Abstract:

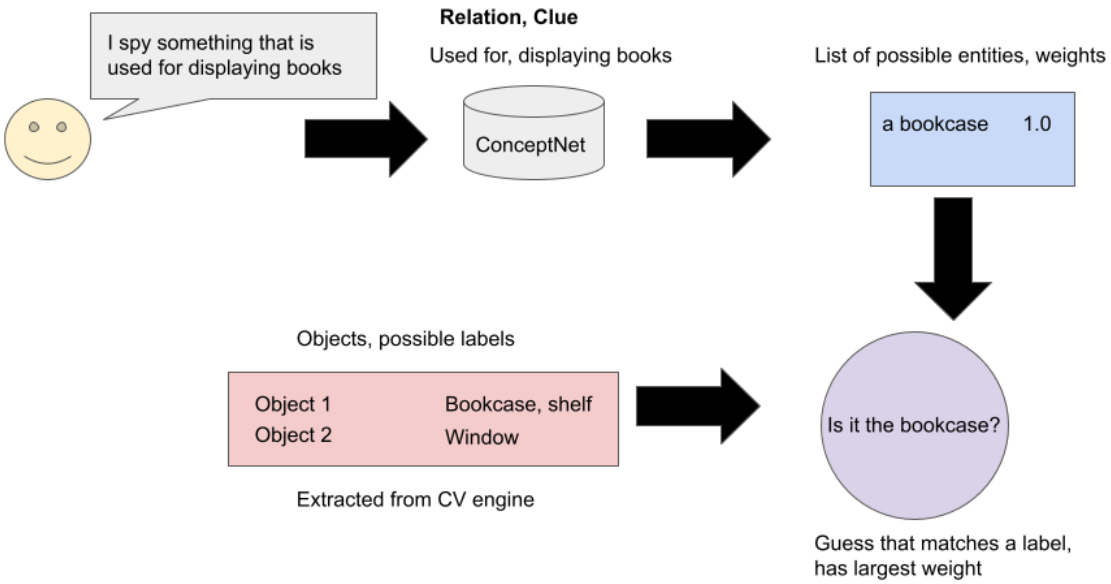
Family car rides have been identified as a main locus for the informal exchange of knowledge between parents and young children as part of everyday conversations and questioning [4 , 5]. However, there is also a trend of in-car time being a source of tension and boredom for children [3,7]. To leverage these learning opportunities while enhancing family experiences in the car, we aim to develop AISpy, a speech agent capable of playing turn-taking, contextually-aware games with kids in order to support conceptual learning, commonsense knowledge acquisition, and promote family bonding. In this paper, we develop new functionality that allows AISpy to query and process commonsense knowledge concepts for the "I Spy" guessing-game style of play and generate question-answer pairs using existing Question-Answer Generation models. Overall, this work will aim to contribute to new generations of interactive technologies that promote curiosity-driven educational experiences that are both fun and effective.

Introduction:

Family car rides are often a source of tension and boredom for children [3,7], yet they have the potential to enable playful, social, curiosity-driven experiences. To address this, Hoffman et al. developed Mileys, "a car game that integrates location-based information, augmented reality and virtual characters" with the goal of making car rides more interesting for children and promoting bonding with both their family and their environment [7]. Another car focused mobile application developed by researchers is nICE: nice In-Car Experience, where passengers work together to uncover the tiles of an image through a series of minigames [3]. Depending on the performance during the minigame, a certain number of tiles will be revealed [3]. In terms of question-answer generation, researchers have developed question-answer generation models trained on the SQuAD dataset [1,6,8]. SQuAD is a reading comprehension dataset composed of crowdsourced questions on a set of Wikipedia articles, where the answers are a segment of text from the corresponding article [9].

We build off of previous research by building an app tailored for use during car rides, capable of pulling in data from the local environment or provided by a user in order to generate content for curiosity-driven learning games. Specifically, to trigger play, the user takes a photo of their local environment and uploads it in the AISpy Android app. Using the information extracted by the app's computer vision engine about entities contained in the photo, AISpy is able to formulate a descriptive clue for the user and guess entities based on a clue given by the user. The aims of my DREU project are to build off of the existing functionality of AISpy by integrating new modules that can (a) query and process commonsense knowledge concepts harvested from a database to (b) retrieve additional information on the entities in the image and (c) offer additional spin-off learning activities (e.g., playful quizzes, informative stories) beyond the basic guessing style game play. Our target user group is children 3-5 years of age who have an inherent curiosity and need to learn fundamental concepts about the world around them.

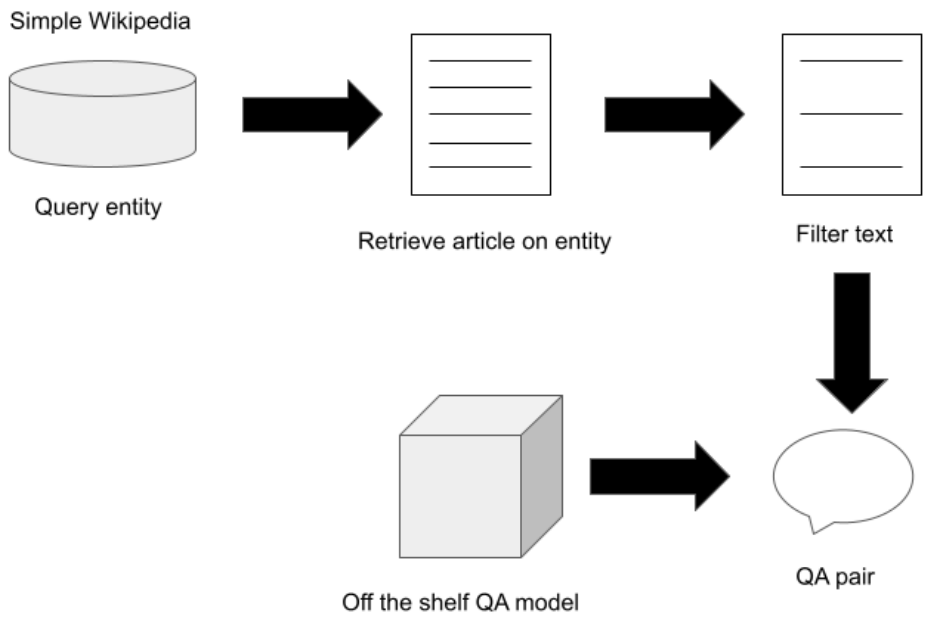
Prototype



Pipeline for querying a user's clue in ConceptNet, with a basic example

ConceptNet is made up of nodes, which are words or phrases and edges, which are units of knowledge that link one node to another node [10]. AISpy processes a given clue by querying the attribute as an end node and relation to the entity as the edge in the ConceptNet API, which returns the result in JSON-LD format [10]. For instance, if the given clue is “I spy something that is used for displaying books”, AISpy will return start nodes that are connected by the edge “used for” to the end node “displaying books”. The start nodes are the entities with the property given by the clue. AISpy compares these entities to the possible labels of each object. If any of the entities match any of the possible labels, the AISpy guesses the entity with the highest weight, an indicator of how believable the information is, specified by ConceptNet.

The main issue that we encountered was that sometimes the possible labels for each object were unrelated to one another. For instance, when we tested the CV engine on a photo of squirrels on the grass, it labeled one object as being both grass and a squirrel. This made it difficult to rank objects for guessing, especially in the case where we wanted to allow the user to provide more clues to the agent. We also tried to program AISpy to also look at synonyms of the clue, but found that there would be a long runtime, so we did not include it.



Overview of the QA generation pipeline

The quiz style game play is introduced when the AISpy makes a correct guess. The question-answer pairs are generated from a Simple Wikipedia article on that entity. Most of the off-the-shelf Question-Answer (QA) models that we tested were trained on the SQuAD dataset or other datasets drawn from Wikipedia articles. SQuAD is a reading comprehension dataset composed of crowdsourced questions on a set of Wikipedia articles, where the answers are a segment of text from the corresponding article [9]. When we tried to generate text from the results of Kiddle, a search engine for kids, the QA models produced very few questions. Moreover, there was not an accessible API for extracting "kid-friendly" text, so we extracted text from Simple Wikipedia. To make the text more "kid-friendly", AISpy is programmed to calculate the modified LIX readability score of each sentence, filtering out sentences that have a LIX readability score less than or equal to 25. The LIX formula is originally defined as the sum of the percentage of long words (more than 6 letters) and the average number of words per sentence [10], offering a coarse yet easily implemented means of filtering out content that would be overly complex for a young learner. However, when testing this filtering system on a Simple Wikipedia article on elephants, we found that very few sentences were meeting the threshold that we set, thus few usable questions were being generated. When we looked back on the text, we felt that there were sentences that were reasonable for our target age group and thought that since "elephant" itself was a long word, it could have caused the readability score of the sentence to be higher. We therefore modified the formula so that if the entity itself is a long word, it would not

be included in the word count when calculating the readability score. The goal of filtering was to exclude concepts or words that may be too difficult for our target user group to understand.

One issue that we encountered was that when we were testing out various off-the-shelf models, the question-answer pairs being generated often did not make sense or were not factually correct. In one instance, when generating a set of questions and answers from an article on elephants, one of the questions that was generated was “what types of elephants are larger than females?” and the answer generated was “males”. While this question is grammatically correct and technically true, it is not formulated fluently, in the way a human would express the same idea. An example of a question that was generated that had an incorrect answer was "Why is an elephant's skull relatively short to provide better support?" and the answer was “the neck”.

Another issue we encountered was that the app would time out because the QA generation would take too long. The main factor was that the text being used to generate the questions was too long and therefore filtering through each sentence would cause the app to time out. We tried to combat this by only filtering through the first 10 sentences of the text. This approach was able to output a question-answer pair without timing out; the downside is that it also limits the number of questions that can be generated, but it typically still produces enough to support a sufficient play experience.

Future Work

As mentioned previously, we encountered an issue wherein many of our utilized QA models generated questions that did not make sense and, in some cases, did not generate any questions at all when given a text fetched from Kiddle. We suspect this can be attributed to the more basic wording of the text, since most QA models are trained on the SQuAD dataset, which has more complex wording than the corpora from which we are trying to generate QA pairs. To address this shortcoming, we might find or curate a dataset of slightly more complex texts in order to enhance the question generation model or we might also incorporate human interaction into the loop, to allow the parent or the child to indicate whether or not a particular question is grammatically incorrect, too difficult, or does not make sense. In the immediate term, this would enable our app to flag that question in a database as a poor question to avoid in the future; and further, this data might act as input to a reinforcement learning approach that could be trained to recognize whether generated questions are appropriate to serve to users.

Another improvement we can explore is rather than dynamically processing the user’s input, we can also “preprocess” images and store them in the database, along with their corresponding QA pairs. That way it will make it easier to rank the objects, since we can manually throw out any labels that are incorrect or do not make sense and the questions can be fetched from the database rather than having the user wait for the questions to be generated on the fly.

Finally, a top future priority is to conduct user testing of these app features to evaluate childrens' engagement levels, learning gains, psychological attitudes about their ability as a learner, and how the family interacts during the game. Using these results, we can understand whether our approach meets our goals of supporting conceptual learning, commonsense knowledge acquisition, and family bonding and modify our app accordingly as well as inform the design of future educational technologies.

References:

[1] Alberti, C., Andor, D., Pitler, E., Devlin, J., & Collins, M. (2019). Synthetic QA Corpora Generation with Roundtrip Consistency. *ArXiv:1906.05416 [Cs]*. http://arxiv.org/abs/1906.05416

[2] Anderson, J. (1983). Lix and Rix: Variations on a Little-known Readability Index. *Journal of Reading*, *26*(6), 490–496. https://www.jstor.org/stable/40031755

[3] Broy, N., Goebl, S., Hauder, M., Kothmayr, T., Kugler, M., Reinhart, F., Salfer, M., Schlieper, K., & André, E. (2011). A cooperative in-car game for heterogeneous players. *Proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, 167–176. https://doi.org/10.1145/2381416.2381443

[4] Callanan, M. A. (1985). How Parents Label Objects for Young Children: The Role of Input in the Acquisition of Category Hierarchies. *Child Development*, *56*(2), 508–523. https://doi.org/10.2307/1129738

[5] Callanan, M. A., & Oakes, L. M. (1992). Preschoolers' questions and parents' explanations: Causal thinking in everyday activity. *Cognitive Development*, *7*(2), 213–233. https://doi.org/10.1016/0885-2014(92)90012-G

[6] Chan, Y.-H., & Fan, Y.-C. (2019). A Recurrent BERT-based Model for Question Generation. *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, 154–162. https://doi.org/10.18653/v1/D19-5821

[7] Hoffman, G., Gal-Oz, A., David, S., & Zuckerman, O. (2013). In-car game design for children: child vs. parent perspective. *Proceedings of the 12th International Conference on Interaction Design and Children*, 112–119. https://doi.org/10.1145/2485760.2485768

[8] Lopez, L. E., Cruz, D. K., Cruz, J. C. B., & Cheng, C. (2020). Simplifying Paragraph-level Question Generation via Transformer Language Models. *ArXiv:2005.01107 [Cs]*. http://arxiv.org/abs/2005.01107

[9] Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). SQuAD: 100,000+ Questions for Machine Comprehension of Text. *ArXiv:1606.05250 [Cs]*. http://arxiv.org/abs/1606.05250

[10] Speer, R., Chin, J., & Havasi, C. (2017, February). Conceptnet 5.5: An open multilingual graph of general knowledge. In Thirty-first AAAI conference on artificial intelligence.