

テキストマイニング

— Part 1 —

2022/7/1

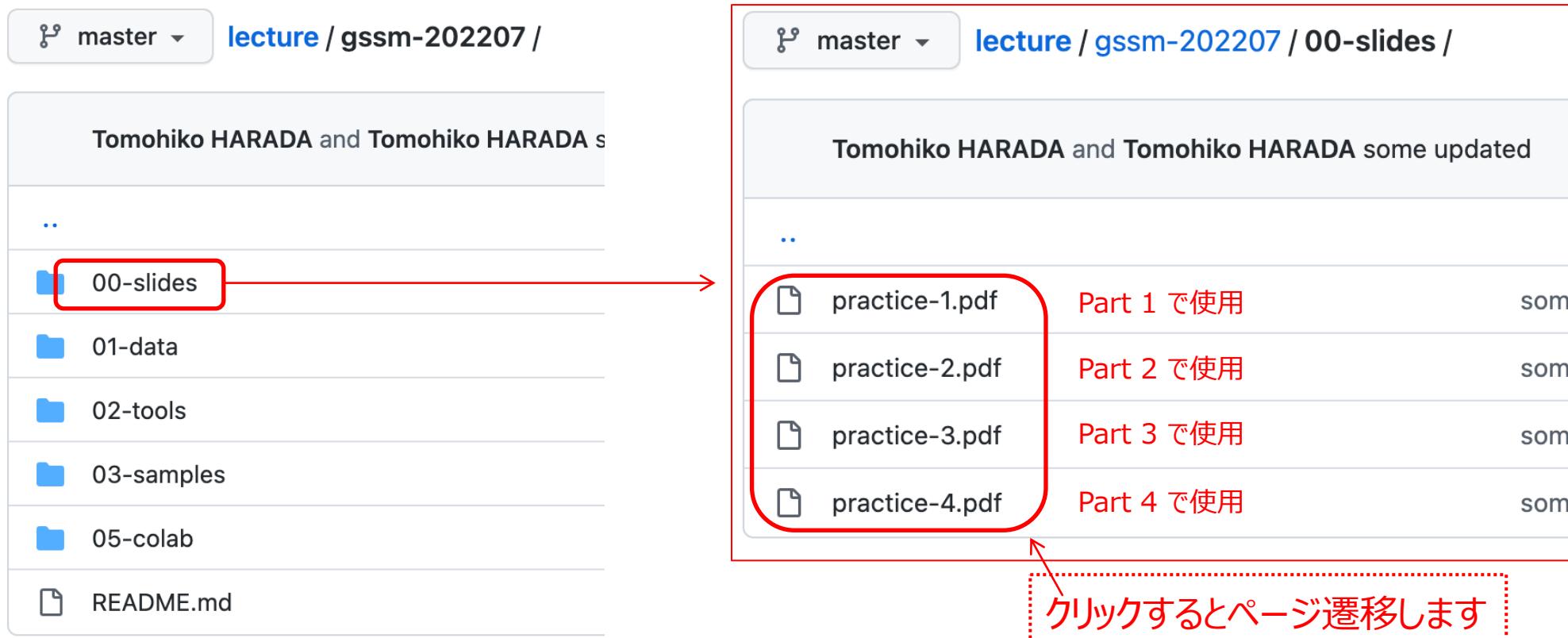
人文社会ビジネス科学学術院
ビジネス科学研究群

スケジュール

- Part 1
 - 説明 — 自然言語処理のトレンド
 - 説明 — 環境説明
- Part 2
 - 説明 — テキストマイニングの手順
 - 説明 — データ理解
 - 実習 — データ理解 (Excel)
- Part 3
 - 説明 — データ分析の実践 (KHCoder)
 - 実習 — データ分析の実践 (KHCoder)
- Part 4
 - 実習 — データ分析の実践 (KHCoder)
 - 実習 — データ分析の実践 (KHCoder)

講義スライド

- <https://github.com/haradatm/lecture/tree/master/gssm-202207>



自然言語処理のトレンド

2022/7/1

人文社会ビジネス科学学術院
ビジネス科学研究群

自然言語処理って何を連想しますか？

どうやって出来ているかの話です

- 小難しい話が沢山出できます
- が、ぼんやりとでも良いのでこれだけ覚えていってください
- **分布仮説**
- 単語の**分散表現**
- 文脈を考慮したテキスト(文)の**分散表現**

分布仮説 [Harris+, 1954]

- 単語の意味はその周囲の単語から形成されるという仮説
→ 似た文脈で出現する単語は意味が似ている

文1: 昨日,りんごを食べた。りんごジュースを飲んだ。りんごの皮をむいた。

文2: 昨日,りんごを食べた。ぶどうジュースを買った。ぶどうの皮をむいた。

文3: 昨日,自転車に乗った。自転車を修理した。自転車を買った。

分布仮説 [Harris+, 1954]

- ・単語の意味はその周囲の単語から形成されるという仮説
→ 似た文脈で出現する単語は意味が似ている

文1: 昨日,りんごを食べた。りんごジュースを飲んだ。りんごの皮をむいた。

文2: 昨日,りんごを食べた。ぶどうジュースを買った。ぶどうの皮をむいた。

文3: 昨日,自転車に乗った。自転車を修理した。自転車を買った。

共起による
ベクトル表現
[Lin, 2002]

	…	食べる	…	飲む	…	修理	…
りんご	0	1	0	1	0	0	0
ぶどう	0	1	0	1	0	0	0
自転車	0	0	0	0	0	1	0
:							

→ 似てる
→ 似てない

← 語彙数(数万～数十万)分の疎なベクトルになる →

分布仮説と単語の分散表現

- ・単語の意味はその周囲の単語から形成されるという仮説 (=分布仮説)
→ 似た文脈で出現する単語は意味が似ている
- ・各意味を複数の次元で分散して表現する (=分散表現)
→ 次元は低次元(例えば100次元)で, 値は実数値

単語埋め込み
(word embedding)
とも呼ばれる

これらの実数値をニューラルネットワークで求める

共起による
ベクトル表現
[Lin, 2002]

次元→	0	1	…	50	…	98	99
りんご	1.07	-1.08		1.48		0.46	0.48
ぶどう	1.95	-1.53		0.36		-0.61	-0.44
自転車	0.67	1.44		-1.50		0.10	0.67
:							

← 数百次元の密なベクトル →

似てる
似てない

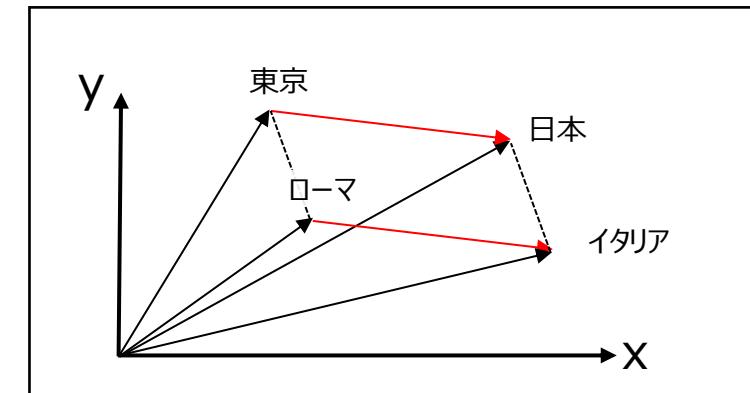
word2vec [Mikolov+, 2013]

- ニューラルネットワークを用いた単語のベクトル化（分散表現）

古典的手法	分散表現
TF-IDF, Okapi BM25 など (分布的, 高次元, スパース)	word2vec, GloVe, fastText など (分散的, 低次元, 密)

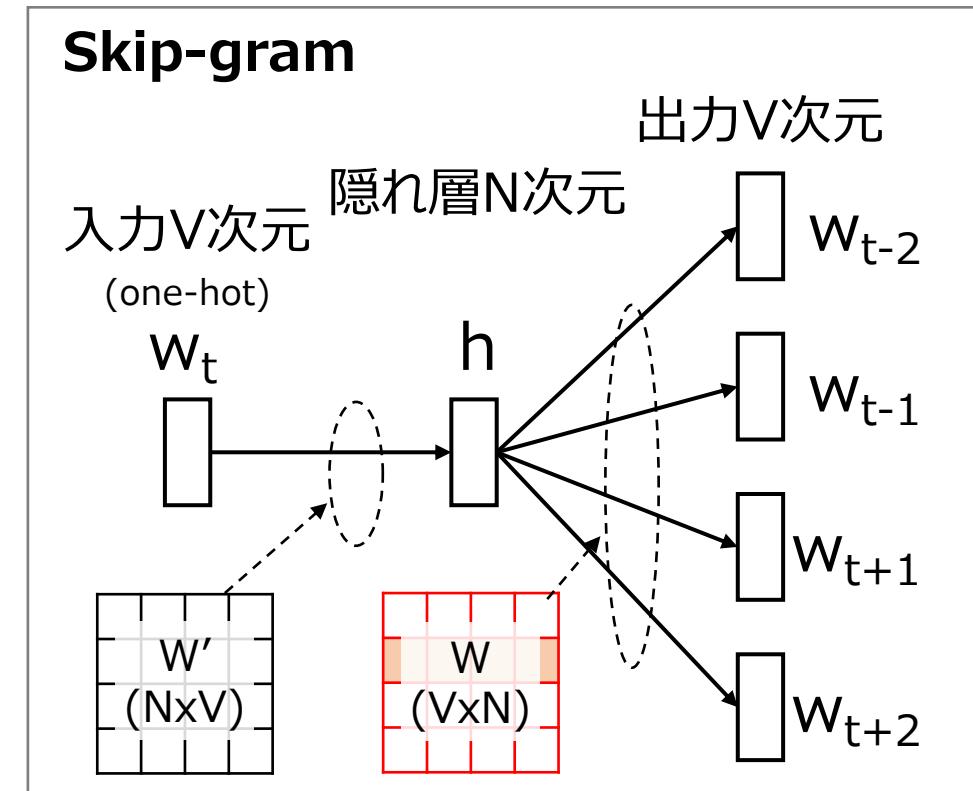
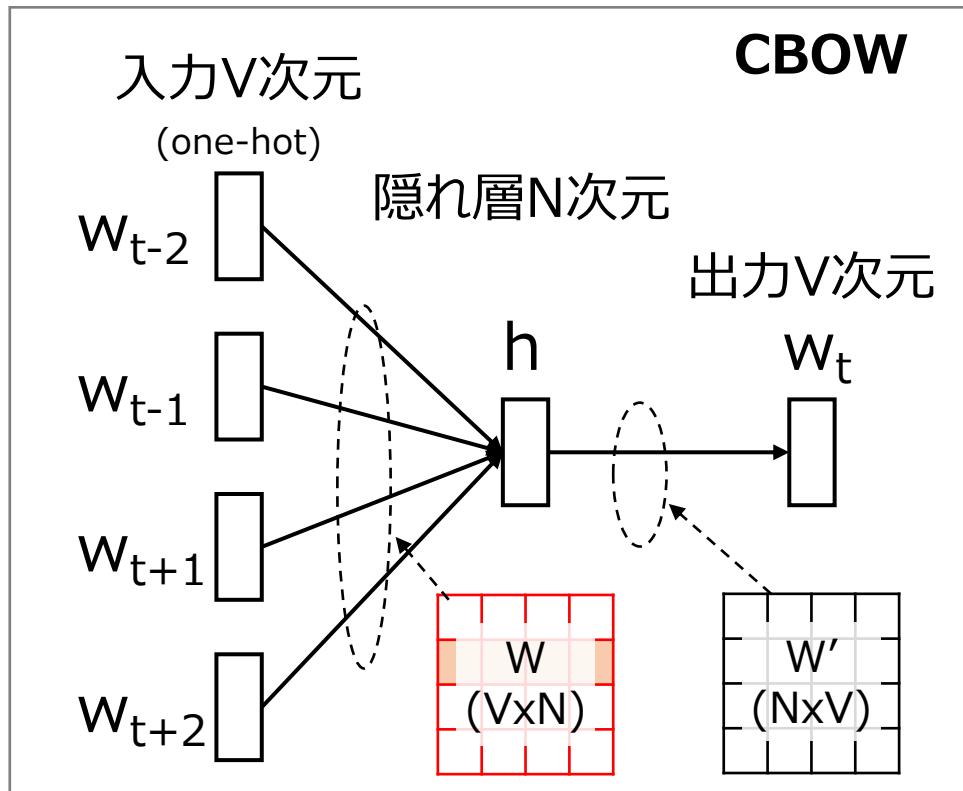
- 代表格は「word2vec」
 - NNによる分布仮説のモデル化
 - king - man + woman = queen で有名
→右の例では、日本- 東京 + ローマ = イタリア

Tomas Mikolov, Wen-tau Yih, Geoffrey Zweig, 2013, NAACL



word2vec の学習

- 周辺の単語から中心の単語を予測(CBOW) or その逆(Skip-gram)



word2vec の問題点

- 分布仮説に起因した以下の問題がある
 - 反義語: 共起する単語が似ているので,似たベクトルになりやすい
 - 多義語: 意味の異なる単語を同じベクトルにしようとする

テキスト(文)の分散表現

- 文脈を考慮することで様々なタスクの性能が向上

文脈に関係なく 一つの単語には一つのベクトルが割り当てられる	周りの文脈によって 同じ単語でも異なるベクトルが割り当てられる
<p>首を痛める</p> <p>首 </p> <p>会社を首になる</p> <p>首 </p>	<p>首を痛める</p> <p>首 </p> <p>会社を首になる</p> <p>首 </p>

近年の自然言語処理

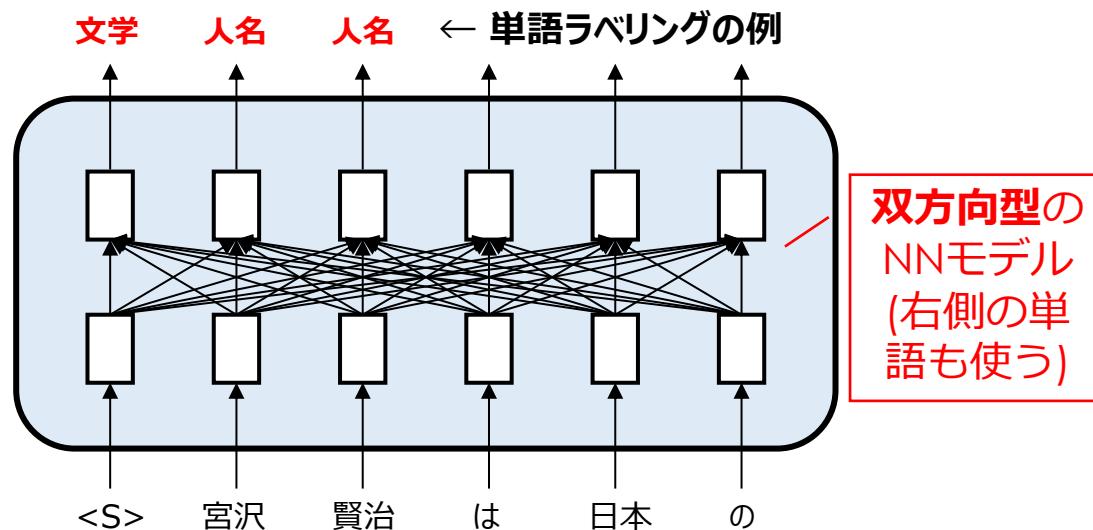
[西田,2022] JSAI2022 チュートリアル
講演資料の一部を修正して作成

- NNにより文脈を考慮した分散表現を獲得し,様々なタスクで性能向上

エンコーダ型:

テキスト(単語系列)のクラス分類や
テキスト(単語系列)単語ラベリングなど

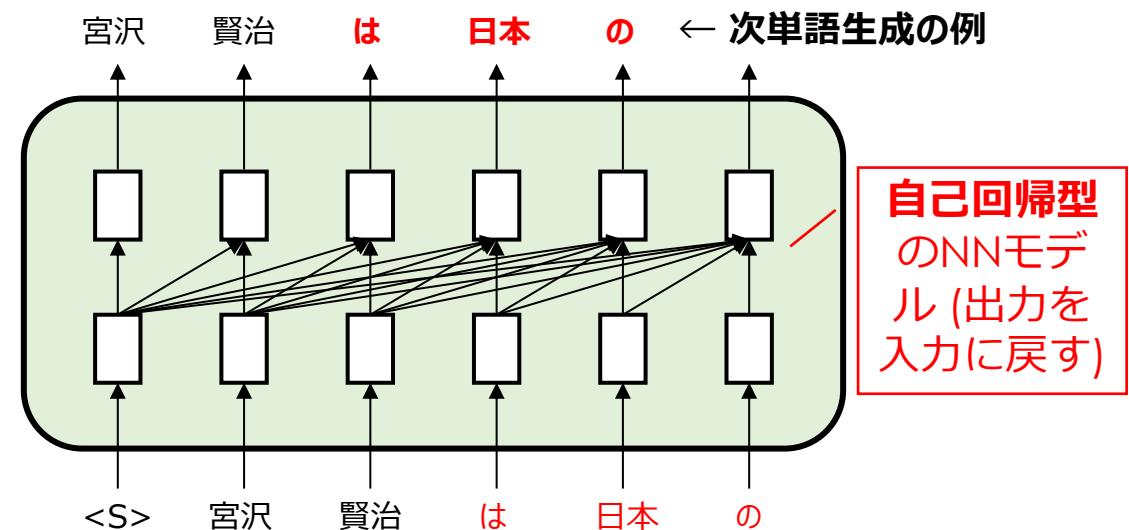
代表モデル: BERT [Devlin+, NAACL'19]



デコーダ型:

テキスト(単語系列)の続きを生成したり
テキストAからテキストBへの変換(翻訳)を行う

代表モデル: GPT-3 [Brown+, NeurIPS'20]



ここまでまとめ

- 分布仮説
 - 似た文脈で出現する単語は意味が似ているという仮説
- 分散表現
 - 単語の意味を複数次元の実数値で分散して表現したもの
 - ニューラルネットワークで学習し, 次元は低次元 (例えば100次元)
 - word2vec が有名 → 反義語や多義語の問題がある
- 近年の自然言語処理
 - 複雑なNNにより文脈を考慮した分散表現を獲得 → 様々なタスクで性能向上
 - BERT や GPT-3 が有名

古典的な自然言語処理

- 基礎タスク

- 言語を応用タスクで利用しやすい形式に変換する

形態素解析

固有表現抽出

構文解析

照応解析

意味解析

談話解析

など

- 応用タスク

- 自然言語処理を応用したアプリケーション

テキスト検索

テキスト分類

テキスト要約

情報抽出

機械翻訳

質問応答

対話

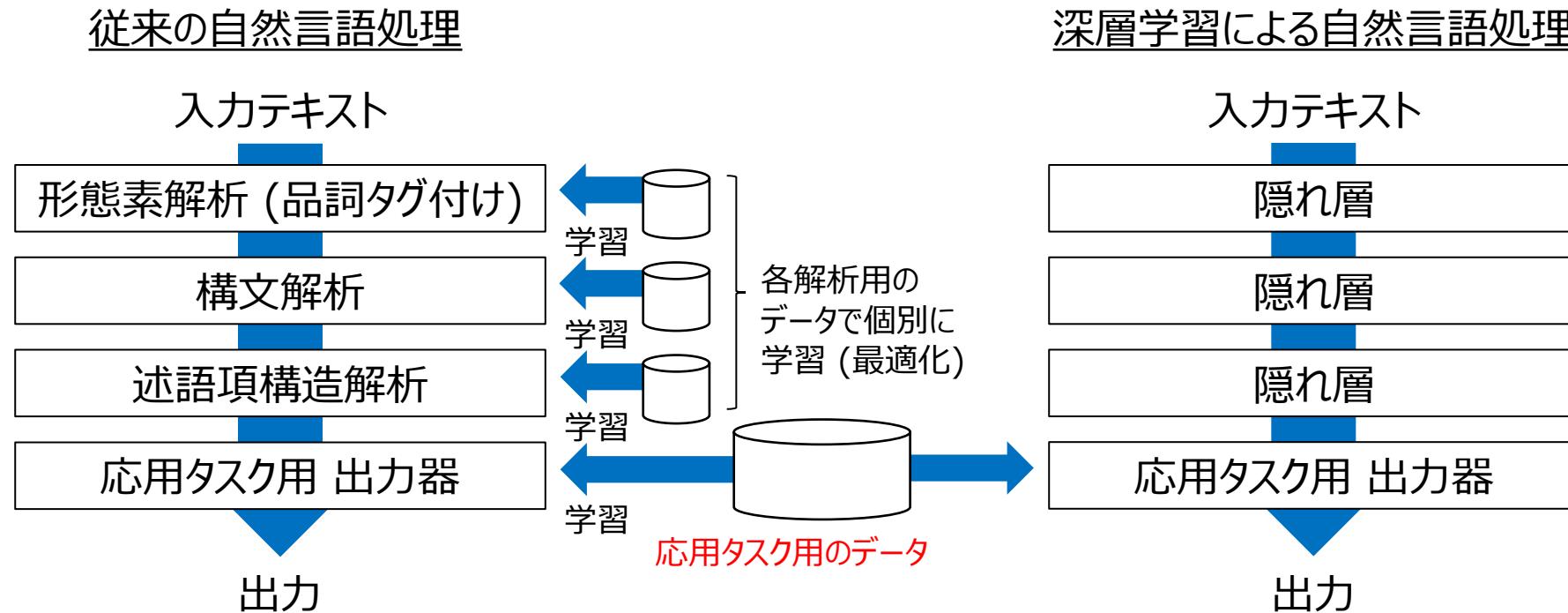
など

(参考) 古典的な自然言語処理タスク

基礎タスク	形態素解析 (品詞タグ付け)	文をそれぞれの意味を担う最小の単位(= 形態素)に分割し,それに品詞などの情報を付与する (例: MeCab, JUMAN++)
	構文解析	形態素解析で分割した単語同士の関連性を解析し,主に文節間の係り受け構造を発見しツリー化する, 文中の単語間の係り受け関係を調べ,どの単語がどの単語に係るのかを構文的に解析する <u>係り受け解析</u> (例: CaboCha, KNP, SpaCy) や, 語および文法的カテゴリを節点とするツリー形式によって文の構造を表現した <u>句構造解析</u> (例: Stanford Core NLP) がある
	意味解析	与えられた文のを明らかにする処理は何でも意味解析と呼ばれる, 格解析, 述語項構造解析, 多義性解消, 比喩理解 などが例として挙げられる
応用タスク	機械翻訳	自然言語によるある言語の文を入力とし,これを違う言語の文に翻訳する
	質問応答	自然言語による質問文を入力として受け取り,適切な回答を返す
	テキスト要約	与えられた文章を短く簡潔にまとめる, 文章の一部を抜粋して要約を作成する <u>抽出型要約</u> と,元の文章に存在しない文章で要約を作成する <u>抽象型要約</u> がある
	対話システム	自然言語により人間と機械が対話をを行う,チャットボットなどに使用されている

深層学習(NN)の導入

- 大規模な訓練データで応用タスク全体を学習 → End-to-end 学習



坪井, 海野, 鈴木. 深層学習による自然言語処理. 講談社, 2017, p.4 の図を一部修正

深層学習(NN)の導入 →つまりこれ

近年の自然言語処理

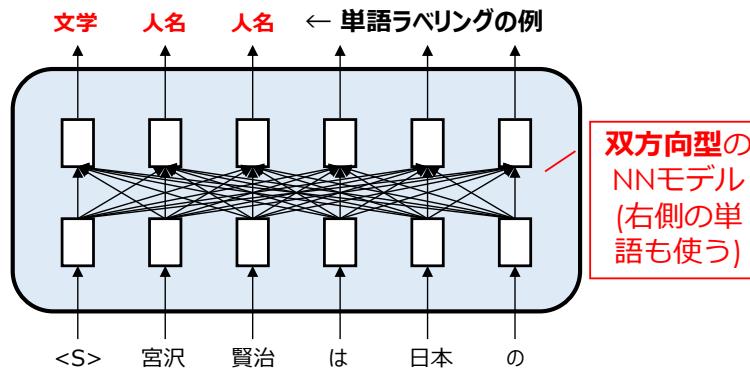
[西田,2022] JSAI2022 チュートリアル
講演資料の一部を修正して作成

- NNにより文脈を考慮した分散表現を獲得し,様々なタスクで性能向上

エンコーダ型:

テキスト(単語系列)のクラス分類や
テキスト(単語系列)単語ラベリングなど

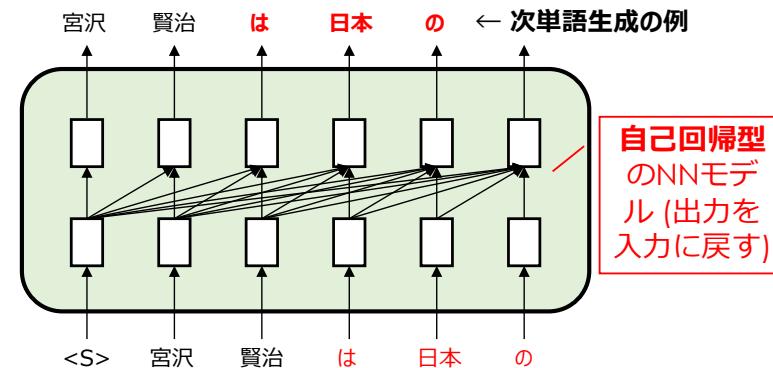
代表モデル: BERT [Devlin+, NAACL'19]



デコーダ型:

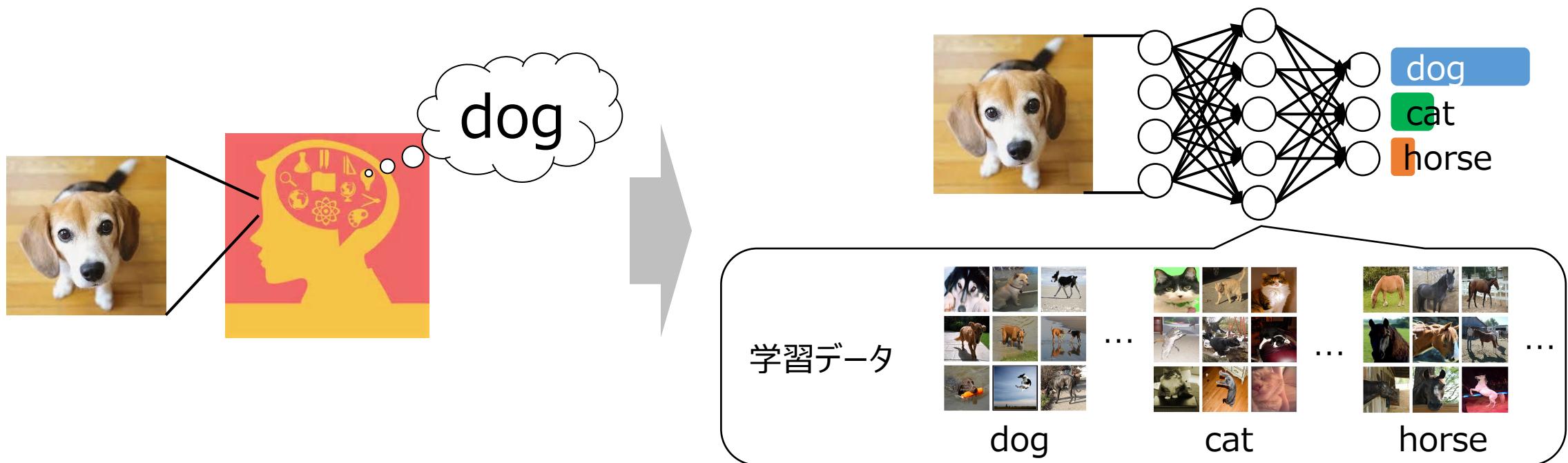
テキスト(単語系列)の続きを生成したり
テキストAからテキストBへの変換(翻訳)を行う

代表モデル: GPT-3 [Brown+, NeurIPS'20]



(参考) 深層学習=ディープラーニング

- ニューラルネットワーク(NN)を用いた機械学習手法
 - 機械学習とは、データを学習し、パラメータを獲得すること
 - 脳の神経細胞(ニューロン)の働きを模した仕組みや構造のこと



(参考) ニューラルネットワーク研究の系譜

- 黎明～終焉を繰り返し,近年は 3度目のブーム

第 1 期	1940～	• McCullochとPitts が形式ニューロンモデルを発表 [McCulloch-Pitts,43]
	1950～	• Rosenblatt がパーセプトロンを発表 [Rosenblatt,57]
	1960～	• MinskyとPapert が単純パーセプトロンの(線形分離不可能問題への)限界を指摘 [Minsky-Papert,69]
冬	1970～	冬の時代 (階層的構造の学習方法が未解決)
第 2 期	1980～	• Fukushima らがネオコグニトロンを提案 [Fukushima,80]
		• Rumelhart らが誤差逆伝播法を提案 [Rumelhart+,86]
		• LeCun らが畳み込みニューラルネット Conv.net を提案 [LeCun,89]
冬	1990～	冬の時代 (学習時間や過学習に課題, 一方でSVMが流行)
第 3 期	2000～	• Hinton らが事前学習とオートエンコーダを導入した多層NNを提案 [Hinton+,06]
	2010～	• Seide らが音声認識のベンチマークで圧勝 [Seide+,11] • Krizhevsky らがReLU を提案し画像認識コンペで圧勝 [Krizhevsky,12]

(参考) 音声認識で成功 [Seide+, 2011]

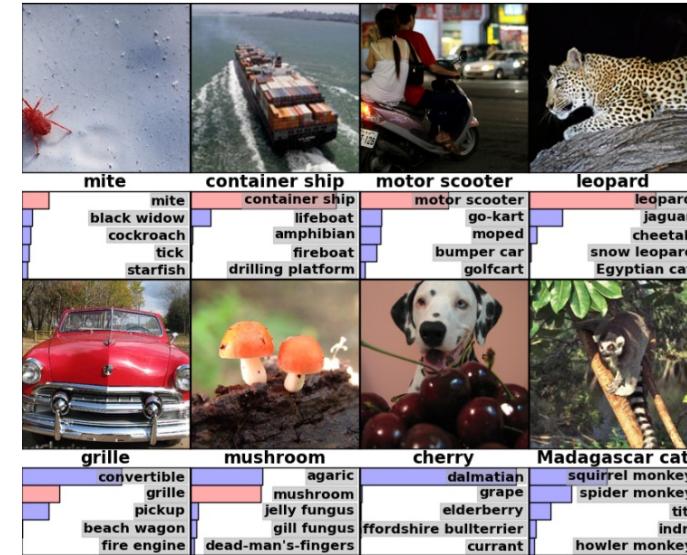
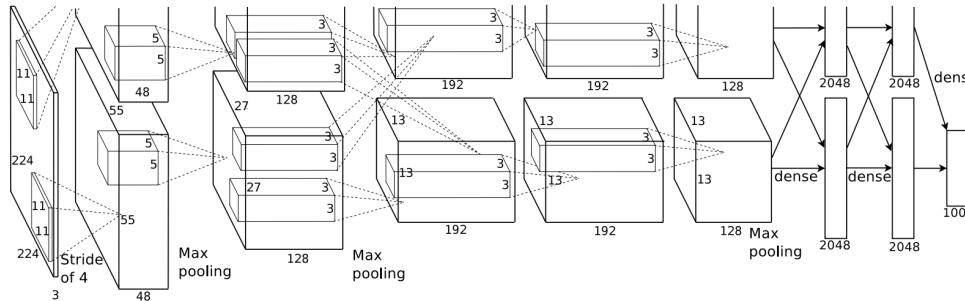
- Microsoft Research のグループ
 - 電話での会話音声の標準データセット
 - 入力(MFCC)-出力(HMM状態変数)の関係をDNNで学習
 - 従来 GMM-HMM → DNN-HMM (全結合7層, 事前学習あり)
 - 単語誤認識率で 10%前後の大幅な精度改善

acoustic model & training	recognition mode	RT03S		Hub5'00 SWB	voicemails		tele- conf
		FSH	SW		MS	LDC	
GMM 40-mix, ML, SWB 309h	single-pass SI	30.2	40.9	26.5	45.0	33.5	35.2
GMM 40-mix, BMMI, SWB 309h	single-pass SI	27.4	37.6	23.6	42.4	30.8	33.9
CD-DNN 7 layers x 2048, SWB 309h, this paper (rel. change GMM BMMI → CD-DNN)	single-pass SI	18.5 (-33%)	27.5 (-27%)	16.1 (-32%)	32.9 (-22%)	22.9 (-26%)	24.4 (-28%)

F. Seide, G. Li and D. Yu, "Conversational Speech Transcription Using Context-Dependent Deep Neural Networks." *Interspeech*. 2011.

(参考) 画像認識で成功 [Krizhevsky+, 2012]

- ・一般物体認識 (Hintonのグループ)
 - ・ImageNet Large-scale Visual Recognition Challenge 2012
 - ・1000カテゴリ×約1000枚 = 100万枚 の訓練画像
 - ・畳込み層5, 全結合層3, 2つのGPUで2週間 (AlexNet)
 - ・誤識別率が10%以上減少 (過去数年間での向上は1~2%)

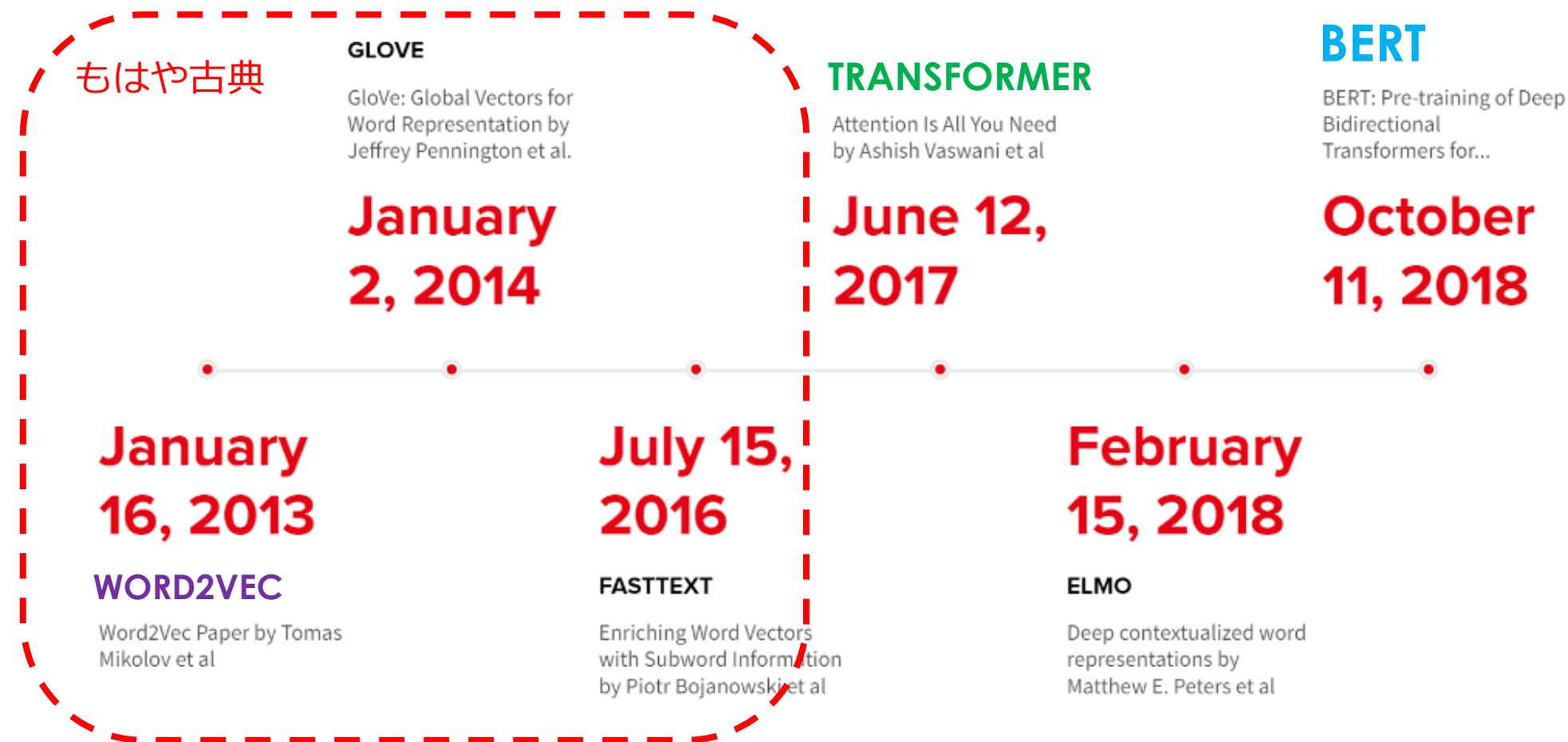


Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton.
"Imagenet classification with deep convolutional neural networks."
Advances in neural information processing systems. 2012.
<http://image-net.org/challenges/LSVRC/2012/supervision.pdf>

(参考) 深層学習 成功の背景

- 一定以上の規模のデータ → 改善
 - WebやIoT(センサ)などから十分な規模のデータを収集可能
- 学習の難しさ → 改善
 - 様々なテクニック (事前学習, dropout 等)
- 誤差逆伝搬法の計算量膨大 → 改善
 - 計算機能能力の飛躍的向上
 - GPU, マルチコアCPU, PCクラスタの登場
- 性能を引き出すのに必要なノウハウ → 未解決
 - 「黒魔術」のまま → Explanable AI (説明可能AI) 研究へ

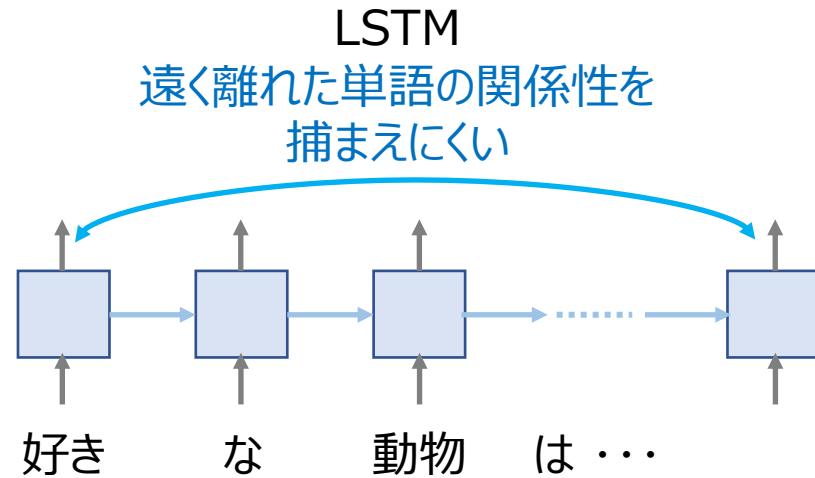
事前学習モデルのタイムライン



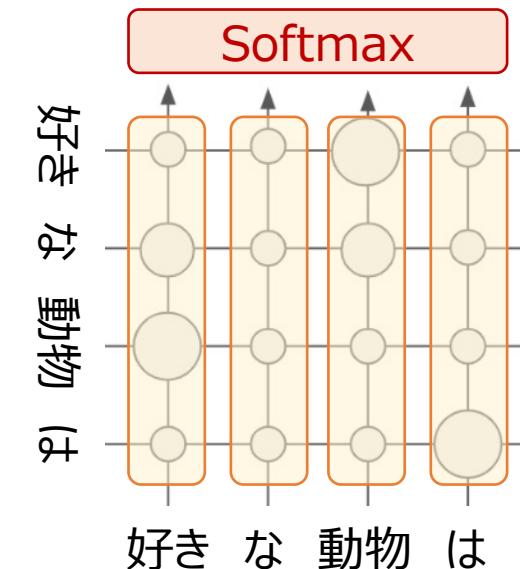
<https://towardsdatascience.com/2019-year-of-bert-and-transformer-f200b53d05b9>

Transformer [Vaswani+, 2017]

- Transformer (RNNやCNNを使わずアテンションのみ使用)がニューラル機械翻訳で圧倒的な SOTA を達成
 - 従来、単語系列の文脈理解は主にLSTM →長期依存性の理解に限界
 - 離れた単語の関係性も直接考慮できる Self-Attention が性能向上に大きく寄与した (しかも省メモリで計算可)

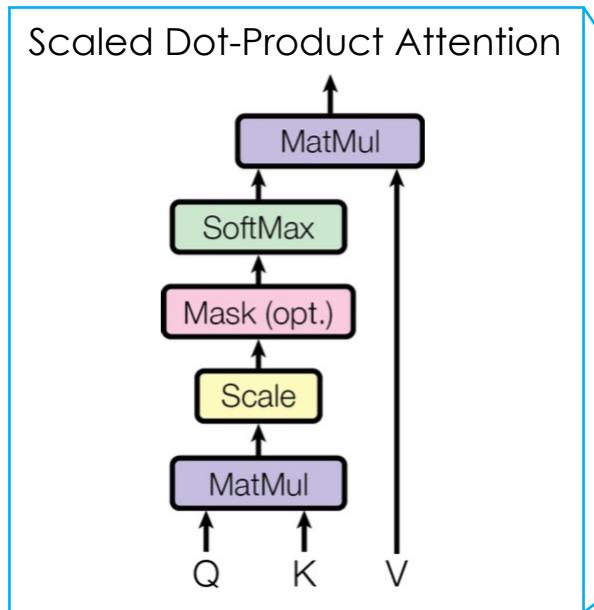


Self-Attention
遠く離れた単語も
直接関係性を考慮できる

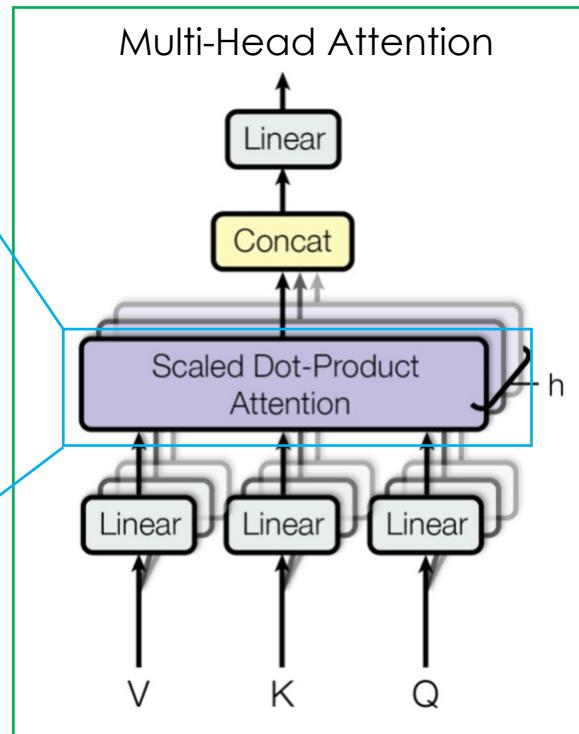


(参考) Transformer の構造

- 例: レイヤーN=6, ヘッドh=8, 長さ=512, 中間層=768



$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$



$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O$$

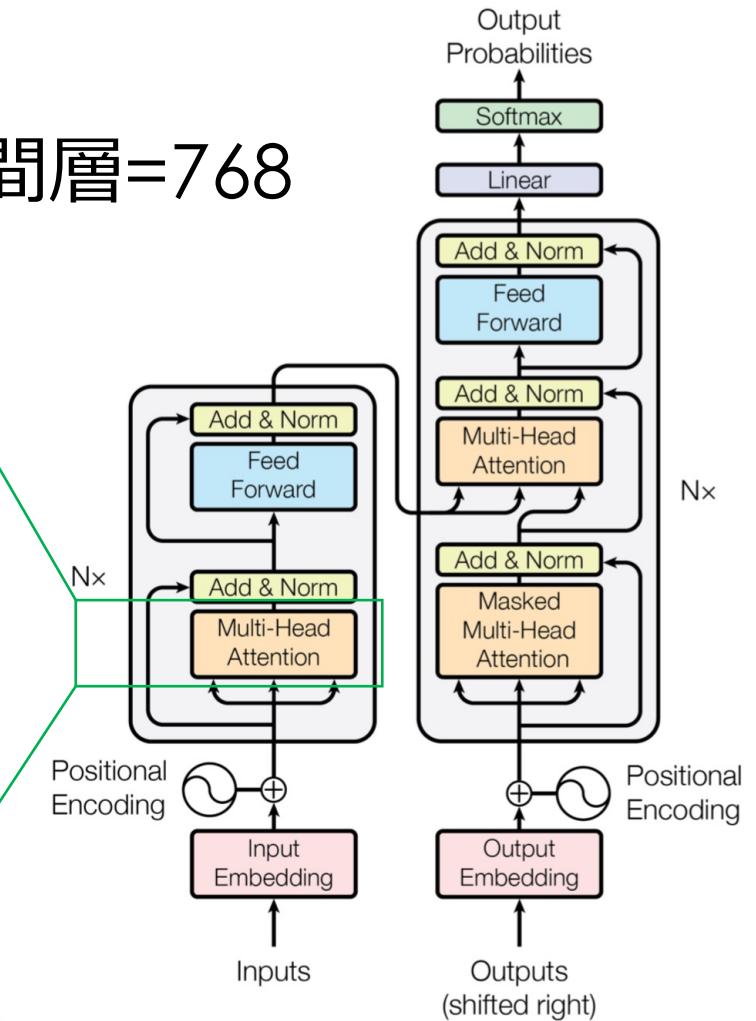


Figure 1: The Transformer - model architecture.

2018年10月：BERT の衝撃

- タスクに特化した構造を持たずに、人間のスコアを大きく超えた

SQuAD1.1 Leaderboard

Since the release of SQuAD1.0, the community has made rapid progress, with the best models now rivaling human performance on the task. Here are the ExactMatch (EM) and F1 scores evaluated on the test set of SQuAD v1.1.

Rank	Model	EM	F1
	Human Performance <i>Stanford University</i> (Rajpurkar et al. '16)	82.304	91.221
1	BERT (ensemble) <i>Google AI Language</i> https://arxiv.org/abs/1810.04805	87.433	93.160
2	BERT (single model) <i>Google AI Language</i> https://arxiv.org/abs/1810.04805	85.083	91.835
2	nlnet (ensemble) <i>Microsoft Research Asia</i>	85.356	91.202

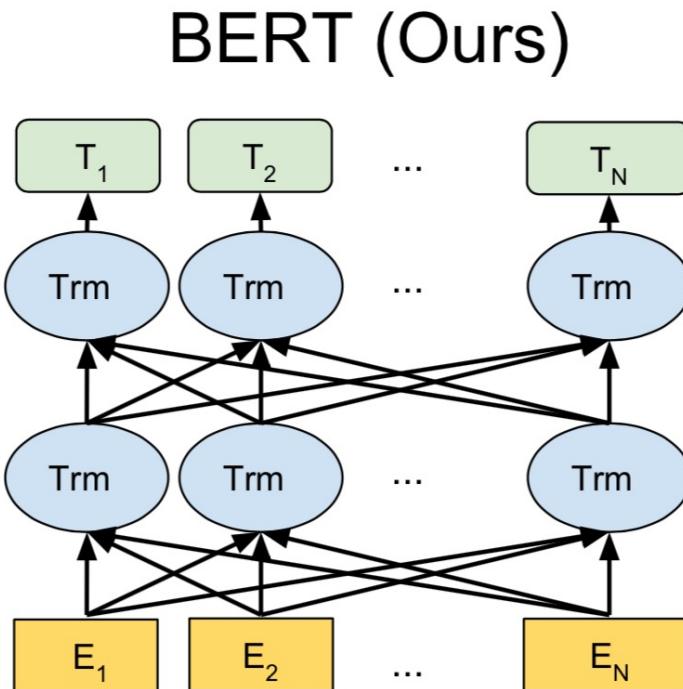
人間
BERT

- 機械読解タスク(左)で、完全一致と部分一致の両指標で最高精度（2018/10/5）
- 様々な自然言語理解タスクでSOTA (QA, 含意, 言い換え, NER等)
- タスク適応は、出力層をタスク毎に1層のみ追加してfine-tuning

<https://rajpurkar.github.io/SQuAD-explorer/>

BERT [Devlin+, 2018] – 自然言語処理のブレイクスルー

- 双方向 Transformer ブロックを24層重ねた言語モデル
- 事前学習モデルが公開



- 英語
 - 本家 Google の事前学習モデル *1
 - Book Corpus 8億語 + 英語 Wikipedia 25億語 (語彙数 3万)
- 日本語
 - 黒橋研の事前学習モデル *2
 - 日本語 Wikipedia 約1,800万文 (語彙数 3.2万)

*1 <https://github.com/google-research/bert>

*2 <http://nlp.ist.i.kyoto-u.ac.jp/index.php?BERT日本語Pretrainedモデル>

BERT 事前学習モデル

公開元	Google Research	京大 黒橋・河原・村脇研	NICT	東北大 乾・鈴木研
日/英	英語	日本語	日本語	日本語
コーパス	14GB (Book Corpus, Wikipedia)	3GB (Wikipedia)	3GB (Wikipedia)	3GB (Wikipedia)
単語数	30K (BPE)	32K (JUMAN & BPE)	32K (MeCab+JUMAN & BPE)	32K (MeCab+Neologd & BPE)
入力長 *1	最大512トークン	最大128トークン	最大512トークン	最大512トークン
パラメータ	24層, 各層1024次元	24層, 各層1024次元	12層, 各層768次元	12層, 各層768次元
学習時間	16Cloud TPUs で 4日間(÷100時間)	1GPU (GTX 1080 Ti) で 約 30日間(÷750時間)*2	32GPU (V100) で約 7日 間(÷175時間)	8Cloud TPUs で 約14日間(÷350時間)

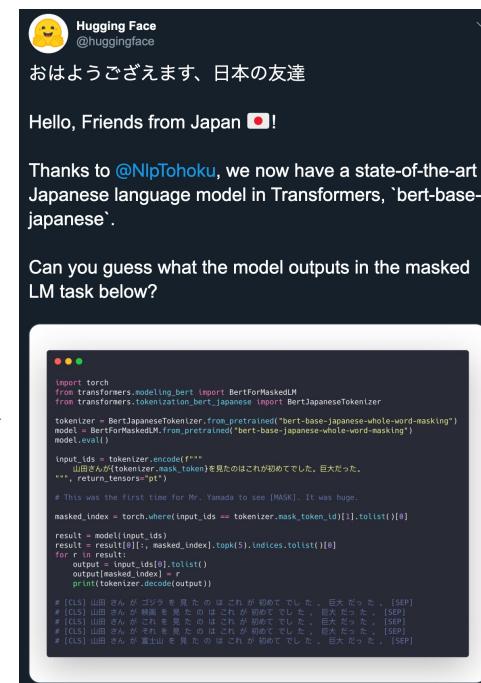
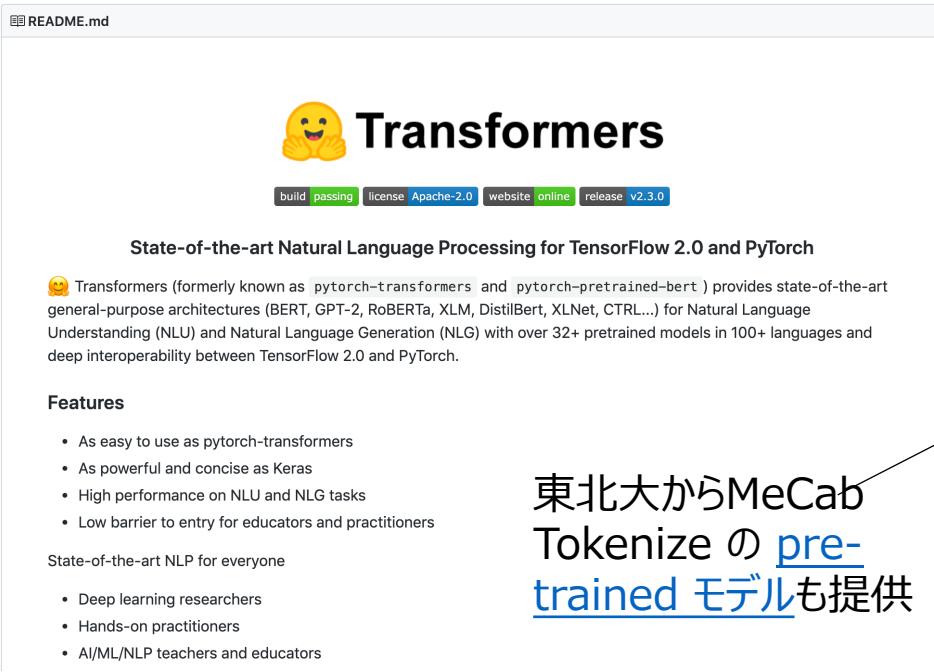
*1 入力できるシーケンスの長さに制限があることに注意

*2 表中のパラメタは LARGEモデル, 学習時間のみ BASEモデル(12層, 768次元)の場合

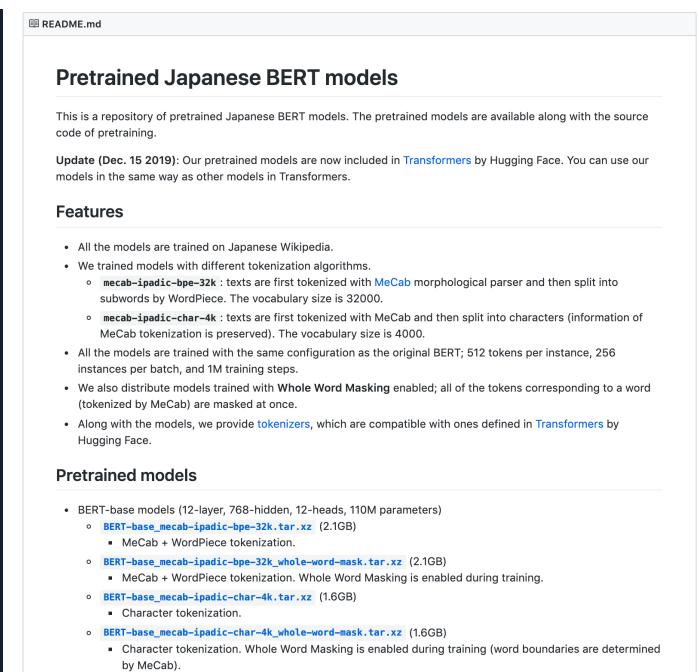
HuggingFace's Transformers

<https://huggingface.co/>

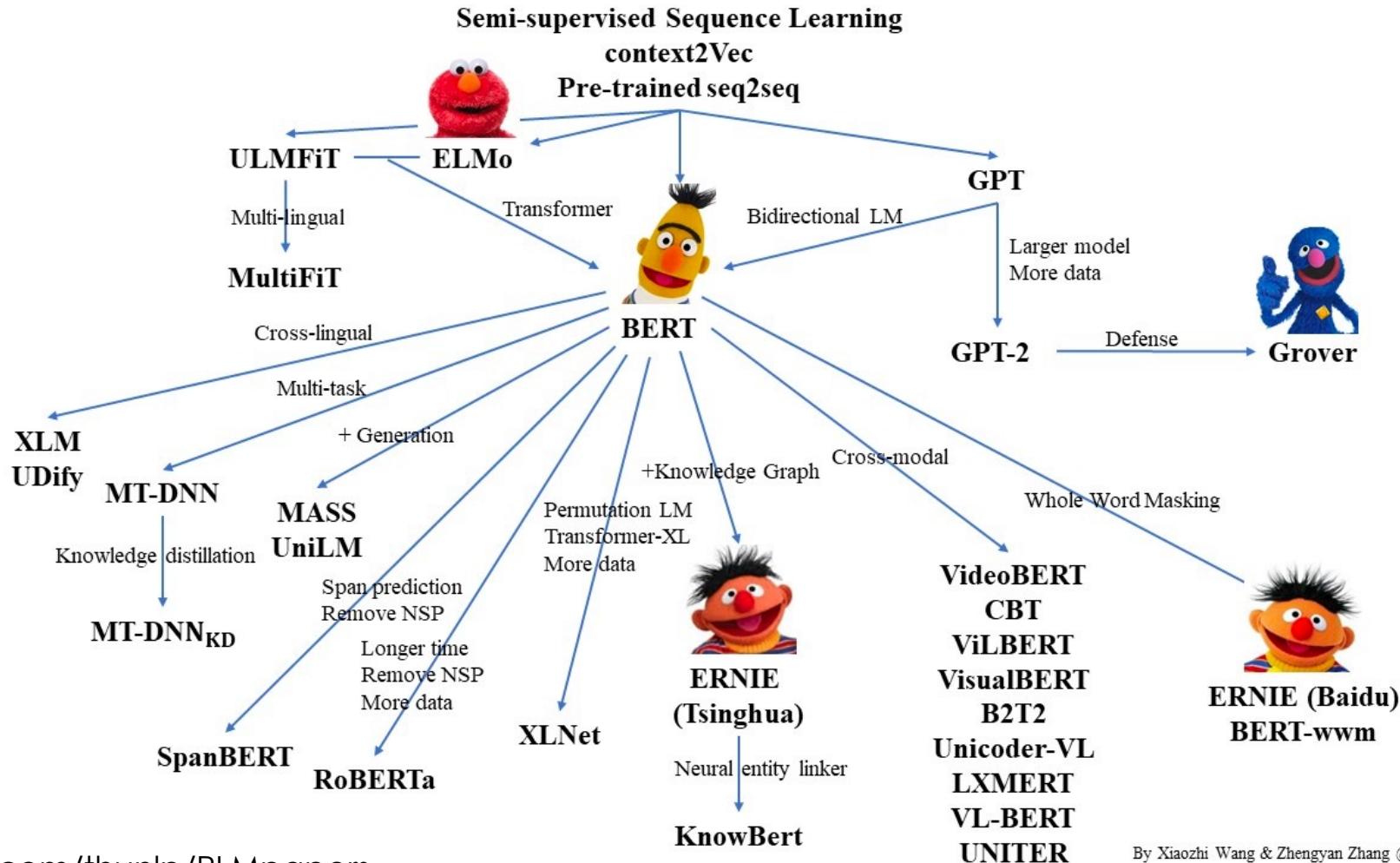
- Huggingface が提供する Pytorch によるフレームワーク
- 簡単にBERTなどの汎用言語モデルを動かせる



東北大からMeCab
Tokenize の pre-trained モデルも提供



1年以内に BERT 改良モデルが続々登場



<https://github.com/thunlp/PLMpapers>

GPT-3 [Brown+ (OpenAI), 2020]

- GPT-1_(1億), GPT-2_(15億)と同じ自己回帰モデルだが, **超大規模**(**1,750億**)
- タスクの説明もテキストとして入力し, マルチタスクを実現
- 少数のデモンストレーションのみで, 転移学習の性能に匹敵
 - **Zero-shot:** タスク説明のみを与え全くサンプルを与えない
 - **One-shot:** タスク説明と1つのサンプルのみを与える
 - **Few-shot:** タスク説明と少数(10から100)のサンプルを与える

The three settings we explore for in-context learning

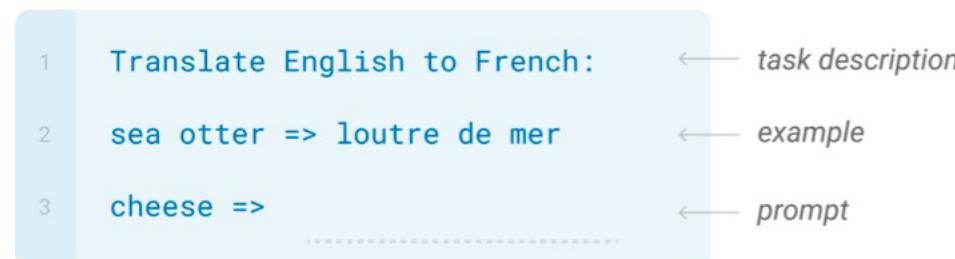
Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.



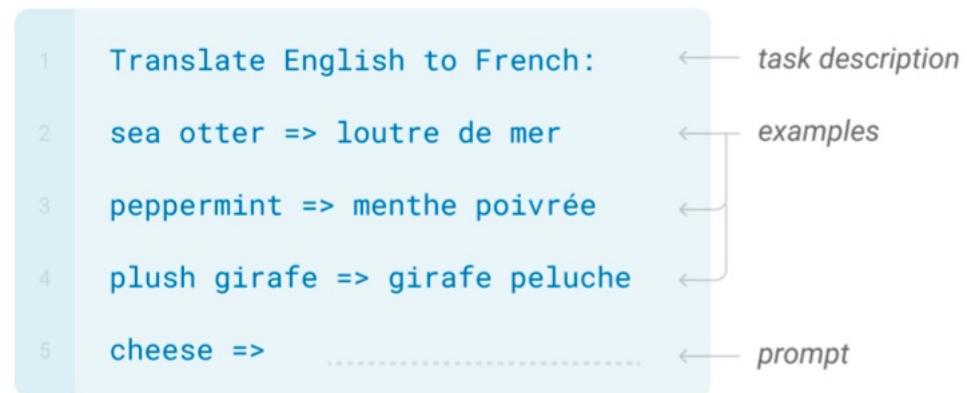
One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.



Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

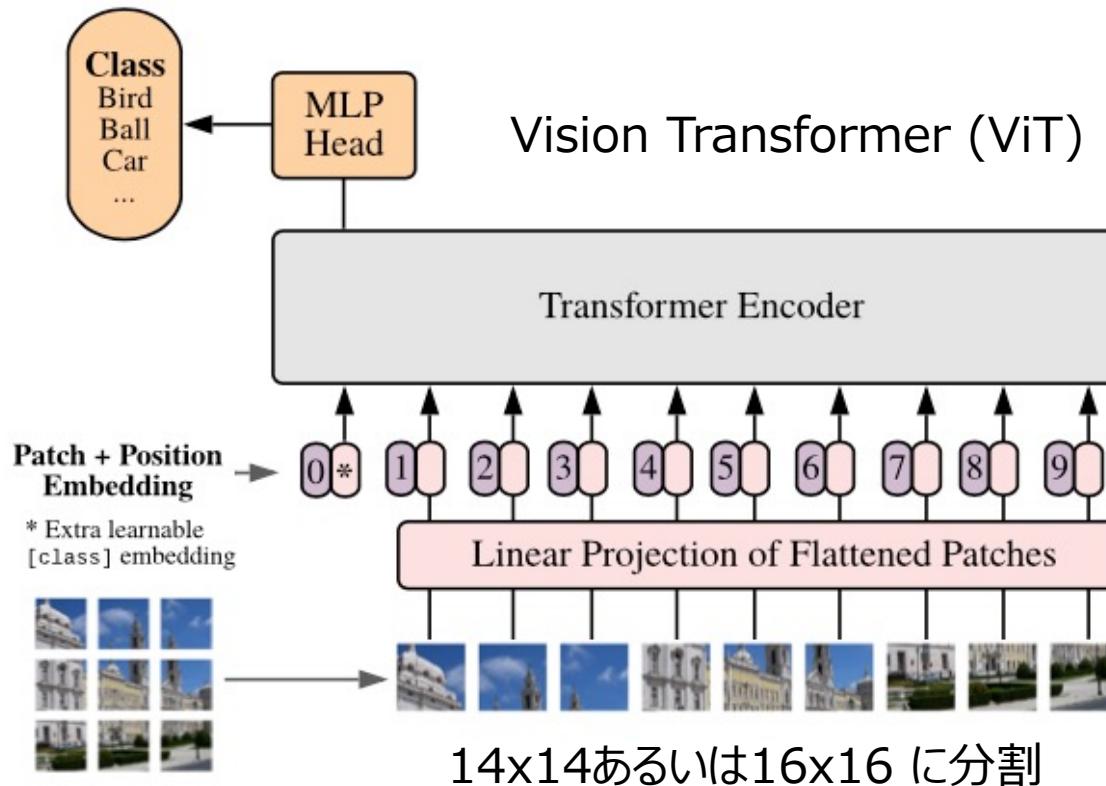


The screenshot shows the OpenAI Playground interface. At the top, there are navigation links: OpenAI Beta, Playground, Documentation, and Examples. Below this is a title bar with "Playground" and a help icon. The main area displays a list of Japanese-to-Japanese calendar conversion examples. The examples are:

- 西暦から和暦に変換します。
- 西暦2021年 => 令和3年
- 西暦1967年 => 昭和42年
- 西暦1900年 => 明治33年
- 西暦1853年 => 光熱元年
- 西暦1804年 => 慶安元年
- 西暦1733年 =>

Vision Transformer (ViT) [Dosovitskiy+, 2021]

- Transformer は画像認識などの NLP以外でも成果を発揮



- 画像パッチを単語とみなす6.32 億パラメタの Transformerエンコーダ
- 画像は最初にパッチに分割した後、線形変換で埋め込み
- 3億枚以上の画像分類で事前学習し、ImageNet 等で SOTA

https://github.com/google-research/vision_transformer

DALL·E [Ramesh+ (OpenAI), 2021]

- OpenAI が発表した文章に忠実な画像を生成するモデル

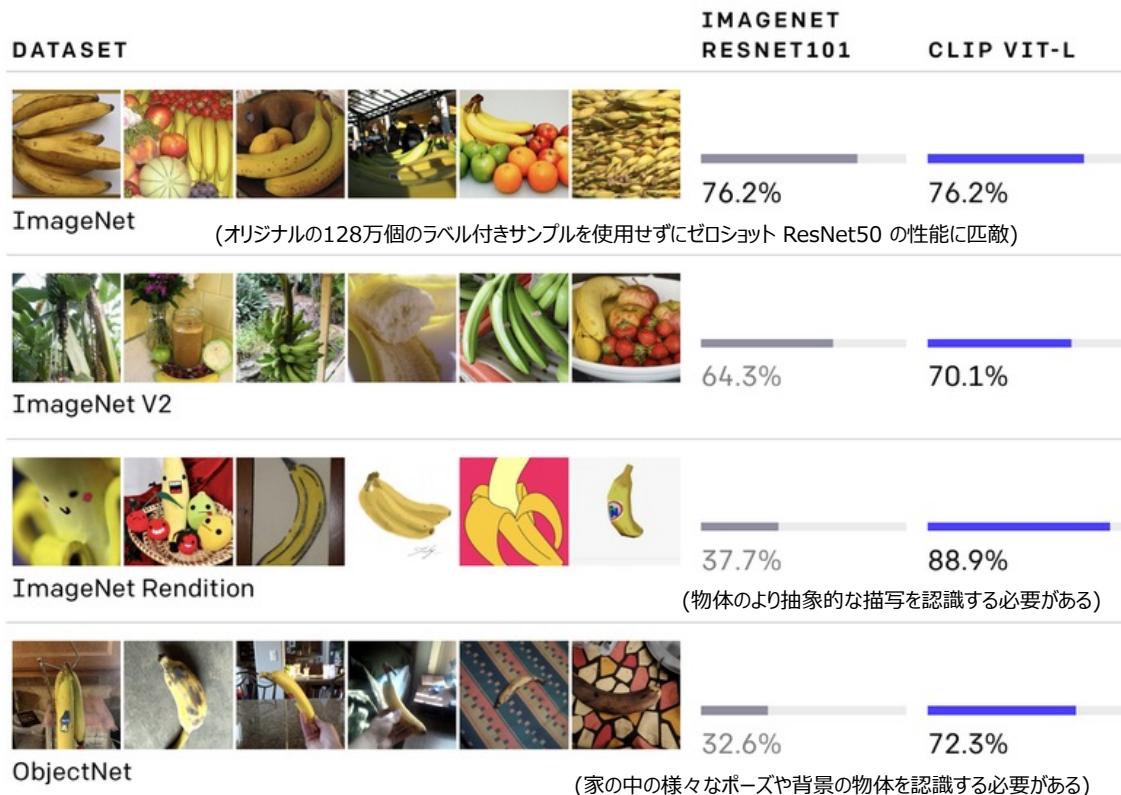


- 巨大な Transformer デコーダによる Text-to-image モデル
 - 最大 120億パラメタ (ViTの約20倍)
- 大量の画像と説明文ペアから学習、生成画像のレベルが高い
- 画像は1024(32x32)のコード系列(8192種)として扱う

<https://openai.com/blog/dall-e/>

CLIP [Radford+ (OpenAI), 2021]

- 大規模な画像とテキストのペアで zero-shot の画像認識を実現



- 画像とテキストのマッチングを4億ペアから事前学習
 - DALL·E の生成画像のランキングにも使われている
- 正しい画像・テキストペアを分類できるように Contrastive pre-training を行う

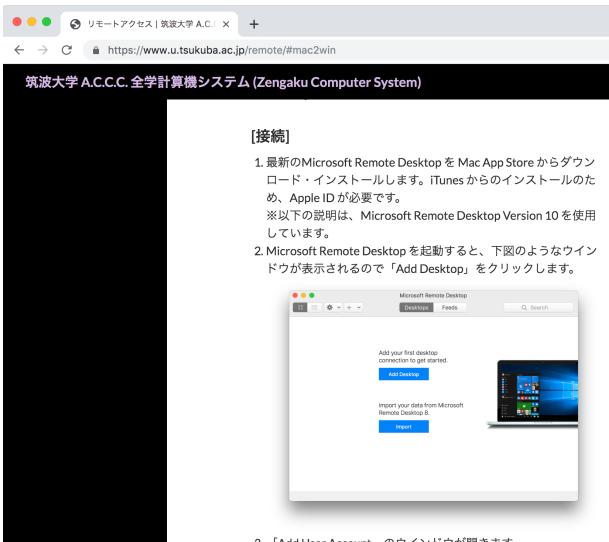
<https://openai.com/blog/clip/>

実習環境について

- 実習では,以下のツールを使用します
 - Part 2 では **Microsoft EXCEL** (用途: データの加工や修正)
 - Part 3 以降では **KHCoder** (用途: テキストマイニング) ※フリーソフト
- **KHCoder** は,全学計算機システムのリモートデスクトップでも動作します
 - 【Win】 <https://www.u.tsukuba.ac.jp/remote/#win2win>
 - 【Mac】 <https://www.u.tsukuba.ac.jp/remote/#mac2win>
- 個人のPCで **KHCoder** を使用しても構いません
 - ただし, Windows OS (11, 10, 8.1) を搭載した PC が必要です

全学計算機システムを利用する場合

- 事前に、下記のページの説明に従い全学計算機システム(Windows)へログインができるることを確認しておいてください
 - 【Win】 <https://www.u.tsukuba.ac.jp/remote/#win2win>
 - 【Mac】 <https://www.u.tsukuba.ac.jp/remote/#mac2win>



Mac の場合:

左記のページにある説明に従って、事前にツール [Microsoft Remote Desktop](#) のインストールが必要です

KH Coder インストール時の注意:

全学の Windows では C ドライブへのファイル保存は禁止されています。ダウンロードした KH Coder を解凍する場合は、保存先を「**C ドライブ以外**」に変更してください。例)「**Z:¥Desktop¥khcoder3**」

KH Coder のインストール (Part 3 以降で使用)

- ・ダウンロードとインストール <https://khcoder.net/dl3.html>



- ① ここをクリックすると遷移先のページからダウンロードが始まります
- ② ダウンロードしたファイルを実行（ダブルクリックし、開いた画面上の「Unzip」ボタンをクリックします。）
- ③ 保存先を「**Cドライブ以外**」(**Cドライブへの保存は禁止されています**)に変更します。例)
'Z:¥Desktop¥khcoder3'
- ④ 指定した保存先フォルダにすべてのファイルが解凍されます。解凍された「**kh_coder.exe**」を実行すると KH Coder が起動します。

Q&A

参考書

(KH Coder)

- [1] 横口耕一. 社会調査のための計量テキスト分析—内容分析の継承と発展を目指して
【第2版】 KH Coder オフィシャルブック. ナカニシヤ出版, 2020.
- [2] 横口耕一. テキスト型データの計量的分析—2つのアプローチの峻別と統合—. 理論
と方法, 数理社会学会, 2004, 19(1): 101-115.
- [3] 牛澤賢二. やってみよう テキストマイニング—自由回答アンケートの分析に挑戦!.
朝倉書店, 2019
- [4] 横口耕一. 動かして学ぶ! はじめてのテキストマイニング: フリー・ソフトウェアを
用いた自由記述の計量テキスト分析 KH Coder オフィシャルブック II. ナカニシヤ
出版, 2022.

New

(Windows環境によるデータ収集方法の参考に)

- [5] テキストマイニングソフトを利用した新未来洞察手法の研究. 第10分科会, (財)市場
創造研究会. http://www.shijo-sozo.org/news/第10分科会_1.pdf

参考書

(Rを使った参考書)

- [6] 金明哲. "テキストデータの統計科学入門." 岩波書店, 2009.
- [7] 石田基広. "RMeCabによるテキスト解析. Rによるテキストマイニング入門." 森北出版, 2008, 51-82.

(他のツールを使った参考書)

- [8] 那須川哲哉. "テキストマイニングを使う技術/作る技術: 基礎技術と適用事例から導く本質と活用法." 東京電機大学出版局, 2006.
- [9] 上田隆穂, 黒岩祥太, 戸谷圭子. "テキストマイニングによるマーケティング調査." 講談社, 2005.

(統計解析を中心とした参考書)

- [10] 前田忠彦; 山崎誠. 言語研究のための統計入門. くろしお出版株式会社, 東京, 2013.