

ゆるふわ系自然言語処理

京都産業大学3回 宮島健太

とりあえず自己紹介タイム

まずは自分から

名前:宮島健太

所属:京産大 コン理CS 3回(C.A.C.)(多分21卒)

技術:C/C++, Python, Go,
競プロ, AI, サーバ, Git

趣味:ゲーム, アニメ, ラーメン

- ・ ほぼ毎日PUBGm
- ・ リゼロ, ジョジョ, SAO は神
- ・ 今期は [鬼滅の刃](#) と [Dr.Stone](#) が個人的におすすめ
- ・ 週1くらいで「あくた川」



@m_k0122b

その他

- ・ 今年のICPCで国内予選通った人
(先輩にキャリーしてもらった)
- ・ CyberAgentのインターン
- ・ 留年しそう。やだやだ。

大学が好きすぎてもう一年通っちゃおうかなwww

では、やっていきましょう！

今回のテーマ！

文章を生成する

どこで使われてるんだろう



チャットボットとかで
使われてる技術ですね。

チャットボットって何？

Chatbot (Chatterbot)

テキストや音声を通じて、
会話を自動的に行うプログラム

AI(人工知能)ではなく、人工無能と言われる。

チャットボットの種類

- ・ Eliza型（聞き役として相づちや会話の要約をする）
- ・ 選択肢型（決められたシナリオによって選択式で会話をする）
- ・ 辞書型（登録された単語とそれに対する応答をする）
- ・ ログ型（会話ログを利用して文脈に近しい応答をする）

代表的なチャットボット

- Siri
- IBM Watson
- Google Now
- Microsoft Cortana
- りんなちゃん

どうやってやるん？

機械学習

と

深層学習

それぞれ解説

(今回の場合)

- **機械学習(人工無能)**

特定のキーワードをデータベースとマッチングさせて文章を生成する。簡単。

- **深層学習(人工知能)**

ニューラルネットワークを使うことで、文章の意味を理解しながら文章を生成できる。難しい。

まあ、こんな感じです。

とりあえず…

頑張ろう！！！

本題に入る前に…

とりあえず脳死で 入れて欲しい物

コマンド(Bash)

- Python3

```
brew install python3
```

- Mecab

```
brew install mecab mecab-ipadic
```

- Janome

```
pip3 install janome
```

- Chainer

```
pip3 install chainer
```

- Requests

```
pip3 install requests
```

- BeautifulSoup

```
pip3 install beautifulsoup4
```

それぞれ解説

- Python3 … プログラミング言語
- Mecab … 形態素解析
- Janome … 形態素解析
- Chainer … 機械学習ライブラリ
- Requests … HTTPライブラリ
- BeautifulSoup … 静的webページの解析(HTML)
 - Chainerだけじゃなく、Scikit-learnやTensorFlowなどもある。
 - 動的Webページの解析(JS)には Selenium

これからの流れ

前半(Chapter1)

- ・機械学習で文章を作る

後半(Chapter2)

- ・スクレイピング
- ・深層学習でやってみる

ここまでで
質問ありますか？

じゃ、やっていきます。

Chapter1

機械学習で文章を作る

マルコフ連鎖を使います。

マルコフ連鎖とは？

確率過程の一種であるマルコフ過程のうち、取りうる状態が離散的なものをいう。

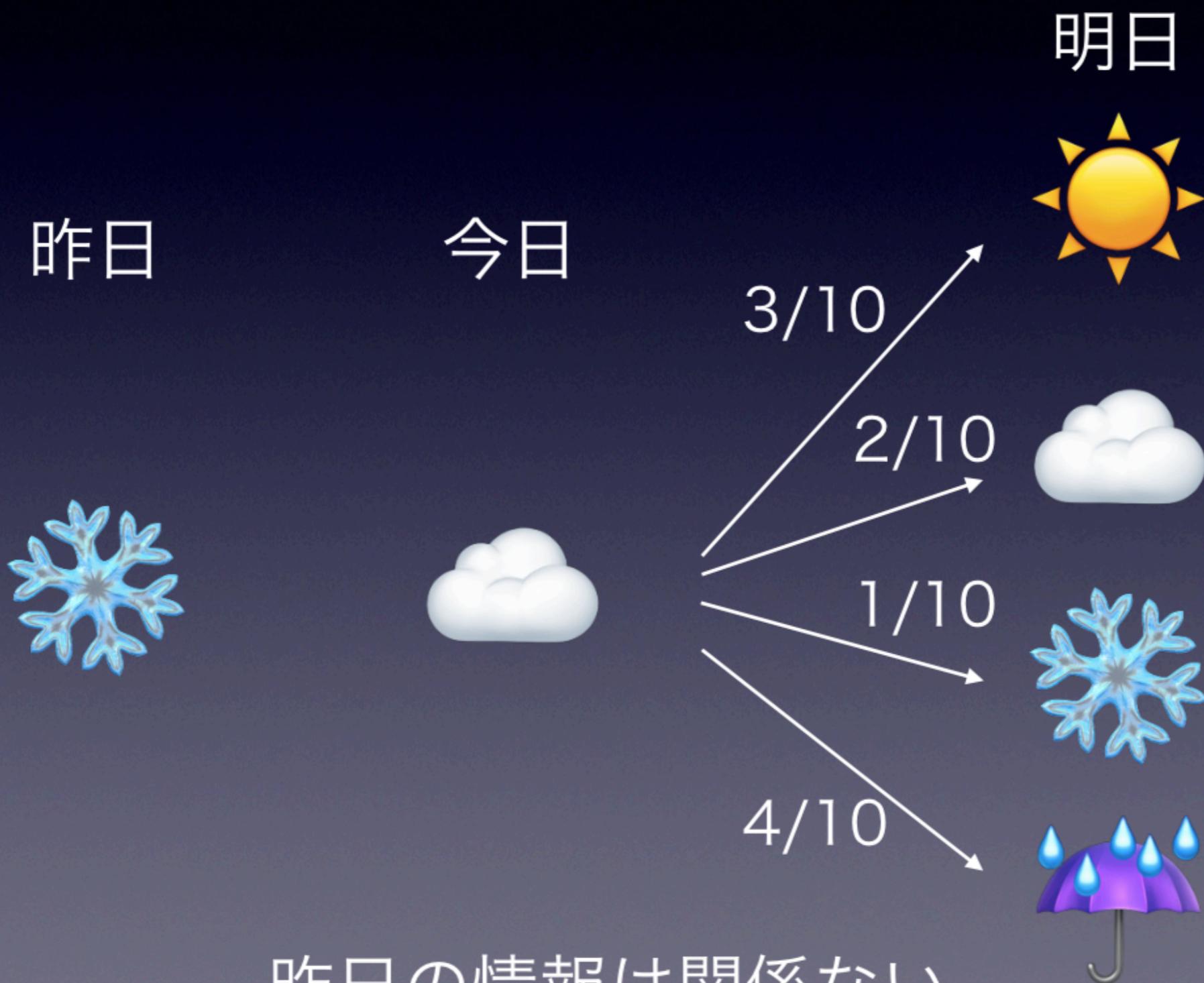
wikipediaから引用

は？

つまり

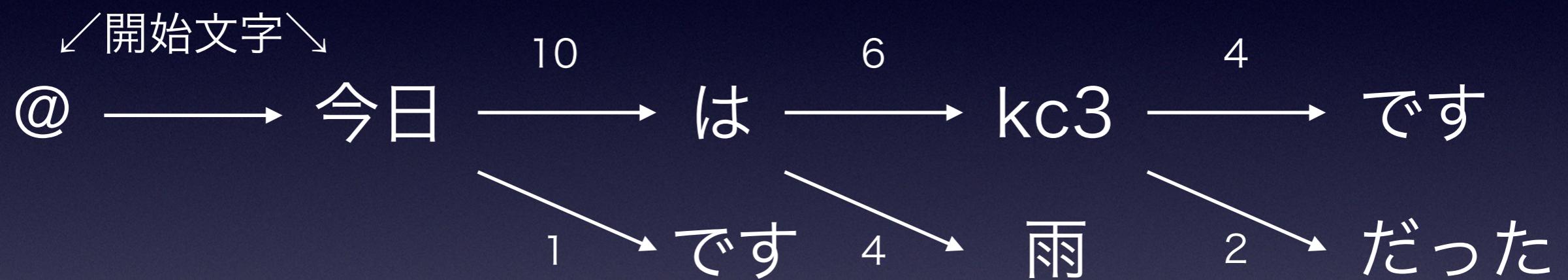
これから起こることは現在の値だけで決定され、
過去の挙動は無関係であること。

イメージ



なるほどな！

今回の場合だと

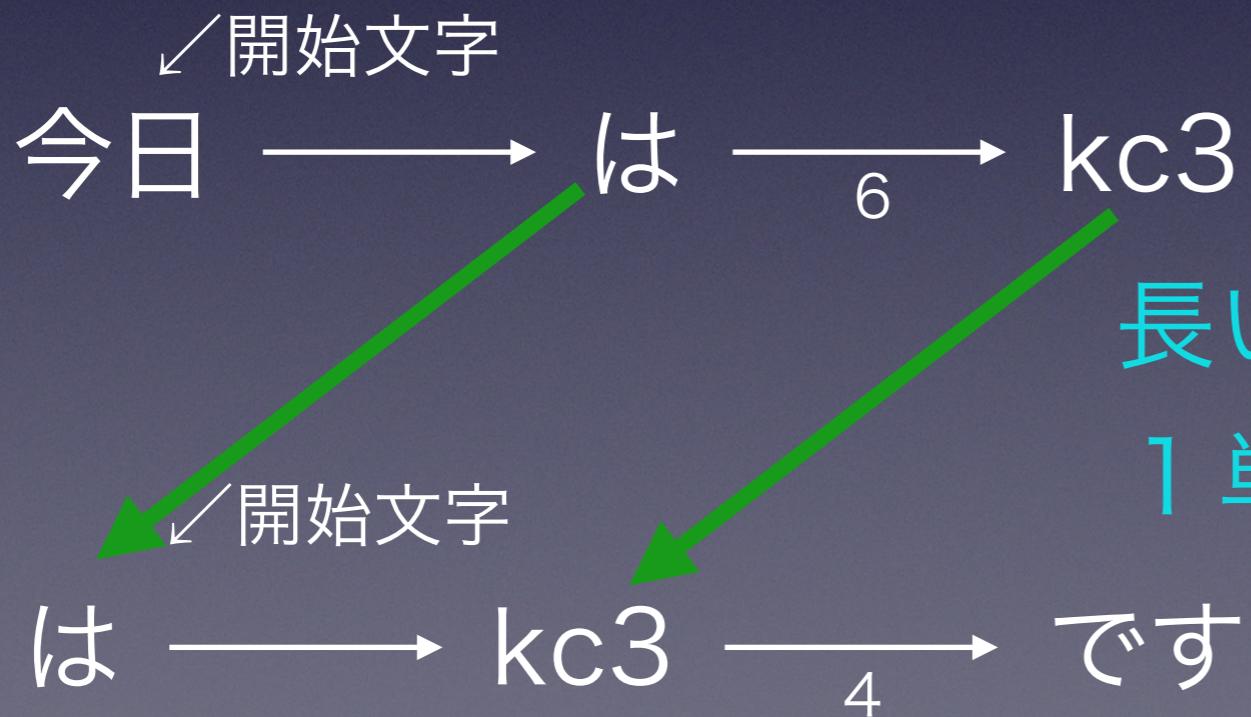


3単語ごとに開始文字が1つずれて、
次の1単語を選ぶ作業を繰り返します。

データベース(json)



短い文は開始文字が@



長い文は3単語のまとまりで
1単語ずつずらして分割

ほら、連鎖してるでしょ！？

では、実装してみましょう！

待て待て、
文章はどうやって
分割すればいいんだよ…

お、ナイス気づき！

形態素解析をしましょう！

形態素解析とは

文を品詞ごとに分割すること

こんな感じ

今日は待ちに待ったkc3です！

今日	名詞, 動詞可能, *, *, *, *, 今日, キヨウ, キヨー
は	助詞, 係助詞, *, *, *, *, は, ハ, ワ
待ち	名詞, 一般, *, *, *, *, 待ち, マチ, マチ
に	助詞, 格助詞, 一般, *, *, *, に, ニ, ニ
待つ	動詞, 自立, *, *, 五段・タ行, 運用タ接続, 待つ, マツ, マツ
た	助動詞, *, *, *, 特殊・タ, 基本形, た, タ, タ
kc	名詞, 一般, *, *, *, *, *
3	名詞, 数, *, *, *, *, *
です	助動詞, *, *, *, 特殊・デス, 基本形, です, デス, デス
！	記号, 一般, *, *, *, !, !, !

EOS

これをやってくれるのが…

JanomeとMeCab !

正直どっちでもいいけど、
今回はJanomeを使つていこうと思うんだ。

やっていきましょう！

コードをCloneしてね

https://github.com/shadowlink0122/kc3_NLP

The screenshot shows a GitHub repository page for 'kc3_NLP'. At the top, there are buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find File', and a green 'Clone or download' button. Below this, there's a list of files: 'shadowlink0122 Initial commit' (image icon), 'README.md' (text icon), and another 'README.md' (text icon). To the right, there's a 'Clone with HTTPS' section with a URL field containing 'https://github.com/shadowlink0122/kc3_NLP'. Below it are 'Open in Desktop' and 'Download ZIP' buttons. A large red arrow points from the bottom right towards the 'Clone with HTTPS' URL field, which is circled in red.

任意のディレクトリで '\$ git clone URL'

作業内容は簡単！

Chapter1/cgi-bin/botengine.py

このファイルの中の **TASK** の条件に従い、
“#” の部分を完成させよう！

README.mdにも書いてあります

では始めちゃってください！

あと5分くらいです！

終了～！

ちょっと難しかったですよね…

ans_botengine.py

追加したので参考程度にどうぞ

git pull してね

ちなみに、LINE Botにするとこんな感じになります



Chapter 1 は
これでおしまい！

Chapter2

深層学習とスクレイピング

RNN(LSTM)を使います。

深層学習初心者にはちょっときついかも…?

RNNとは？

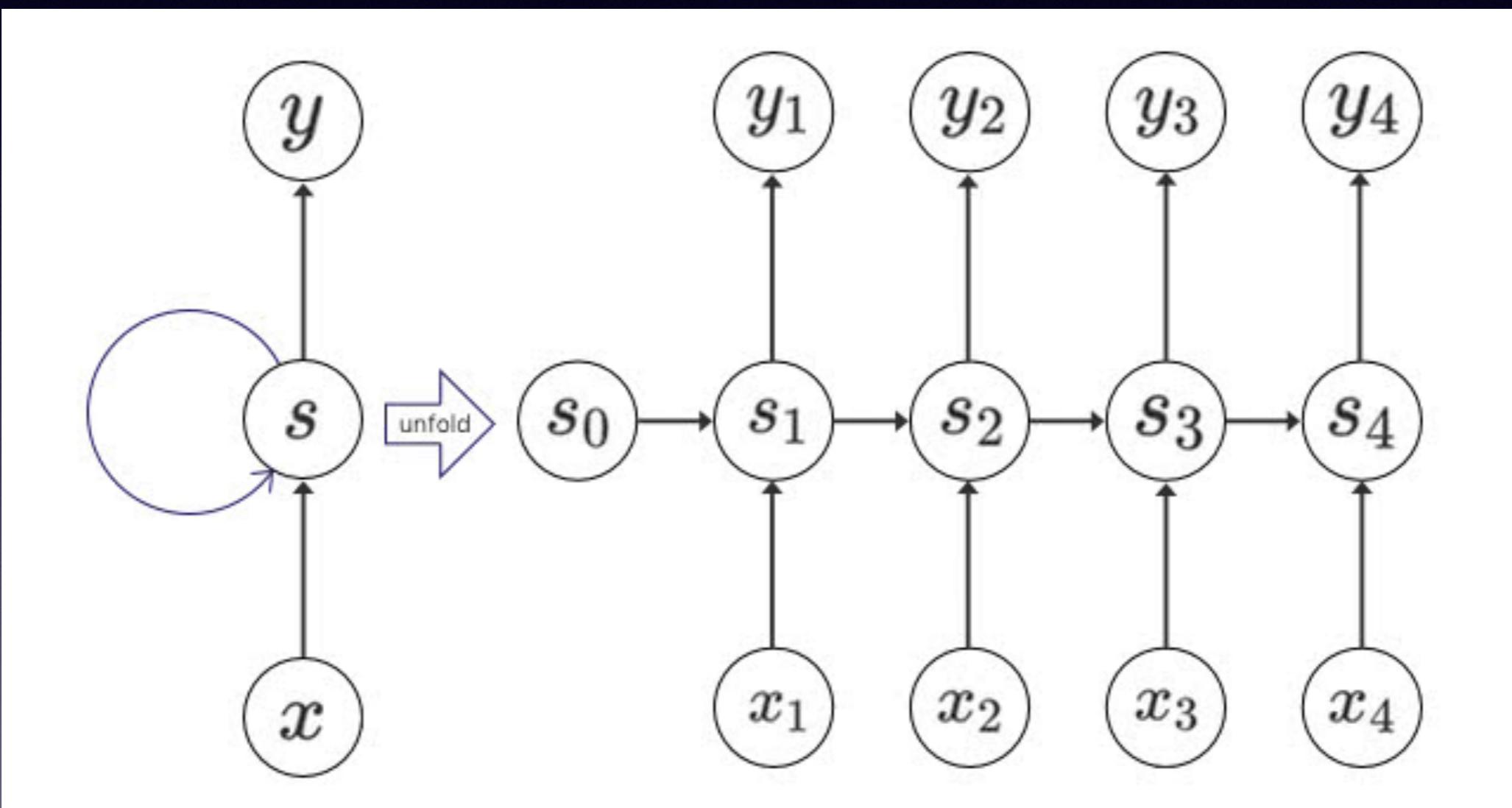
RNN(Recurrent Neural Network)

再帰型ニューラルネットワークともいう。

NNの出力を、次のNNの入力に使用する。

未来の予測などによく使われる。

RNNのイメージ



では、LSTMとは？

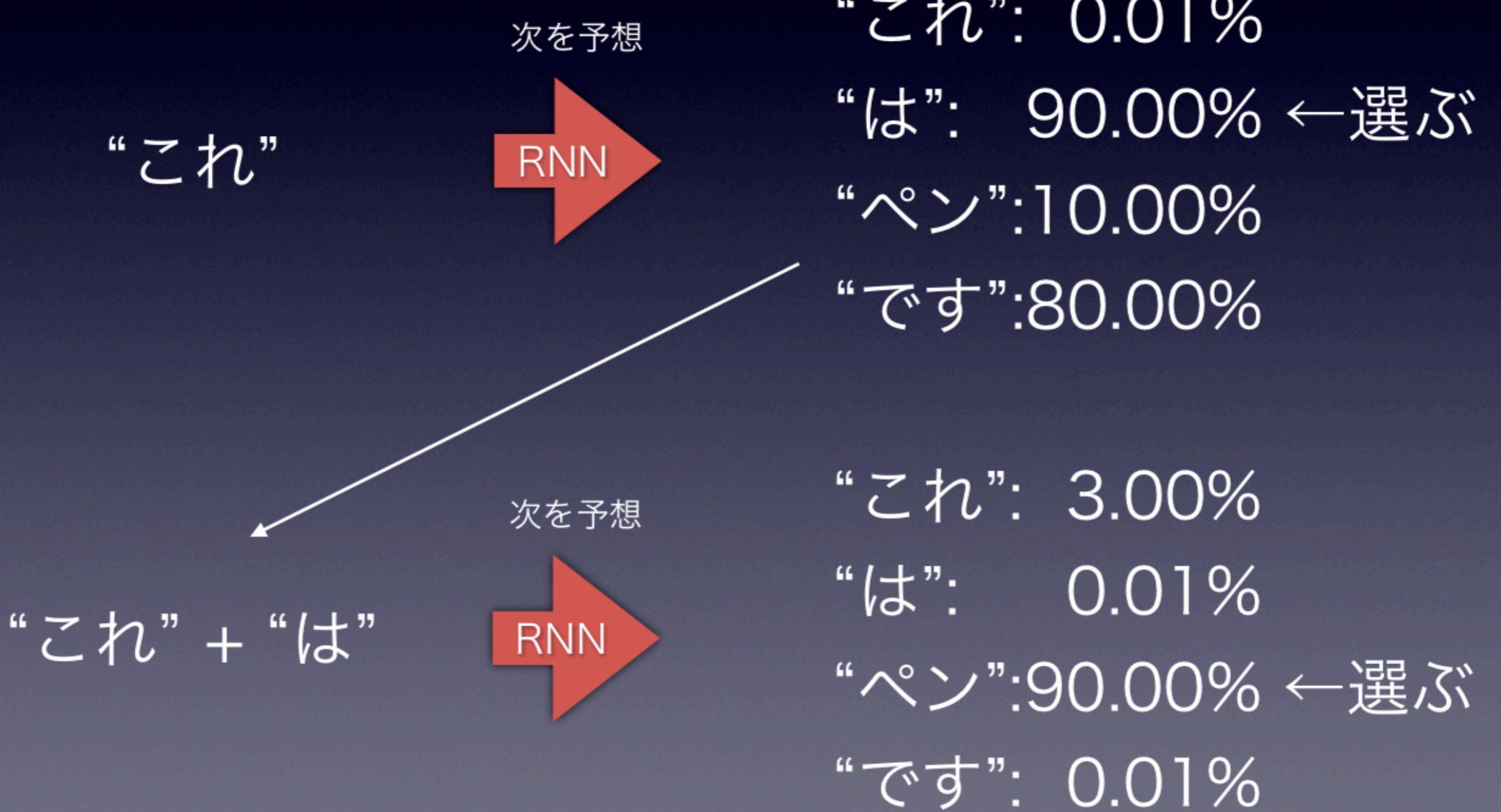
LSTM(Long Short Term Memory)

RNNから派生したNNで、勾配消失問題を
解決したモデル。

勾配消失問題

NNの層が深いと損失関数のパラメータ勾配が0や
無限大に発散します。そうすると、誤差逆伝播が
上手くいかなくなり、古い情報を忘れてしまうこと。

LSTMの使い方



ここまでで、
「何もわかんないぜ」
って人はいますか？

※ ここら辺からだんだん
説明が雑になっていくので
わからないことがあつたら
遠慮なく聞きましょう！

(答えれるとは言ってない)

学習のやり方は
わかったと思うので

学習させるために必要な
データを集めたいと思います！

スクレイピング

chapter2/scraping/scrape.py

今回はアリス物語シリーズのあらすじの文章を
wikipediaから集めたいと思います。

そして、これを学習データ兼生成する文章の土台とします。

実行してみよう！

chapter2/

ここで ‘\$ make scrape’

すると！

chapter2/corpus/raw.txt

が生成される！

さらにさらに！

文章を品詞ごとにスペースで区切っちゃおう！

‘\$ make wa’

俺は人間だ → 俺_は_人間_だ
こうなる。

ある程度

下準備はできましたけどね、

まだこれNNの学習データとしてふさわしくないんよね。
単語、全部数字に置き換えちゃいましょうか。

数字に置き換えるのは簡単！

出現した単語に番号を振っていく。
あとはそれを置き換える。

たとえ 火の中 水の中 土の中 森の中



1, 2,3,4, 5,3,4, 6,3,4, 7,3,4,

ってことで

ここでまた TASK のお時間です！

Chapter2/corpus/generate.pyを編集して
正しい学習データを作成しましょう！

\$ make data

TASK終了！

ans_generate.pyをあげたので参考程度にどうぞ

git pull

では、学習させて
いきましょう。

ここからはずっと
デモになります

学習中暇なので

文章生成のアルゴリズムについて

木探索

その中でもビームサーチを使う。

結果はこんな感じ

0.00018694038 キャロル自筆の原本は現在大英博物館に収蔵されている
0.00018608986 アリスがそれを飲むと、身長が約に縮んだ
3.3501434e-05 アリスはどうやら夢を見ていた
2.7118393e-05 アリスがそれを食べると、今度は身体が大きくなりすぎてしまう
2.1547747e-05 アリスは、陪審員の動物たちに混じって裁判を見物する
1.5762826e-05 アリスはどうやら夢を見ていたらしい
1.22637775e-05 『パブリッシャー・サーキュラー』
9.561816e-06 またパロディ詩
6.175095e-06 また児童小説
3.0908795e-06 前作同様
2.7796455e-06 また児童

もっと精度を上げるには

- ・学習データの数を増やそう
- ・マルコフ連鎖とLSTMを組み合わせると良い

かもしれない

とりあえず今回は
こんな感じです。

ここでみなさんに謝罪を

全然ゆるふわじゃなくて
ごめんなさい



でも、そういう日もあるわな！ww

みなさんも、
自分で集めたデータで学習させてみたり
何かwebサービスを作ってみたり
面白いチャットボットを作ってみてほしい！

以上で
ゴリゴリ自然言語処理の講座を
終わります！