

Lazy Literature

Introduction to Computer Programming

Dr. Paul Vrbik

November 16, 2018

Lazy Literature

Goal

Write a program that **consumes** text (like Alice in Wonderland) towards the goal of **generating** text in the style of that book.

Plan

We will break our work into two **phases**

- I. **Learn** – consume a text and store information in dictionaries.
- II. **Generate** – used the stored information to generate gibberish text in the style of the consumed text.

and write two versions over two lectures.

Learn

What information is pertinent in the following text?

Dan played ball.

Dan walked fast.

Dan said so.

How can we codify this information?

```
{'Dan': ['played', 'walked', 'fast'], 'played':['ball'], ... }
```

The following text

```
Did you get the sword from the old man on top of the
waterfall?
```

should produce the following dictionary

```
{'': ['Did'], 'Did': ['you'], 'you': ['get'],
 'get': ['the'], 'the': ['sword', 'old', 'waterfall?'],
 'sword': ['from'], 'from': ['the'], 'old': ['man'],
 'man': ['on'], 'on': ['top'], 'top': ['of'],
 'of': ['the']}
```

Implement

Question

Implement

```
make_dictionary(f: TextIO) -> Dict[str, List[str]]:
```

which returns a dictionary where the keys are words in `f` and the value for a key is the list of words that were found to follow that key.

```
mimic_text(word_dict: Dict[str, List[str]], num_words: int) -> str:
```

which based on the word patterns from the word dictionary, returns a string that mimics that text, and has `num_words` words.

Next Lecture

1. Develop 2.0 version.

Issues to Consider

What if the word dog were followed in the original text by barked ten times and food twice? We'd like to prefer to choose barked most of the time, since that's what was done by the original author. How can we do this?

If we start with ' ' every time, our "story" will always start with the first word of the story on which ours was based. How can we start at a random word?

Issues to Consider

We'll have to fix that `waterfall?` is not keyed in our dictionary.

What should we do when we hit the last word of the text (which has no follow-up words)? Its unlikely to happen to us in large texts, but it is still a bug and must be fixed!

We used only one word (the current word) to determine the word that should follow. But what if we used more context to decide what word to choose next? If we took more text into account, our follow-up words might be more reasonable. For example, we could use the most recent three words to decide the next word.