

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” を数理モデル化しよう！
 - ▷ 確率的言語というのを考えます

確率的言語

- 「その文が起きる確率」というのを考える
 - ▷ 文の確率 $P(\text{文})$ を考える
 - ▷ $P(\text{toki pona li toki pona!})$ とか
- なにがうれしいの？

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
 - ▷ 確率的言語は文法を扱えるかも？

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 : ma tomo Tokijo li lon ma Nijon.
 - ▷ 文 B : ma tomo Lanten li lon ma Tosi.
 - A は正しいが, B は間違い
 - $P(A) > P(B)$ となるはず
 - ▷ 確率的言語は意味や常識を扱えるかも？

確率的言語

- 確率で言語を近似することができれば, 文法や意味を捉えたモデルを作成できるのでは?
 - ▷ “良い文”・“悪い文” を定性的に評価できる
 - チャットボットも作れる
 - ▷ もっと広く言語処理の様々な場面で使われている
 - 機械翻訳などもこの考えを元に作られている
- 文の確率をどうやって計算するの?

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

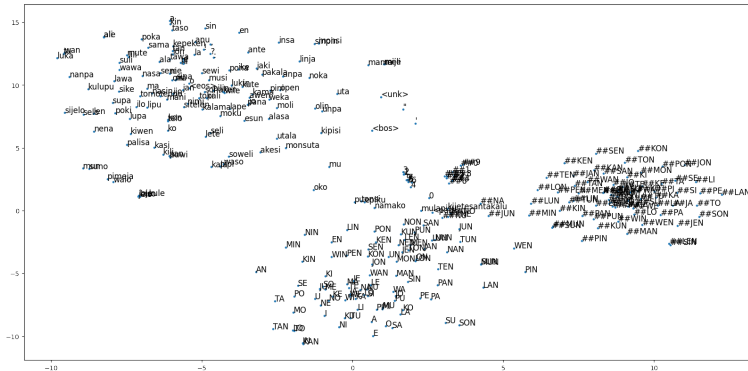
- Mopumopu で採用したニューラルモデル
 - ▷ 2017 年に Google の人たちが考案した
 - Google 翻訳のなかみもこれ
- 次からのスライドで細かいことを説明します
 - ▷ 注: ここで説明するのは、一般によく言われている Transformer Encoder-Decoder モデルのことではなく、Transformer 言語モデルのことです。

Transformer モデル

- Transformer モデルは次の部分からなっている
 - ▷ 単語 Embedding 層
 - ▷ 位置 Embedding 層
 - ▷ Self-attention 層
 - ▷ Feed-forward 層

單語 Embedding 層

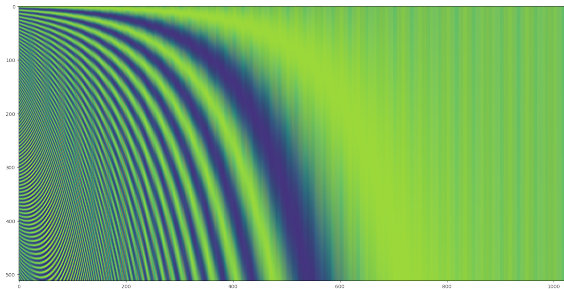
- 単語をベクトルに変換する
 - ▷ Mopumopu では 1024 次元
 - ▷ 逆(ベクトル-> 単語)もできる



位置 Embedding 層 (難しい)

- 位置(何番目の単語か)をベクトルで表す
 - ▷ 三角関数の回転を利用している
 - ▷ t 番目の i 次元が

$$\mathbf{p}_t^{(i)} = \begin{cases} \sin(\omega_k t) & (i = 2k) \\ \cos(\omega_k t) & (i = 2k + 1) \end{cases}$$

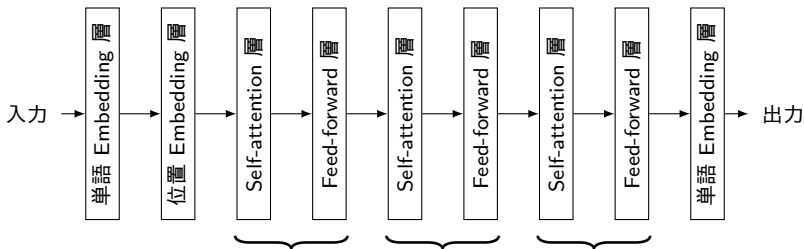


Self-attention 層 (難しい)

Feed-forward 層

Transformer モデル

- それをこんな感じで組み合わせると



- なんと言語モデルができる

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

確率的勾配降下法

- クロスエントロピーを微分した方向の逆にモデルのパラメータを動かしていけば、小さくなっていくのでは？
 - ▷ 「標高が低い方に進めば下山できるのでは？」
遭難するのでやってはいけない
 - ▷ 言語モデルが学習できる！
- ほんとにこんなんでも学習できるのか？
 - ▷ 厳しいので工夫する

Adam アルゴリズム

- a

(SDG の式に 2 つ修正点があって, 1 次の向きをおさえてるのと, 2 次の大きさをおさえてるのがあるんだって話をかけばいい)

自己回帰生成

ランダムサンプリング

ランダムサンプリングは厳しい

top-p サンプルリング

nymwa/ponapt

実際に動かす

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

おわりに
