

# 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)

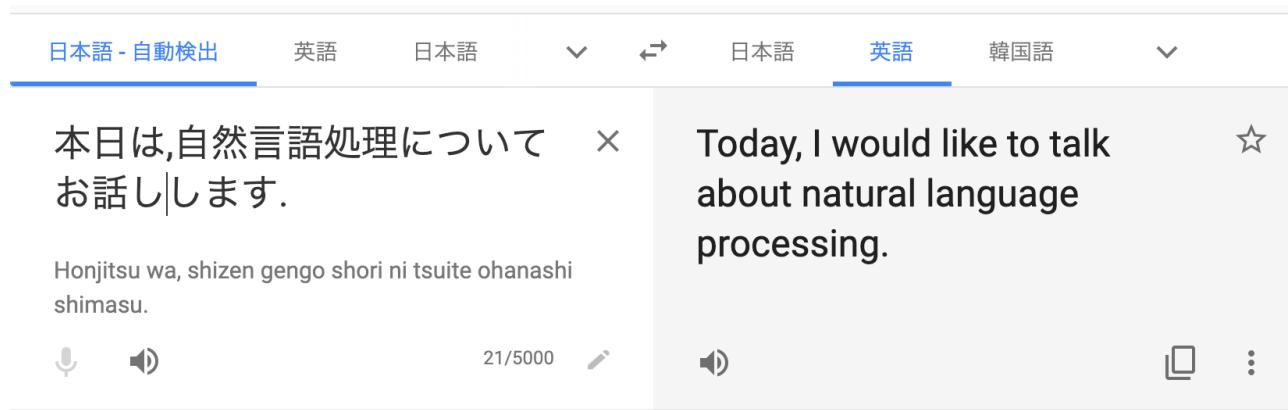
# 自然言語処理のトレンド

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

- 機械翻訳



- AIアシスタント



- 検索



自然言語をコンピュータ  
で処理するための技術

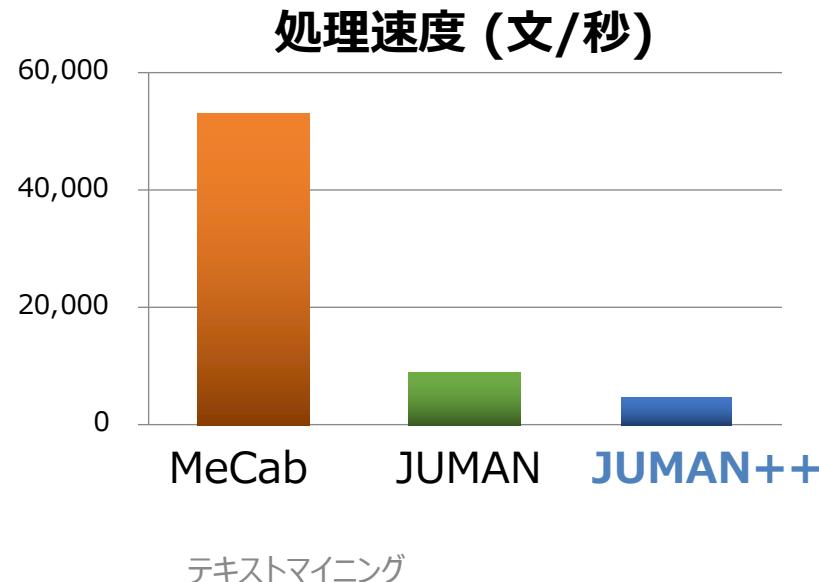
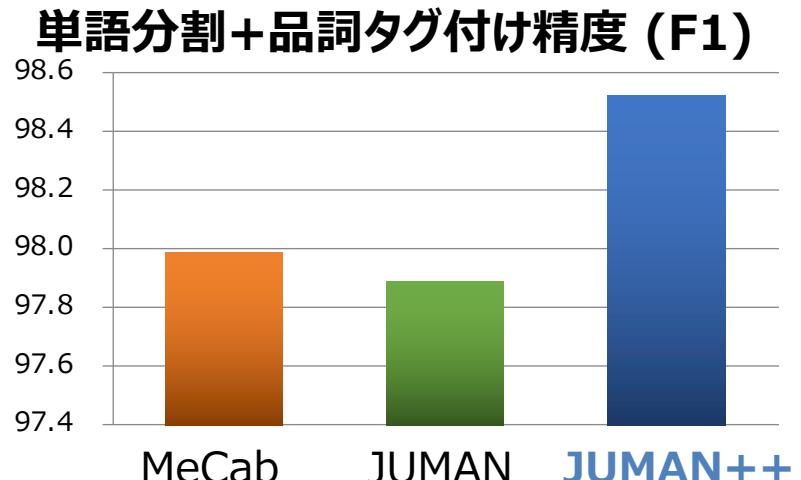
# 自然言語処理

- 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  
例: 形態素解析(品詞タグ付け), 構文解析, 意味解析など
- 応用タスク
  - 自然言語処理を応用したアプリケーション  
例: 機械翻訳, 質問応答, テキスト要約, 対話システムなど

# 形態素解析器

出所: <https://taku910.github.io/mecab/> をもとに作成

形態素解析器	ChaSen	MeCab	JUMAN	JUMAN++
コスト推定	HMM	CRF	人手	RNNLM
探索方法	接続コスト最小法 (ビタビアルゴリズム)			
連携する構文解析器	Cabocha	Cabocha	KNP	<b>深層学習</b> を使った手法で、 <b>自然な言葉の繋がり</b> を考慮



**学習・評価データ**  
京都大学テキストコーパス (NEWS),  
京都大学ウェブ文書リードコーパス (WEB)

**RNN言語モデルの学習**  
Webコーパス 1000万文

出所:  
[https://drive.google.com/file/d/1DVnrsWw4skRgC8jU6\\_RkeofOQEHFwctc/view?usp=sharing](https://drive.google.com/file/d/1DVnrsWw4skRgC8jU6_RkeofOQEHFwctc/view?usp=sharing)

# (参考) 形態素解析の辞書

辞書	JUMAN辞書	Ipadic (NAIST-jdic)	UniDic	NEologd
コーパス	京都大学テキスト コーパス	RWCコーパス <sup>*2</sup>	BCCWJ コアデータ <sup>*3</sup>	RWCコーパス
形態素解析器	JUMAN MeCab	Chasen MeCab	MeCab	MeCab
単語長	長い	やや短い	短い	とても長い

\*1 毎日新聞 1995年の記事や社説 4万文

\*2 旧通産省主導のプロジェクト,毎日新聞1994年3000記事 約3万7千文(約91万語)

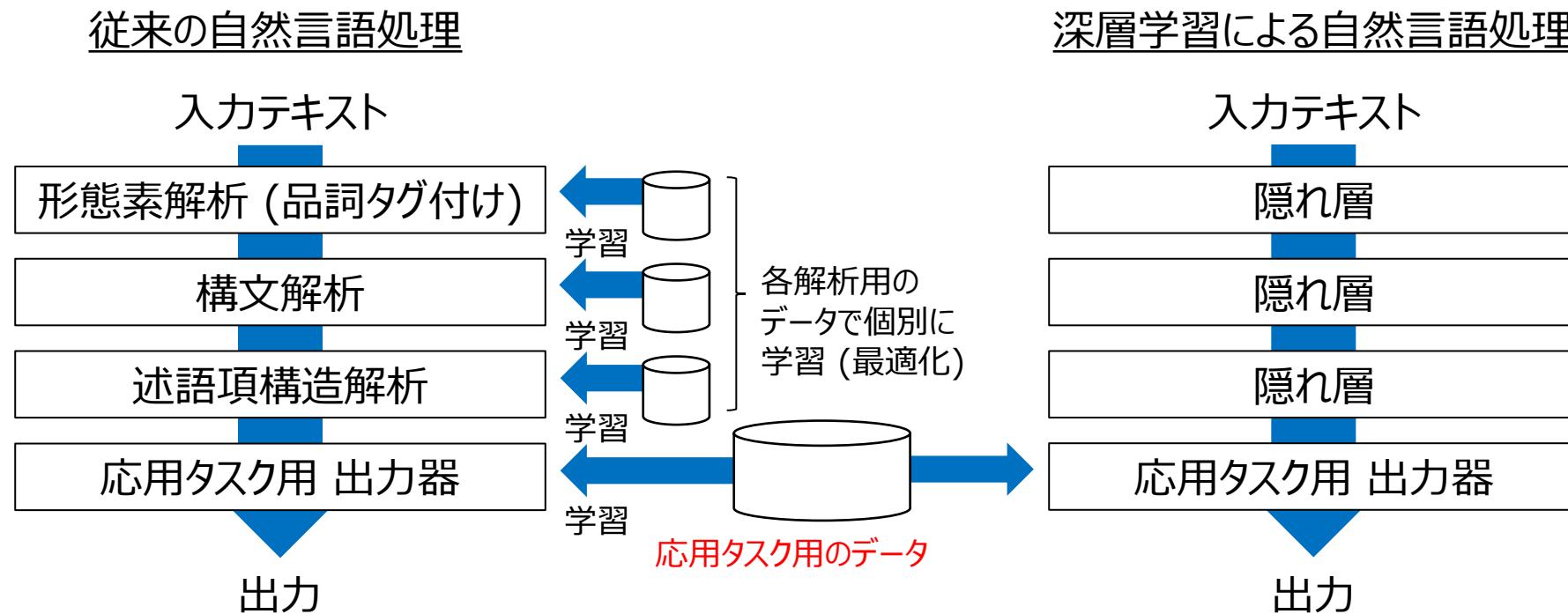
\*3 現代日本語書き言葉均衡コーパス, 国語研が中心となり開発, 書籍,雑誌,新聞,Webなど, 9万単語

# 自然言語処理タスクの例

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

# トレンドは、深層学習の導入

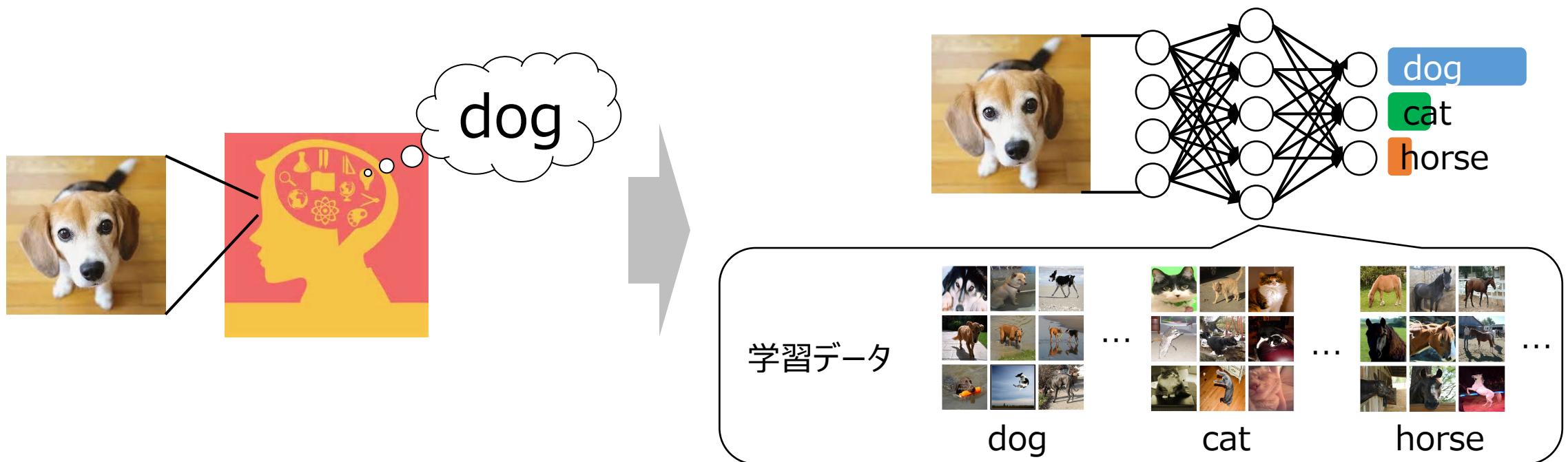
- 大規模な訓練データで応用タスク全体を学習 → End-to-end 学習



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（ディープラーニング）

- ニューラルネット(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

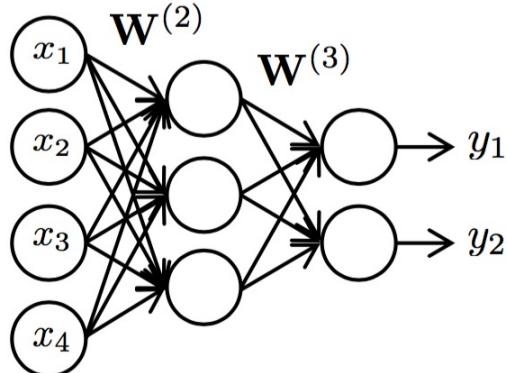


# ニューラルネットの歴史

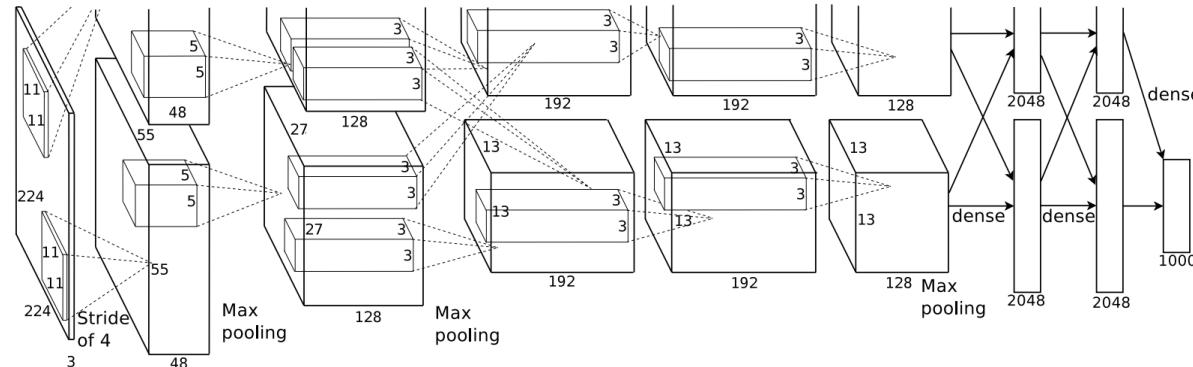
- 黎明～終焉を繰り返し,近年は 3度目のブーム

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]
第 3 期	2000～	• LeCun らが畳み込みニューラルネット Conv.net を提案 [LeCun,89]
		冬の時代 (学習時間や過学習に課題, 一方でSVMが流行)
		• 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]

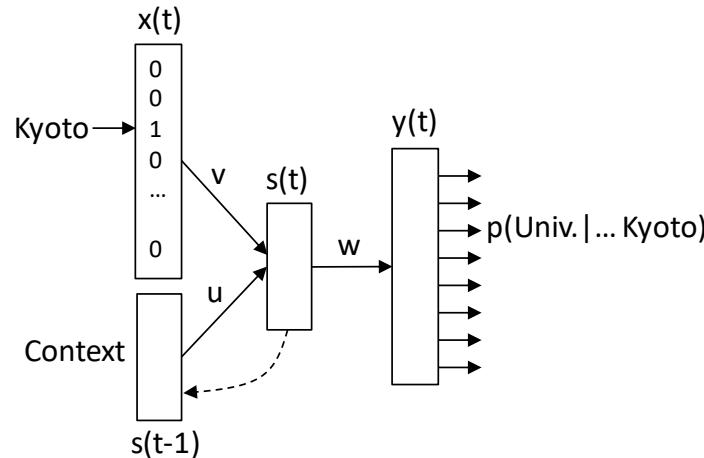
# 様々なニューラルネット



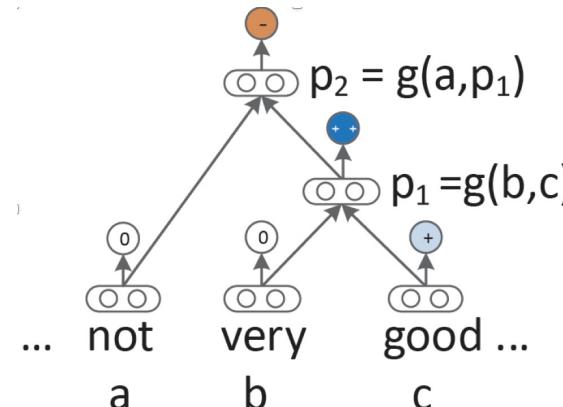
Feed forward NN



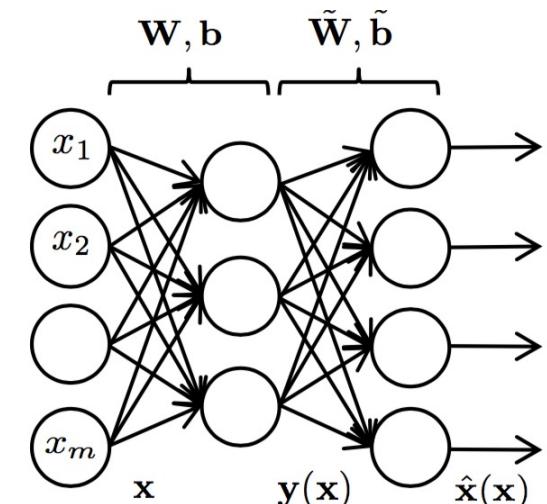
CNN (畳み込みNN)



RNN (Recurrent NN)



Recursive NN



AutoEncoder

# 音声認識で成功 [Seide+, 2011]

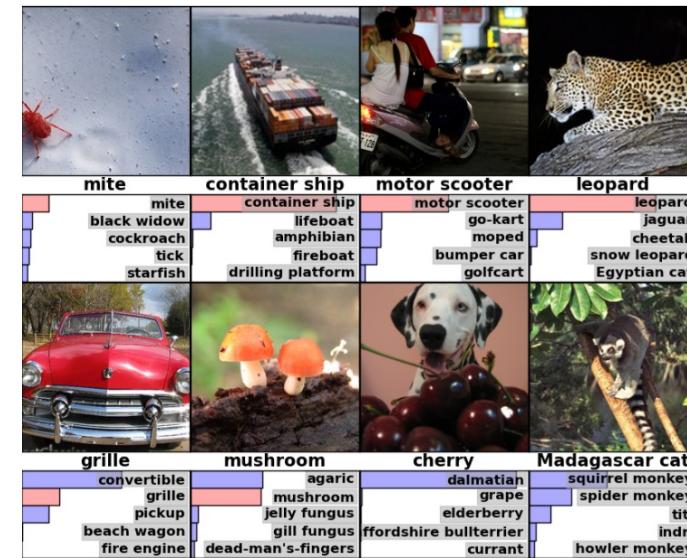
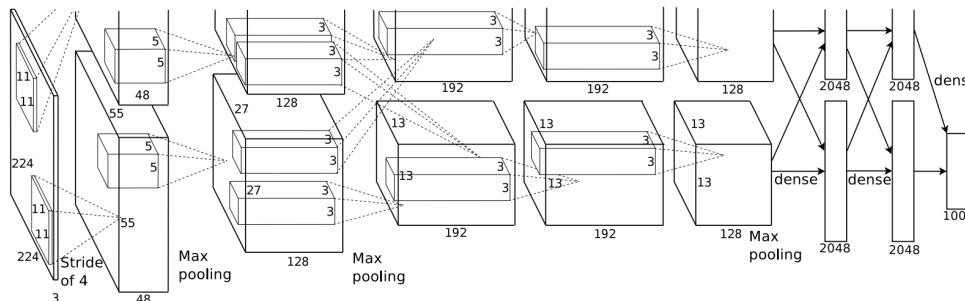
- Microsoft Research のグループ
  - 電話での会話音声の標準データセット
  - 入力(MFCC)-出力(HMM状態変数)の関係をDNNで学習
    - 従来 GMM-HMM → DNN-HMM (全結合7層, 事前学習あり)
  - 単語誤認識率で 10%前後の大幅な精度改善

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.

# 画像認識で成功 [Krizhevsky+, 2012]

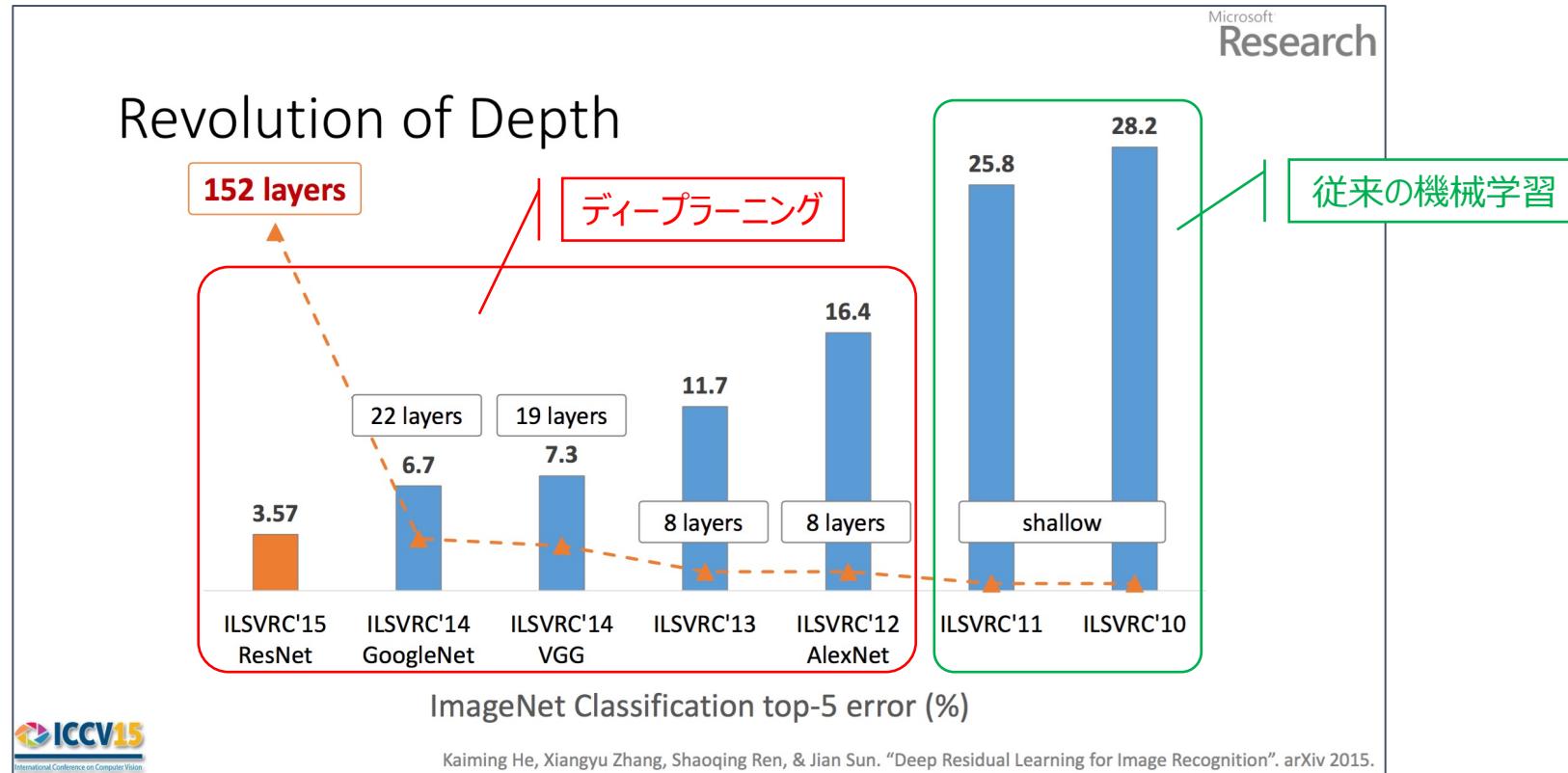
- ・一般物体認識 (Hintonのグループ)
  - ・ImageNet Large-scale Visual Recognition Challenge 2012
    - ・1000カテゴリ×約1000枚 = 100万枚 の訓練画像
    - ・畳込み層5, 全結合層3, 2つのGPUで2週間 (AlexNet)
    - ・誤識別率が10%以上減少 (過去数年間での向上は1~2%)



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>

# 画像認識における認識精度の変遷

- 2015年、人の認識精度(5.1%)を超えたことが話題に



# 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 → **Explainable AI (説明可能AI)**として研究が盛ん

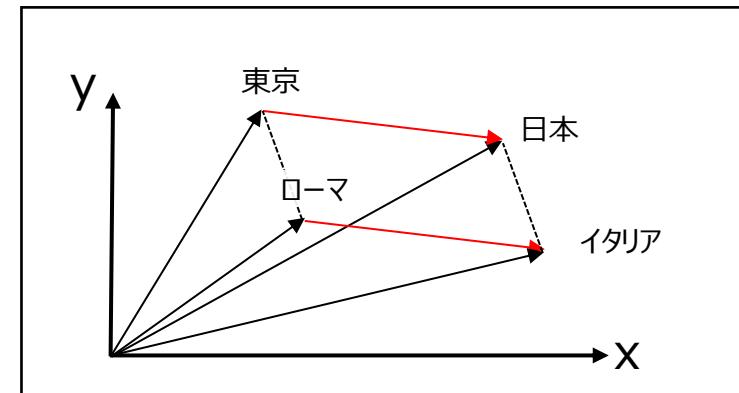
# 事前学習モデル –自然言語処理のブレイクスルー(ひと昔前)

- 大規模コーパスによる事前学習 → 単語のベクトル化 (分散表現)

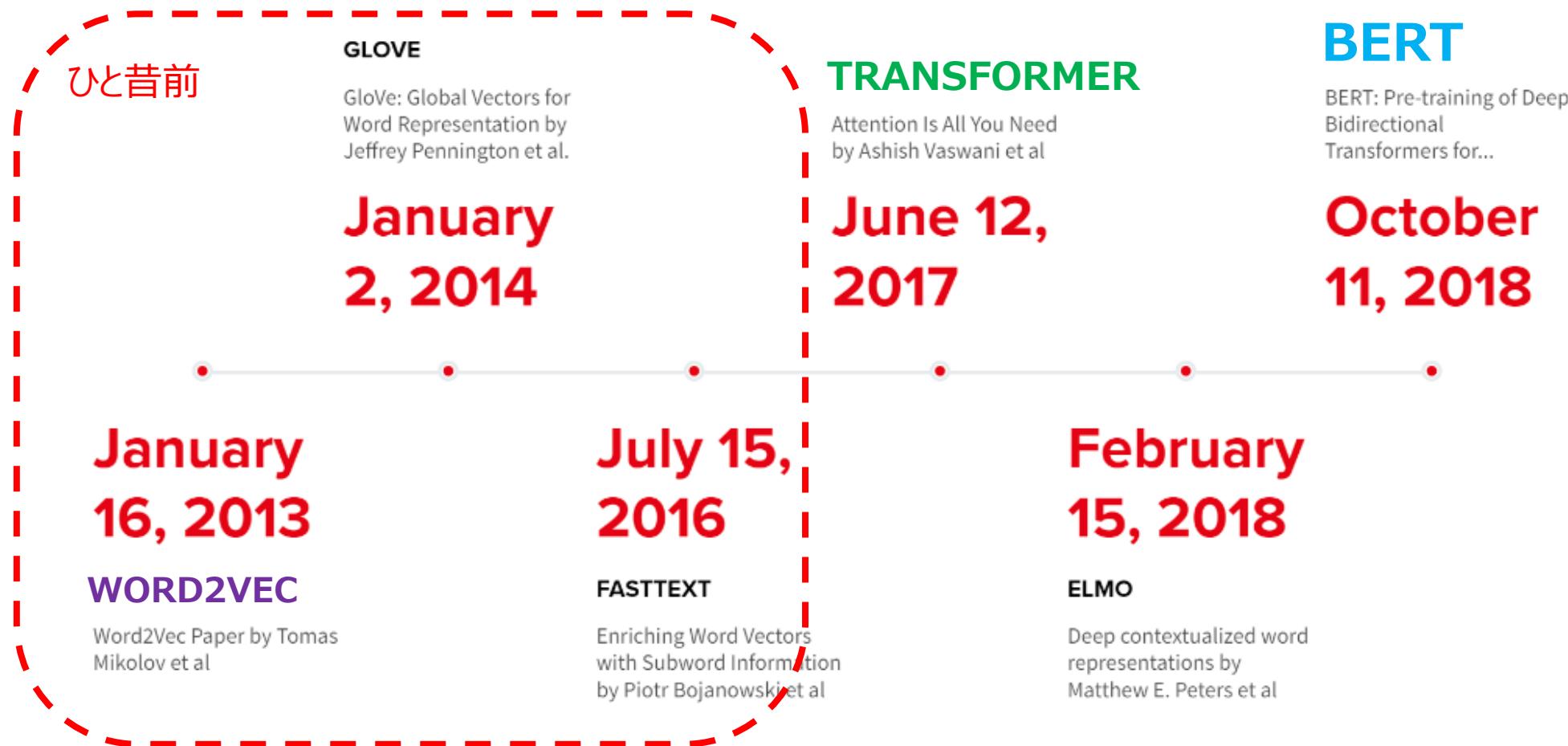
既存手法	近年→事前学習モデル
<b>TF-IDF</b> , Okapi BM25 など (分布的, 高次元, スパース)	<b>word2vec</b> , GloVe, fastText など (分散的, 低次元, 密)

- 代表格は「**word2vec**」
  - 深層学習による分布仮説のモデル化
  - $\text{king} - \text{man} + \text{woman} = \text{queen}$  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



# 事前学習モデルのタイムライン



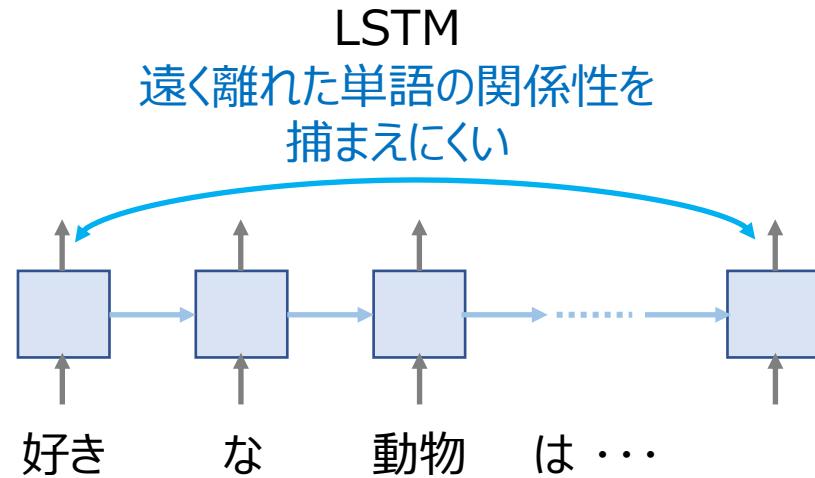
<https://towardsdatascience.com/2019-year-of-bert-and-transformer-f200b53d05b9>

# (参考) 深層学習の適用

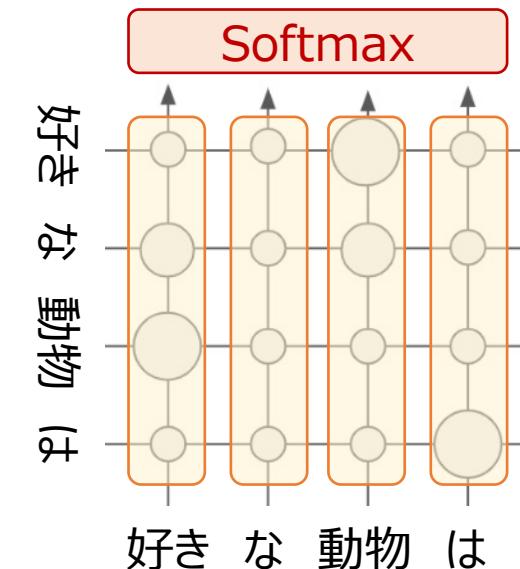
分類	タスクの例	既存手法	深層学習による手法
基礎タスク	言語モデル	• N-gram	• Recurrent NN (RNNLM)
	分散表現	• TF-IDF • Okapi BM25	• word2vec → <b>BERT</b>
	品詞タグ付け	• CRF • SVM	• Encoder-Decoder → <b>BERT</b> ※ Seq2Seq や Attention機構を含む
応用タスク	文書分類	• TF-IDF • Okapi BM25	• Recurrent NN ※前の語を考慮 • Recursive NN ※木構造を考慮 • Convolutional NN ※付近の語を考慮 } → <b>BERT</b>
	機械翻訳	• 統計的機械翻訳	• Encoder-Decoder → <b>Transformer</b> ※ 対訳コーパスを end-to-end で学習する
	文書要約	• SVM • 最大被覆問題	• Encoder-Decoder → <b>Transformer</b> ※ 原文と要約文を end-to-end で学習する

# 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)

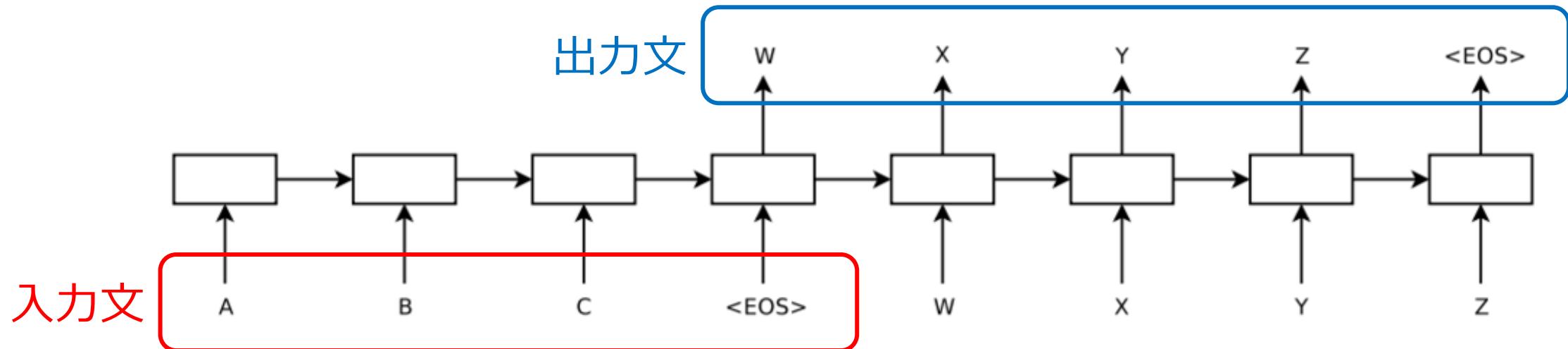
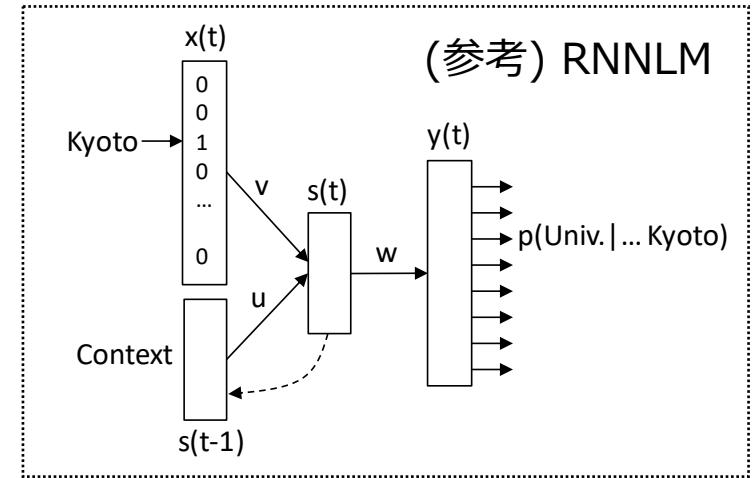


Self-Attention  
遠く離れた単語も  
直接関係性を考慮できる



# (参考) Transformer 以前

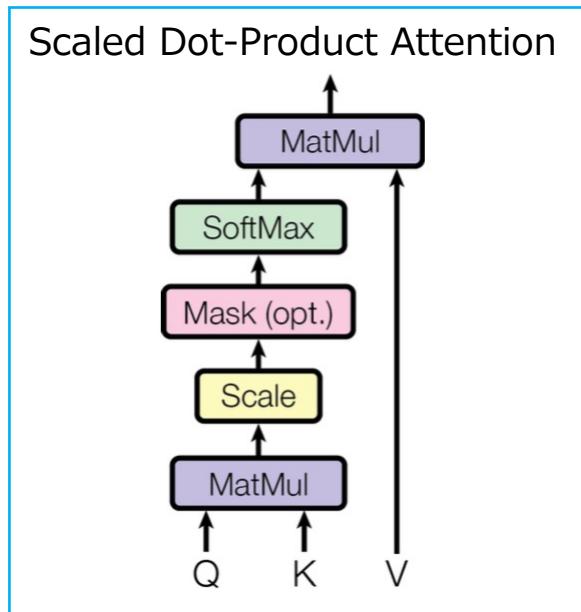
- ニューラル機械翻訳の基本となったモデル
  - Seq2Seq [Sutskever+, NIPS2014]



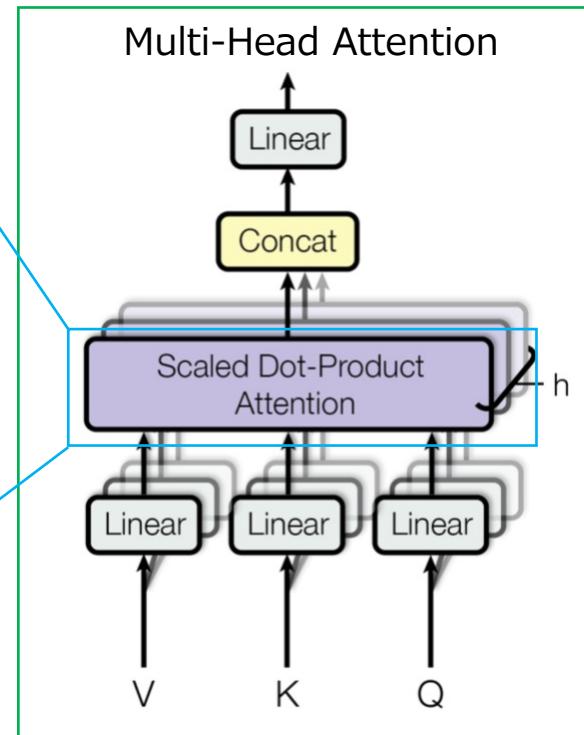
“ABC”という単語列から“WXYZ”という単語列への翻訳

# Transformer [Vaswani+, 2017]

- 例: レイヤーN=6, ヘッドh=8, 長さ=512, 中間層=768



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



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

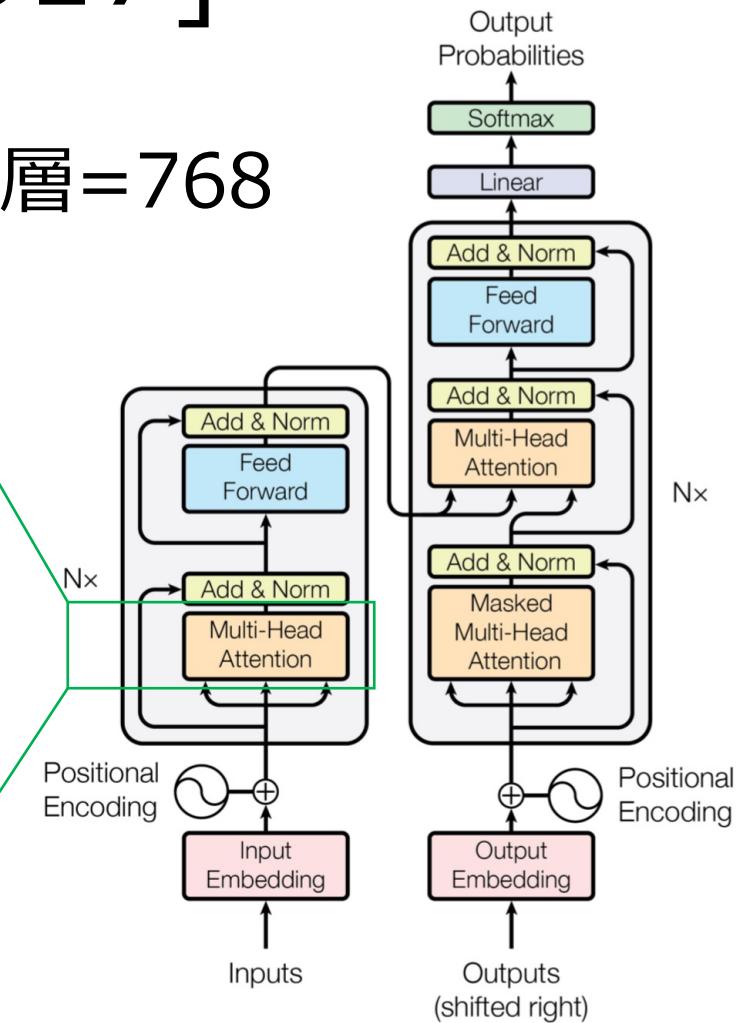
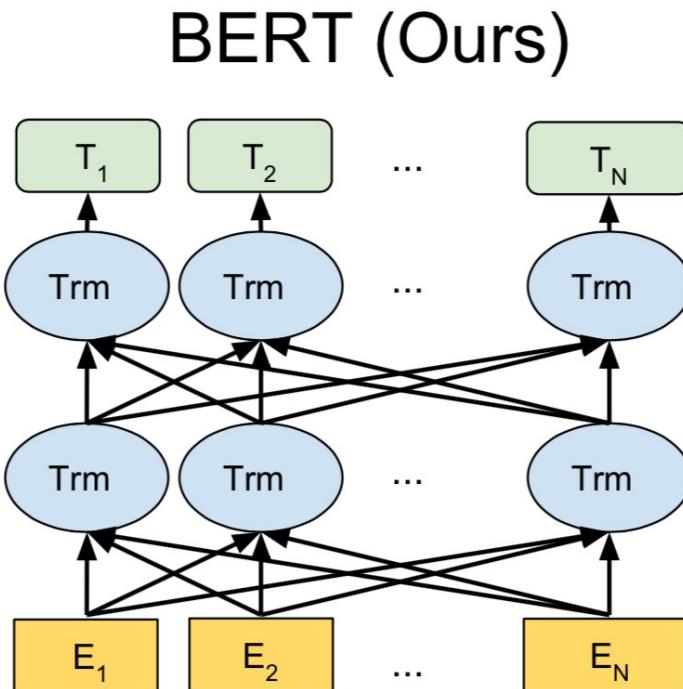


Figure 1: The Transformer - model architecture.

# 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>

# 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> <a href="https://arxiv.org/abs/1810.04805">https://arxiv.org/abs/1810.04805</a>	87.433	93.160
2	BERT (single model) <i>Google AI Language</i> <a href="https://arxiv.org/abs/1810.04805">https://arxiv.org/abs/1810.04805</a>	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/>

# 文脈を考慮した表現

- 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 style="text-align: center;">↓ 同じ</p>	<p>首を痛める</p> <p>首 </p> <p>会社を首になる</p> <p>首 </p> <p style="text-align: center;">↓ 異なる</p>

# BERT 事前学習モデル

公開元	<a href="#">Google Research</a>	<a href="#">京大 黒橋・河原・村脇研</a>	<a href="#">NICT</a>	<a href="#">東北大 乾・鈴木研</a>
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)

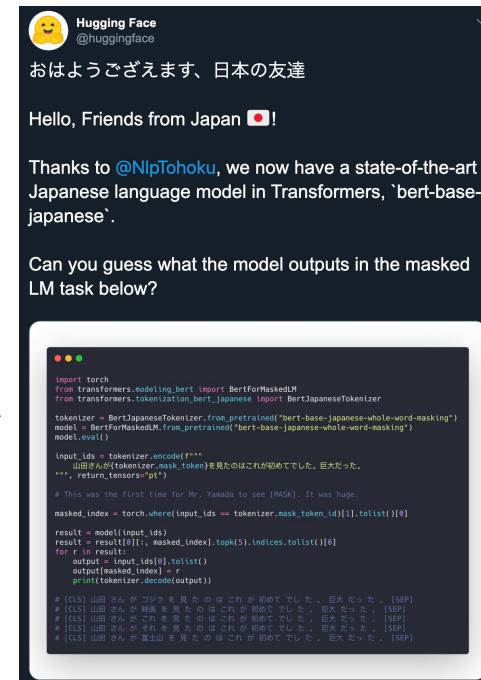
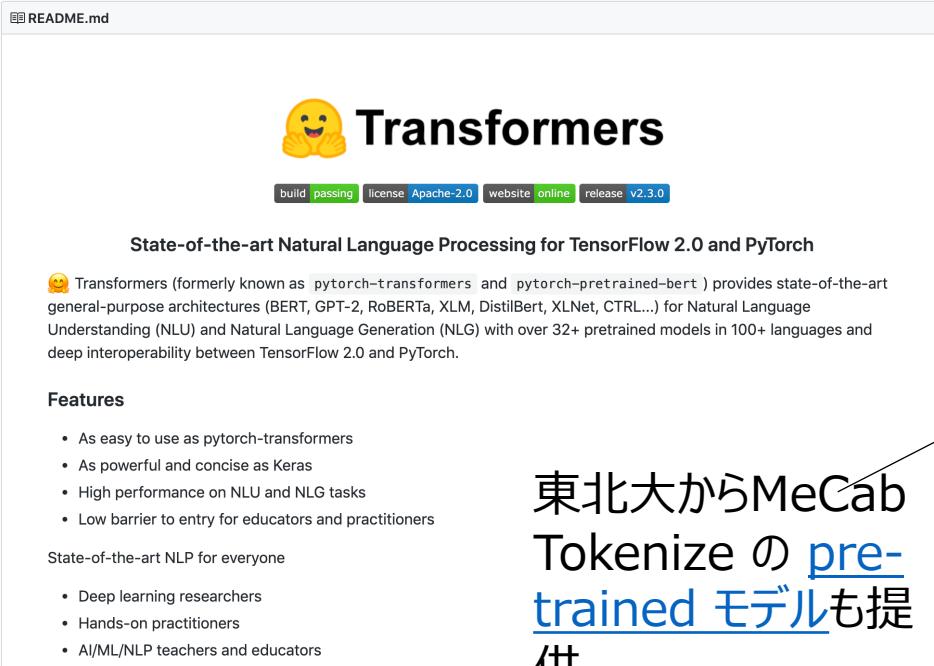
\*1 入力できるシーケンスの長さに制限があることに注意

\*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

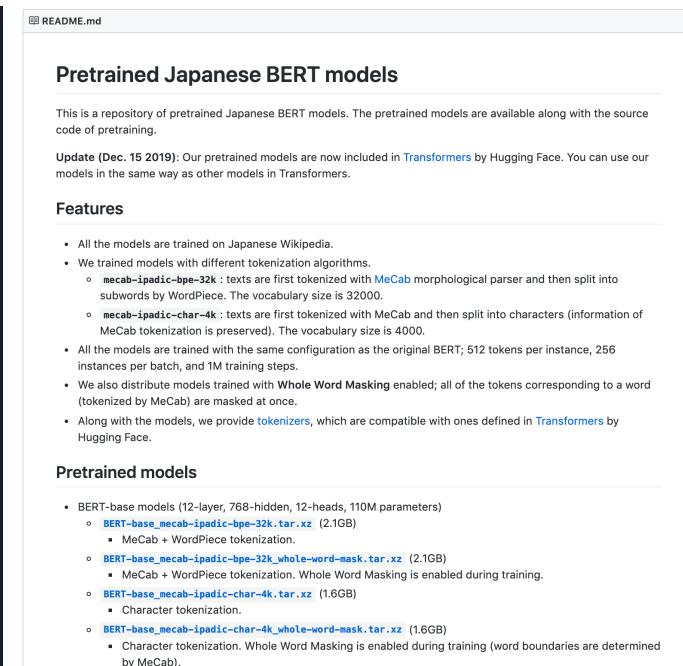
# HuggingFace's Transformers

<https://huggingface.co/>

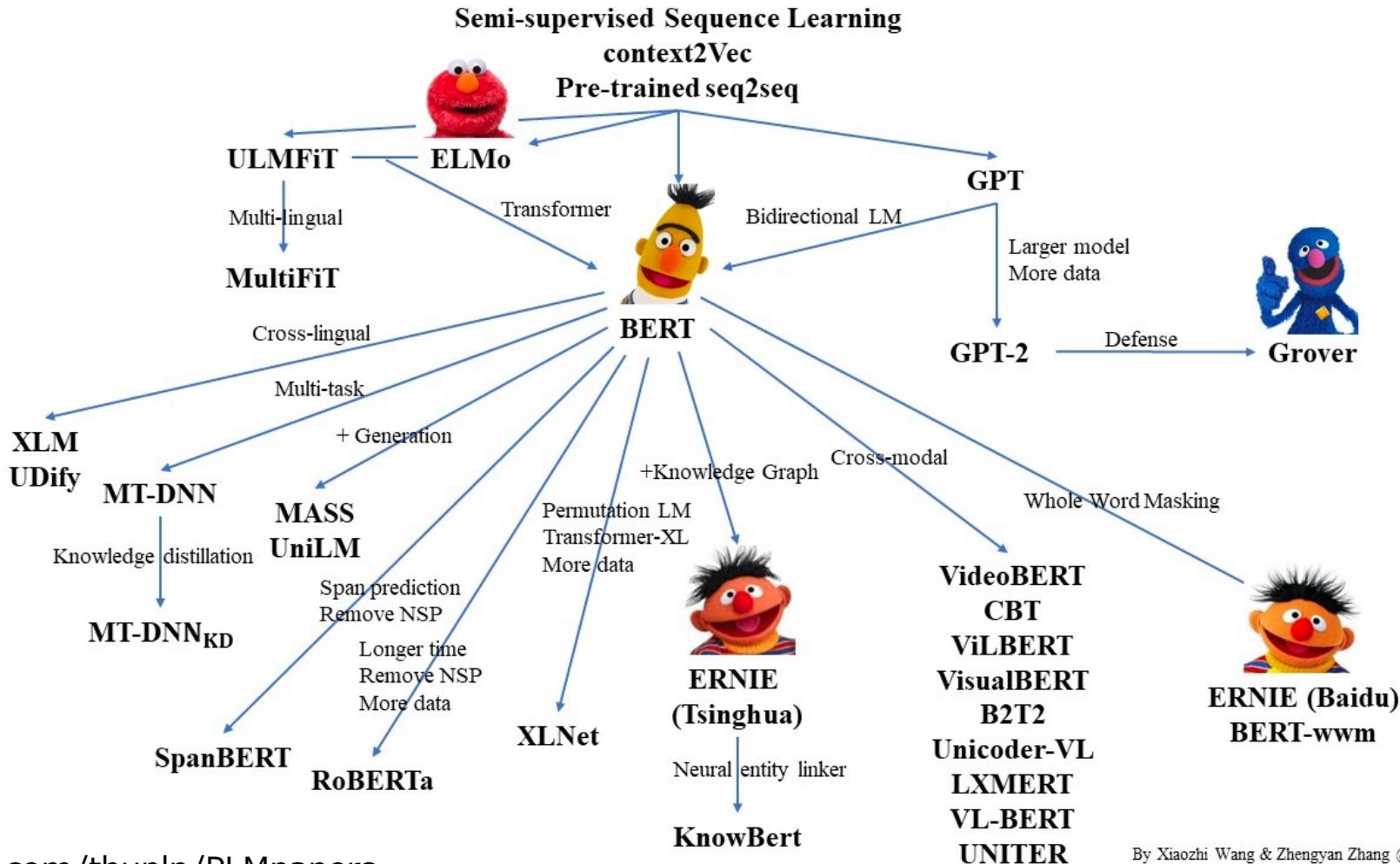
- Huggingface が提供する Pytorch によるフレームワーク
- 簡単にBERTなどの汎用言語モデルを動かせる



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



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

- GPT-1<sub>(1億)</sub>, GPT-2<sub>(15億)</sub>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

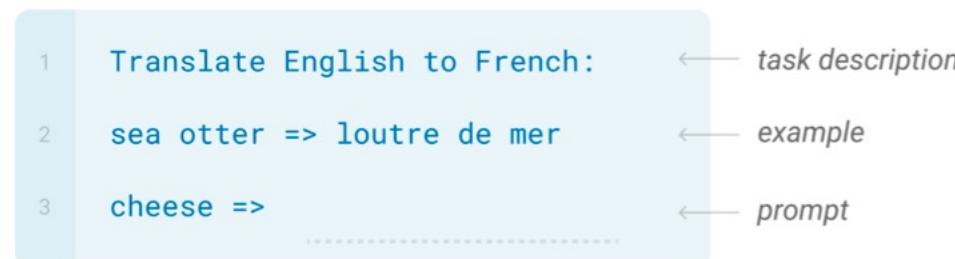
### Zero-shot

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



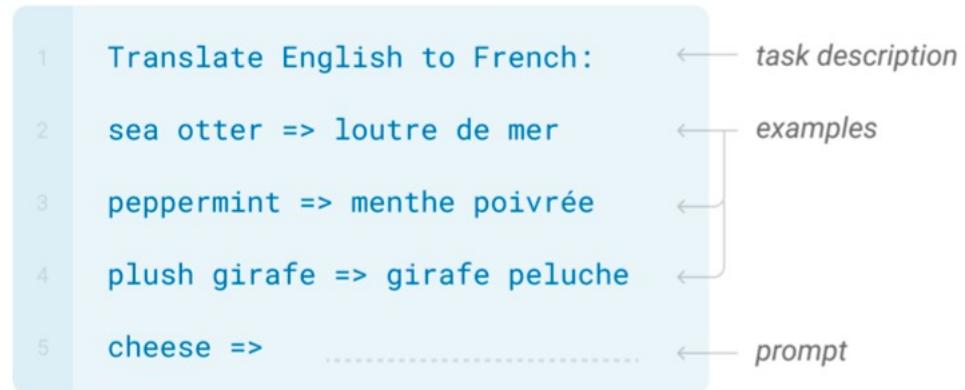
### One-shot

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.

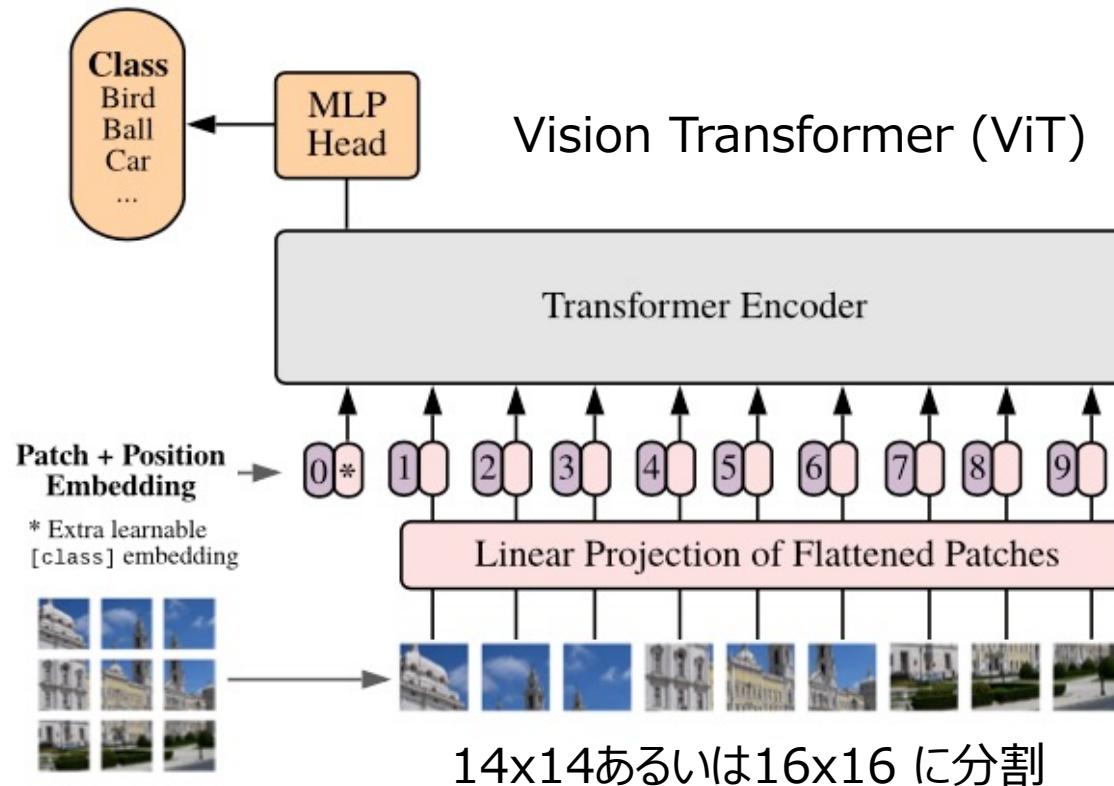


The screenshot shows the OpenAI Playground interface. At the top, there are navigation links: OpenAI Beta, Playground, Documentation, and Examples. Below that, the title "Playground" is displayed with a help icon. The main content area contains a list of Japanese calendar conversions:

- 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](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

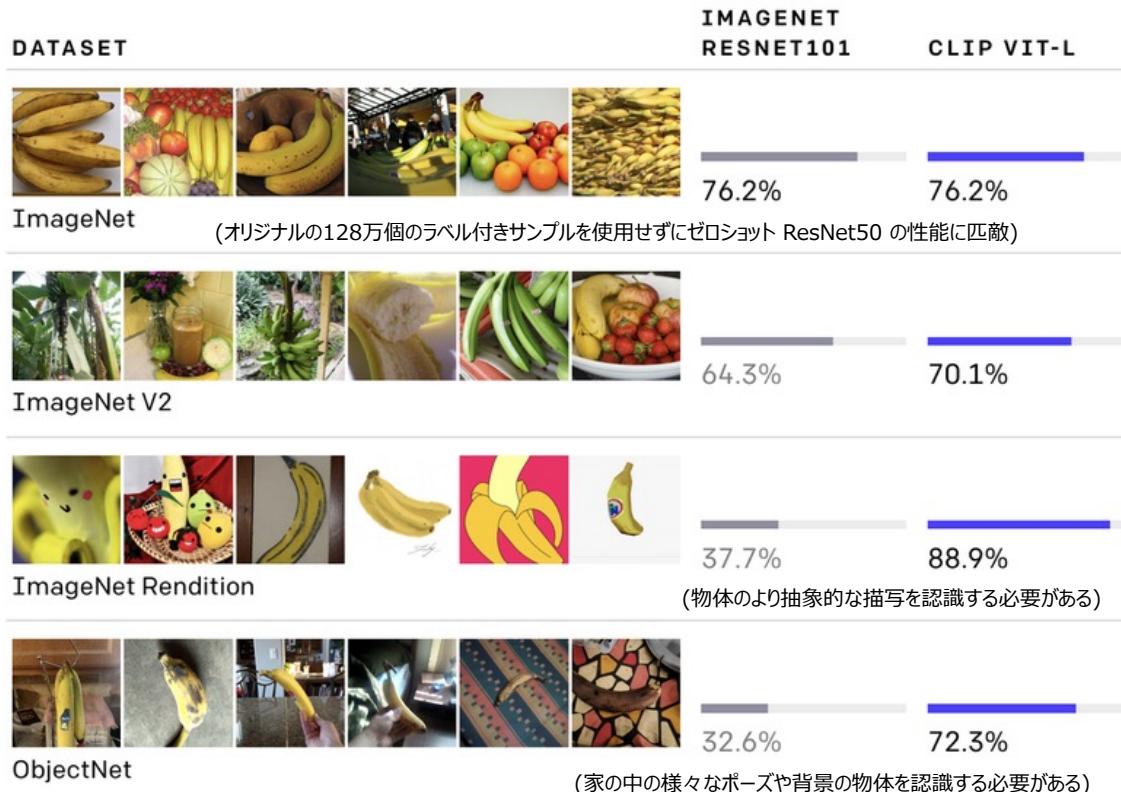


- 巨大な Transformer デコーダによる Text-to-image モデル
  - 最大 120 億パラメタ (ViT の約20倍)
- 大量の画像と説明文ペアから学習、生成画像のレベルが高い
- 画像は1024(32x32)のコード系列(8192種)として扱う

<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

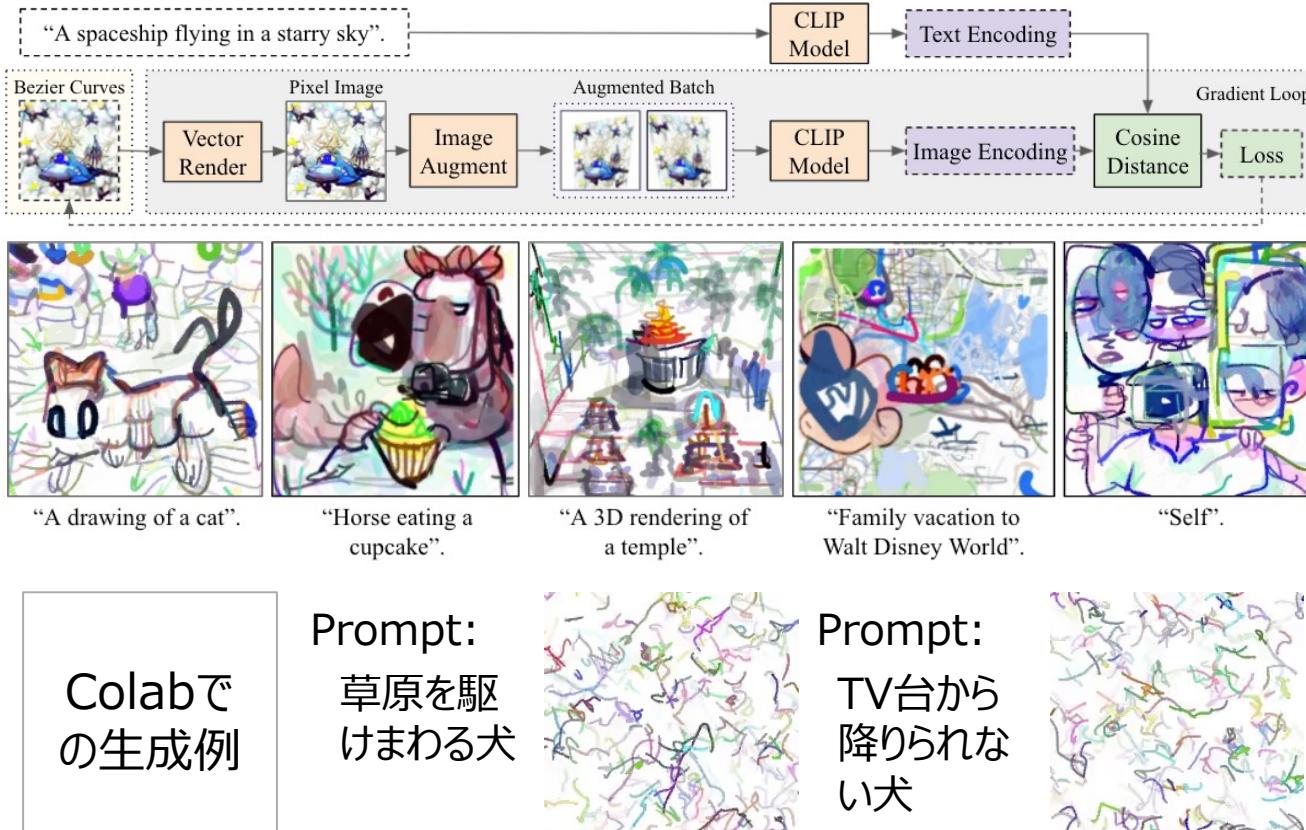


- 画像とテキストのマッチングを4億ペアから事前学習
  - DALL·E の生成画像のランキングにも使われている
- 正しい画像・テキストペアを分類できるように Contrastive pre-training を行う

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

# CLIPDraw [Frans+, 2021]

- CLIPを使って、言葉の指示に合わせたスケッチアートを作成



- 言葉は CLIP で符号化, 絵は固定数のベジエ曲線で初期化した後に CLIP で符号化
- 言葉と絵の間のcos距離が合うようベジエ曲線のパラメータを誤差逆伝搬で最適化
- 抽象的な言葉も扱え, 絵が生成される過程も面白い
- Google Colab で試せる

<https://kvfrans.com/clipdraw-exploring-text-to-drawing-synthesis/>

# (参考) Colaboratory とは

- 機械学習の教育・研究を目的とした研究用ツール



- **設定不要** (最初から Python や機械学習に必要なものが入っている)
- **無料で使える** (Googleアカウントさえあれば良い)
- **ブラウザで動作する** (PCのスペックが低くても関係なし)
- **GPUが無料で使える** (計算時間を大幅に短縮できる)
- **ただし、90分&12時間ルールあり** <sup>\*1</sup>

\*1 Colab Pro (1,072円/月)にすることで各種制限を緩和できます

# まとめ

- ・**深層学習の発展とともに自然言語処理も進化**
  - ・応用タスクの学習データで End-to-end で学習可能
  - ・ブラックボックスのため、出力の解釈が難しい等の課題もある
- ・**自然言語処理研究を超えて Transformer が席卷中**
  - ・**BERT** (Transformer) テキスト分類、機械翻訳、質問応答(機械読解)、要約、文生成 など様々な自然言語処理の応用タスクの性能を向上
  - ・**Transformer** が、画像認識など自然言語処理以外の領域でも成果を発揮しつつある
- ・トレンドは **大規模モデル** と **視覚+言語の事前学習**、研究も盛ん

# 文献

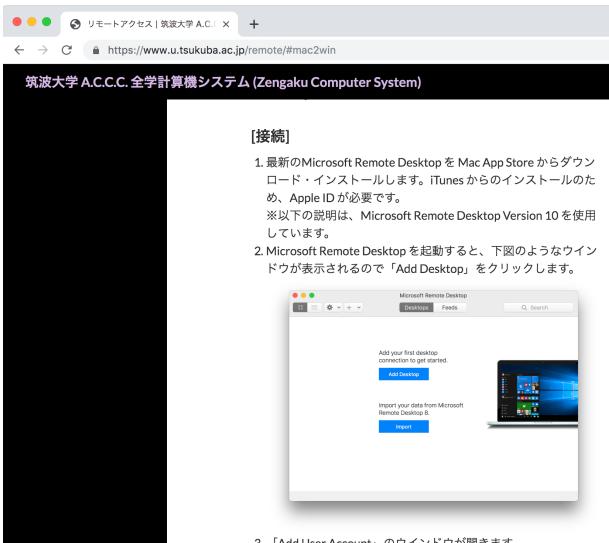
- [Seide+,2011] F. Seide, G. Li and D. Yu, "**Conversational Speech Transcription Using Context-Dependent Deep Neural Networks.**" Interspeech. 2011.
- [Krizhevsky+,2012] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "**Imagenet classification with deep convolutional neural networks.**" Advances in neural information processing systems. 2012.
- [Sutskever+,NIPS2014] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "**Sequence to sequence learning with neural networks.**" Advances in neural information processing systems. 2014.
- [Vaswani+,2017] Vaswani, Ashish, et al. "**Attention is all you need.**" Advances in neural information processing systems. 2017.
- [Devlin+,2018] Devlin, Jacob, et al. "**Bert: pre-training of deep bidirectional transformers for language understanding.**" arXiv preprint arXiv:1810.04805 (2018).
- [Brown+ (OpenAI), 2020] Brown, Tom B., et al. "**Language models are few-shot learners.**" arXiv preprint arXiv:2005.14165 (2020).
- [Dosovitskiy+, 2021] Dosovitskiy, Alexey, et al. "**An image is worth 16x16 words: Transformers for image recognition at scale.**" arXiv preprint arXiv:2010.11929 (2020).
- [Ramesh+ (OpenAI), 2021] Ramesh, Aditya, et al. "**Zero-shot text-to-image generation.**" arXiv preprint arXiv:2102.12092 (2021).
- [Radford+ (OpenAI), 2021] Radford, Alec, et al. "**Learning transferable visual models from natural language supervision.**" arXiv preprint arXiv:2103.00020 (2021).
- [Frans+,2021] Frans, Kevin, et al. "**CLIPDraw: Exploring Text-to-Drawing Synthesis through Language-Image Encoders.**" arXiv preprint arXiv:2106.14843 (2021).

# 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](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.