

そうえり もぷもぷ

soweli Mopumopu

- Twitter 上のトキポナボット

▷ @soweli_mopumopu



soweli Mopumopu

- 15 分おきに何か言葉を発します
- 会話もできます たのしいね



なんの話をするの

- soweli Mopumopu はどうやって動いてるのか？
 - ▷ 実はすこし高度なことをしている
- もぷもぷの技術について説明します

トキポナとは

- みんなが知ってるミニマル言語
 - ▷ 語彙数: 120 語
 - ▷ 習得がかんたん たのしい!
 - ▷ できる人が割と多い たのしい!
- 知らない人のために
 - ▷ 初級トキポナ文法簡介
 - ▷ 2 分 40 秒で世界一簡単な言語を紹介して伝授する
 - ▷ トキポナレッスン1「トキポナってなに」
 - ▷ jan Pije's lessons (注)
 - ▷ 【日本語訳】toki pona li toki pona - トキポナソング

トキポナをモデル化する

- トキポナをコンピュータで扱いたい!
 - ▷ コンピュータには言語がわからぬ...
- “文” を数理モデル化して, “文の良さ” を評価しよう!

確率的言語

- 文を「その文が起きる確率」として捉える
 - ▷ 言語を確率 P によって, $P(\text{文})$ で表す
 - ▷ $P(\text{toki pona li toki pona!})$ とか
- なにかうれしいの？

確率的言語

- 次の 2 つの文 A, B はどちらが “良い” 文だろうか？
 - ▷ 文 A : toki pona li toki pona.
 - ▷ 文 B : a akesi ala alasa ale.
 - 明らかに A のほうがよい
 - $P(A) > P(B)$ となるはず
- 確率で言語を近似することができれば, 文法・意味的に “良い文”・“悪い文” を定性的に評価できる
 - ▷ チャットボットが作れる
 - ▷ 機械翻訳などもこの考えを元に作られている

言語モデル

- 文に確率を与えるモデルのこと
- つまり,
 - ▷ 文 $w_1^n = w_1 w_2 \cdots w_n$ を入力して,
 - ▷ $P(w_1^n)$ を計算・出力する関数のようなもの
- どのように計算する?
 - ▷ そもそも入力の長さが文ごとに違うしつらい...

分解する

- 確率論の乗法定理を用いて, $P(w_1^n)$ を分解

$$\begin{aligned}P(w_1^n) &= P(w_1, w_2, \dots, w_n) \\&= P(w_1)P(w_2, w_3, \dots, w_n|w_1) \\&= P(w_1)P(w_2|w_1)P(w_3, w_4, \dots, w_n|w_1, w_2) \\&= P(w_1)P(w_2|w_1)P(w_3|w_1^2)P(w_4^n|w_1^3) \\&= \prod_{i=1}^n P(w_i|w_1^{i-1})\end{aligned}$$

- 1 単語ずつ計算してかければ文の確率になる!

$P(w_i | w_1^{i-1})$ を計算する

- 1 番目から $i - 1$ 番目の単語がわかってて, i 番目の単語が起こる確率
- $P(\text{li} | \text{toki pona}) > P(\text{wile} | \text{toki pona})$
- Mopumopu では, ニューラルネットを用いている
 - ▶ ニューラル言語モデルっていう

Transformer モデル

- Mopumopu で採用したニューラルモデル
 - ▷ 2017 年に Google の人たちが考案した
 - Google 翻訳のなかみもこれ
- 次からのスライドで細かいことを説明します

Transformer モデル

どのように学習するのか？

- モデルの確率分布を言語の確率分布に近づける
 - ▷ モデルと言語の相対エントロピーを小さくする

言語のエントロピー

相対エントロピー

クロスエントロピー

確率の勾配降下法

Adam アルゴリズム

自己回帰生成

ランダムサンプリングは？

top-p サンプルリング

PyTorch を使って学習・推論をする

言語モデルでチャットボットをつくる

- ちょっと無理がある
- すこし怪しいことをしている

Twitter API を使う
