SPRINT21

自然言語処理入門

自然言語処理とは

私たち人間が日常的に用いている日本語やフランス語、中国語のような言語は **自然言語** と呼ばれ、プログラミング言語や記号論理学、エスペラント語のように開発された言語（**人工言語**）と区別することができます。

自然言語処理（NLP, Natural Language Processing） とは、このような **自然言語** をコンピュータに処理させる技術のことです。Sprin21ではその中でも、機械学習の入力として自然言語を用いることを考えていきます。

自然言語を取り扱う機械学習モデルとは、どのようなモデルでしょうか？

自然言語処理で用いられる機械学習モデル：

RNN（Sprint22で登場）

Recurrent Neural Networks（RNN）とは

主にシーケンスデータの分析（固定の順序で提供される配列データの分析）に使用されるディープラーニングモデルの一種。日本語では、再帰型ニューラルネットワークと呼ばれます。

このRNNをベースとしたモデルは、自然言語処理に限らず、

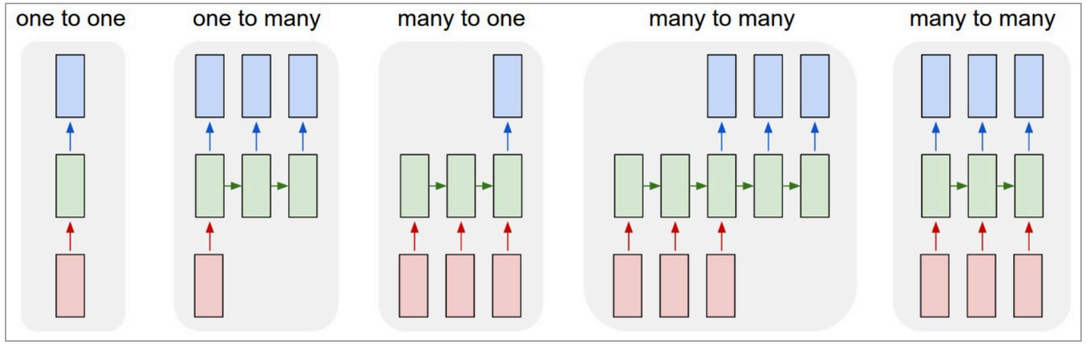
以下のようなタスクに適用されている。

適用タスク例：

言語生成、機械翻訳、音楽生成、時系列予測、画像キャプショニングなど

タスクによって入力ベクトルと出力ベクトルの数を変えることができる

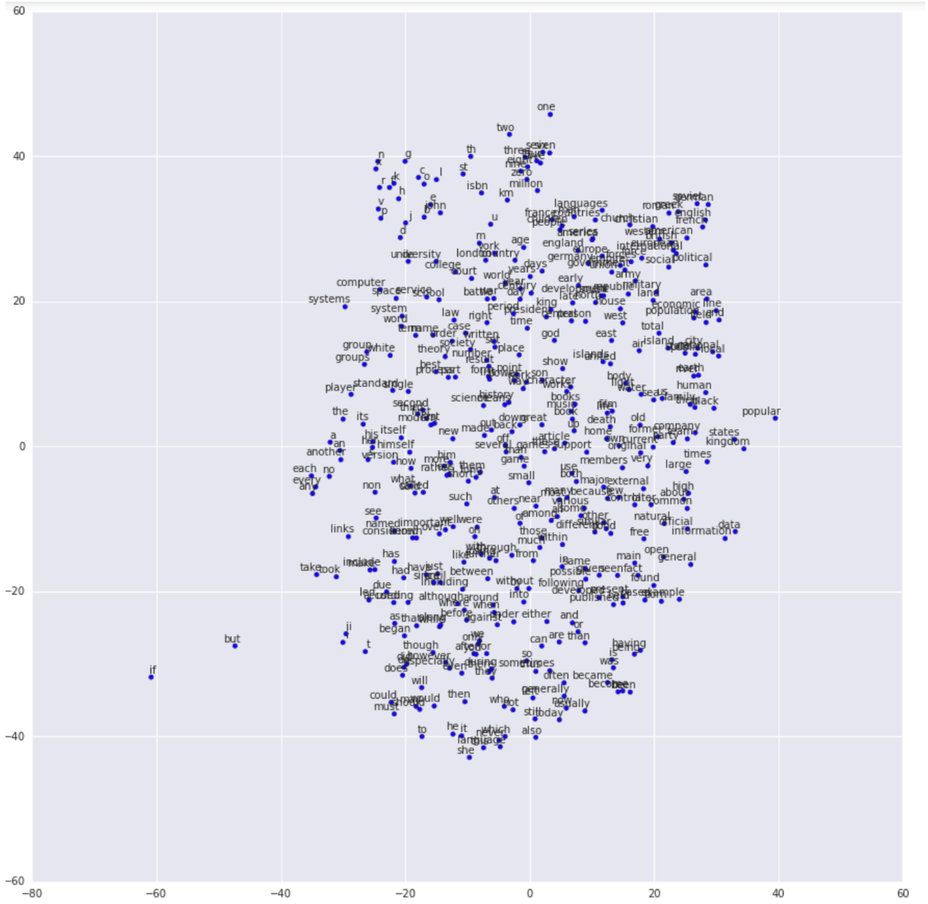
http://karpathy.github.io/2015/05/21/rnn-effectiveness/



自然言語処理の入力には
どんなデータを使いますか？

機械が処理しやすいように**文字列（記号）をベクトル化したデータ**

言語モデルにおける入力データには、
「分散表現」（あるいは **単語埋め込み** Word Embedding ）
と呼ばれる「ベクトル表現」が用いられます。



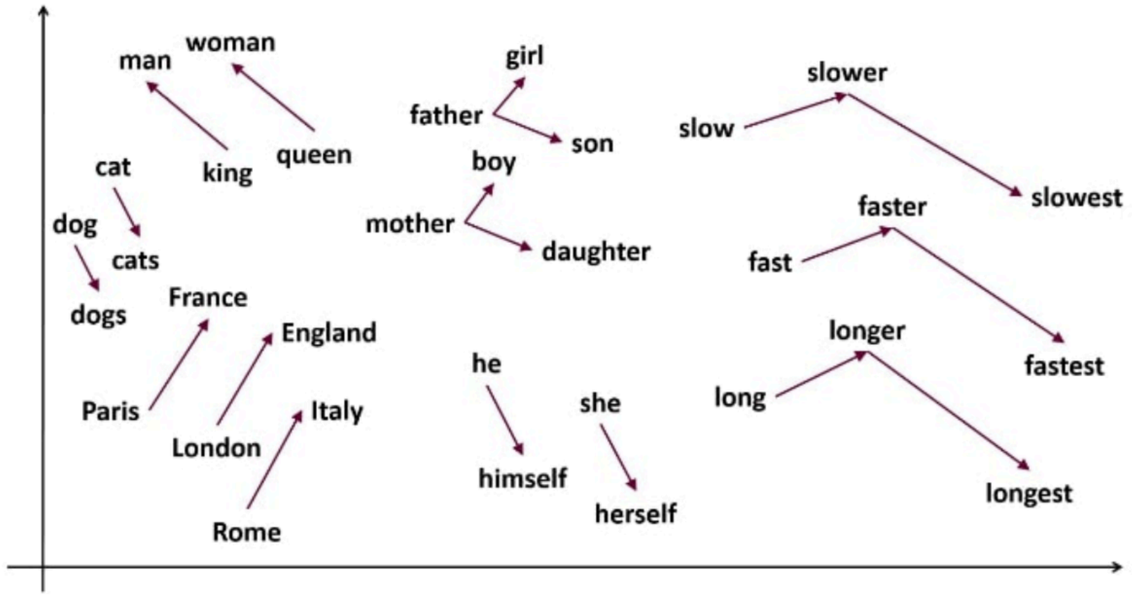
分散表現って何？

「**意味的な**類似性」を「**文脈における**類似性」に置きかえた表現

アイデア：

「2つの単語が、**似たような文脈**（その単語の周りに生起する単語郡のこと）において使われているのなら、それらの単語の**意味は似ている**」と考える、

分布仮説（Distributional Hypothesis [1]）に基づいています。



[1] https://www.tandfonline.com/doi/abs/10.1080/00437956.1954.11659520

どんなベクトル表現なの？

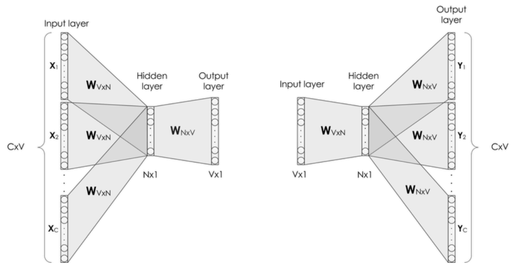
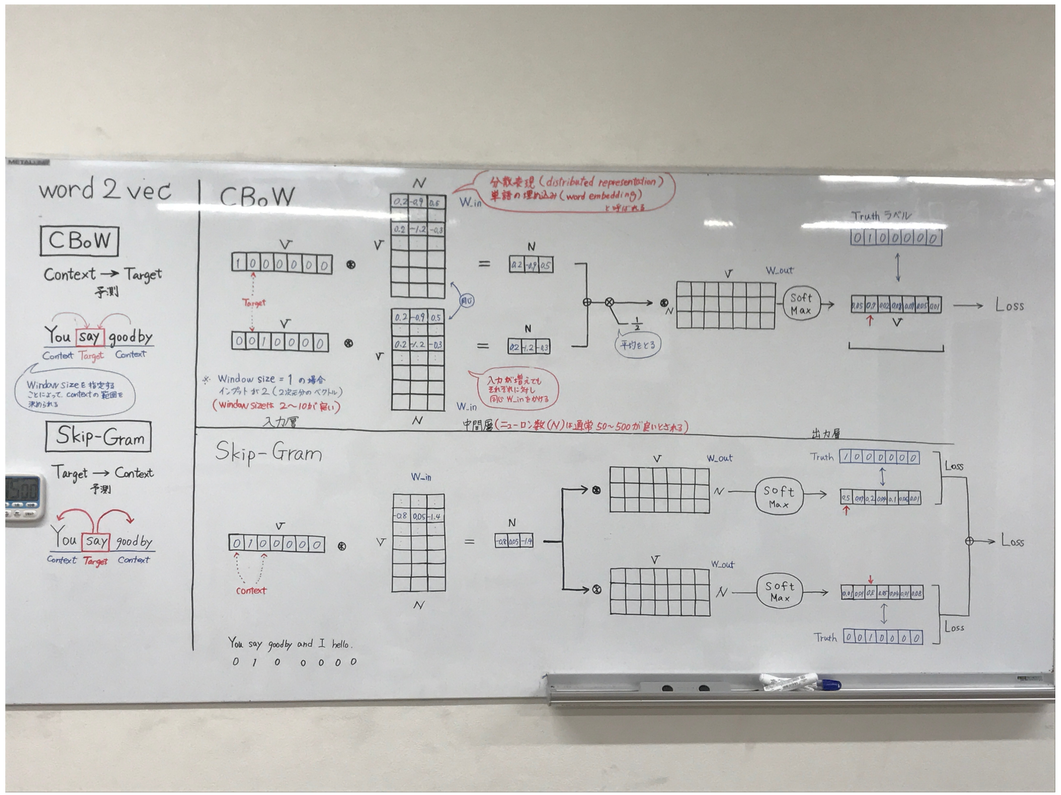


Illustration of the word2vec models: (a) CBOW, (b) skip-gram [16, 33].

ニューラルネットワークの重み行列から取得される、任意の次元のベクトル

「分散表現」は、**Word2Vec**（ニューラルネットワーク）の学習において生成される**重み行列からなるベクトル**です。世の中に何万語と存在する単語を識別するためのユニークな表現として、**任意の次元**のベクトルを用います。

似たような文脈を持つ単語ほど近しい位置を占めるベクトル空間（前ページの図参照）に、分散表現が最小単位として埋め込まれます（一般的に、50〜500次元の固定長ベクトルとする）。
分散表現の個々の**次元**（軸）は通常、固有の意味を持ちません。ベクトル間の位置と距離によって決まる**全体的なパターン**（組み合わせ）こそが、**単語の意味の代わりになる**のです。



単語を代替する手段として、
分散表現以外は考えられないの？

分散表現に対して、語彙のインデックスのベクトルで単語を表現する**「局所表現」**というものがある

分散表現 VS 局所表現：

分散表現に対して、**BoW**（Bag of Words）と呼ばれる古典的手法（ニューラルネットワークではない）によってベクトルを取得することができます。これを「**局所表現**」と呼びます。
この手法では、個々の単語をIDに置き換えることによって符号化し、ワンホットベクトルを得ます。
この **ワンホットベクトル** は、各単語が1つの次元を占め、他の次元とは無関係な（言い換えると、他の次元に対し写像がない）多次元空間に、互いに独立して存在している、と捉えることができます。

このため、単語同士の類似性を表すことができず、さらにベクトルの**次元数**（長さ）がその**分析対象の全ボキャブラリ数**に依存して決まるため（しかも、疎な高次元ベクトルになる）、後からボキャブラリを増やすこともできません。

	a	bad	film	good	is	movie	this	very
0	0	0	0	1	1	1	1	1
1	1	0	1	1	1	0	1	0
2	0	2	0	0	0	0	0	3

行方向が文書(Document)、列方向が単語(Words)を表している

局所表現にも種類があるよ

BoWではなくTF-IDFという手法で作られる

もっとマシな局所表現がある

	a	bad	film	good	is	movie	this	very
0	0	0	0	1	1	1	1	1
1	1	0	1	1	1	0	1	0
2	0	2	0	0	0	0	0	3

BoW

	a	bad	film	good	is	movie	this	very
0	0.00000	0.00000	0.00000	0.417796	0.417796	0.549351	0.417796	0.417796
1	0.51742	0.00000	0.51742	0.393511	0.393511	0.000000	0.393511	0.000000
2	0.00000	0.65918	0.00000	0.000000	0.000000	0.000000	0.000000	0.751985

TF-IDF

BoWによるワンホットベクトルの問題点：

① 多くの文章内に非常に一般的な単語が登場する（「the」、「and」、「or」など）。

② 同じ単語が反復される文章は入力データとしてふさわしくない。

これらの問題点は、**TF - IDF**（ BoWの発展的手法 ）で作られる局所表現では解消されます。

TF - IDFは、文章内の単語の総数（ TF ）を、すべての文章中での出現頻度（*df(t)* ）によって 重みづけ します。

出現頻度の数値が大きいほど、逆文書頻度（ IDF ）が小さくなり、その単語の価値を低く見積もります（つまり、値を小さくする）。

分散表現には問題点はないの？

一方、分散表現にも言語特有の構文構造が失われるといった

問題点は残ります。