Markov Solution

Download our Solution

markov.js

```
/** Textual markov chain generator */


class MarkovMachine {

  /** build markov machine; read in text.*/

  constructor(text) {
    let words = text.split(/[ \r\n]+/);
    this.words = words.filter(c => c !== "");
    this.makeChains();
  }

  /** set markov chains:
   *
   *  for text of "the cat in the hat", chains will be
   *  {"the": ["cat", "hat"], "cat": ["in"], "in": ["the"], "hat": [null]} */

  makeChains() {
    let chains = new Map();

    for (let i = 0; i < this.words.length; i += 1) {
      let word = this.words[i];
      let nextWord = this.words[i + 1] || null;

      if (chains.has(word)) chains.get(word).push(nextWord);
      else chains.set(word, [nextWord]);
    }

    this.chains = chains;
  }


  /** Pick random choice from array */

  static choice(ar) {
    return ar[Math.floor(Math.random() * ar.length)];
  }


  /** return random text from chains */

  makeText(numWords = 100) {
    // pick a random key to begin
    let keys = Array.from(this.chains.keys());
    let key = MarkovMachine.choice(keys);
    let out = [];

    // produce markov chain until reaching termination word
    while (out.length < numWords && key !== null) {
      out.push(key);
      key = MarkovMachine.choice(this.chains.get(key));
    }

    return out.join(" ");
  }
}


module.exports = {
  MarkovMachine,
};
```

makeText.js

```
/** Command-line tool to generate Markov text. */


const fs = require("fs");
const markov = require("./markov");
const axios = require("axios");
const process = require("process");


/** Make Markov machine from text and generate text from it. */

function generateText(text) {
  let mm = new markov.MarkovMachine(text);
  console.log(mm.makeText());
}


/** read file and generate text from it. */

function makeText(path) {
  fs.readFile(path, "utf8", function cb(err, data) {
    if (err) {
      console.error(`Cannot read file: ${path}: ${err}`);
      process.exit(1);
    } else {
      generateText(data);
    }
  });

}


/** read URL and make text from it. */


async function makeURLText(url) {
  let resp;

  try {
    resp = await axios.get(url);
  } catch (err) {
    console.error(`Cannot read URL: ${url}: ${err}`);
    process.exit(1);
  }
  generateText(resp.data)
}


/** interpret cmdline to decide what to do. */

let [method, path] = process.argv.slice(2);

if (method === "file") {
  makeText(path);
}

else if (method === "url") {
  makeURLText(path);
}

else {
  console.error(`Unknown method: ${method}`);
  process.exit(1);
}
```

Further Study: Bigrams

bigram.js

```
/** Textual markov chain generator using bigrams. */


class MarkovMachine {

  /** build markov machine; read in text.*/

  constructor(text) {
    let words = text.split(/[ \r\n]+/);
    this.words = words.filter(c => c !== "");
    this.makeChains();
  }

  /** set markov chains:
   *
   *  for text of "the cat in the hat", chains will be
   *  {"the cat": ["in"], "cat in": ["the"], "in the": ["hat"], "the hat": [null]} */

  makeChains() {
    let chains = new Map();

    for (let i = 0; i < this.words.length - 1; i += 1) {
      let bigram = this.words[i] + " " + this.words[i + 1];
      let nextWord = this.words[i + 2] || null;

      if (chains.has(bigram)) chains.get(bigram).push(nextWord);
      else chains.set(bigram, [nextWord]);
    }

    this.chains = chains;
  }


  /** Pick random choice from array */

  choice(ar) {
    return ar[Math.floor(Math.random() * ar.length)];
  }


  /** return random text from chains */

  makeText(numWords = 100) {
    // pick a random key to begin
    let keys = Array.from(this.chains.keys());
    let key = this.choice(keys);
    let out = [];

    // produce markov chain until reaching termination word
    while (out.length <= numWords && key !== null) {
      let [w1, w2] = key.split(" ");
      out.push(w1);
      key = w2 + " " + this.choice(this.chains.get(key));
    }

    return out.join(" ");
  }
}


module.exports = {
  MarkovMachine,
};
```