A Markov Chain Tool for Grade 6-12 Learners to Explore Generative AI

Rebecca Ellis[1], Jie Chao[1], Carolyn Rosé[2], and Shiyan Jiang[3]
[1]Concord Consortium, [2]Carnegie Mellon University, [3]North Carolina State University

Abstract

The *AI Education Across the Curriculum* Project (also known as *StoryQII*) has developed a digital tool to support students to represent, inspect, and generate text using a Markov chain. The tool is designed for use in an English Language Arts (ELA) class and does not require coding or statistics. Using this tool and its accompanying curriculum module, secondary students learn the basics of text generation and how it relates to the core ELA concepts of voice, authorship, and creativity. This tool has been tested in ninth grade, eleventh, and twelfth grade ELA classes with promising results for teaching students about generative AI.

Background

By September 2024, OpenAI had amassed over one million paying users, and over 200 million weekly active users for its Generative Artificial Intelligence (GAI) tool (Reuters, 2024). As GAI becomes more popular and the controversies over its role in K-12 education continue to grow, it is imperative that students gain an understanding of what GAI can, and cannot, do, and why. *AI Education Across the Curriculum* is a National Science Foundation project (DRL-2241669) created to embed the teaching of AI into core disciplinary subject areas in high school. As part of this project, the team created a tool that leverages Markov chains to teach students about how AI can represent and generate text.

Markov chains are "the simplest mathematical models for random phenomena evolving in time" (Norris, 1988, p. 1). These stochastic models describe memoryless sequences of possible events. At each step, the next event depends on the previously attained state, and the choice for the following step is determined via probabilities calculated from an initial input of sequences. In essence, a Markov chain serves as a "next-state generator," which parallels next-word prediction used for GAI models. Modern large language models now use more sophisticated machine learning models, but the history and overall process remain. Our project has found that lessons and activities for students leveraging a Markov chain model provide an effective entry point for introducing students to the processes of GAI.

Artificial Intelligence for Georgia is one of the early projects working to bring AI to students (https://ai4ga.org/). As part of their extensive project, they have been developing a Markov chain model that uses statistics to teach students about next-word prediction in GAI. Our project similarly aims to bring AI into K-12 education, but with a twist. Instead of designing special AI courses or elective modules, we look for spaces within core disciplinary courses to introduce AI concepts. Our tools are designed to be led by non-computer science teachers within their subject-area courses. As a result, we develop less technical and more visual tools to convey critical AI concepts. Our Markov chain tool has been designed for use in an English Language Arts (ELA) classroom, for typical high school ELA students, and taught by a traditional ELA teacher.

The Markov chain tool

To support secondary students to learn about GAI, our project has created two resources: the Markov chain tool and a week-long ELA curriculum module. We created our module along with two ELA teachers who not only codesigned with us, but brought pilot materials to their classrooms for testing and data collection. This paper focuses primarily on the tool itself. Other papers address the course material.

Our Markov chain tool is a plugin for CODAP, the popular, open-source platform that is used in classrooms around the world for visualizing data (Common Online Data Analysis Platform, 2014). CODAP allows students to harness large data sources and explore the data though dynamically-linked graphs and tables. By building the Markov chain tool within CODAP, we leverage these technological affordances along with the affordances of the tool itself.

The major functionalities of our Markov chain model tool are representing text, inspecting graphs, and generating sequences. In this paper, we will use the short sentence, "My name is Jimmy and my cat's name is Alfredo." as the input data to explore how the tool helps students to understand how text can be represented and used to create both meaningful and nonsensical new text. Because of the sentence's mix of unique and non-unique words, it demonstrates how the Markov chain tool generates loops and variation.

Representing text

As taught in an ELA class, speaking and writing is the act of putting words and punctuation in a sequence. These strings of words and punctuation follow certain rules and patterns, as well as allow for a flexibility and creativity.

Bringing these sequences into the Markov tool plugin visibly represents text in a manner that mimics human speech while also demonstrating rudimentary GAI techniques.

The tool converts input text into a bubble-and-arrow graph that represents states and transition probabilities to show how words are connected to each other. To help students understand how this represents text, we have created three ways for students to create the graph: using a table, manually drawing, and using the Text-to-Graph function. Each method of creation encourages students to think about the training text in slightly different ways, which contributes to their understanding of the representation.

Using a table

Using the first method, students need to first create their text as a list of values in a table. Each row contains a single word or punctuation mark. This table creation takes thought, as there are different ways to slice and consider the data (which hits on the AI concept of "tokens"). Students can make each word and punctuation its own row, or they can choose to combine certain words or phrases. Students can decide if a possessive (such as in the word, "cat's") should be part of the word thereby creating a single row for "cat's," or if the "'s" should be separate, or even if the apostrophe and the "s" should each get their own row. Each choice will have an impact on the resulting graph and subsequent generated text.

Once students have created their table, they drag and drop it into the tool where it will autogenerate a graph (Figure 1A). Bubbles are created for each state, and arrows are drawn between states to show the transitions. The probabilities for these transitions are automatically calculated based on the relative frequency of each transition within the table data, and can be viewed by double clicking on a state.

Manually drawing

The plugin also provides students with the capability to manually create the graph. The first step is to determine all the unique words and punctuation in the sentence. Like with the table method, determining what deserves its own bubble is a thinking task with consequences. One way to parse the sample sentence, "My name is Jimmy and my cat's name is Alfredo.", is to determine that there are 7 unique tokens. Students would then create 7 "bubbles" or states, and type the token text in each (Figure 1B). The tool does not pay attention to capitalization, by design, and therefore students need not worry about including that in their drawing.

Students then draw arrows to represent transitions between the words based on the input data. The easiest way to do this is to follow the sentence, placing arrows to show the path of tokens throughout the input data. Repeated, non-unique tokens may be passed through more than once. For example, in the sample sentence, after the word "my" the words "name" and "cat's" both come next, so "my" will have two arrows coming out of it (Figure 1C). This act of manually drawing the graph can be tedious, but it is a good learning experience for students to really understand how the graphs are created and how they represent text.

As the arrows are created, implicit probabilities are set with equal weight for each transition coming out of a state. For example, in our sample data, the transition probability of going from "my" to "name" is initially set to 50% because there are two outgoing arrows. Students can manually override these probabilities, however, by double clicking on the starting state and adjusting the probabilities in a popup (Figure 1D). If, for example, the input data included "my name" twice and "my cat's" once, then the student could drag the slider to reflect this, and set the probabilities to 67% and 33%. In this popup, the student can also rename the state, which helps in instances of typos or a change in the input data. In larger and more significant datasets, adjusting the transitions and state may have grand impacts on the eventual generated text, which leads to important discussions about responsible model creation and bias.

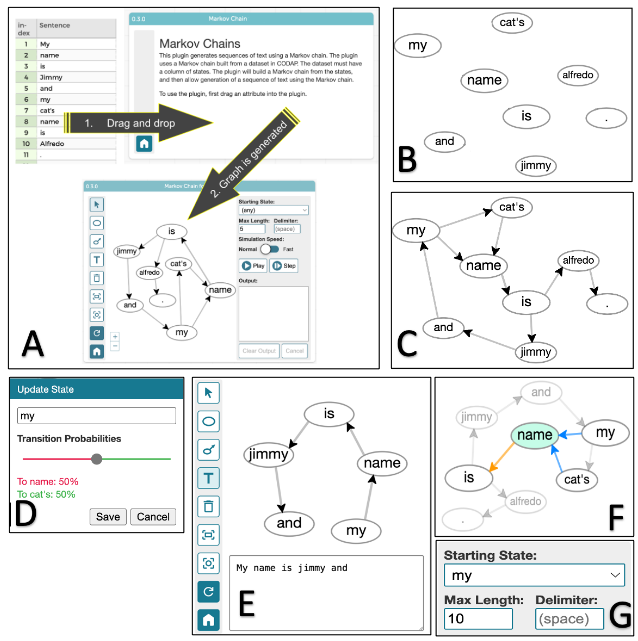

Figure 1. A. Converting a table into a Markov chain. **B.** Drawn bubbles to represent states. **C.** Drawn arrows to connect states. **D.** Renaming states and adjusting transition probabilities. **E.** Using the text tool to generate a Markov chain model. **F**. Selecting a state and its transitions into the foreground. **G.** Settings for generating the output text.

Using the Text-to-Graph functionality

The third way to convert from written text to the graphical representation, and perhaps the easiest, is to use the text-to-graph feature. When activated, students view a textbox where they can type or copy in their text. As the text is written, the corresponding graph is automatically generated in the workspace with transition probabilities updating as additional text is drafted (Figure 1E). This functionality is particularly useful when representing large bodies of text, for students who are uncomfortable with the fine motor skills required for manual construction, or as a check for comparing a hand-drawn graph with computer-generated one.

Inspecting graphs

Once the graphical representation of the text has been created, students can modify and inspect the graph for further comprehension. Each method encourages students to consider how language is used and shared to convey meaning.

Rearrange for ease

Students can click and drag states on the graph into different arrangements. Some students adjust the states to improve readability, while others arrange the graph to look like a decision tree. The rearrangement impacts only the visual representation, and has no impact on the underlying model.

Inspect states

Students can select any state in the graph to forefront it for more inspection. (Figure 1F). The selected state becomes highlighted in a pale green, while all unconnected states fade. Transitions toward the selected state are highlighted in blue, transitions away in orange, and reflexive transitions, if any, in purple. All unconnected arrows also are temporarily faded. As students inspect the states, the corresponding module encourages discussion about how words are sequenced in the English language. In longer input datasets, these inspections draw attention to words that are used in many different ways, and students explore the different parts of speech that can come next.

Inspect transition probabilities

Students can also double click on any state to open the popup that lists all the outgoing transitions and their probabilities (Figure 1D). As students review the transition probabilities, they can quantitatively see how different words are more or less likely to come after a given word within the input text. This leads to an engaging discussion about word choice, word order, and meaning.

Generating text

The most exciting part of the tool is using it to create new text. The Markov chain graph serves as the playground for the tool to follow the paths and create sequences of words and punctuation. Using the tool's righthand panel (Figure 1G), students can select a starting state, or allow the model to randomly choose for them using a weighted probability. For the purpose of this example, we will suppose that the students select "my" as the starting state.

Next, students choose the max length of the sequence. The system has no limit and the model will continue for that length or until it reaches a point where it has no outgoing transitions. For example, our sample statement has no state after the period, and so once a sequence reaches that state, it will automatically terminate even if the desired length has not been met. Students may also choose a delimiter between states, or can leave the default as a space.

To generate the sequence, students have the option to "Play" or "Step." Play will work its way through the full sequence on its own, while step will allow the user to stop and inspect throughout. Students can switch back and forth between settings at will.

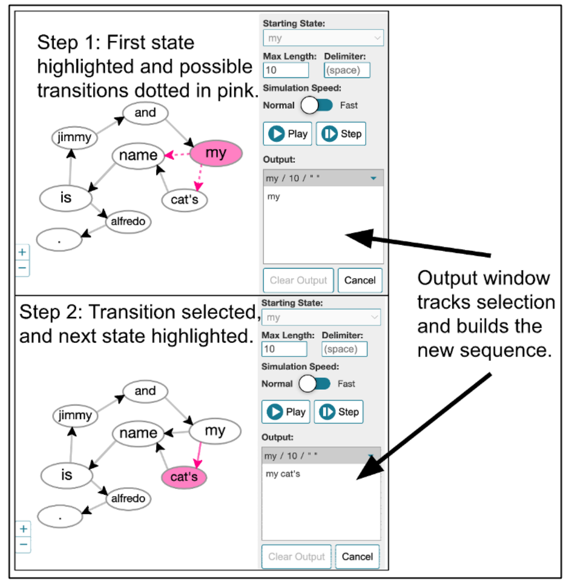

Figure 2. Students can step through text generation as the choices are highlighted and recorded.

When a student first presses "Step," the starting state in the graph is highlighted in pink, and all its possible transitions are dotted with pink. Additionally, this first state is recorded in the "Output" window (Figure 2). When "Step" is clicked again, the model uses a random selection weighted by the calculated transition probabilities to choose the next state, which is indicated by becoming a solid pink arrow and the other options return to gray, and the chosen state highlighted in pink. The selected next state is also recorded in the Output window.

Once the sequence has reached its max length or termination, students can review the full sequence in the Output window. The Output window also includes a gray bar with the run settings, which allows students to change their settings, run the model again, and review the different outputs. Clicking on any sequence in the Output window will also cause the full path to light up in pink on the graph.

Since the sequences are generated using probabilities, repeated runs with the same settings often lead to different sequences. This leads to important discussions about determinism and creativity in writing. It also can lead to interesting discussions about plagiarism.

After the students run the simulation several times with the starting text as "my" and a high max length, they inspect the output. First, whenever something is named "Alfredo" the sequence concludes. This is because there is only one next place to go, which is the final period. Second, after someone is named "Jimmy" the text continues because the only next state is "and," which can lead to a repeating loop.

Next, based on the probabilities, the second word in the sequence is always "name" or "cat's" and this appears in about equal frequency. With only a few runs, this 50% chance will not likely be apparent, but with a larger sample size this will tend to be true. If time allows, we encourage teachers to delve into explorations about tendency toward the mean despite individual variance.

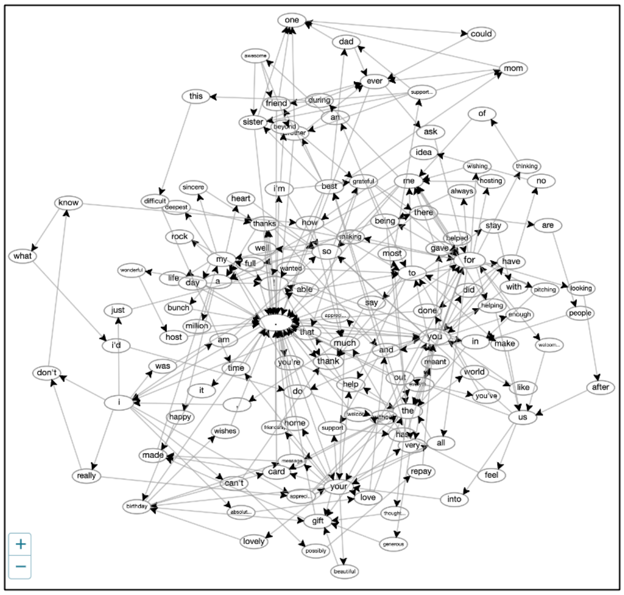

Figure 3. This graph is generated from a dataset of 70+ ways to say Thank You. Students use this model to explore limitations of a bi-gram Markov chain model.

When using a larger input text, rather than our simple sentence, the model navigates more transition choices and generates more creative outputs. We have an activity where students use the text from 70+ ways to express gratitude, and attempt to use that to generate new phrases (Figure 3). While some outputs are viable, many are not. This leads to inspection of the text and the graph to find which states are involved in the strange outputs. For instance, students find that having a unique state for the punctuation leads to disjointed clauses and phrases. They also discover that a context window of a single word is often not sufficient to generate long and coherent text, which is a limitation of using a memoryless system to generate text, especially when using only a bi-gram model. Students further recognize the impact of homonyms and multi-use words on text generation.

AI Learning Goals & Expected Outcomes

Our Markov chain tool teaches students about the basics of GAI and its limitations, as the tool demonstrates rudimentary bi-gram text generation. The accompanying ELA module is aligned with the AI4K12 standards (Touretsky et al., 2023). While the bulk of the materials focus on *Big Idea 2: Representation & Reasoning*, the module also touches on the other ideas, particularly *Big Idea 5: Societal Impact*.

Students discover they need a robust starting text dataset in order to get effective and interesting outputs. When only a single sentence is used, the number of new generated sequences is limited and uncreative (Aligned to 3-C-ii).

Students explore how certain words and punctuation cause confusion in the output, such as how the word "you" can be the subject of a sentence, the object, or part of the phrase "Thank you." This causes the transitions from the word "you" to be quite varied, and due to the probabilistic nature of the model, rather than context-based understanding, the outputs may not make sense. Generated phrases may be something like "Thank you are the best" (Aligned to 4-A-i).

As students explore the possibilities and limitations of generated text via a Markov chain, students embark on meaningful conversations about when it is and is not appropriate to use GAI. They explore concepts of authorship, plagiarism, creativity, and authenticity, and wrestle with how to redefine and reassert these key concepts in a world with AI (Aligned to 5-A-ii).

Discussion and Implementation Results

We have tested our tool and module during both the 23-24 and 24-25 school year in schools on both coasts of the United States. Our dataset includes just over 140 students in a mix of ninth, eleventh, and twelfth grade classes. Across these tests, we have found promising results. Students show the ability to describe and explain a Markov chain model with respect to generated text. They demonstrate comfort with the technology and navigating the tool. Students talk about GAI as next word prediction and have meaningful conversations about the strengths and limitations. Our analysis of the pre- and post- questions revealed that students gained knowledge in describing, reading, and explaining a Markov chain model.

The affordances of this new tool show potential to help students to visualize what it means to represent and generate text. That students can create the Markov chain graphs from tables, by drawing, and writing, activates different ways to conceptualize what it means to read and represent text. These different strategies serve to meet different learning needs, and triangulate the graph into a coherent understanding. The ability to highlight specific states and explore the transition probabilities, especially in reference to paragraph text is also quite powerful. Furthermore, the connection between the tool and CODAP allows students and teachers to inspect and analyze the inputs and outputs in many different ways, allowing for flexibility and creativity.

Final Thoughts

In the coming years, we will add more functionalities to the tool and create additional corresponding modules and resources. At the point of publication, a new cohort of teachers across three states is bringing this tool to their ELA classes.

The current version of the tool, along with its curricular materials, can be viewed at learn.concord.org/storyq. With a free teacher account, teachers gain access to support materials. Anyone may also reach out to the project team directly if they would like to receive a live demo.

Overall, our data suggests that the Markov chain tool has the potential to be useful for teaching students about representing text data, analyzing models, and thinking about the possibilities and limitations of GAI.

Acknowledgements

The work presented in this paper would not be possible without the fabulous technology team at The Concord Consortium, as well as the hard work of all members of the *AI Education Across the Curriculum* Project, our codesign teachers, and their hardworking students who gave us extremely helpful feedback.

References

Artificial intelligence for Georgia. https://ai4ga.org/. Accessed: 2024-09-16.

Common Online Data Analysis Platform. 2014. [Computer software.] Concord, MA: Concord Consortium.

Norris, J.R., 1998. Markov chains (No. 2). Cambridge university press.

Reuters. 2024. OpenAI hits more than 1 million paid business users. https://www.reuters.com/technology/artificial-intelligence/openai-considers-pricier-subscriptions-its-chatbot-ai-information-reports-2024-09-05/. Accessed: 2024-09-05.

Touretzky, D., Gardner-McCune, C. and Seehorn, D., 2023. Machine learning and the five big ideas in AI. International Journal of Artificial Intelligence in Education, 33(2), pp.233-266. https://doi.org/10.1007/s40593-022-00314-1.

Appendix

The following three substandards of the AI4K12 Standards are addressed in our module and in this paper.

- 3-C-ii. “A large dataset is typically required to capture the diversity of a complex domain and narrow down the range of possible reasoner behaviors.”
- “4-A-i: “Identify portions of a text that would be difficult for a computer to understand, and explain why.”
- 5-A-ii: “To ensure that AI systems are helpful and not harmful, ethical design criteria include: fairness, transparency, explainability, accountability, respect for privacy, and adherence to societal values.”