

マルコフ連鎖を使った文章生成

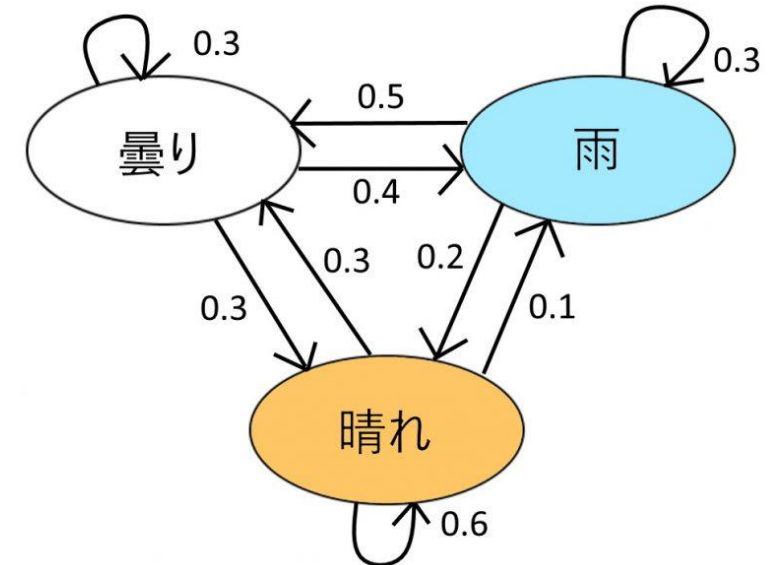
2-3-27 小泉海斗

マルコフ連鎖とは

簡単にいうと…

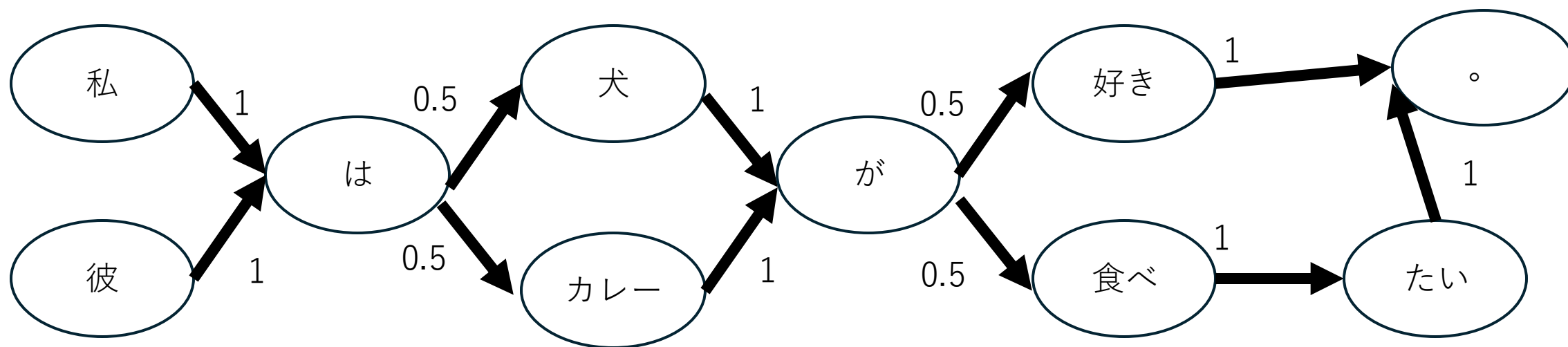
「現在の状態のみが次の状態に影響を与え、過去の状態には依存しない」

これを文章生成に活用する

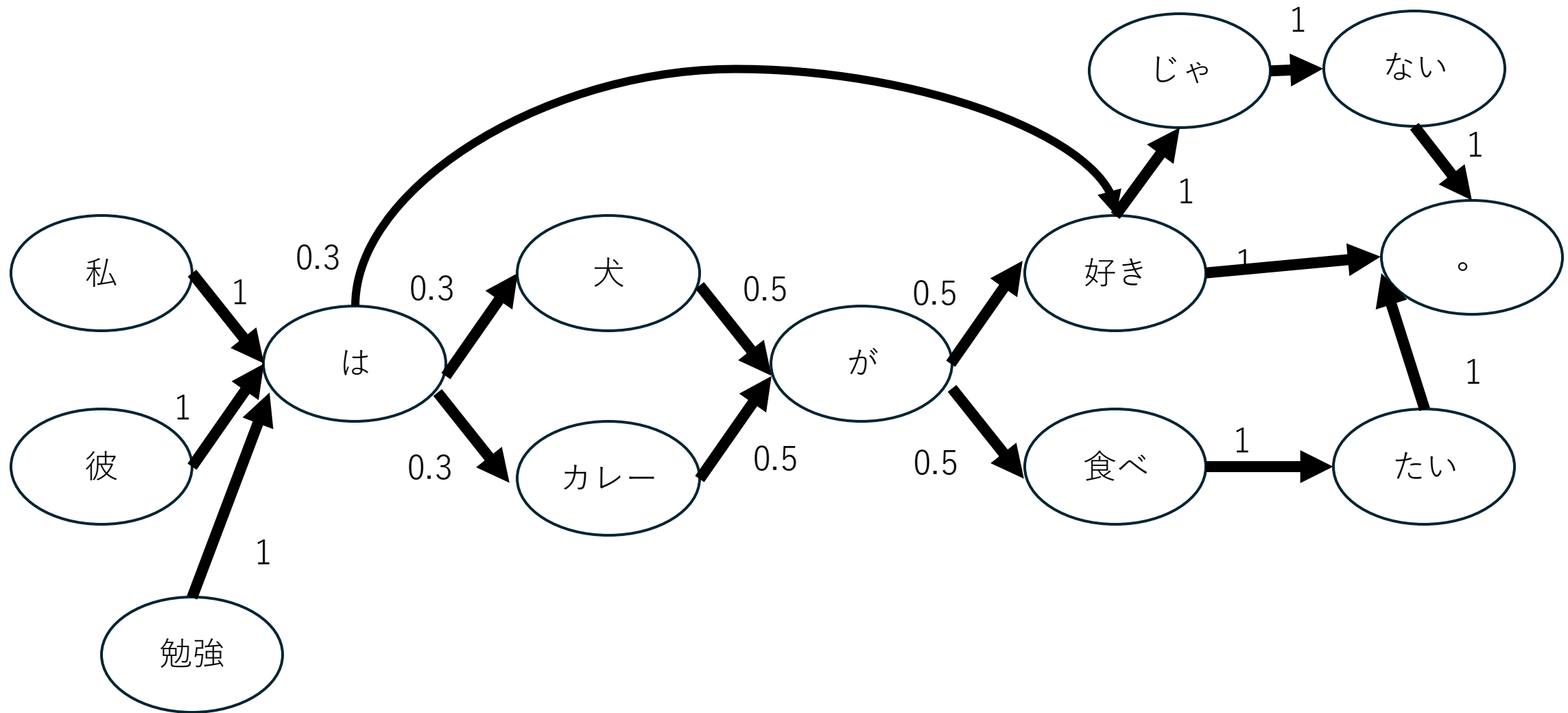


参照:<https://math-exploration.com/archives/212>

「私は犬が好き。」と「彼はカレーが食べたい。」という2つの文章をマルコフ連鎖にかけると



さらに「勉強は好きじゃない。」という文章を加えると



松岡修造の名言を生成してみよう

BoSとEoSについて

BoS(Begin of Sentence)とEoS(End of Sentence)とは文の始まりと文の終わりを示すものです。マルコフモデルでは、次の単語が前の単語に依存しているため、文の開始と終了を明示的に管理することで、文意が成り立つようにしています。BoSがあることで、どこから文章を始めるかが明確になり、EOSがあることで、生成された文章が自然に終了します。

まずはスクレイピング

スクレイピングを用いてサイトからstrongタグに収納されている松岡修造の名言を取得する

```
import requests
from bs4 import BeautifulSoup
```

```
# URL
```

```
url = 'https://www.sidebizz.net/blog/business/shuzo-matsuoka/'
```

```
# Webページの内容を取得
```

```
response = requests.get(url)
page_content = BeautifulSoup(response.text,
                              'html.parser')
```

```
# <strong>タグの内容を抽出
```

```
strong_tags = page_content.find_all('strong')
text_parts = []
for strong_tag in strong_tags:
    text_parts.append(strong_tag.text)
```

```
<div class="sc_frame" style="background-color:#fff;border-color:#2ac113">
  <p>
    <strong>ナイスボレー、修造！</strong> == $0
  </p>
</div>
</div>
```

次に取得したデータの加工

単語ごとに分け、リストに格納された複数の文字列を一つの大きな文字列に結合して形態要素解析

```
import MeCab
```

```
mecab = MeCab.Tagger('-Owakati')
```

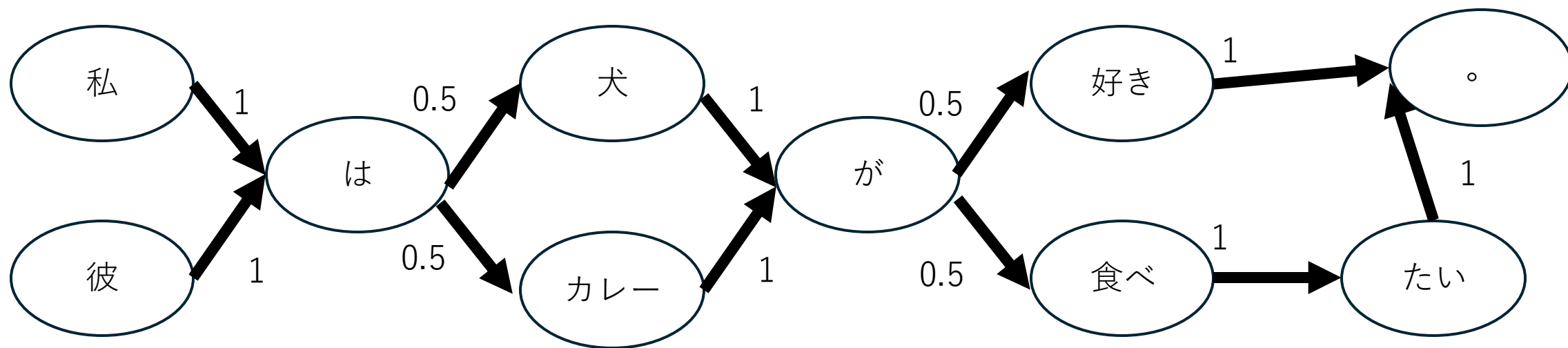
```
input_text = " ".join(text_parts)
input_text = mecab.parse(input_text).replace('。', '。')
input_text = input_text.replace('!', '!').rstrip()
```

```
In [63]: print(input_text)
ナイス ボレー、 修造！ 緊張してきた。
よっしゃー！ ヤッター！ 俺の脳、いまブルブル震えて喜ぼうとしているよ。
何万球打ってきたんだ、思い出せ！ わがままではなく、あるがままに。
自分を創るのは自分だ！ 僕はただ明るいだけ。
そして、神経質なところがある。
でも、それが僕だ！ ベストを尽くすだけでは勝てない。
僕は勝ちにいく。
OK！ ナイス TRY！ 崖っぷちありがとう最高だ！ そうだ、僕は心から本当にテニスが大好きなんだ！ ネガティブになったら、心の中でストップ。
プレッシャーを感じられることは幸せなことだ。
今の僕には勢いがある。
苦しいか？ 修造、笑え！ 迷ったら負ける！ 自分を信じる、決断しろ！ 自分を好きになれ！ 悔しがればいい、泣けばいい、喜べばいい。
```

マルコフモデルの作成

先ほど取得した修造の名言の文章ごとにループをかけて、その文章ごとに単語で分け、リストに単語を追加します。

```
def create_markov_model(input_text):  
    markov_model = {}  
    sentences = input_text.split('\n')  
    for sentence in sentences:  
        words = sentence.split(' ')  
        for word in words:  
            if previous_word in markov_model:  
                markov_model[previous_word].append(word)  
            else:  
                markov_model[previous_word] = [word]  
            previous_word = word  
    return markov_model
```



ランダムな文章を作成

BoSからランダムに単語を取得し、現在の単語に続く単語をランダムに選び、文章に追加し続けます

```
def generate_random_sentence(markov_model, min_length, max_length):
    eos_marks = '。 ! ?'
    start_words = markov_model['BoS']
    while True:
        current_word = random.choice(start_words)
        result_sentence = current_word
        while current_word not in eos_marks and len(result_sentence) < max_length:
            current_word = random.choice(markov_model[current_word])
            result_sentence += current_word

        if len(result_sentence) >= min_length:
            return result_sentence
```

結果(文字数制限なし)

- 僕はポイズンだ！
- 夢を言われてイライラしたらガッツポーズ。
- それが変わればいい、泣けば、全部人たちをピンチになったら、なんで力をピンチには必ず来る！
- 今の僕は勝ちにはお前の最後まで、いまブルブル震えているんだ！
- プレッシャーを言われて喜ぼうよ！

結果(文字数制限10~50)

- 不平・不満はタートルだよ！
- 君が次に叩く一回では、決断しろ！
- ナイスボレー、心の中でやめん坊、ばんざーい！
- 大事なことは幸せなことに勝てない。それが次に叩く一回でやめんだ、なんで力を好きになったら、頑張らなきゃいけないんだ！
- 何度でも、なんとなくフィニッシュだ！

考察

- 修造っぽくないこというときがある
- 元の文章が短いから元の文章がそのまま出力されることがあった
「ナイストライ！」などマルコフ連鎖での分岐がそもそも少ないやつ
- 元の文章の量が少ないので、生成される文章が単純だったが、生成される文字数の下限を変更すると文章の傾向が大きく変わって面白いと思った
- 名言系はマルコフ連鎖での文章生成に向いてなさそう