Automatic Hint Generation

Adam Jatowt
adam.jatowt@uibk.ac.at
Department of Computer Science
University of Innsbruck
Austria

Calvin Gehrer
calvin.gehrer@student.uibk.ac.at
Department of Computer Science
University of Innsbruck
Austria

Michael Färber
michael.faerber@kit.edu
Karlsruhe Institute of Technology
Karlsruhe
Germany

ABSTRACT

At times when answers to user questions are readily and easily available (at essentially zero cost), it is important for humans to maintain their knowledge and strong reasoning capabilities. We believe that in many cases providing hints rather than final answers should be sufficient and beneficial for users as it requires thinking and stimulates learning as well as remembering processes. We propose in this paper a novel task of automatic hint generation that supports users in finding the correct answers to their questions without the need of looking the answers up. As the first attempt towards this new task, we design and implement an approach that uses Wikipedia to automatically provide hints for any input question-answer pair. We then evaluate our approach with a user group of 10 persons and demonstrate that the generated hints help users successfully answer more questions than when provided with baseline hints.

CCS CONCEPTS

• **Information systems** → **Question answering**; *Information extraction*; **Specialized information retrieval**.

KEYWORDS

Hint Generation, Question Answering, Question Generation

ACM Reference Format:
Adam Jatowt, Calvin Gehrer, and Michael Färber. 2023. Automatic Hint Generation. In *Proceedings of the 2023 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR '23), July 23, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3578337.3605119

1 INTRODUCTION

Automatic question answering (QA) and question generation (QG) have been recently studied quite extensively in the NLP and IR communities leading to excellent outcomes and ground-breaking systems [6, 8, 9, 17, 20, 21]. In this context, the development of large language models in particular allowed easy access to the accumulated human knowledge on unprecedented scale. However, automatically providing hints to help humans successfully answer questions has not been researched yet, despite hints being a common vehicle for humans to infer correct answers. Hinting does not only raise a chance that the answer will be found but is also an effective way of teaching as it requires a person to start thinking (often through an interplay of complex processes of association, comparison, abstraction, etc.) to come up with a correct answer or at least to narrow down the scope of potential candidates. For this reason, questions in pupil and student exams or homework assignments are sometimes complemented with supportive hints. Indeed, merely presenting the right answer after a wrong one (or when no answer) was given, usually offers weaker learning effect than when a user is aided with a hint to find the answer by herself. We believe that with the current trend of increasing reliance of users on ready answers given by chatbots and question answering systems, it will be important in the future to make sure that humans still keep learning and maintain their critical thinking as well as good reasoning and remembering skills. Additionally, given the well-known problem of the reliability of answers provided by the current large language models like ChatGPT [2, 19], solutions that involve humans in answer verification should be quite useful.

Furthermore, in psychology and cognitive science it is known that self-efficacy, or positive perceptions of one's competence, impacts students' motivation [1, 16]. Letting users come up with the correct answers by themselves should then also contribute to the positive psychological effect, potentially increasing the users' self-confidence and their motivation for learning. AI systems that tend to always know all the answers better than their users, could actually demotivate the users to learn and reason, which could actually harm the users' life-long learning process.

Finally, besides a clear educational benefit, providing a series of hints to difficult questions can have applications in entertainment such as supporting complementing various quizzes (e.g., Jeopardy) which tend to be liked by many.

The objective of our work is to formulate and define a novel task of *automatic hint generation*. Our second contribution is designing, implementing and testing a proof-of-concept approach[1] which takes a <question, answer> pair as an input, and which generates effective hints that could be used for asking users to guess the answer to the question. For example, for the question: "*What country won the very first FIFA World Cup in 1930?*" our approach produces the following hints:

- "*The searched location is on continent South America*"
- "*The searched location shares border with Brazil*"
- "*Spoken language in searched location is Spanish*"
- "*The capital of searched country is Montevideo*"
- "*Head of state of searched country is Luis Lacalle Pou*"

ICTIR '23, July 23, 2023, Taipei, Taiwan

ACM ISBN 979-8-4007-0073-6/23/07.
https://doi.org/10.1145/3578337.3605119

[1] The code and experimental data are available at https://github.com/calvingehrer/automaticHintGeneration, while the extended and further improved version of the original code can be found at https://github.com/AlexWalcher/automaticHintGeneration.

Based on one or at least a few of such hints a user should be able to come up with the correct answer ("*Uruguay*" in this case). In our approach, we utilize Wikipedia as a convenient source of reasonably high-quality knowledge about a large number of concepts and entities that should be useful for generating helpful hints. We make sure that generated hints are relevant to user questions and easy enough so that users are able to benefit from the provided clues. In the experiments we ask a group of users to answers questions with unknown to them answers on variety of topics and we show that our pilot approach is indeed useful. In particular, we focus on questions about persons, locations and dates.

To sum up, we make the following contributions in this paper:

- We define and propose a novel research task of automatic hint generation for natural language questions.
- We design and implement a solution for this task using Wikipedia and Wikidata focusing on three common question types.
- We conduct user experiments to determine the usefulness of generated hints, and we outline several promising directions for further research.

The remainder of this paper is structured as follows. In the next section, we describe the related work. Section 3 provides the formal definition of the hint generation task. We outline our approach in Section 4, while in Section 5 we describe the experimental settings and present the evaluation of generated hints. Section 6 briefly discusses the limitations of our work. We conclude the paper in the last section where we also outline our future plans.

2 RELATED WORK

While the fields of automatic question answering (QA) [6, 21] and generation (QG) [8, 9, 20] have advanced quite much in the last years, the research related to hints is scarce and largely limited to analyzing the impact and helpfulness of hints in the area of assisting with writing code [10], or to augmenting computer aided instructional tools such as logic tutors [3, 15]. Price et al. [12] studied what constitutes a good hint in the context of learning to program when using intelligent tutoring systems. Such systems usually attempt to identify a desirable coding path through a space of previously observed code states. The authors focused especially on next-step hints for users to develop their code, finding among others that data-driven hints are poor when students write code that diverges from common solutions. Through educational experiments, Price et. al [13] pointed out that not every hint is equally useful to every student for learning programming. The authors investigated the impact of the hint quality on the help-seeking behavior of the students. They also showed that the better the first hints a student received, the more hints were later requested. So it could be said that better hints encourage the students to seek more help. If students received a hint they could not follow, they rather stopped asking for help. Earlier studies showed also that students with lower prior knowledge are more likely to ask for help [18]. The research focused on two homeworks the students needed to solve on their own. Pino and Eskénazi [11] proposed to study the relation between hints and response accuracy for measuring and customizing the amount of information provided by hints.

Klein-Braley and Raatz [7] investigated generating hints for a cloze task; hence not for standalone questions. The authors improved the cloze task through the controlled word uncovering by revealing consecutive first letters of masked words. Progressive and partial character uncovering of words' first characters is, of course, a rather limited way in which hints can be generated, and corresponds in fact to a word completion task used in cloze tests. In contrast, we propose providing natural language hints that supplement users with information other than the question content, as a more interesting and educationally useful way to generate hints.

3 AUTOMATIC HINT GENERATION TASK

We formally define the task of Hint Generation (HG) as follows:

> Given a pair of **question** q and its **correct answer** a, the task is to generate a **hint** h such that $P(a|q,h) - P(a|q) > \epsilon$, where $P(a|q,h)$ and $P(a|q)$ denote respectively the probability of a user successfully answering q after h is given, and the same probability without the hint h being provided. ϵ is a threshold parameter ($\epsilon > 0$).

We assume here that the user initially does not know the answer a to the presented question q, but she can request a hint h to be generated to help in answering the question. If the user still does not know the correct answer, the system could provide follow-up hints to let her come up with the right answer or at least "come closer" to it.

Note that in the above setup, a hint is generated without considering any contextual information. A natural extension, which we plan to focus in the future, is to consider also the previously given wrong answers by the user (if any), and perhaps the prior hints issued to the user for the same question, if there were any. This would form a richer input that could lead to generating customized hints. Naturally, hints could be also adapted based on any information about the user such as the level of possessed knowledge on the topics of the asked questions, or general user's interests, etc.[2]

While we defined the hint generation above as the problem involving a pair of a question and its correct answer, in a more general setup, either the question or answer could be assumed missing from the input. In case when no question is given, hint generation would focus on helping users to find the answer without the need to pay attention to any constraints or context that the presence of the question could provide (e.g., the requirement of not repeating information already present in the question or a need for a topical match of the generated hints with the question). In the case, when the answer is missing, QA systems would then essentially "collaborate" with the user in jointly finding the correct answer. This scenario could for example involve situations when a user forgot certain information (e.g., personal information, or information that was earlier told to the user by someone else).

Finally, the hint generation task can be regarded more broadly as the problem of providing missing knowledge to let users recall or come up with the required information, or to increase the comprehension of certain topics or information. This entails situations in which instead of a question, a document or a concept might form

[2]In this work, we focus however on the most basic setup, given the lack of prior research on the HG task.

an input, and the generated hints should then support users in the understanding or learning process.

4 APPROACH

To generate effective hints, we propose to use Wikipedia and Wikidata, since they contain broad content on many world topics and entities. Out of diverse types of information related to a given question and its answer, we attempt to select one that has the best chance to become useful for a user trying to come up with the correct answer. We focus in our method on three fundamental question types to approach HG task: "Who?", "Where?" and "When?" questions. Non-factoid question types like "How?" and "Why?" are not covered at the moment since they are more complex to answer, and their answers are typically not just a single entity but rather longer (and sometimes complex) text content. In the following subsections, we describe our approach for generating hints for the three above-listed question types.

4.1 Generating Hints for "When?" Questions

For questions that require date as an answer (e.g., a year) we generate hints in the form of major events that occurred on the same date as the answer. The assumption is that users may remember dates of key past events, and mentioning such events should help them come up with the correct date being the answer to the target question. For this, we propose to use Wikipedia's year pages[3] that provide brief accounts of major events worldwide which occurred in each particular year together with listings of famous or important persons that were born or died in that year. Each year has its dedicated article and all the year articles are structured in the same way. The first section of the year pages contains "Events" followed by the "Births" and then "Deaths" sections. In Fig. 1 we show sample events contained in the "Events" section of 1991's article[4].

We assume for simplicity that an answer is in the form of a year, that is, a user is asked to tell in which year a certain event described in the question happened. Note that the extension to serve questions requesting answers in month or day granularities is possible and easy, since, most of the time, events listed in the Wikipedia year pages contain also finer granularity temporal information, that is, the months and days of listed events.

Since multiple events are always listed for each year on Wikipedia year pages (e.g., especially many events are mentioned in the years within the last few decades), we need a way to select events that would be useful to serve as hints. We decided to follow an approach based on the popularity estimation of each event. In particular, we estimate how well-known an event is by analyzing the popularity of entities associated with this event. For example, when a well-known entity such as Barack Obama is mentioned, the event has a high probability to be known by many users. Events are extracted from the Wikipedia year pages using regular expressions, and an entity is assumed to be a string within the event description that contains a link to another Wikipedia article. We estimate the popularity of entities using the approach based on the combination of backlinks' counting and page view analysis. Backlinks are links pointing to the target article from other Wikipedia articles. We first obtain the number of backlinks for each entity mentioned in an event. For this, we utilize the Wikipedia API[5] and count the number of returned articles that point to the Wikipedia article representing the input entity. Since for each call, we can obtain information on up to 500 linking articles, for efficiency, if a target article has more backlinks than a pre-defined threshold l (we choose l = 1,000 based on the pilot study), it is deemed as one representing a popular entity. In such a case, we proceed to count its page views. Otherwise, the entity is dropped and its popularity is assumed to be zero.[6] To collect information on the access frequency to Wikimedia articles we use the Wikipedia's pageview API[7] and calculate the average page views per month over the last year. This value is finally used as a proxy for the popularity of the entity being the topic of the article.

The popularity of an event is then computed as an average popularity of its entities. We apply also filtering conditions such as dropping entities that are locations since many locations such as countries or major cities typically have a large number of backlinks and page views, thus any minor event occurring in these locations would be considered as important. Another problem that we found occurs when a popular entity is not the subject of a sentence. For example, while many heard of "Ronald Reagan", few people would know much about his mother, Nelle Wilson Reagan. An event involving this person as a main actor (such as the one listed in the Wikipedia article on 1962 that starts with "Nelle Wilson Reagan, mother of United States President Ronald Reagan…") would then receive an unfairly high score, but is not very useful. To solve this issue, we consider only entities that are subjects[8] of event-describing sentences. Another necessary filtering step was to prevent *answer leakage* by removing events whose names already contain the answer year such as "2001 Gerry Weber Open – singles", "2001 Argentina rugby union tour of New Zealand and Great Britain", "2001 Formula One season", or ones for which the year is mentioned in the event description. We have also decided to discard recurring events that are represented only by their ordinal numbers like "51st Berlin International Film Festival" as it is probably difficult for users to remember in which year a given numbered edition of a periodical event occurred.

4.1.1 Topical Relevance. We think that useful hints should be related to the topic of the question. The question relevance is important as hints from other topical categories and domains than the one of the original asked question could distract users and may require other kinds of knowledge. Since one of the primary objectives of hinting relates to education, we believe that hints should not only be effective in terms of letting users find the correct answer but should also be related to the asked questions to provide a better learning effect. Relevance is then also considered for computing the hint utility. Sorting of hints is then performed by the combination of their estimated popularity scores (PV), subject to

[3] E.g., https://en.wikipedia.org/wiki/1977
[4] https://en.wikipedia.org/wiki/1991
[5] https://www.mediawiki.org/wiki/API:Main_page
[6] The limitation of this choice is of course with the case of entities which become popular only recently without having sufficient time to "accumulate" a high number of Wikipedia articles linking to the articles corresponding to those entities. We leave this issue as a future work.
[7] https://wikitech.wikimedia.org/wiki/Analytics/AQS/Pageviews
[8] We use the *spaCy* library to parse sentences.


Events

January

- January 1 - Czechoslovakia becomes the second Eastern European country to abandon its command economy.[2]
- January 4 – The UN Security Council votes unanimously to condemn Israel's treatment of Palestinians.[citation needed]
- January 5 – Georgian troops attack Tskhinvali, the capital of South Ossetia, starting the 1991–92 South Ossetia War.[3]
- January 7 – 1991 Haitian coup d'état: An attempted coup by the Tonton Macoute, a paramilitary force under former dictator Jean-Claude Duvalier, is thwarted in Haiti.[4] On July 30, he is convicted by a jury of attempting to overthrow the country's first democratically elected government.[citation needed]

Figure 1: Example of events included in the "Events" section of a Wikipedia's article about the year 1991 (https://en.wikipedia.org/wiki/1991).

normalization, and the degree of their similarity to the target question:

$$\alpha * (PV/maxPV) + (1 - \alpha) * simScore$$

where α is set to 0.5.[9] We compute the similarity between the question and the candidate hint as a cosine similarity score between the vectors obtained by applying BERT [4] on the text content of a question and one of generated candidate hints.

4.1.2 Formulating Final Hints. We process the lists of births and deaths of famous persons in a similar way as with the events from the "Events" section (as described just above). The only difference is that we make sure that now only person entities are considered for popularity computation. To construct the final hints we use a template with a slot to be filled, e.g., "*The following event happened in the searched date: <event>*." In Table 1, we show several top generated hints.

4.2 Generating Hints for "Where?" Questions

For "Where?" type questions, we use Wikidata which offers large-scale and detailed information about locations (e.g., cities, states, and countries). Note that this was not feasible for "When?" questions since there is relatively little information on events in Wikidata in comparison to the data on entities such as locations or persons [5].

Locations present in Wikidata may have quite a large number of possible relations. We thus have manually selected predicates that we believe are useful for generating hints for location entities, such as ones that indicate unique and well-known properties like a country's population, currency, capital, etc. We list them in Table 5 in the Appendix. Note that sometimes a property of a target entity may have multiple object entities. For example, "shares border with" will usually return multiple countries or states as objects. In such a case, we list the first 5 object entities in the hint.

To construct hints for "Where?" questions we use a set of templates with a slot to be filled for each used property. For example "*Head of state of the searched country is <headOfState>*.". In Tab 2, we show the top generated hints for an example "Where?" question.

[9]We found this value to result in quite good hints following a small-scale manual assessment of sample generated hints.

4.3 Generating Hints for "Who?" Questions

Same as in the case of hint generation when asking about locations, we use Wikidata to help answer "Who?" questions as it contains detailed information on a large number of persons. We carefully select relevant properties which could lead to meaningful and useful hints. The complete set of properties is listed in Table 6 in the Appendix. Similarly to the "Where?" question type, we utilize templates that are filled with found properties. We present a few samples of generated hints in Table 3.

5 EXPERIMENTS

We first discuss in Section 5.1 the experimental settings followed by the approach for generating baseline hints in Section 5.2, and then we describe the experimental results in Section 5.3.

5.1 Experiment Setup

Since automatic hint generation is a novel task, no datasets or benchmarks are available. We have thus conducted a task-oriented evaluation to see if the generated hints were indeed useful. The experiments were performed with 10 users with ages ranging from 20 to 54 years old. For the evaluation, we used 30 question-answer pairs that were randomly selected from the SQUAD dataset [14]. In particular, for each question type, we selected 10 questions by making sure the type of their answers was correct.

We then generated 5 hints for each question-answer pair using the approach described above. We have also generated hints based on a baseline which is described in Sec. 5.1.1. The hints generated by our approach and the ones given by the baseline were then used by two groups consisting of 5 different evaluators who tried to answer the questions. In total, considering the number of questions and the assessors, each different type of question was answered 50 times by both groups of users. The evaluators were first asked if they know the answer to each question, and we made sure that the answers to all the asked questions were previously unknown to the users taking part in the experiment. The evaluators were then requested to read the first hint and to try to come up with the correct answer. If the answer was wrong, they could see the second generated hint, after seeing which, they were asked to enter the answer again. If the answer was correct, the evaluators could move to the next question, otherwise, we showed up to 5 hints following the above procedure. We made sure that small differences such as the differing case of an answer word were properly taken care of. We finally counted the success rate of the two user groups defined

Table 1: Example top-scored hints generated for an example "When?" question.

Question: "In which year was the Hubble Space Telescope launched?" (answer: 1990)
The following event occurred on the searched date: Russian aircraft carrier Admiral Kuznetsov is commissioned.
The following event occurred on the searched date: The World Health Organization removes homosexuality from its list of diseases.
The following event occurred on the searched date: Singing Revolution: The Soviet Union announces that Lithuania's declaration of independence is invalid.
The following event occurred on the searched date: Mikhail Gorbachev is elected as the first executive president of the Soviet Union.

Table 2: Example top hints generated for an example "Where?" question.

Question: "General Franco became leader of which country in 1939 after a Civil War?" (answer: Spain)
Currency in the searched location is euro.
The next body of water of the searched location is Atlantic Ocean.
The searched location shares border with Portugal.
Head of government of the searched location is Pedro Sánchez.
The highest point in the searched location is Teide.

Table 3: Example top hints generated for an example "Who?" question.

Question: "Who became the most respected entrepreneur in the world according to Financial Times in 2003?" (answer: Bill Gates)
The searched person held the position of chief executive officer.
The searched person has 2 siblings.
The searched person has following citizenship United states of America.
The searched person was born in Seattle.
The searched person has gender male.

as the number of correctly answered questions. Furthermore, the users were later asked to assess the utility of the presented hints. The rating scores were as follows:

(1) not useful
(2) so so
(3) useful

5.2 Baseline Hints

We prepared baseline hints as follows. For "Who?" and "Where?" question types we randomly collected sentences from Wikipedia abstracts of the articles about the corresponding answers to be used as hints. Wikipedia abstracts contain the most descriptive and relevant information about the described entities. Note that no utility score was computed in this case, so the relevance of the hint to the question was not considered. We also made sure that the answer to the question itself does not occur in the selected sentences to prevent the answer leakage problem and avoid generating obvious hints.

For "When?" type questions we randomly selected events from the corresponding Wikipedia year pages.

5.3 Results

Table 4 shows the results obtained for both user groups. We see that the hints generated by the proposed approach were more useful than the baseline hints. Out of 30 questions with a maximum of their corresponding 5 hints, the evaluators group using our hints were able on average to answer correctly 24.2 (i.e., 80%) questions. On the other hand, the control group, which received the baseline hints, could answer 16 (i.e., 53%) questions on average. This observation is corroborated by the average hint rates (see Figure 3). The average score for our hints is 1.97, compared to 1.5 for baseline hints.

If we look at the rate of questions that could be answered correctly per question type, shown in Figure 2, we can notice that the difference is the largest for "When?" questions. For example, out of 50 times when the "When?" questions were asked we received only 4 correct answers when using the baseline hints, whereas the users could correctly answer 39 times using our generated hints. We also deduce that the hints generated by the proposed approach were quite helpful for "Where?" questions, too, judging from the average number of correct answers. The rates of right answers for "Who?" questions are however nearly the same. One reason for this

Table 4: The average numbers of correct answers and the average ratings of hints.

Method	#Correct answers	Rating: *Who? quest.*	Rating: *Where? quest.*	Rating: *When? quest.*	Total rating
Baseline hints	16	1.75	1.66	1.08	1.50
Proposed hints	**24.20**	**1.84**	**2.26**	**1.79**	**1.97**

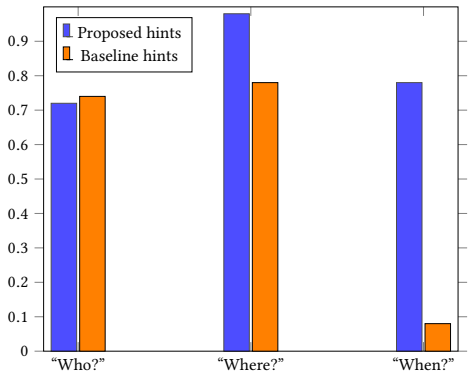

Figure 2: Average rate of correct answers per question type

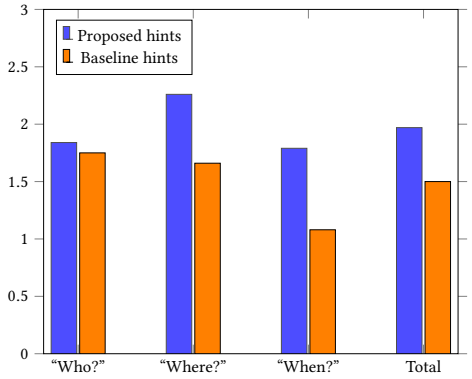

Figure 3: Average rating per question type (the higher the score, the better)

might be that Wikipedia abstracts about persons such as celebrities already contain quite useful information. Another reason might be related to the choice of properties, as will be also discussed later.

Next, we look into the ratings given by the users to hints generated for questions of a different type shown in Figure 3. The average rating of the hints for "When?" type questions is 1.79 when using our approach, and 1.08 for baseline. This agrees with the previous observation for this question type and suggests that event popularity is an important factor to consider when generating hints, despite that Wikipedia should actually already contain relatively important events in its year pages. The average rating of the hints for "Who?" type questions is similar for the proposed method (1.84) and the baseline method (1.75) indicating that the properties which were used to extract information about persons could be selected in a better way. While we took care to choose meaningful properties, more investigation is needed to find suitable properties. For example, hints like: "*The searched person has a height of ...*" and "*The searched person has ... siblings*" were often rated as "not useful" by participants. Also, hints like "*The searched person was educated at ...*" do not seem to restrict the number of potential answers sufficiently.

The hints for "Where?" questions were rated as best with an average rating of 2.26 compared to the average rating of 1.66 for the baseline hints. These results suggest that the properties in Wikidata about locations could be more useful than the ones used for persons. For example, predicates like "Shares border with ..." restrict the possible locations to a few, or ones like "Head of state is ..." lead users towards a concrete single location, while usually not resulting in an obvious or trivial hint.

6 LIMITATIONS

Our work has the following limitations.

First, the current approach generates hints without considering educational objectives. Hence, the output hints do not provide strong cues to create long-lasting associations nor to foster understanding and learning regarding the questioned entities. We have rather focused on demonstrating that using Wikipedia it is possible to generate in easy way multiple hints that lead users to correct answers.

Second, we have focused on three types of questions (leaving other types for future work), and assumed that answers are in the fowe rm of named entities. The latter allows directly using Wikipedia and Wikidata for generating hints.

Third, the evaluation method used in our experiments focuses only on whether users could guess the answers (success rate), and on the general notion of hint usefulness, without specifying detailed criteria related to good hints such as interestingness, obviousness, learnability, readability, or others.

7 CONCLUSIONS

The omniscient large language models like ChatGPT that have been popular lately are likely to pose risks in reducing the motivation of average persons to learn and to critically think. The objective of this research is to propose a novel task of automatic hint generation (HG) for helping users with answering their questions and to outline its benefits and opportunities for further research. Advanced HG systems could foster the process of learning and understanding when interacting with chatbot and question answering systems. Our belief is that automatic hint generation should be considered as an important, complementing research direction for the established tasks of question answering and question generation.

The second objective was to design, implement as well as test the proof-of-concept approach to assess if the task is feasible. We proposed an approach that takes a question-answer pair as input and generates hints based on processing the Wikipedia/Wikidata

content. The evaluation showed that these data sources are appropriate for our objective and the hints generated by our approach are useful. The difference in the rate of questions answered correctly for both user groups, and the difference in the obtained ratings indicate that the proposed method produces effective hints that lead users closer to the right answers.

What should be investigated next, besides researching the previously mentioned customization and serialization extensions of HG task (see Section 1), is designing a dedicated evaluation approach for the HG task. Although this was out of the scope of the current paper, it forms a part of our future plans. An effective evaluation framework would drop the requirement for human-based evaluation, which although highly reliable, is obviously also quite costly and limits reproducibility.

Regarding our introduced method, we plan to ensure the presence of pedagogical aspects in generated hints to allow users to learn or at least better remember the answers. Second, we will investigate incorporating the obviousness degree of hints to make sure that the generated hints are not too simple. For this, one could evaluate the number of potential answer candidates to which each generated hint confines the initial candidate pool. For example, if the answer is "Ireland", the hint "*The searched location shares border with UK.*" would be too obvious as leaving only a single possible answer. Similarly, properties like "headOfState" might be also sometimes too obvious (e.g., of USA), although one may still not know the answer for less-popular countries, which suggests that the problem is not that trivial and may require additional steps (e.g., popularity assessment of involved entities and properties).

REFERENCES

[1] Albert Bandura. 2013. The role of self-efficacy in goal-based motivation. (2013).

[2] Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023* (2023).

[3] Tiffany Barnes and John Stamper. 2010. Automatic hint generation for logic proof tutoring using historical data. *Journal of Educational Technology & Society* 13, 1 (2010), 3–12.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[5] Michael Färber, Frederic Bartscherer, Carsten Menne, and Achim Rettinger. 2018. Linked data quality of dbpedia, freebase, opencyc, wikidata, and yago. *Semantic Web* 9, 1 (2018), 77–129.

[6] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. (2020), 6769–6781.

[7] Christine Klein-Braley and Ulrich Raatz. 1984. A survey of research on the C-Test1. *Language Testing* 1, 2 (1984), 134–146.

[8] Ghader Kurdi, Jared Leo, Bijan Parsia, Uli Sattler, and Salam Al-Emari. 2020. A systematic review of automatic question generation for educational purposes. *International Journal of Artificial Intelligence in Education* 30, 1 (2020), 121–204.

[9] Chao-Yi Lu and Sin-En Lu. 2021. A Survey of Approaches to Automatic Question Generation:from 2019 to Early 2021. In *Proceedings of the 33rd Conference on Computational Linguistics and Speech Processing (ROCLING 2021)*. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP), Taoyuan, Taiwan, 151–162.

[10] Jessica McBroom, Irena Koprinska, and Kalina Yacef. 2021. A survey of automated programming hint generation: The hints framework. *ACM Computing Surveys (CSUR)* 54, 8 (2021), 1–27.

[11] Juan Pino and Maxine Eskénazi. 2009. Measuring Hint Level in Open Cloze Questions. In *Proceedings of the Twenty-Second International Florida Artificial Intelligence Research Society Conference, May 19-21, 2009, Sanibel Island, Florida, USA*, H. Chad Lane and Hans W. Guesgen (Eds.). AAAI Press. http://aaai.org/ocs/index.php/FLAIRS/2009/paper/view/69

[12] Thomas W Price, Yihuan Dong, Rui Zhi, Benjamin Paaßen, Nicholas Lytle, Veronica Cateté, and Tiffany Barnes. 2019. A comparison of the quality of data-driven programming hint generation algorithms. *International Journal of Artificial Intelligence in Education* 29 (2019), 368–395.

[13] Thomas W. Price, Rui Zhi, and Tiffany Barnes. 2017. Hint Generation Under Uncertainty: The Effect of Hint Quality on Help-Seeking Behavior. In *Artificial Intelligence in Education*, Elisabeth André, Ryan Baker, Xiangen Hu, Ma. Mercedes T. Rodrigo, and Benedict du Boulay (Eds.). Springer International Publishing, Cham, 311–322.

[14] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250* (2016).

[15] John Stamper, Michael Eagle, Tiffany Barnes, and Marvin Croy. 2013. Experimental evaluation of automatic hint generation for a logic tutor. *International Journal of Artificial Intelligence in Education* 22, 1-2 (2013), 3–17.

[16] Ellen L Usher and Frank Pajares. 2006. Sources of academic and self-regulatory efficacy beliefs of entering middle school students. *Contemporary educational psychology* 31, 2 (2006), 125–141.

[17] Zhen Wang. 2022. Modern question answering datasets and benchmarks: A survey. *arXiv preprint arXiv:2206.15030* (2022).

[18] H. Wood and D. Wood. 1999. Help seeking, learning and contingent tutoring. *Computers & Education* 33, 2 (1999), 153–169. https://doi.org/10.1016/S0360-1315(99)00030-5

[19] Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A. Smith. 2023. How Language Model Hallucinations Can Snowball. arXiv:2305.13534 [cs.CL]

[20] Ruqing Zhang, Jiafeng Guo, Lu Chen, Yixing Fan, and Xueqi Cheng. 2021. A review on question generation from natural language text. *ACM Transactions on Information Systems (TOIS)* 40, 1 (2021), 1–43.

[21] Tiancheng Zhao, Xiaopeng Lu, and Kyusong Lee. 2021. SPARTA: Efficient Open-Domain Question Answering via Sparse Transformer Matching Retrieval. In *Proceedings of the 2021 NAACL-HLT*. 565–575.

APPENDIX

Table 5: Predicates used for generating hints for "Where?" type questions.

P35: head Of State, P30: continent, P36: capital, P610: highest point, P1082: population, P421: located in time zone, P38: currency, P17: country, P1376: capital of, P47: shares border with, P131: located in territorial, P1830: owner of, P6: head of government, P793: significant event, P37: official language, P463: member of, P206: located in or next to body of water

Table 6: Predicates used for generating hints for "Who?" type questions.

P21: sex or gender, P27: country or citizenship, P569: date of birth, P19: place of birth, P1971: number of children, P106: occupation, P1340: eye color, P1884: hair color, P2048: height, P39: position held, P69: educated at, P512: academic degree, P102: member of political party, P3602: candidacy in election, P800: notable work, P166: awards received, P3373: siblings, P1412: languages spoken, written or signed, P413: position played in team / speciality, P118: league, P54: member of sports team