

Generating Random Text

Order-Zero, Order-One

Markov Text Generation

- Order-zero Markov text generation
 - Use training text to generate text randomly
 - Choose character at random; don't use any characters to "predict" next character

```
wa nitser nibsm epex uatcdfaaaleohdi f roise e.tj  
gthgb lutosth nid, loa qnednegee oIwr dasti tdnihertv  
nao o astvnscna nnrl tmba oteidpi ttfti eion nia  
beTtlies0 aod eura spnb utee y dnkmo
```

Markov Text Generation

- Order-zero Markov text generation
 - Use training text to generate text randomly
 - Choose character at random; don't use any characters to "predict" next character

```
wa nitser nibsm epex uatcdfaaaaleohdi f roise e.tj  
gthgb lutosth nid, loa qnednegee oIwr dasti tdnihertv  
nao o astvns cna nnrl tmba oteidpi ttfti eion nia  
beTtlies0 aod eura spnb utee y dnkmo
```

Markov Text Generation

- Order-zero Markov text generation
 - Use training text to generate text randomly
 - Choose character at random; don't use any characters to "predict" next character

```
wa nitser nibsm epex uatcdfaaaaleohdi f roise e.tj  
gthgb lutosth nid, loaqnednegee oIwr dasti tdnihertv  
nao o astvnscna nnrl tmba oteidpi ttfti eion nia  
beTtlies0 aod eura spnb utee y dnkmo
```

Markov Text Generation

- Order-zero Markov text generation
 - Use training text to generate text randomly
 - Choose character at random; don't use any characters to "predict" next character
 - In order-one, use one character to predict next

wa nitser nibsm epex uatcdfaaaaleohdi f roise e.tj
gthgb lutosth nid, loa qnednegee oIwr dasti tdnihertv
nao o astvnscna nnrl tmba oteidpi ttfti eion nia
beTtlies0 aod eura spnb utee y **dnkmo**

itstious acron ary, chom prondy at h) s hes cromindeathay,
on bests s orion terinomicof id cthat gul asojedafincos
fimis in. cut toffroutheail cthespend stesurd bl
ike ibst tico nat ts mpsthe cen cecy

Markov Text Generation

- Order-zero Markov text generation
 - Use training text to generate text randomly
 - Choose character at random; don't use any characters to "predict" next character
 - In order-one, use one character to predict next

```
wa nitser nibsm epex uatcdfaaaaleohdi f roise e.tj  
gthgb lutosth nid, loa qnednegee oIwr dasti tdnihertv  
nao o astvnscna nnrl tmba oteidpi ttfti eion nia  
beTtlies0 aod eura spnb utee y dnkmo
```

```
itstious acron ary, chom prondy at h) s hes cromindeathay,  
on bests s orion terinomicof id cthat gul asojedafincos  
fimis in. cut toffroutheail cthespend stesurd bl  
ike ibst tico nat ts mpsthe cen cecy
```

Markov Text Generation

- Order-zero Markov text generation
 - Use training text to generate text randomly
 - Choose character at random; don't use any characters to "predict" next character
 - In order-one, use one character to predict next

```
wa nitser nibsm epex uatcdfaaaaleohdi f roise e.tj  
gthgb lutosth nid, loa qnednegee oIwr dasti tdnihertv  
nao o astvnscna nnrl tmba oteidpi ttfti eion nia  
beTtlies0 aod eura spnb utee y dnkmo
```

```
itstious acron ary, chom prondy at h) s hes cromindeathay,  
on bests s orion terinomicof id cthat gul asojedafincos  
fimis in. cut toffroutheail cthespend stesurd bl  
ike ibst tico nat ts mpsthe cen cecy
```

Markov Text Generation

- Order-zero Markov text generation
 - Use training text to generate text randomly
 - Choose character at random; don't use any characters to "predict" next character
 - In order-one, use one character to predict next

wa nitser nibsm epex uatcdfaaaaleohdi f roise e.tj
gthgb lutosth nid, loa qnednegee oIwr dasti tdnihertv
nao o astvnscna nnrl tmba oteidpi ttfti eion nia
beTtlies0 aod eura spnb utee y dnkmo

itstious acron ary, chom prondy at h) s hes cromindeathay,
on bests s orion terinomicof id cthat gul asojedafincos
fimis in. cut toffroutheail cthespend stesurd bl
ike ibst **tico** nat ts mpsthe cen cecy

Developing MarkovZero

- We'll think about behavior or methods first
 - That will help think about state or instance variables
- We need to set the training text
- We need to generate random text
- Could combine into one method, but in general making methods do one thing is a good idea

Training Text in MarkovZero

- Setting the training text to be used when generating random text
 - Might want to generate several random "texts" from the same training text

```
public class MarkovZero {  
    private String myText;  
  
    public void setTraining(String s){  
        myText = s;  
    }  
}
```

Training Text in MarkovZero

- Setting the training text to be used when generating random text
 - Might want to generate several random "texts" from the same training text
 - Store text in instance variable to set in one method and access in another

```
public class MarkovZero {  
    private String myText;  
  
    public void setTraining(String s){  
        myText = s;  
    }  
}
```

Training Text in MarkovZero

- Setting the training text to be used when generating random text
 - Might want to generate several random "texts" from the same training text
 - Store text in instance variable to set in one method and access in another

```
public class MarkovZero {  
    private String myText;  
  
    public void setTraining(String s){  
        myText = s;  
    }  
}
```


Generating Text at Random

- Choose random character from training text

```
public String getRandomText(int numChars){  
    if (myText == null) return "";  
  
    StringBuilder sb = new StringBuilder();  
    for(int k=0; k < numChars; k++) {  
        int index = myRandom.nextInt(myText.length());  
        sb.append(myText.charAt(index));  
    }  
    return sb.toString();  
}
```

Generating Text at Random

- Choose random character from training text
 - Use java.util.Random

```
public String getRandomText(int numChars){  
    if (myText == null) return "";  
  
    StringBuilder sb = new StringBuilder();  
    for(int k=0; k < numChars; k++) {  
        int index = myRandom.nextInt(myText.length());  
        sb.append(myText.charAt(index));  
    }  
    return sb.toString();  
}
```

Generating Text at Random

- Choose random character from training text
 - Use java.util.Random

```
public String getRandomText(int numChars){
    if (myText == null) return "";

    StringBuilder sb = new StringBuilder();
    for(int k=0; k < numChars; k++) {
        int index = myRandom.nextInt(myText.length());
        sb.append(myText.charAt(index));
    }
    return sb.toString();
}
```

Generating Text at Random

- Choose random character from training text
 - Use java.util.Random
 - Use StringBuilder to build random text

```
public String getRandomText(int numChars){  
    if (myText == null) return "";  
  
    StringBuilder sb = new StringBuilder();  
    for(int k=0; k < numChars; k++) {  
        int index = myRandom.nextInt(myText.length());  
        sb.append(myText.charAt(index));  
    }  
    return sb.toString();  
}
```


Generating Text at Random

- Choose random character from training text
 - Use java.util.Random
 - Use StringBuilder to build random text

```
public String getRandomText(int numChars){
    if (myText == null) return "";

    StringBuilder sb = new StringBuilder();
    for(int k=0; k < numChars; k++) {
        int index = myRandom.nextInt(myText.length());
        sb.append(myText.charAt(index));
    }
    return sb.toString();
}
```

Generating Text at Random

- Choose random character from training text
 - Use java.util.Random
 - Use StringBuilder to build random text

```
public String getRandomText(int numChars){  
    if (myText == null) return "";  
  
    StringBuilder sb = new StringBuilder();  
    for(int k=0; k < numChars; k++) {  
        int index = myRandom.nextInt(myText.length());  
        sb.append(myText.charAt(index));  
    }  
    return sb.toString();  
}
```

Constructors and Methods

- Constructors should initialize fields

```
public class MarkovZero {  
    private Random myRandom;  
    private String myText;  
    public MarkovZero() {  
        myRandom = new Random();  
    }  
    public void setRandom(int seed){  
        myRandom = new Random(seed);  
    }  
}
```

Constructors and Methods

- Constructors should initialize fields
 - Random number generator

```
public class MarkovZero {  
    private Random myRandom;  
    private String myText;  
    public MarkovZero() {  
        myRandom = new Random();  
    }  
    public void setRandom(int seed){  
        myRandom = new Random(seed);  
    }  
}
```


Constructors and Methods

- Constructors should initialize fields
 - Random number generator

```
public class MarkovZero {  
    private Random myRandom;  
    private String myText;  
    public MarkovZero() {  
        myRandom = new Random();  
    }  
    public void setRandom(int seed){  
        myRandom = new Random(seed);  
    }  
}
```

Constructors and Methods

- Constructors should initialize fields
 - Random number generator
 - Sometimes useful to set seed

```
public class MarkovZero {  
    private Random myRandom;  
    private String myText;  
    public MarkovZero() {  
        myRandom = new Random();  
    }  
    public void setRandom(int seed){  
        myRandom = new Random(seed);  
    }  
}
```

Constructors and Methods

- Constructors should initialize fields
 - Random number generator
 - Sometimes useful to set seed
- String that's training text

```
public class MarkovZero {  
    private Random myRandom;  
    private String myText;  
    public MarkovZero() {  
        myRandom = new Random();  
    }  
    public void setRandom(int seed){  
        myRandom = new Random(seed);  
    }  
}
```

Toward MarkovOne

- Can test MarkovZero with MarkovRunner
 - User selects training text file
 - Replaces '\n' with ' '
 - Repeatedly generates random text
- What changes with MarkovOne?
 - Use same methods, same state
 - Must change getRandomText
 - Use one character to predict next

MarkovOne

- One character predicts the next
 - From training text we see that 'a' follows 't' 12% of the time
 - We see 'y' follows 7%
- Don't create probabilities
 - Search for every 't'
 - Add next characters to list
 - 'a', 'e', 'a', 'r', 'a', 'e', 'y' ...

