowledge is hierarchically organized. Based on this idea, he proposed that meaningful learning involves understanding the relationships between concepts and identifying new relations. When meaningful learning is done, knowledge is easily retained and applied, whereas rote learning lets learners just memorize all scattered knowledge [4, 38]. Meaningful learning is achieved when the instructional design considers these hierarchical relationships between prior knowledge and new knowledge. Ausubel also described three learning processes by which new knowledge is assimilated into the existing cognitive structure. The first is **superordinate learning**, where learning of a concept is facilitated by connecting it to many well-acquainted examples. For example, when learning deciduous trees, knowing about instances of deciduous trees, such as maples, oaks, and apple trees, would help understanding the concept of deciduous trees. The second is **subordinate learning** which occurs when learners subsume new information to the prior knowledge in a hierarchical manner. This type includes two subtypes of subsumptions which are **correlative subsumption** and **derivative subsumption**. **Correlative subsumption** occurs when learners have to alter or extend their previously learned concept to include the possibility of new information. For example, when learners encounter a tree that has red leaves but only know those with green leaves, then they need to extend the concept of trees to include the cases of red leaves. This process enriches the higher-level concept. **Derivative subsumption** is where new knowledge is an instance or an example of a previously learned concept so learners can leverage existing knowledge to learn the new one. For example, a learner who knows that a tree has a trunk would be able to use that knowledge when learning about a new tree, that the new tree would also have a trunk. The last type is **combinatorial learning**, where learners relate previously acquired knowledge to learn new information that is neither more inclusive nor more specific than the previously acquired one. For example, to learn something about pollination in plants, a learner might relate it to the previously acquired knowledge of how fish eggs are fertilized. While previous work has designed prompts based on Ausubel's high-level lessons [25], to the extent of our knowledge, our work is the first to directly make use of pedagogically effective subsumption patterns to create scaffolding prompts.

## 2.3 Generation of Prompts

Question-answering (QA) is a conversational activity that takes a similar form to scaffolding prompts. Algorithmic QA generation has been an active area of research in NLP and computational linguistics. To drive research in this area, researchers have constructed large crowdsourced conversational QA datasets (e.g. CoQA [44], QuAC [13]), which collected dialogues between crowd workers asking and answering a sequence of questions about a source document. With collected datasets, researchers investigated approaches to generate questions [42] or answers [7] in conversations. These datasets and generated artifacts are close to scaffolding prompts format-wise, but they do not consider educational effects in question answering. On the other hand, QA systems that are designed for pedagogical purposes consider educational effects but are less diverse in terms of concepts dealt with and require a high manual load for QA authoring. These systems were developed as dialogue agents that help students learn programming [59], math [9], and factual knowledge in science, safety, and English vocabulary [48]. Our approach aims to meet the goal of increasing the diversity of concepts while considering educational effects and reducing authoring load.

## 2.4 Knowledge Representation in Learning Context

Learning is a process of integrating new information into existing prior knowledge [37], and structured knowledge representations, such as concept maps, flow diagrams, knowledge graphs, and tree diagrams have often been used to support the process [1, 46]. Various systems in the HCI domain were also designed to help learners structure knowledge with these representations. For example, ConceptScape [32] provided learners with concept maps created through crowdsourcing to help their comprehension and navigation. Similarly, Texsketch [53] supported readers to design diagrams in the process of reading texts to allow them to integrate concepts into a cohesive mental model. In this work, we leverage knowledge representation for another purpose, to generate scaffolding prompts in a scalable manner. To facilitate pedagogical effects of generated prompts, we structure knowledge into knowledge graphs with hierarchical relationships between concepts [39, 40]. Moreover, we facilitate the process of structuring knowledge graphs with a human-AI hybrid annotation tool.

## 2.5 Graph Annotation from Text Data

We introduce a knowledge graph annotation approach that builds upon previous work on graph annotation tools and human-machine hybrid annotation. Graph annotation tools allow annotators to extract how entities in the source medium (e.g., document) relate to each other. Early tools, such as BRAT [52], visualized these annotated relations on the text itself, and they could get visually complex when many relations are annotated. More recent tools accompanied the visualization of graphs to allow sensemaking of annotated relations [54]. Among tools that support visual representation, some

supported the annotation of similar representations such as concept maps or knowledge diagrams [32, 53] and were designed for educational purposes.

To reduce human effort for annotating complex knowledge structures out of text documents, machine assistance can be a viable solution. In natural language annotation, machine assistance has been leveraged in many tools [17, 51], some of them providing machine learning-based recommendations [26, 27]. In this work, we extend existing work to Grannotate, a knowledge graph annotation tool that adopts AI to recommend candidate entities, if any of the entities relate to each other, and how those entities relate. Moreover, Grannotate is designed to assist the accurate generation of prompts, by showing annotators how prompts would be generated out of the currently annotated knowledge graph.

# 3 CHALLENGES IN DESIGNING SCAFFOLDING PROMPTS

In this work, to support diverse learners in video learning, we aim to design a scalable approach to creating diverse scaffolding prompts. To learn requirements for effective scaffolding prompts and difficulties in creating them, we conducted semi-structured interviews with four instructors from a variety of domains ranging from mathematics to computer science to history. In these interviews, we asked about (1) types of scaffolding prompts they mainly generate and use, (2) how they make prompts for various learners, (3) challenges they face while authoring scaffolding prompts, and (4) types of interventions that can alleviate their effort when authoring prompts. Interviews were conducted remotely, and the audio was recorded. After transcribing the audio, one of the authors conducted iterative coding with inductive analysis. Coded results were reviewed with another author.

## 3.1 Findings

Instructors prefer creating shallow follow-up prompts [12], which focus on knowledge pieces that could be answered by directly referring to a specific sentence given in the lecture content (e.g., what passes through the human heart?). Instructors emphasized that effective shallow prompts enable instructors and learners to interact more actively and motivate learners by letting them answer the question easily. They would avoid using hard questions with deep prompts [12], like discussing the student's mental model about the learned content (e.g., how would membrane being permeable to certain substances relate back to capillary walls?). A series of effective shallow prompts let learners relate each single knowledge piece to other pieces and structure them in learners' scheme.

Instructors mentioned that scaffolding prompts should reveal more and more information with multi-turn in an adaptive fashion, elaborating the learner's answer over time and successively letting them elicit knowledge with a instructor-given guide [12]. However, as the turn goes on between the instructor and the learner, deciding which knowledge to provide in each turn becomes challenging. They noted that provided information needs to be related to target knowledge and covered in the lecture. Moreover, multi-turn prompts should be dependent on each other. Instructors felt that considering all these factors makes the authoring of multi-turn scaffolding prompts effortful. Sometimes they could not prepare multi-turn prompts in their lecture despite all the benefits because of not enough time.

Instructors said they usually provide the same prompt to all learners, but they were concerned if the prompt would not be effective to learners with little prior knowledge who might require more guides. To address this issue, P1 sometimes surveys learners' prior knowledge and prepares a few different types of scaffold, with a spectrum of knowledge granularity. However, it is very time-consuming, so in most cases, to learners who cannot get the right answer, P1 just would give answers instead of alternative prompts.

Finally, when lectures get longer, instructors often focus on prompting about main concepts and fail to address minor concepts that are difficult to understand without any scaffolds. Instructors said they usually could not be prepared for all those details when the lecture has too much content. They mentioned that they usually realize the need for scaffolding those details only after seeing learners experiencing difficulties.

## 3.2 Design Goals

Based on the formative study, we present design goals for creating scaffolding prompts.

- G1. Reduce the required time and effort in creating scaffolding prompts that progressively give information related to the target knowledge in multi-turn.
- G2. Create diverse prompts to provide adequate support for learners with varying prior knowledge.
- G3. Create prompts that can comprehensively cover the lecture, even for long ones.

# 4 PROMPTIVERSE: GENERATING PROMPTS WITH KNOWLEDGE GRAPH

We introduce Promptiverse, a scaffolding prompt authoring approach that uses knowledge graphs to generate quality scaffolding prompts in a scalable manner. Effective prompts should consider relations between knowledge elements [12, 24], but, as the formative study revealed, creating diverse prompts while considering knowledge relations is challenging. Promptiverse uses knowledge graphs to computationally and scalably design diverse scaffolding prompts while considering knowledge relations. That is, lecture designers can structure lecture content into a knowledge graph, and the Promptiverse uses the graph to generate prompts with pedagogical patterns. Traversability of a knowledge graph is key in achieving our design goals: multiple entities and relations in a knowledge graph can be automatically traversed in multiple ways, which can produce diverse scaffolding prompts (G1, G2). Moreover, once the knowledge graph thoroughly covers the knowledge within a lecture, it would create a comprehensive set of prompts that cover most of the lecture content (G3).

Specifically, Promptiverse codifies 1) how a single prompt can be generated, 2) how a dyad of prompts can be generated with single prompts while keeping the coherency in prompt content, and 3) how multi-turn prompts can be formulated with dyads of prompts in pedagogically meaningful ways. As building knowledge graphs involves the lecture designer's effort, in a later section (Section 5), we introduce our knowledge graph annotation system that provides machine learning (ML) recommendations to facilitate the process.

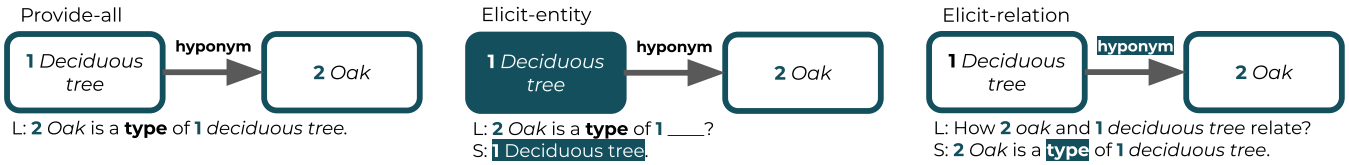



**Figure 2: Different types of single prompts that vary by which knowledge elements in the triplet are provided or hidden. Elicited knowledge elements are shaded in dark teal. In *provide-all* (left), both knowledge entities and the connecting relation are given in the prompt. In *elicit-entity* (middle), one of the entities is asked while giving the other entity and the relation as hints. *Elicit-relation* (right) asks about the relation while giving both entities as hints.**

## 4.1 Representing Lecture Content into a Knowledge Graph

As a preliminary, we describe how we represent lecture content into a knowledge graph that consists of entities and relations [40]. Entities are concepts from the lecture, which are considered educationally important by the lecturer. On the other hand, relations explain how those entities are related to each other. With one relation, there would be two connected entities, and these three elements constitute a triplet. By combining multiple triplets, a knowledge graph is formed (knowledge graph in Figure 1). Our knowledge graph structure mainly focuses on hierarchical relations between knowledge entities, as leveraging such relations is known to facilitate learning [4].

To formulate our knowledge graph structure around hierarchical relations, we referred to existing knowledge relation taxonomies [18, 20, 22, 34, 36, 40]. Considering previous work, we identified seven cross-hierarchy relations and four in-hierarchy relations. Cross-hierarchy relations explain how entities in different hierarchy levels relate, hence one being higher than the other (e.g., "machine learning" (hypernym) and "supervised learning" (hyponym)). Cross-hierarchy relations span over a spectrum, from causal relations (Cause-Effect) to examples (Abstract-Instance), subtypes (Hypernym-Hyponym), features (Object-Attribute), inclusion (Whole-Part), means (Purpose-Used), and substeps (Process-Steps) relations. On the other hand, in-hierarchy relations explain how entities in the same hierarchy level relate to each other (e.g., "supervised learning" is comparable to "unsupervised learning"). For this type, we identified Sequence, Compare/Contrast, Identification, and Coreference. Note that Identification and Coreference are different in that Identification is used when two different entities are considered to have the same meaning in the lecture, while Coreference is used to indicate that two entities are the same thing. We name any relations that fall into cross- or in-hierarchy relations as *class relations*. For cases where the knowledge relations are not best explained with this taxonomy, we also allow *open relation*, which is a relation freely definable by the lecture designer. In the next sections, we explain how this knowledge structure is used to generate prompts.

## 4.2 Mechanism for Generating Prompts

We explain our novel mechanism of generating scaffolding prompts from knowledge graphs on lecture content. Promptiverse focuses on prompts that correspond to shallow questions according to Chi et al. [12]. Our scope sets a lower participation threshold to elicit more learner participation. Among shallow follow-ups, we more specifically focus on prompts that explain or ask about the relations between knowledge elements.

*4.2.1 Generating a Single Prompt.* The most basic mechanism for generating prompts out of a knowledge graph is to derive a single round of prompts from a triplet. For example, when there is a triplet that consists of the entities of "shape of distribution" and "modality", and a connecting relation of "attribute", we can design a prompt that asks about the relation between the entities, like "How are modality and shape of distribution related?" In a prompting sentence, a knowledge element can serve two functions—1) be *provided* in the sentence or 2) be a subject to be *elicited* from learners [24]. For example, in the previous example statement, "modality" and "shape of distribution" are entities provided and the relation of "attribute" would be elicited from learners. Note that in a prompting sentence, at least two knowledge elements should be provided. For example, if more than two knowledge elements are hidden and are to be elicited from learners, the question would be too challenging as there is little information to derive the answer to be elicited (e.g., Supervised learning has **which** relation to **what**?).

Based on how knowledge elements are elicited or provided, a single triplet can turn into three versions of single prompts (Figure 2). If the online learning system is aware of the learner's level of understanding, these different prompts can be presented adaptively to learners. For example, if the learner barely understands the lecture content, it would be more effective to provide them with all the information rather than eliciting it.

Among the three versions (Figure 2), the first type is *provide-all*, where all information in a triplet is provided to the learner. The other types all involve elicitation. These types are *elicit-entity*, and *elicit-relation*. *Elicit-entity* provides one entity and a relation, and elicits the other entity in the triplet. *Elicit-relation* provides both entities and asks students about the relation between those entities.

*4.2.2 Generating a Dyad of Prompts.* Prompting can be more effective by combining multiple prompting sentences into one coherent set of multi-turn scaffolding prompts. One basic mechanism for generating such multi-turn prompts is to share an entity between the two prompts (Figure 3a), as that entity would bridge the context between the prompts. Another basic mechanism is not to elicit already provided or elicited knowledge elements again (Figure 3b), as asking about the already mentioned information would be pointless in many cases. The only exception is when the learner could not answer a specific entity and the prompt asks for the entity in other ways. In this case, the following prompt would give more

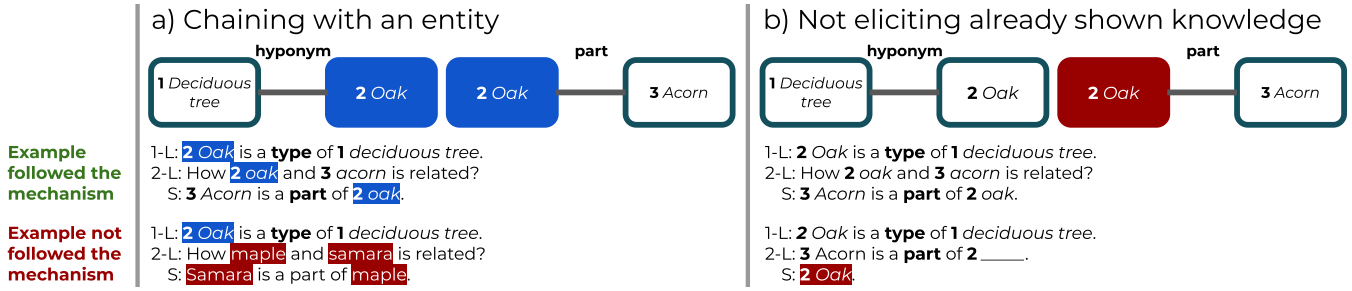



**Figure 3: Two mechanisms for generating a dyad of prompts. Examples that do and do not follow the mechanisms are both shown. a) A dyad of prompts share an entity. In the figure, *oak* is being shared between two rounds of prompts. However, in the counterexample, the following prompt does not have any entity shared with the preceding one. b) The following prompt should not elicit a knowledge entity from the preceding prompt. In the given example, *oak* is not elicited, but provided in the second turn. On the other hand, in the counterexample, *oak* is elicited even though it has been mentioned before, thereby breaking coherence of prompts.**

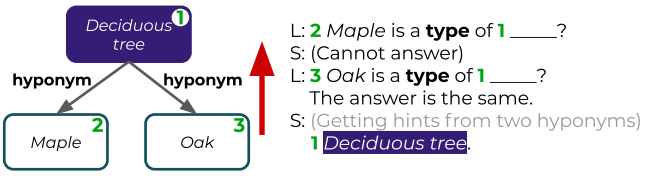



**Figure 4: Example prompts from a superordinate learning pattern. Subordinate knowledge entities (white) are provided first and then the superordinate knowledge entity (purple) is elicited after. It facilitates the learning of the superordinate entity by connecting it back to many subordinates.**

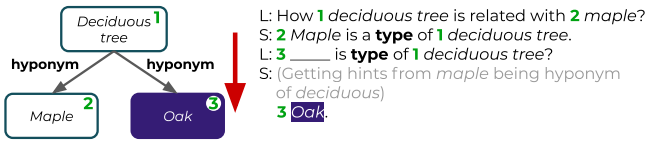



**Figure 5: Example prompts from a correlative learning pattern. One superordinate entity (white), its subordinate entities (white), and connecting relations are first given as prompts, then a new subordinate entity (purple) is introduced in the next prompt. This learning pattern enriches the knowledge about one superordinate knowledge by adopting the new subordinate.**

information to help learners to answer the entity. With these two mechanisms, we can generate diverse dyads from three entities and two relations, by permutating on whether to elicit or provide in each prompt turn and, if eliciting, on which entity to elicit.

*4.2.3 Generating Educationally Effective Prompts.* While dyads of prompts can seem syntactically coherent, it is not clear which series of dyads would be educationally effective. To expand dyads of prompts to educationally meaningful multi-turn prompts, we took Ausubel's theory as inspiration and designed mechanisms for them [3, 5]. Ausubel's theory explains meaningful learning happens when considering how each knowledge would be organized with respect to other related knowledge in cognitive structure. Specifically, Ausubel emphasized the role of hierarchical relations between knowledge entities and the order in which they are introduced to learners. With our mechanism, we adopt these patterns into traversals on a knowledge graph and describe how these traversals can turn into prompts.

There are four types of multi-turn scaffolding prompting mechanisms inspired by Ausubel's theory: *superordinate*, *correlative*, *derivative*, and *combinatorial* prompting. First, superordinate learning (Figure 4) happens when learners first learn subordinate knowledge elements (**Hypernym-Hyponym**, *maple* and *oak* in Figure 4), and then relate that to one superordinate entity (*deciduous tree* in Figure 4). Hence, in superordinate prompting, each subordinate entity is provided gradually first, and then the superordinate entity is provided or elicited, as in the example in Figure 4. Here, the learning would be more effective if all subordinate entities have the same relation to the superordinate entity, as learners would more easily consider subordinate entities together compared to when relations are all different.

The second is correlative learning (Figure 5), where the learner has a model of a superordinate entity (*deciduous tree* in Figure 5), and learns new related subordinate knowledge (**Hypernym-Hyponym** and *oak* in Figure 5) related to that superordinate entity. In correlative prompting, this pattern would be realized by consecutively providing or eliciting other subordinate entities and relations to a superordinate entity as in the example of Figure 5. The subordinate relations do not have to be consistent, as correlative learning is about adding new related knowledge to one superordinate entity.

In derivative learning (Figure 6), there is a superordinate (*deciduous tree* in Figure 6) and a subordinate(*maple* in Figure 6) entity in hierarchical relations of either Abstract-Instance or Hypernym-Hyponym. These entities have relations of the same class (**Whole-Part** in Figure 6) stemming out of them. In derivative prompting, the prompt expects the learners to leverage the hierarchical relationship between the superordinate and subordinate entities when learning the shared knowledge relations. For example, both in Figure 6-1 and 2, the hierarchical relationships are provided or elicited

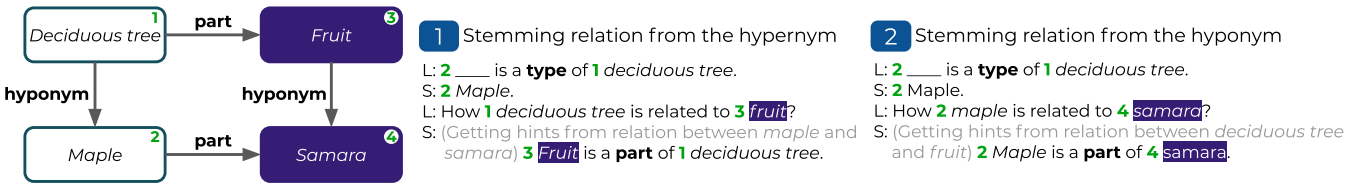



**Figure 6: Example prompts from a derivative learning pattern. First, a prompt about entities (white) in a hierarchical relation (either in Hypernym-Hyponym or Abstract-Instance) is introduced, followed by a prompt about how one of those entities relates to an entity with a stemming relation (purple). This pattern expects learners to leverage the relation that stems from one of the entities in the hierarchy relation to understand the other stemming relation in the counterpart entity in the hierarchy relation.**

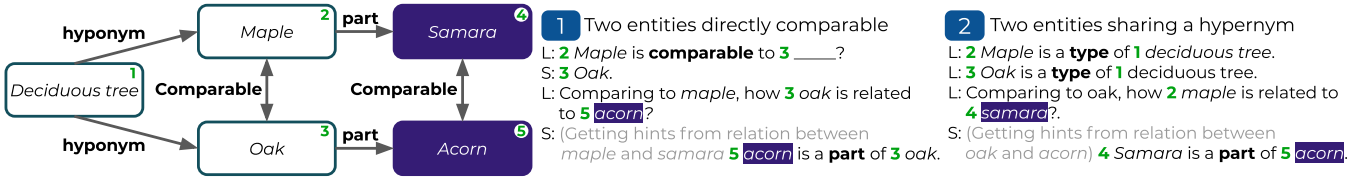



**Figure 7: Example prompts from a combinatorial learning pattern. First, a prompt (or prompts) about entities in a comparable relation (white) is introduced, followed by a prompt about how one of those entities relates to an entity with a stemming relation (purple). This pattern expects learners to leverage the relation that stems from one of the entities in the comparable relation to understand the other stemming relation in the counterpart entity in the comparable relation.**

first. Then, the relation stemming from either superordinate or subordinate entity (**Whole-Part**) would be subsequently provided or elicited. Here, the hierarchical relations are restricted to Hypernym-Hyponym and Abstract-Instance, as stemming relations can be shared in these two.

The last type is combinatorial learning (Figure 7). In this, there are comparable knowledge entities (*maple* and *oak* in Figure 7). These entities have relations of the same class (**Whole-Part** in Figure 7) stemming from them, similar to derivative learning. Comparable entities can share the same superordinate entity (*deciduous tree* in Figure 7), related in Hypernym-Hyponym or Abstract-Instance. It is because one superordinate's hyponyms or instances would be comparable to each other. In combinatorial prompting, the prompt expects the learners to use the comparable relations when learning the knowledge relations that stem out of comparable entities. As in Figure 7, first, the comparable relations are prompted, either by directly referring to comparable relation or to the shared superordinate. Then, the relation stemming from one of the comparable entities (**Whole-Part**) is provided or elicited, expecting the comparable relation would help learn the stemming relation.

# 5 GRANNOTATE: KNOWLEDGE GRAPH ANNOTATION SYSTEM

While Promptiverse can create diverse prompts by traversing a knowledge graph, the effort required to build a knowledge graph [2] can be a bottleneck. To reduce this barrier, we introduce Grannotate, an annotation tool that leverages a visualization of a knowledge graph and AI-powered recommendations. We first investigated challenges in building knowledge graphs with an existing baseline tool. Based on this investigation, we designed our tool.

## 5.1 Challenges in Graph Annotation

To identify challenges in graph annotation, we conducted a pilot study with an existing tool. We recruited five participants (graduate students). Among the three lecture topics that we prepared (Machine Learning, Chemistry, Data Structure), each participant picked one topic that they were most familiar with. Each study session lasted for 40 minutes. During the session, participants were first given instruction on the annotation guideline. Then, they used BRAT [52], which allows users to annotate knowledge entities and their relations on a document. The tool shows annotated triplet relations on the document as two highlighted text snippets with an arrow connecting them. Each participant watched a lecture video on the topic they picked and annotated the lecture transcript on BRAT. At the end of the study, participants provided feedback through a semi-structured interview [41]. The sessions were video recorded for analysis.

*5.1.1 Findings.* Videos were coded for different annotation activities, including reading, adding a node, adding a relation, labeling a relation, and editing. One of the authors did iterative coding with inductive analysis, and the resulting codes were reviewed by the other authors. At a high level, we found four critical challenges with the existing graph annotation tool. First, only with the text, it is hard to identify the whole structure of the graph **(C1)**. Participants said that this was because the graph gets more complex as they annotated more entities and relationships. Second, it is hard to verify whether the annotated entities and relations are correct based only on the examples in the guideline **(C2)**. For example, they were not sure if they annotated entities accurately—i.e., with the correct span (e.g., "electronic structure in an atom" vs. "electronic structure"?). They also wanted some examples on how to use the relation classes

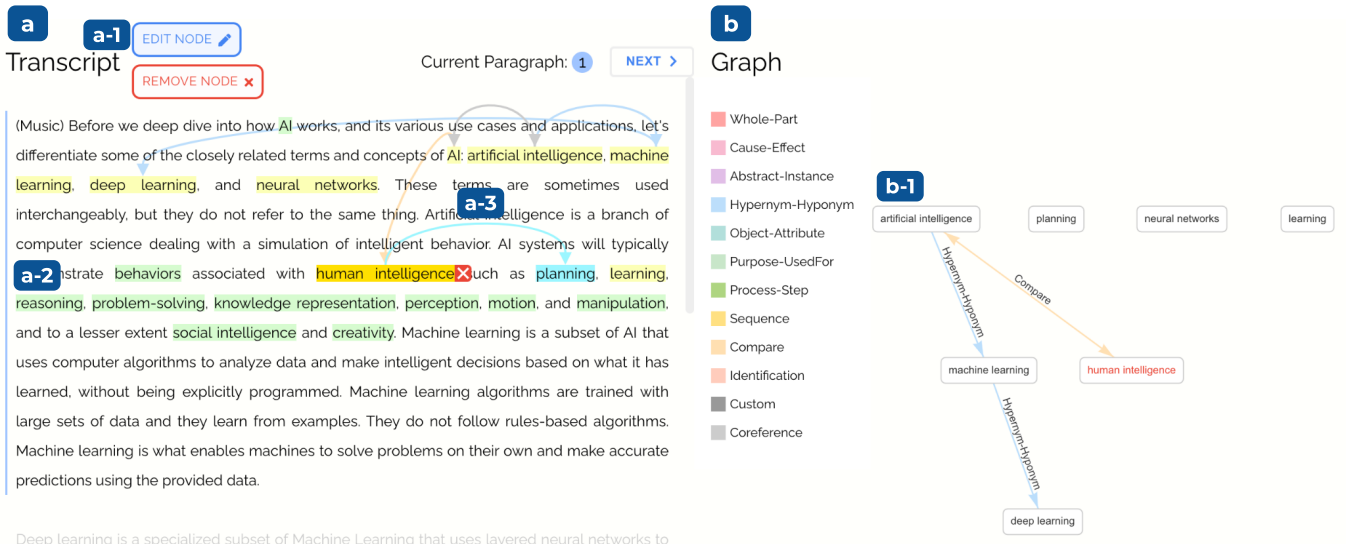



**Figure 8: The interface of Grannotate. On the left, a) transcript is shown with annotations. On the right, b) the color codes for the relation classes are shown, and b-1) annotations are visualized in a graph. On the transcript, the user can select a part of the text to annotate an entity, or select two entities to annotate a relation. The user can also annotate the relation by connecting two entities on the graph. a-1) With an entity or relation selected, the user can also edit or remove them. The tool also provides AI recommendations to lower the load of users. a-2) Based on the currently annotated entities, the tool recommends other candidate entities. a-2) If the annotator selects an entity, the tool recommends a potential candidate entity that the selected one could be related to.**

in the guideline with the entities that they annotated. Third, it is hard to assure if all important entities are covered, as there can be many entities **(C3)**. Additionally, participants were sometimes not sure about what entities should be considered important enough to be annotated. For example, P2 said "Did I pick every entity that I should do? I might have missed some of them..." Lastly, it is hard to choose the relation class because they are many and unfamiliar to annotators **(C4)**. Participants needed to refer back to the whole guideline again when annotating a new relation.

## 5.2 Annotation Interface

Based on the challenges (C1-4) identified from the pilot study, we designed our hybrid human-AI annotation tool. First, our tool addresses the problem of sensemaking about the currently annotated results (C1) by having two representations of an annotated graph: overlaid on the transcript and a graph visualization (Figure 8). The visualization is designed to remove clutter and help the user more easily perceive the relations between the annotated concepts. The user can use both representations to add and edit entities and relations. On the transcript, the user can add knowledge entities by selecting a portion of the text. Then, the annotated part is highlighted in yellow (Figure 8a) and the entity is also visualized on the graph-side (Figure 8b-1). By selecting this entity either on the transcript or the graph, the user can edit the name of the entity or delete that entity (Figure 8a-1). Edited names are only shown on the graph. The user can also add a relation by either selecting two entities on the transcript or connecting one entity to another on the graph with a dragging motion. When two entities are connected, the modal for specifying the class of relation is shown (Figure 9). In the modal, the user can select a class from either the cross-hierarchy classes or the in-hierarchy classes. If the user thinks that no class adequately explains the relation between the selected entities, they can specify the relation as an open relation with free text input (Figure 9c-2). If the user wants to switch the order of entities, they can click on the "Switch Order" button.

When selecting a relation class, it can be difficult for the user to understand how the annotated class will be used in prompts (C2). Hence, our tool also shows how prompts would be created out of the annotation (Figure 9e). For example, if "planning" is annotated to be a hyponym of "human intelligence", the system shows example prompts like "Planning is a type of human intelligence", or "What can be a type of human intelligence?". When they select the "Abstract-Instance" class, annotators can also add the setting of the instance, which explains the specific setting in which the instance appears (e.g., "In the setting of classifying images with a cat, a image set with or without a cat is an example training data."). Once the user confirms the relation class, the annotated relation is visualized on both the transcript and the graph (Figure 8). The class of "Coreference" is the only exception, as entities related with this class are shown as a single node on the graph. The user can edit or delete relations by selecting them on the graph-side.

To alleviate the burden of having many options for entities and relation classes, our tool provides AI-driven recommendations (C3 and C4). There are three types of recommendations: 1) entity recommendation, 2) relation existence recommendation, and 3) relation class recommendation. Entity recommendations and relation existence recommendations highlight potentially important entities


**Figure 9: Modal for selecting a relation class. a) The chosen entities are shown, with a direction from left to right. b) Button for switching the direction of the relation. c) Section where the annotator can decide the relation class. Classes for cross-hierarchy relations and in-hierarchy relations are shown. c1) AI recommended classes are accompanied by a "Recommended" highlight below the class button. The chosen class is shown in its color code. c2) When the annotator thinks that none of the classes is adequate, they can come up with a custom open relation. d) When Abstract-Instance is chosen as the class, the annotator can also add a setting to give more context to the prompt that would be generated. e) To help annotators understand how their class selection would impact the prompt generation, an example prompt for the annotated class is shown.**

and relations among all the possible options. Entity recommendations are provided after the user annotates five initial entities. These recommendations are shown as light-green highlights on the transcript-side, that the annotators can click on when they want to add them (Figure 8a-2). Edge existence recommendations are provided when the user selects one of the entities. On the transcript these recommendations are shown as light-blue edges (Figure 8a-3). Finally, the edge class recommendations are shown when the user enters the modal for specifying relation classes, by highlighting the recommended classes (Figure 9c1). These recommendations are the top three classes predicted by an edge class classification model we trained based on a GPT-based large-scale natural language model. Details will be presented in Section 5.3.

## 5.3 AI Recommendation Architecture

We designed three different pipelines to support each of the three recommendations mentioned in Section 5.2, which are entity, relation existence, and relation class recommendations.

To provide entity recommendations, we use the DYGIE++ framework [57], which supports three information extraction tasks with state-of-the-art performances: named entity recognition, relation extraction, and event extraction. With DYGIE++, we trained a model on the SCIERC dataset [35], which is a collection of 500 scientific abstracts annotated with scientific entities, their relations, and coreference clusters. With the trained model, we first extracted entities from the whole script. When the user annotates five initial entities, from extracted entities, our pipeline identifies entities that co-occur with these initial entities in the same sentence. These identified entities are recommended to the user as entity recommendations. As the user annotates more entities, this process is repeated with the newly added entities. For relation existence recommendation, we also extracted relations using the same framework. When the user selects an entity, the pipeline identifies if that entity is included in one of the relations found by the framework. If such a relation is found, the user is recommended to relate the selected entity with the other entity found in that relation.

Our relation class recommendation is enabled by a Transformer-based classification model which takes two entities as input and predicts the top three classes that best explain the relation between those two entities. Specifically, we adopted p-tuning approach [30, 31, 33, 50], which tunes "prompts" that can guide Transformer-based language models to serve the targeted task. GPT-based models could be guided to serve different tasks according to different natural language "prompts". For example, by inputting a natural language prompt "Estimate relational classes between two entities" to the model, the model can be guided to serve the task. Instead of hand-designing these natural language tokens, the p-tuning approach learns the optimal prompt tokens on the continuous embedding space, to replace the natural language prompts. We adopted this approach as it has shown reliable performance even with a small number of training data instances [33]. We used GPT-Neo with 2.7 billion parameters [8] as our language model. To train the model that classifies relation, we used 180 samples of data points from lectures on five different domains (Networking, Design and Product, Probability and Statistics, Electrical Engineering, and Machine Learning). Two authors annotated knowledge entities and relations separately and merged individual results through discussion. As our corpus has a low number of instances on Sequence, it was excluded from the recommendation. Moreover, as Coreference is a straightforward relation that indicates the same entity, we assumed that annotators would not struggle with selecting this class and excluded it also from the recommendation. It is partly also because having more classes would lower the performance of the algorithm. We used 70% of the dataset as the training set and

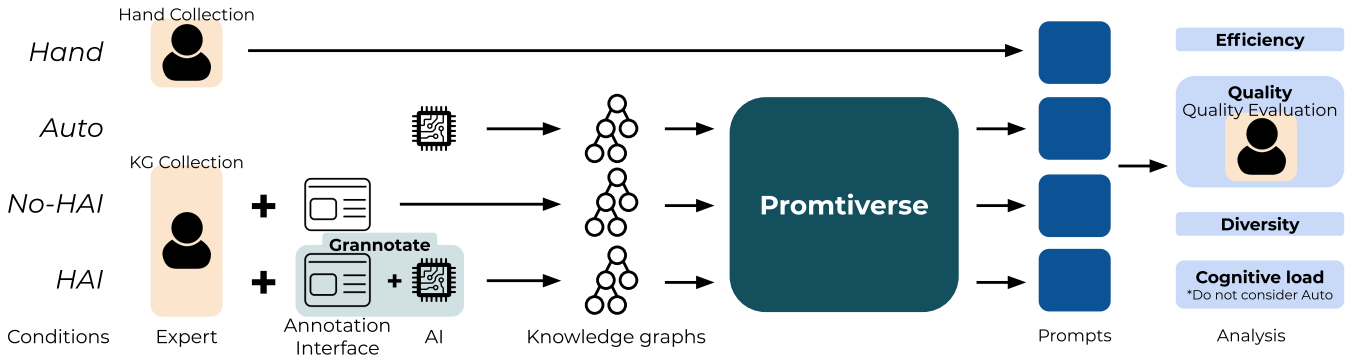



**Figure 10: The design of evaluation. We collected prompts from four conditions, which vary with the involvement of expert participant, AI components, and Promptiverse. In *Hand*, all annotations are manually created by experts from *Hand Collection*. In the other three conditions, Promptiverse was used to generate prompts from knowledge graphs. In *Auto*, knowledge graphs were created only with AI algorithms. In the other two conditions, experts annotated knowledge graphs with our annotation interface in *KG collection*. In *No-HAI*, experts did not get AI support while experts in *HAI* got AI recommendations from Grannotate. *KG Collection* was designed to be within-subject design, where experts annotated graphs with both *No-HAI* and *HAI* conditions. Collected prompts were analyzed in four analyses, which scoped on efficiency, quality, diversity, and cognitive load. In quality analysis, evaluation was done with expert evaluators in *Quality Evaluation*. In cognitive load, *Auto* was not analyzed as it does not involve any expert.**

30% as the test set. For training, we split the training set into a 7 : 3 ratio for training and validation. Training details are explained in the Appendix. Our model had 64% of accuracy on test data, which is (the number of true relation class being included in our top-3 predictions)/(the number of samples in the test set)×100. While the accuracy is not extremely high, it is above the random chance of the true relation being included in the recommendation (33.3%) and hence could give meaningful support to users. Moreover, the model would make annotators not over-rely on the algorithm, because its below-perfect accuracy would require annotators to consider the recommendation.

### 5.4 Interface Implementation

Our graph annotation interface is implemented as a web application by using HTML, CSS, and JavaScript. We used React and Node.js as our front-end and back-end frameworks, respectively. For storing annotation data, we used MongoDB. We implemented the AI recommendation server separately, as a Flask-based API.

## 6 EVALUATION AND RESULTS

To assess if Promptiverse and Grannotate lead to the scalable generation of diverse scaffolding prompts, we conducted a series of experiments. Specifically, we ask the following two questions:

- RQ1. How the quantity and quality of prompts from Promptiverse would be different from those created fully manually? Which approach would impose more load on doing the task?
- RQ2. How the quality and quantity of prompts from Promptiverse would be impacted by different levels of automation involved in creating knowledge graphs? Which approach would impose more load in doing the task, if annotators are involved?

To answer these questions, we conducted data collection and analyses considering the following four conditions:

- *Hand*: Prompts are manually designed.
- *Auto*: Prompts are generated with Promptiverse, and input knowledge graphs are created fully automatically with models used in AI recommendation features.
- *No-HAI*: Prompts are generated with Promptiverse, and input knowledge graphs are manually created with the tool that does not have AI recommendation features.
- *HAI*: Prompts are generated with Promptiverse, and input knowledge graphs are created in a hybrid manner using Grannotate.

Our evaluation design is summarized in Figure 10.

### 6.1 Method

For video materials from which prompts are created, we used videos from two sub-domains of computer science, AI and IoT. We chose these two sub-domains as they are topically distant enough that they cover different knowledge entities, hence being able to show the generalizability of our approaches.

To collect and evaluate prompts, we conducted three rounds of data collection (Orange boxes in Figure 10). The first round focused on collecting manually designed prompts (*Hand Collection*). The second round focused on collecting knowledge graphs for *No-HAI* and *HAI* conditions (*KG Collection*). The last, third round of data collection evaluated the quality of collected prompts (*Quality Evaluation*). Note that for three conditions that involve human annotators, we adopted a mix of within and between-subjects design, where *No-HAI* and *HAI* are combined as within-subject design while they are combined with *Hand* as between-subject. While this approach is not conventional, we argue that this approach still allows us to reliably answer our questions. First, our design would give a penalty

to our proposed conditions, *No-HAI* or *HAI*, as participants would have felt more fatigue by going through two task conditions. Hence, if our approach turns out to have higher quality and quantity, then it means that our approach shows benefit despite penalties in the study design. Second, as *No-HAI* and *HAI* are compared in the same study condition, it would not impact answering our second question.

*6.1.1 Hand Collection.* For *Hand Collection*, we recruited twelve experts (age $M = 26.3$ and $SD = 2.8$, 7 females and 5 males, 4 undergraduate students, 4 graduated students, and 4 industry workers) and asked them to create prompts on one lecture video. We recruited them by using word of mouth and posting advertisements in online forums such as Twitter, Facebook, and the online communities of several colleges. Participants had expertise in one of the domains of our target videos. They also had the experience of teaching using teaching materials that they made ranging between 6 months and three years. For example, these experts included TAs who have taught undergraduate students and made learning materials. Since the videos were not created by the participants, we asked them to watch the videos two times before the study. During the study session, we first gave participants instructions on the scope of prompts to be created, which are scaffolding prompts that deal with relations between knowledge entities in the lecture. Then, participants created prompts for 25 minutes. They are then asked to do a NASA-TLX survey on their cognitive load in creating prompts. The session ended with a short interview asking about their experience of doing the task. Participants were paid 20,000 KRW (approximately USD 17) for their participation.

*6.1.2 KG Collection.* For *KG Collection*, we recruited another twelve experts (age $M = 26.5$ and $SD = 3.2$, 7 females and 5 males, 4 undergraduate students, 6 graduate students, and 2 industry workers) using the same recruiting process. In this collection, with our graph annotation system, participants were asked to annotate two lecture videos we selected. Hence, each participant should have expertise in both domains while having teaching experience in at least one of the domains. Their experiences are ranging from 6 months to 3 years. Similar to *Hand collection*, participants were asked to watch the subject videos two times before joining the study session. They were first given instructions on the purpose of the study and usage of the graph annotation tool. Then, participants annotated graphs from lecture scripts first with one of *No-HAI* or *HAI* condition, assigned randomly, and then with the other condition. For each condition, participants were given 25 minutes to create the knowledge graph. The session ended with a NASA-TLX survey and a short interview on their annotation experience. Knowledge graphs from this collection are fed into Promptiverse and used as prompts for *No-HAI* and *HAI* conditions. Participants were given 30,000 KRW (approximately USD 25.5) for their participation.

*6.1.3 Quality Evaluation.* For *Quality Evaluation*, we recruited two experts as evaluators. They were asked to evaluate prompts from both domains, and hence, should have expertise in both domains while having teaching experience. For this quality evaluation, we first sampled a subset of prompts from each condition. Prompts collected in *Hand Collection* were used for *Hand*, and those generated with knowledge graphs from *KG Collection* were used for *No-HAI* and *HAI*. To collect prompts for *Auto*, we ran our recommendation algorithms, created knowledge graphs only out of them, and then fed those knowledge graphs into Promptiverse. From each video for each condition, we randomly sampled 10 prompts, hence resulting in 80 prompts in total ($10 \times 2(video) \times 4(conditions)$).

The evaluators are asked to watch the video before joining the session. During the session, they were given sample prompts in a blind condition, and we asked them to score questions according to a provided scoring rubric, which is based on the framework for analyzing scaffolding strategies [55] and our goal of creating accurate prompts. The rubric had six criteria including **Direction maintenance**, **Cognitive structuring**, **Reduction of degrees of freedom**, **Recruitment**, **Contingency management and frustration control**, and **Accuracy of knowledge**. These criteria evaluate how well these prompts support students' metacognitive activities, cognitive activities, and student affect, while reflecting accurate knowledge conveyed in the video. Details are explained in the Appendix. These rubrics were asked on a 5-point scale (Not satisfied to Satisfied). We paid evaluators 30,000 KRW (approximately USD 25.5) for their participation in 1.5 hours session.

## 6.2 Results

To answer our research questions on efficiency, quality, and diversity of generated prompts, we conducted three analyses on the created prompts. We also answer questions about the cognitive load of experts and their experience through the NASA-TLX survey and qualitative analysis. For each analysis, we describe our method of analysis, and then the results.

*6.2.1 Efficiency Analysis.* To assess efficiency in creating prompts, we did a statistical test on the number of prompts created by each expert. Note that *Auto* does not have an expert, hence only has one data point for each video. To test if a difference exists among conditions, as there were few data instances that are not in normal

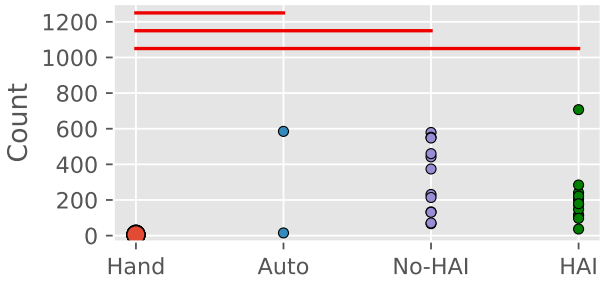



**Figure 11: The number of prompts generated in each condition. For *Hand*, *No-HAI*, and *HAI*, the number of prompts generated from each participant is plotted. Note that each participants' data in *Hand* are close and overlap with each other in this plot. For *Auto*, the counts of prompts from two videos are shown. Red connecting lines indicate that the difference between two connected conditions is significant. *Auto*, *No-HAI*, and *HAI* generated significantly more prompts compared to *Hand*.**

distributions, we conducted a non-parametric Kruskal-Wallis test. Then, as a posthoc analysis, we conducted Dunn's test.

*Result: Participants generate significantly more prompts with Promptiverse than by hand.* The four conditions had a significant difference in the number of prompts generated ($H = 24.59, p < 5e-5$). In post hoc test, we found that *Auto* ($AVG = 300.00, n = 2$), *No-HAI* ($AVG = 316.83, n = 12$), and *HAI* ($AVG = 210.00, n = 12$) generated significantly more prompts compared to *Hand.* ($AVG = 4.92, n = 12$) ($p < 0.05$).

*6.2.2 Quality Analysis.* To analyze the quality of generated prompts, we conducted a statistical analysis on quality evaluation results. For each evaluation question of each prompt, we first averaged scores from evaluators. Then, for each evaluation question, we conducted a Kruskal-Wallis test against four conditions, each of which had 20 prompts. We chose a non-parametric test as the data were ordinal. As a posthoc test, we conducted Dunn's test.

*Result: Prompts generated with HAI are not significantly different from those generated by Hand in quality, except for contingency management/frustration control.* The result of the quality analysis is presented in Figure 12. From a Kruskal-Wallis test, we found that four conditions were significantly different in five criteria, Direction maintenance ($H = 18.2, p < 5e-3$), Reduction of degrees of freedom ($H = 12.85, p < 0.005$), Recruitment ($H = 10.36, p < 0.05$), Contingency management/frustration control ($H = 11.84, p < 0.01$), and Accuracy of knowledge ($H = 30.60, p < 5e-06$). Only in Cognitive structuring, a Kruskal-Wallis test result did not show significance ($H = 4.21, p > 0.1$). With pairwise comparisons, we found out that *Hand* showed higher quality compared to *Auto* and *No-HAI* in four criteria (Direction maintenance, Reduction of degrees of freedom, Recruitment, Contingency management/frustration control, $p < 0.05$). However, when *Hand* is compared to *HAI*, except for Contingency management/frustration control ($p < 0.05$), *Hand* had no significant difference from *HAI* ($p > 0.05$). Moreover, *Auto* showed significantly low Accuracy of knowledge than all other conditions ($p < 0.05$). *Auto* is also outperformed by *HAI* in Direction maintenance, Reduction of degrees of freedom, and Accuracy of knowledge ($p < 0.05$).

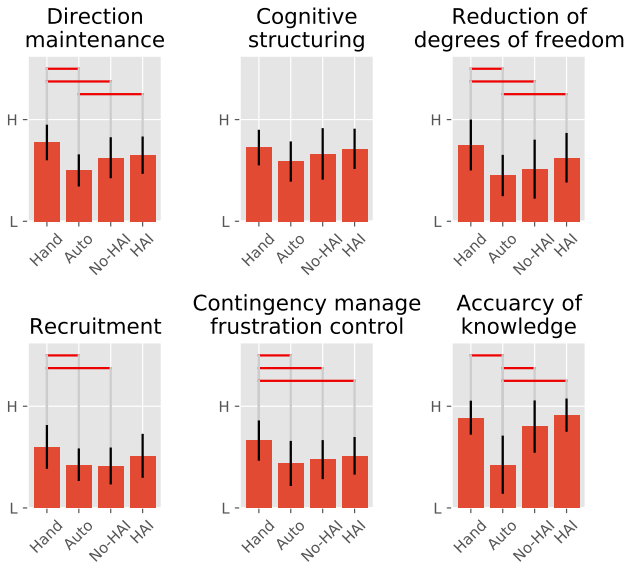



**Figure 12: Quality of prompts measured according to six criteria. Error bars indicate the standard deviation. The red connecting line indicates that the difference between the two conditions is significant. *Hand* outperformed *Auto* and *No-HAI* in direction maintenance, reduction of degrees of freedom, and recruitment, while *Hand* and *HAI* were not significantly different in those criteria. The only significant difference between *Hand* and *HAI* was observed in contingency management/frustration control.**

*6.2.3 Diversity Analysis.* We analyzed the diversity of generated prompts from each condition by embedding them into the vector space, as it is an effective way to quantify the semantic distance between textual data, or how different they are [10, 45]. We used BERT to embed prompts [16]. With embedded prompts, we conducted both quantitative and qualitative analyses. For quantitative analysis, we measured the "area" on the vector space that is covered by prompts from each condition. While the "distance" between elements has been widely used as the metric for diversity [10, 45], it is a metric about "how two elements are different". As we are more curious about "how many elements span over semantic space", we instead measured "area". We calculated the area in the PCA-reduced vector space with the dimension of five, as high dimensionality increases the computation cost. To measure the area, we ran K-means clustering for prompts from each condition with K of 2 to more accurately measure the area. Here, we fixed K for all conditions to have a fair comparison between them. Moreover, we set K to be two as it assured clusters to be distinguishable to each other in most conditions (with high silhouette values [47]). After that, to get the area metric, we ran a convex hull algorithm on each cluster and summed areas from all clusters.

We also conducted a qualitative analysis of the visualization and underlying prompts. To visualize them in two-dimensional space, we took PCA of all vectorized prompts from all conditions (d=2). To analyze the visualization and the underlying prompt, one of the authors first inspected the overall pattern in the visualization, retrieved several prompts (at least 10 samples) from the pattern of interest, and iteratively analyzed them with inductive analysis. The coded result was reviewed with another author. We conducted this analysis on two corpora of texts: one with the full prompts generated from Promptiverse and the other only with a series of triplets from the prompts of the first corpus. We included the second corpus to investigate knowledge-wise diversity without considering linguistic features of prompts. For this analysis, one of the authors manually extracted knowledge entities and relations from hand-designed prompts.

*Result: Promptiverse creates more diverse prompts than hand-designing.* From the area analysis (Figure 13), we could learn that area covered by *Hand* is smaller than *No-HAI* and *HAI* in all videos and embedding approaches. It would be partly due to a small number of prompts from *Hand*, but also because prompts from *Hand*

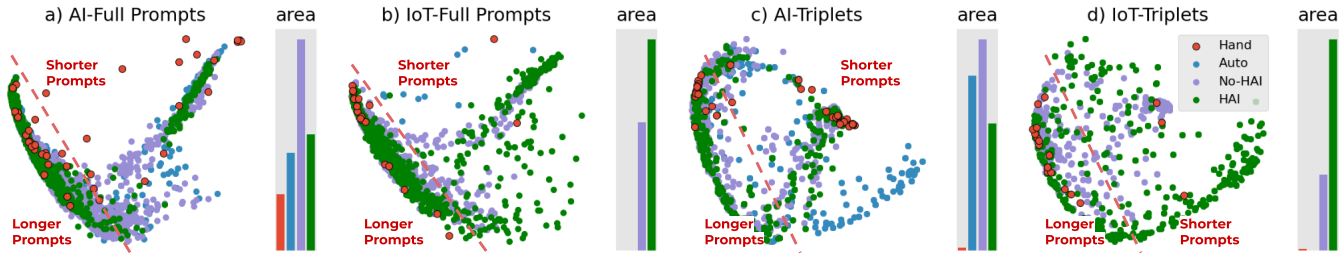



**Figure 13: Visualization of prompt embeddings and their covered area for each video (AI and IoT) and embedding approach (Full Prompts and Triplets). Prompts were embedded with BERT, dimension-reduced with PCA algorithm, and then visualized on 2D planes as scatter plots. Embedding was done either on the raw texts of prompts (Full Prompts) or only on the triplets of knowledge used in prompts (Triplets). The area was calculated for each condition, by taking convex hull area of all clusters of K-means algorithm (K=2) when the dimension is reduced to five. *HAI* and *No-HAI* has higher diversity compared to *Hand*, when compared in covered area.**


**Figure 14: Results of NASA-TLX survey. Error bar indicates the standard deviation. The red connecting line indicates that the difference between the two conditions is significant. In temporal demand, *Hand* participants reported higher load than those in *No-HAI* and *HAI*.**

could not cover the wide area, which can be notably seen in Figure 13b-d. In the AI video, *Auto* showed the area comparable to *No-HAI* or *HAI* based on the prompt embedding approach. However, in the IoT video, *Auto* had a small area, even similar to that of *Hand*. Comparing *No-HAI* and *HAI*, in the AI video, *No-HAI* occupied a larger space, but the trend was opposite in IoT.

From the qualitative analysis of visualization and prompts, we found various patterns of how these prompts are distributed in semantic space. At a high level, prompts divide into long and short prompts in all videos and all embedding approaches. When not considering linguistic features in prompts (hence, in Triplets), from the AI video (Figure 13c), we could observe that *Auto* prompts are separated from *No-HAI* and *HAI* in shorter prompts, which was due to inaccurate prompts generated from machine errors in *Auto* (e.g., "artificial intelligence" is a type of "AI"). In IoT video (Figure 13d), due to the low number of *Auto* prompts, this pattern was not observed. However, in IoT, when did not consider linguistic features, short prompts for *HAI* and *No-HAI* were separated, each to the bottom right and center of the visualization, respectively. In these, similar knowledge elements are annotated in different ways between *No-HAI* and *HAI* (e.g., 'RFID' has an attribute of 'have a processor inside there' in *No-HAI* vs. 'RFID tag' uses 'processor' in *HAI*). In the AI video, some of the full prompts from *Hand* (Figure 13a) were occupied in a space that no other conditions reside, meaning that either knowledge or linguistic features of prompts *Hand* are far different from those of other conditions. When we looked into triplet visualization (Figure 13c), most hand-designed prompts resided close to prompts from other conditions, which indicates that diversity of *Hand* in AI-Full Prompt was due to linguistic features.

*6.2.4 Cognitive load Analysis.* To analyze how prompt creator's cognitive load differs between conditions, we conducted a statistical test on NASA-TLX survey results. We excluded the question on physical demand as our tool is less about exerting physical tasks. As questions are asked on an ordinal scale, for each asked NASA-TLX question, we conducted a non-parametric Kruskal-Wallis test on three conditions that involved experts (*Hand*, *No-HAI*, *HAI*). For the posthoc test, we conducted Dunn's test.

*Result: Experts felt less temporal demand when using Promptiverse than hand-designing prompts.* The survey result is presented in Figure 14. From the Kruskal-Wallis test, we found that a significant difference was only found in temporal demand ($H = 8.05, p < 0.02$). For the rest, significance was not found ($p > 0.05$). In pairwise comparisons for temporal demand, significance was found between *Hand* and the other conditions ($p < 0.01$).

*6.2.5 Qualitative Analysis.* We qualitatively analyzed video recordings on participants' tool usage and interview data. One of the

authors did iterative coding with inductive analysis, and the other authors reviewed the coding result.

*Result: With AI recommendations and samples of possible prompts, participants self-reflect on their annotations to create high-quality prompts.* Participants said that AI recommendation helped them initially focus on a smaller number of candidates when classifying relation classes, from eleven (the number of relation classes) to three (the number of recommended classes). After they learned what relations mean, their experiences on relation class recommendation depend on the inclusion of participants' initial decision in the recommendation and their confidence in their initial decision. When what participants initially considered is included in the recommendation, they would follow it without hesitation. However, when their initial option is not in the recommendation, their behaviors would differ with their confidence. If they are confident, they would not follow the recommendation without hesitation. However, when they were less confident, they self-reflected on their decisions, went back to the guideline, and checked confusing relations multiple times. P6 said "It takes more time and I should think more when recommendations are different from mine, but it could give a chance to reconsider my choice and remind me of examples and definitions in the guideline."

Participants mentioned that entity recommendations capture important entities they missed. Entity recommendations also reduced the load in deciding and specifying the span of entities, as participants could simply click the recommended entities. Sample prompts from participant's annotations made them check if they annotated entities and relations correctly, and if prompts to be generated would be coherent and accurate.

*Result: Participants manually create prompts similarly to how Promptiverse generates prompts.* To create prompts, many *Hand* participants chose which entity to elicit first and specified relations between the elicited entity and the answer of the previous prompt. This is similar to how Promptiverse generates a dyad of prompts as described in Section 4.2.2. They also made multiple turns when all subordinate entities have the same relation to the superordinate entity as Promptiverse used correlative learning pattern. However, they had struggles that resonate with the formative study results (Section 3.1): 1) choosing which entity to ask, 2) creating an initial prompt that has the potential to bring more turns, and 3) building a prompt that can be related to answer of previous prompt. This load might have resulted in low efficiency, low diversity, and high temporal demand compared to approaches in Promptiverse.

*6.2.6 Summary.* Putting all analysis results together, we answer our RQs.

For RQ1, *HAI* generates much more prompts than *Hand* while showing similar prompt quality except for contingency management. Moreover, experts felt less temporal demand in *HAI* than in *Hand*. *HAI* also generates more diverse prompts including many different entities, relations, and their combinations compared to *Hand*. Therefore, we conclude that the combination of Promptiverse and Grannotate enables the efficient creation of diverse prompts that are comparable in quality to hand-designed ones.

For RQ2, comparing *HAI* against *Auto* and *No-HAI*, there is no significant difference in the number of prompts generated. However, in terms of quality, when compared to *Hand*, only the quality of *HAI* prompts is not significantly lower than the quality of *Hand* prompts. Moreover, prompts from *Auto* showed lower quality in direction maintenance, reduction of the degree of freedom, and accuracy of knowledge compared to *HAI*. The level of diversity between these conditions depends on the video. While no significant difference was found in cognitive load, *HAI* had a trend of having higher mental demand than *No-HAI*, which might be because AI recommendations led participants to self-reflect. However, as reported in quality analysis (Section 6.2.2), self-reflection might have increased the quality of prompts. Overall, *HAI* guided participants to create higher quality prompts with on-par efficiency compared to *Auto* and *No-HAI*.

# 7 DISCUSSION

We discuss human-AI interactions in annotation process, how our approach would enrich prompting systems, the role of knowledge graphs in prompt generation, the generalizability of our approach in other learning domains, and limitations.

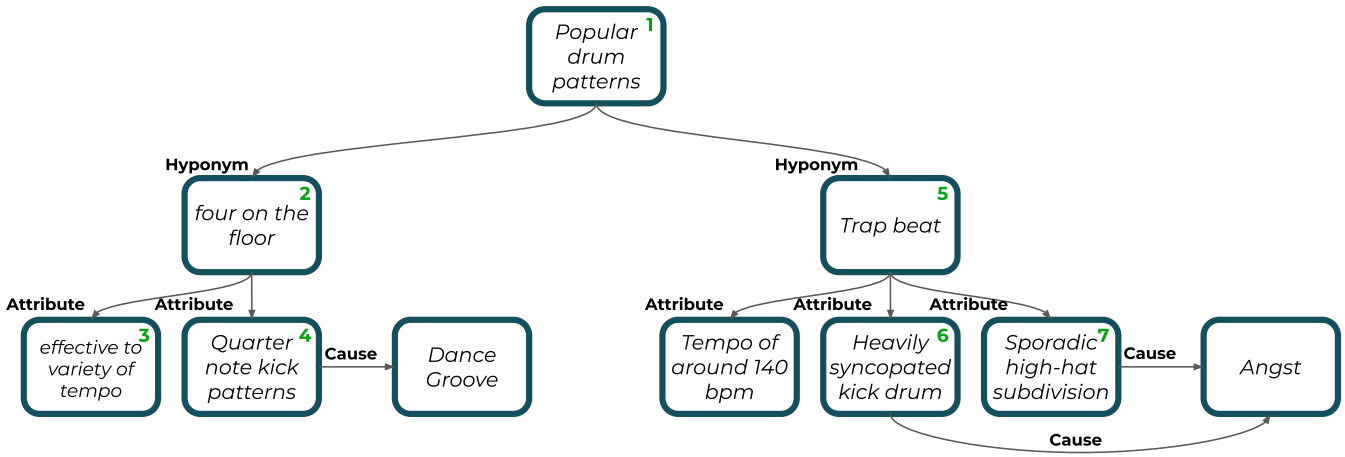
## 7.1 Human-AI Interaction as Learning Process

In our study, experts constructed knowledge graphs by annotating lecture transcripts both manually and with AI-support from Grannotate. In relation class annotation, the experts' behavior was different in these two conditions—without AI, they tended to choose between relation classes without much hesitation or learning, but with AI, the experts appeared to learn through the process. Disagreements with the AI recommendations made them cast doubt on recommended classes, re-check the guidelines, and self-reflect, which helped them gain a deeper understanding of the classes and more certainty about their choices.

Interestingly, this human-AI process parallels how scaffolding prompts help online learners: the experts are learners of a knowledge graph annotation task and the AI recommendations act as scaffolds that help them learn about classes that they find challenging. Based on these observations, one future work direction can be a workflow for an instructor-assistant-AI collaboration that helps assistants follow the instructor's annotation method for prompt generation. In this workflow, the instructor would provide initial annotations to train the AI model and then the assistants would receive HAI support that recommends annotations similar to the instructor's. Then, similar to our study findings, we hypothesize that the machine recommendations would scaffold assistants' learning so that their knowledge graphs would be aligned with the instructor's mental model.

## 7.2 Promptiverse-driven Prompting Applications

We showed Promptiverse with Grannotate generates a greater number and diversity of prompts. Now we suggest how our approach could be used to provide different prompts to learners with varying prior knowledge to alleviate the problem of the one-size-fits-all design of lecture videos. First, as Promptiverse can generate prompts with a different number of turns (e.g., subordinate and correlative




**Combinatorial learning pattern.**
T: **2** *Four on the floor* is a **type** of **1** *popular drum patterns.*
T: **5** *Trap beat* is also a **type** of **1** *popular drum patterns.*
T: Comparing to **5** *trap beat*, what **attribute** does **2** *four on the floor* has?
S: **3** *Effective to variety of tempo.* / **4** *Quarter note kick patterns.*

**Superordinate learning pattern.**
T: **5** ____ has an **attribute** of **6** *heavily syncopated kick drum.*
S: (Learner cannot answer)
T: **5** ____ has an **attribute** of **7** *sporadic high-hat subdivision.*
The answer is the same.
S: **5** *Trap beat.*

**Figure 15: Knowledge graph created from a music lecture video. With Promptiverse, in the music domain also, hierarchical knowledge could be structured in a graph and diverse scaffolding prompts could also be generated out of it.**

prompts), these can be used to gradually provide hints based on how learners answer to the prompts. For example, if a learner could not answer a prompt that asks about the characteristics of "machine learning", the prompting system can provide more questions or hints to facilitate the learner's understanding. Second, with Promptiverse's diverse prompts and a component that recognizes learners' levels of understanding [14, 43], Promptiverse can adaptively provide scaffolding prompts to elicit each learner's understanding. For example, if the system knows that a learner is struggling with a concept, it can give an explanation about the concept instead of asking questions. Moreover, by considering whether the concepts in prompts are in a higher hierarchy or in a lower hierarchy in the knowledge graph, the learning system can give prompts about the details of the lecture to learners who already have a good understanding, and prompts about higher-level concepts to those who are struggling. Third, with a comprehensive set of prompts from Promptiverse, the learning system can provide prompts when a learner knows that they are struggling and requests scaffolding. This would minimize the chance of learner misunderstanding propagating to other lecture content. Fourth, knowledge graphs of multiple lectures can be merged together with the small additional effort of maintaining entity consistency across lectures, and these merged graphs can give further support to learners. For example, cross-lecture prompts can allow learners to more comprehensively relate concepts over the course.

## 7.3 Role of Knowledge Graphs in Prompt Creation

Promptiverse's way of creating prompts appeared to simulate the strategies used by lecture designers when hand-designing prompts. For example, they considered the relations between knowledge entities to create a dyad of prompts or correlative prompts in a similar way Promptiverse does. However, hand-designing is limited as lecture designers struggled in deciding which entity to proceed to when designing more turns and giving more hints about target entities. On the other hand, Promptiverse allows lecture designers to only focus on structuring knowledge by handling the task of turning them into prompts with meaningful learning patterns. With traversability, many different concepts and relations can be covered in diverse ways to create a spectrum of pedagogically effective prompts. However, Promptiverse currently uses knowledge graphs in a limited way, as it only considers triplets to create a round of prompts. Hence, more complex prompts, like those that involve more than two entities or ask about 'why' and 'how', would be difficult to generate with Promptiverse. To use knowledge graphs more richly, the Promptiverse can be combined with other approaches to allow more flexible traversals that are not limited to triplets. For example, recent approaches that combine knowledge graphs with pre-trained language models [21, 58, 60] learn how to turn complex knowledge structures into natural language, and these can potentially be used to generate prompts about complex knowledge incorporated in knowledge graphs. However, future work must first validate the effectiveness and accuracy of these methods in knowledge-guided tasks.

## 7.4 Generalizing to Other Learning Domains

We conducted our study using two videos from different sub-domains in computer science. While our study showed that experts who used Promptiverse and Grannotate generated significantly more prompts than when they did so manually, the two chosen videos might not represent the broad spectrum of topics that can be covered in online lecture videos.

Still, we believe Promptiverse can be generally used for domains other than the ones we evaluated it for. For demonstration, one of the authors annotated a lecture video[1] on music. The lecture video explains popular drum patterns, from what they are to what characteristics and effects they have. Figure 15 shows the resulting knowledge graph which has a rich hierarchical knowledge structure based on the video. This example graph can create diverse pedagogically meaningful prompts. For example, by adopting a combinatorial learning process, the learning system can first inform the learner about the hyponyms of popular drum patterns, and then ask about what are the characterizing attributes of a drum pattern compared to those of other drum patterns (left in Figure 15). Figure 15 also shows how a superordinate learning could be applied on this graph. Generalization into the music domain also reveals one limitation of Promptiverse: it only considers linguistic content when prompting, not other media. Employing such multimedia content in prompts can be interesting future work.

To generalize Promptiverse's prompt-generating capabilities, however, Grannotate should also be generalizable. While our version of Grannotate is tuned for STEM fields, we can expand it to other domains by changing the underlying models. For entity and relation existence detection, we can expand it to other domains by using similar pretrained models that have been trained on datasets from a variety of domains, including GENIA (biomedicine), ChemProt (chemistry), WLPC (biology), MECHANIC (general science), ACE05 (newswire, broadcast news, broadcast conversation, weblog, and discussion forums) [23, 57]. While these datasets cover many different domains and text styles, some domains, such as web programming, are not covered with these datasets. However, for domains which these datasets cannot cover, a potential future direction could be to gather a small amount of data and use the P-tuning approach to detect entities and relations in such domains. As our work showed, the same approach can also be used to expand the edge class classification component to other domains with relatively little effort for data collection.

While the generalizability of our approach seems promising, it is still unclear whether our study findings would also hold for other domains. Future work can conduct studies with lecture videos from various domains to confirm our findings or to identify which domains can or cannot be covered with our approach.

### 7.5 Limitations

Our work has a couple of limitations which we address in this subsection.

The experience and frequency of using prompting strategies in experts' everyday teaching could affect their speed of creating prompts as well as the prompt quality. While we provided participants' information of teaching experience, it cannot explicitly show how much they used prompting strategies while teaching.

It is hard to use our approach for generating prompts with complex knowledge that consider more than two entities (hence, more than one relation). Injecting knowledge graphs into pre-trained language models [21, 58, 60] can present an opportunity to explore how to handle such complex knowledge.

[1]https://www.youtube.com/watch?v=c7ffMObdxro

## 8 CONCLUSION

This paper introduces Promptiverse, a framework that generates diverse scaffolding prompts by traversing knowledge graphs on lecture content in pedagogically meaningful patterns. To facilitate the usage of Promptiverse, we support lecturer designers' annotation processes with Grannotate, which provides AI recommendations and samples of possible prompts based on the user's annotations. In our evaluation, participants who used Promptiverse with Grannotate produced 40 times more prompts than those who hand-designed, with on-par quality and higher diversity in prompts. When compared to other graph construction methods with either full automation or full manual effort, only graphs made with Grannotate generated prompts that had comparable quality to the hand-designed prompts. We hope our work can open up more opportunities in supporting a spectrum of learners by creating diverse learning strategies.

## ACKNOWLEDGMENTS

This work was supported by the KAIST-NAVER Hypercreative AI Center, Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government (MSIT) (No.2020-0-02237, Personalized Progress Analysis and Exercise Recommendation for Remote Language Learning Using AI and Big Data), the DGIST Start-up Fund Program of the Ministry of Science and ICT(2021070007), and the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (NRF-2020R1I1A1A01072385).

## REFERENCES

[1] Shaaron Ainsworth. 2006. DeFT: A conceptual framework for considering learning with multiple representations. *Learning and Instruction* 16, 3 (2006), 183–198. https://doi.org/10.1016/j.learninstruc.2006.03.001

[2] Sören Auer, Viktor Kovtun, Manuel Prinz, Anna Kasprzik, Markus Stocker, and Maria Esther Vidal. 2018. Towards a Knowledge Graph for Science. In *Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics* (Novi Sad, Serbia) *(WIMS '18)*. Association for Computing Machinery, New York, NY, USA, Article 1, 6 pages. https://doi.org/10.1145/3227609.3227689

[3] D. Ausubel. 2000. The Acquisition and Retention of Knowledge: A Cognitive View.

[4] David G. Ausubel. 1963. Cognitive Structure and the Facilitation of Meaningful Verbal Learning1. *Journal of Teacher Education* 14, 2 (1963), 217–222. https://doi.org/10.1177/002248716301400220 arXiv:https://doi.org/10.1177/002248716301400220

[5] David P. Ausubel. 1962. A Subsumption Theory of Meaningful Verbal Learning and Retention. *The Journal of General Psychology* 66, 2 (1962), 213–224. https://doi.org/10.1080/00221309.1962.9711837 arXiv:https://doi.org/10.1080/00221309.1962.9711837 PMID: 13863333.

[6] David Paul Ausubel. 2012. *The acquisition and retention of knowledge: A cognitive view*. Springer Science & Business Media.

[7] Ashutosh Baheti, Alan Ritter, and Kevin Small. 2020. Fluent Response Generation for Conversational Question Answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 191–207. https://doi.org/10.18653/v1/2020.acl-main.19

[8] Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. *GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow*. http://github.com/eleutherai/gpt-neo

[9] William Cai, Hao Sheng, and Sharad Goel. 2020. MathBot: A Personalized Conversational Agent for Learning Math.

[10] Joel Chan, Pao Siangliulue, Denisa Qori McDonald, Ruixue Liu, Reza Moradinezhad, Safa Aman, Erin T. Solovey, Krzysztof Z. Gajos, and Steven P. Dow. 2017. Semantically Far Inspirations Considered Harmful? Accounting for Cognitive States in Collaborative Ideation. In *Proceedings of the 2017 ACM SIGCHI Conference on Creativity and Cognition* (Singapore, Singapore) *(C&C '17)*. Association for Computing Machinery, New York, NY, USA, 93–105. https://doi.org/10.1145/3059454.3059455

[11] Michelene T.H. Chi, Nicholas De Leeuw, Mei-Hung Chiu, and Christian Lavancher. 1994. Eliciting self-explanations improves understanding. *Cognitive Science* 18, 3 (1994), 439–477. https://doi.org/10.1016/0364-0213(94)90016-7

[12] Michelene T.H Chi, Stephanie A Siler, Heisawn Jeong, Takashi Yamauchi, and Robert G Hausmann. 2001. Learning from human tutoring. *Cognitive Science* 25, 4 (2001), 471–533. https://doi.org/10.1016/S0364-0213(01)00044-1

[13] Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question Answering in Context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 2174–2184. https://doi.org/10.18653/v1/D18-1241

[14] Albert T. Corbett and John R. Anderson. 1995. Knowledge Tracing: Modelling the Acquisition of Procedural Knowledge. *User Model. User-Adapt. Interact.* 4, 4 (1995), 253–278. http://dblp.uni-trier.de/db/journals/umuai/umuai4.html#CorbettA95

[15] Stephen Cummins, Alastair R. Beresford, and Andrew Rice. 2016. Investigating Engagement with In-Video Quiz Questions in a Programming Course. *IEEE Transactions on Learning Technologies* 9, 1 (2016), 57–66. https://doi.org/10.1109/TLT.2015.2444374

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[17] Oscar Ferrández, Brett R South, Shuying Shen, F Jeffrey Friedlin, Matthew H Samore, and Stéphane M Meystre. 2013. BoB, a best-of-breed automated text de-identification system for VHA clinical documents. *Journal of the American Medical Informatics Association* 20, 1 (2013), 77–83.

[18] Kata Gábor, Davide Buscaldi, Anne-Kathrin Schumann, Behrang QasemiZadeh, Haïfa Zargayouna, and Thierry Charnois. 2018. SemEval-2018 Task 7: Semantic Relation Extraction and Classification in Scientific Papers. In *Proceedings of International Workshop on Semantic Evaluation (SemEval-2018)*. New Orleans, LA, USA.

[19] Vicki S Gier and David S Kreiner. 2009. Incorporating active learning with PowerPoint-based lectures using content-based questions. *Teaching of Psychology* 36, 2 (2009), 134–139.

[20] Roxana Girju, Preslav Nakov, Vivi Nastase, Stan Szpakowicz, Peter Turney, and Deniz Yuret. 2007. SemEval-2007 Task 04: Classification of Semantic Relations between Nominals. In *Proceedings of the 4th International Workshop on Semantic Evaluations* (Prague, Czech Republic) *(SemEval '07)*. Association for Computational Linguistics, USA, 13–18.

[21] Bin He, Di Zhou, Jinghui Xiao, Xin Jiang, Qun Liu, Nicholas Jing Yuan, and Tong Xu. 2020. BERT-MK: Integrating Graph Contextualized Knowledge into Pre-trained Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 2281–2290. https://doi.org/10.18653/v1/2020.findings-emnlp.207

[22] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, Uppsala, Sweden, 33–38. https://aclanthology.org/S10-1006

[23] Tom Hope, Aida Amini, David Wadden, Madeleine van Zuylen, Sravanthi Parasa, Eric Horvitz, Daniel Weld, Roy Schwartz, and Hannaneh Hajishirzi. 2021. Extracting a Knowledge Base of Mechanisms from COVID-19 Papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 4489–4503. https://doi.org/10.18653/v1/2021.naacl-main.355

[24] Gregory Hume, Joel Michael, Allen Rovick, and Martha Evens. 1996. Hinting as a Tactic in One-on-One Tutoring. *Journal of the Learning Sciences* 5, 1 (1996), 23–47. https://doi.org/10.1207/s15327809jls0501_2

[25] Jeffrey D. Karpicke and Phillip J. Grimaldi. 2012. Retrieval-Based Learning: A Perspective for Enhancing Meaningful Learning. *Educational Psychology Review* 24, 3 (2012), 401–418. http://www.jstor.org/stable/43546799

[26] Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The inception platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*. 5–9.

[27] Jan-Christoph Klie, Richard Eckart de Castilho, and Iryna Gurevych. 2020. From zero to hero: Human-in-the-loop entity linking in low resource domains. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 6982–6993.

[28] Geza Kovacs. 2016. Effects of In-Video Quizzes on MOOC Lecture Viewing. In *Proceedings of the Third (2016) ACM Conference on Learning @ Scale* (Edinburgh, Scotland, UK) *(L@S '16)*. Association for Computing Machinery, New York, NY, USA, 31–40. https://doi.org/10.1145/2876034.2876041

[29] Timothy J Lawson, James H Bodle, Melissa A Houlette, and Richard R Haubner. 2006. Guiding questions enhance student learning from educational videos. *Teaching of Psychology* 33, 1 (2006), 31–33.

[30] Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. arXiv:2104.08691 [cs.CL]

[31] Xiang Lisa Li and Percy Liang. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. arXiv:2101.00190 [cs.CL]

[32] Ching Liu, Juho Kim, and Hao-Chuan Wang. 2018. *ConceptScape: Collaborative Concept Mapping for Video Learning*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3173574.3173961

[33] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. GPT Understands, Too. arXiv:2103.10385 [cs.CL]

[34] Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 3219–3232. https://doi.org/10.18653/v1/D18-1360

[35] Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-Task Identification of Entities, Relations, and Coreferencefor Scientific Knowledge Graph Construction. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*.

[36] William C. Mann and Sandra A. Thompson. 1987. *Rhetorical Structure Theory: A Theory of Text Organization*. Technical Report ISI/RS-87-190. Information Sciences Institute.

[37] Roxana Moreno and Richard Mayer. 2007. Interactive Multimodal Learning Environments. *Educ Psychol Rev* 19 (09 2007), 309–326. https://doi.org/10.1007/s10648-007-9047-2

[38] J. Novak. 2002. Meaningful learning: The essential factor for conceptual change in limited or inappropriate propositional hierarchies leading to empowerment of learners.

[39] Joseph D. Novak and Alberto J. Cañas. 2006. *The Theory Underlying Concept Maps and How to Construct and Use Them*. research report 2006-01 Rev 2008-01. Florida Institute for Human and Machine Cognition. http://cmap.ihmc.us/Publications/ResearchPapers/TheoryCmaps/TheoryUnderlyingConceptMaps.htm

[40] Angela M. O'Donnell, Donald F. Dansereau, and Richard H. Hall. 2002. Knowledge Maps as Scaffolds for Cognitive Processing. *Educational Psychology Review* 14, 1 (01 Mar 2002), 71–86. https://doi.org/10.1023/A:1013132527007

[41] Judith S Olson and Wendy A Kellogg. 2014. *Ways of Knowing in HCI*. Vol. 2. Springer.

[42] Boyuan Pan, Hao Li, Ziyu Yao, Deng Cai, and Huan Sun. 2019. Reinforced Dynamic Reasoning for Conversational Question Generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 2114–2124. https://doi.org/10.18653/v1/P19-1203

[43] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. 2015. Deep Knowledge Tracing. In *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (Eds.), Vol. 28. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2015/file/bac9162b47c56fc8a4d2a519803d51b3-Paper.pdf

[44] Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A Conversational Question Answering Challenge. *Transactions of the Association for Computational Linguistics* 7 (March 2019), 249–266. https://doi.org/10.1162/tacl_a_00266

[45] Samuel Rhys Cox, Yunlong Wang, Ashraf Abdul, Christian von der Weth, and Brian Y. Lim. 2021. Directed Diversity: Leveraging Language Embedding Distances for Collective Creativity in Crowd Ideation. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 393, 35 pages. https://doi.org/10.1145/3411764.3445782

[46] D. Robinson and Kenneth A. Kiewra. 1995. Visual argument: Graphic organizers are superior to outlines in improving learning from text. *Journal of Educational Psychology* 87 (1995), 455–467.

[47] Peter J. Rousseeuw. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* 20 (1987), 53–65. https://doi.org/10.1016/0377-0427(87)90125-7

[48] Sherry Ruan, Liwei Jiang, Justin Xu, Bryce Joe-Kun Tham, Zhengneng Qiu, Yeshuang Zhu, Elizabeth L. Murnane, Emma Brunskill, and James A. Landay. 2019. *QuizBot: A Dialogue-Based Adaptive Learning System for Factual Knowledge*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3290605.3300587

[49] Hyungyu Shin, Eun-Young Ko, Joseph Jay Williams, and Juho Kim. 2018. *Understanding the Effect of In-Video Prompting on Learners and Instructors*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3173574.3173893

[50] Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational

Linguistics, Online, 4222–4235. https://doi.org/10.18653/v1/2020.emnlp-main.346
[51] Brett South, Shuying Shen, Jianwei Leng, Tyler Forbush, Scott DuVall, and Wendy Chapman. 2012. A prototype tool set to support machine-assisted annotation. In *BioNLP: Proceedings of the 2012 Workshop on Biomedical Natural Language Processing*. 130–139.
[52] Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. BRAT: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*. 102–107.
[53] Hariharan Subramonyam, Colleen Seifert, Priti Shah, and Eytan Adar. 2020. *TexSketch: Active Diagramming through Pen-and-Ink Annotations*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376155
[54] Hrishikesh Terdalkar and Arnab Bhattacharya. 2021. Sangrahaka: A Tool for Annotating and Querying Knowledge Graphs. In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (Athens, Greece) *(ESEC/FSE 2021)*. Association for Computing Machinery, New York, NY, USA, 1520–1524. https://doi.org/10.1145/3468264.3473113
[55] Janneke van de Pol, Neil Mercer, and Monique Volman. 2019. Scaffolding Student Understanding in Small-Group Work: Students' Uptake of Teacher Support in Subsequent Small-Group Interaction. *Journal of the Learning Sciences* 28, 2 (2019), 206–239. https://doi.org/10.1080/10508406.2018.1522258 arXiv:https://doi.org/10.1080/10508406.2018.1522258
[56] Omer Faruk Vural. 2013. The Impact of a Question-Embedded Video-Based Learning Tool on E-Learning. *Educational Sciences: Theory and Practice* 13, 2 (2013), 1315–1323.
[57] David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, Relation, and Event Extraction with Contextualized Span Representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, 5783–5788. https://doi.org/10.18653/v1/D19-1585
[58] Chenguang Wang, Xiao Liu, and Dawn Song. 2020. Language Models are Open Knowledge Graphs. *CoRR* abs/2010.11967 (2020). arXiv:2010.11967 https://arxiv.org/abs/2010.11967
[59] Rainer Winkler, Sebastian Hobert, Antti Salovaara, Matthias Söllner, and Jan Marco Leimeister. 2020. *Sara, the Lecturer: Improving Learning in Online Education with a Scaffolding-Based Conversational Agent*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376781
[60] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced Language Representation with Informative Entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 1441–1451. https://doi.org/10.18653/v1/P19-1139

# A P-TUNING METHOD

To recommend relation classes, we trained soft prompts that can guide GPT-Neo model with 2.7 billion parameters [8], which is an open-source GPT-based language model. Instead of manually designing natural language prompts to guide this type of models, P-tuning approach [30, 31, 33, 50] automatically searches for high-performing prompt token embeddings in a continuous space. To employ this technique, we composed prompts as $[P_{0:1}, C, P_{2:5}, E_1, P_6, E_2, R]$ where $P_i$ is the $i$th prompt token, $C$ is the context sentences that include both knowledge entities, $E_1$ and $E_2$ are the knowledge entities, and $R$ is the edge class. When $E_1$ and $E_2$ are in different sentences, we concatenate their sentences to form $C$. P-tuning regards $P_i$ as pseudo tokens and the embeddings of these tokens are trained by backpropagating the CrossEntropy loss from the GPT-Neo model.

On our training dataset (which is 88 data samples), We used the Adam optimizer, the learning rate of 0.001, weight decay of 0.0005, and batch size of 4. We also used an early stopping technique to prevent overfitting on the training set. Our model's prediction result on the test was 64%, which is (the number of true relation class being included in our top-3 predictions)/(the number of samples in the test set)×100.

# B CRITRIA OF RUBRIC FOR QUALITY EVALUATION IN 6.1.3

To score questions we provided a scoring rubric, which is based on the framework for analyzing scaffolding strategies [55] and our goal of creating accurate prompts. The rubric had six criteria as follows:

- Direction maintenance: The lecturer's prompts keep the learning on target and maintain the learner's pursuit of a particular objective.
- Cognitive structuring: The lecturer's prompts provide explanatory structures that organize and justify lecture content.
- Reduction of degrees of freedom: The lecturer's prompts take over parts of a task that the student is not yet able to perform and thereby simplify the task for the student.
- Recruitment: The lecturer's prompts get students interested in the lecture content and help them adhere to it.
- Contingency management and frustration control: The lecturer's prompts concern the facilitation of student performance via a system of rewards and punishments as well as keeping students motivated via the prevention or minimalization of frustration.
- Accuracy of knowledge: The lecturer's prompts accurately reflect knowledge conveyed in the video.