自然言語処理の基礎

単語埋め込みを学ぶ

Sakarush (@sakaiine) / Ishii Akira

単語の表現方法

単語をどう表現するか

- “You” という単語はどのように表現すれば良い？
- 例えば “59 6F 75” (ASCIIコード) は “You”
- もっと “You” の特徴を捉えた表現が良い

→　実ベクトルで表現しようぜ！

- これを単語の埋め込みという
  単語ベクトル / 埋め込みベクトル / 埋め込み

人間の手で作ろう（素性ベクトル）

- 丹精込めて人手でベクトルを設計する
- 素性関数：ある条件を満たす時に1を
  それ以外の時は0を返す
- V＝（名詞？, 形容詞？, 代名詞？,…, ）

素性ベクトルの例：ワンホットベクトル

- 各単語に1成分を対応させる素性関数を考える
- V = (dog?, cat?, apple?, drink?, …,)
  “dog”なら　V = (1,0,0,0,…)
  “cat” なら　V = (0,1,0,0,…)
- この他にも色んな素性関数が…
  - V = (animal?, food?, machine?, …)
- 人手で設計するにはあまりにしんどい　→　NNへ

Word2Vec

基本的な考え方：筋トレ

- 太ももを鍛えるためにサイクリングをする
  → サイクリングは真の目的ではない
- 良いベクトルを得るためにタスクを解く
  → このタスクを解くのは真の目的ではない
- このときの学習用タスクに求められる性質
  - データが沢山準備出来ること
  - いい感じのベクトル生成器が得られること

分布仮説

- 単語の意味は、その周辺の単語 (=文脈) によって定まる

“You shall know a word by the company it keeps”

例)

I drink beer. / I drink wine.
I guzzle beer. / I guzzle beer.

- 分布仮説を元にしたタスクを用意して、筋トレする

CBoWモデル

- 穴埋め問題を解く
- 単語の意味が、文脈によって定まる
  → 文脈から空欄の単語を予測出来るだろう

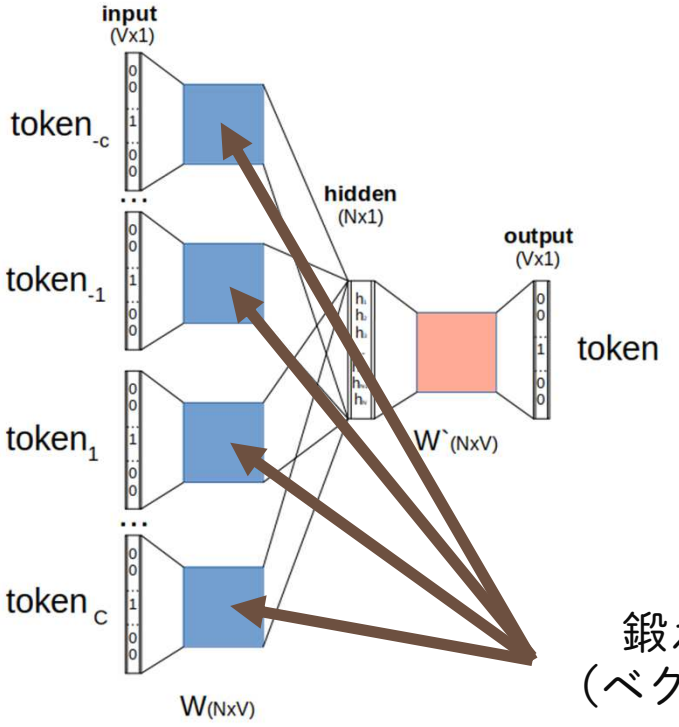
例）

“I < ??? > beer.”

「もうあなたとは <???> するわ！」

- 入力：周辺単語 (=文脈)
  出力：対象の単語の予測結果

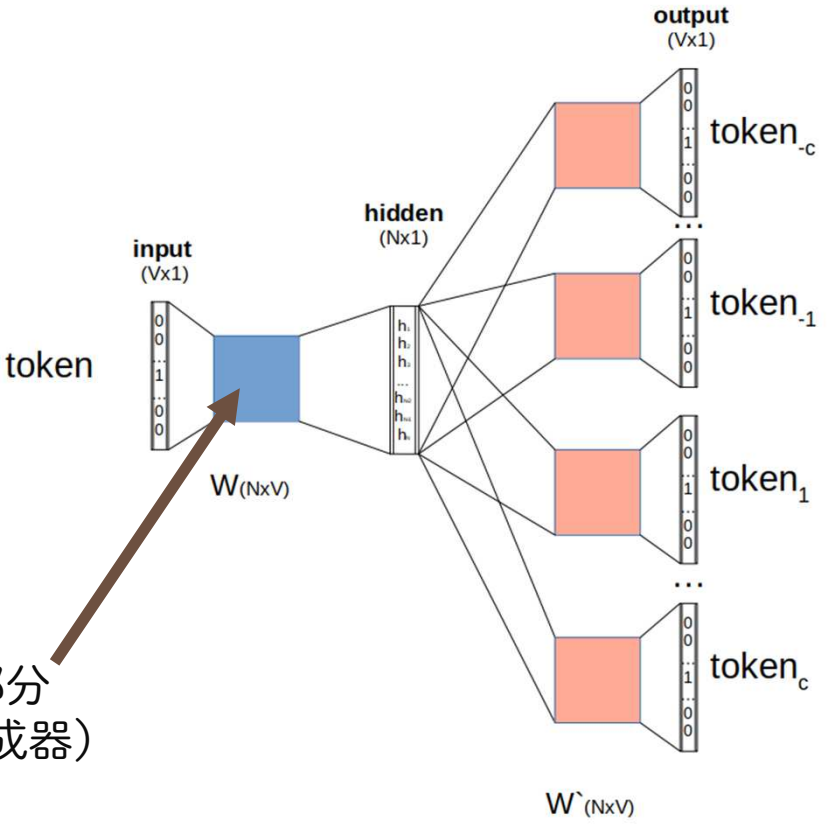
CBoWモデル



鍛えたい部分
（ベクトル生成器）

SkipGramモデル

- 入力：単語
  出力：周辺単語（文脈）
- CBoWの逆
- あくまで欲しいのは
  ベクトル生成器の部分



得られたベクトルの性質

- 意味の演算が出来る

$$\mathrm{King} - \mathrm{Man} + \mathrm{Woman} = \mathrm{Queen}$$

- 類似度の計算が出来る
  - 2つの単語のベクトルのcos類似度を計算する
  - 似た意味の単語のベクトルは似たベクトルとなる

限界・課題

- 同じ単語は同じベクトルになる

→ 文脈によって違うベクトルが得られるモデル
（BERT、ELMo）

→ 出現する位置によって違うベクトルが得られる
（位置埋め込み）