ニューラルネットワークによる自然言語処理 –後半-

後半の内容

word2vecの利用・作成
BERTの概要と利用

word2vec概要

2013年発表、「ベクトル空間における単語の表現の効率的な推定」

従来のOne-hot形式では無く、単語を数百次元程度の特徴ベクトルとして表す方法

One-hotだと、語句が1万語なら1万の長さのOne-hotになるが、word2vecなら数百次元ですむ

このベクトル表現方式として、CBoWとSkip-gramの2方式

Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean,
Efficient Estimation of Word Representations in Vector Space, https://arxiv.org/abs/1301.3781

word2vec：具体例

単語の特徴量を（極端だが） 4次元で次のベクトルとすると

俳優 = (0.9, 0.8, 0.8, 0.0)
女優 = (0.1, 0.8, 0.8, 0.0)
子役 = (0.9, 0.1, 0.0, 0.0)
男性 = (1.0, 0.1, 0.0, 0.0)
女性 = (0.2, 0.1, 0.0, 0.0)

ここで、女優－女性＋男性=(0.1, 0.8, 0.8, 0.0) - (0.2, 0.1, 0.0, 0.0) + (0.9, 0.8, 0.8, 0.0) = (0.8, 0.7,0.8, 0.0) となり、これは俳優のベクトル(0.9, 0.8, 0.8, 0.0)とほぼ一致する。このように、単純な計算で単語の関係性を表現できる

word2vec：作成原理

ある例文に対して、入力語と周辺語のセットを作成

I am writing to confirm our meeting on September 8th.
入力：I → [I, am] , [I, writing], [I, to], [I, confirm] ….
入力: am → [am, I], [am, writing], [am, to] …
入力: our → [our, confirm], [our, meeting], [our to] …

周辺語をどの程度集めるかは14-10個前後が多い
このようなセットをコーパス内文章量だけ作成する
語数分のベクトル空間が作成される

ベクトル表現：CBoW

Continuous Bag-of-Words：周りの単語から挟まれている単語を推測

※Continuous：「連続した」の意味だが、時々Countinuousと書かれた本やWebサイトがある。おそらくタイポ

例えばショパンに対して適応する場合、元文章にマスクし、そこを推測する

ポーランド/の/作曲家/の/ショパン/は/幻想/即興/曲/など/で/知られて/いる

↓

ポーランド/の/作曲家/の [???] は/ 幻想/即興/曲/など/で/知られて/いる

このとき、Window幅の設定によってその前後の範囲を調整する

Window幅1: の [???] は

Window幅4: ポーランド/の/作曲家/の [???] は/ 幻想/即興/曲

さらに、周辺単語は1, その他は0とする

CBoWの学習

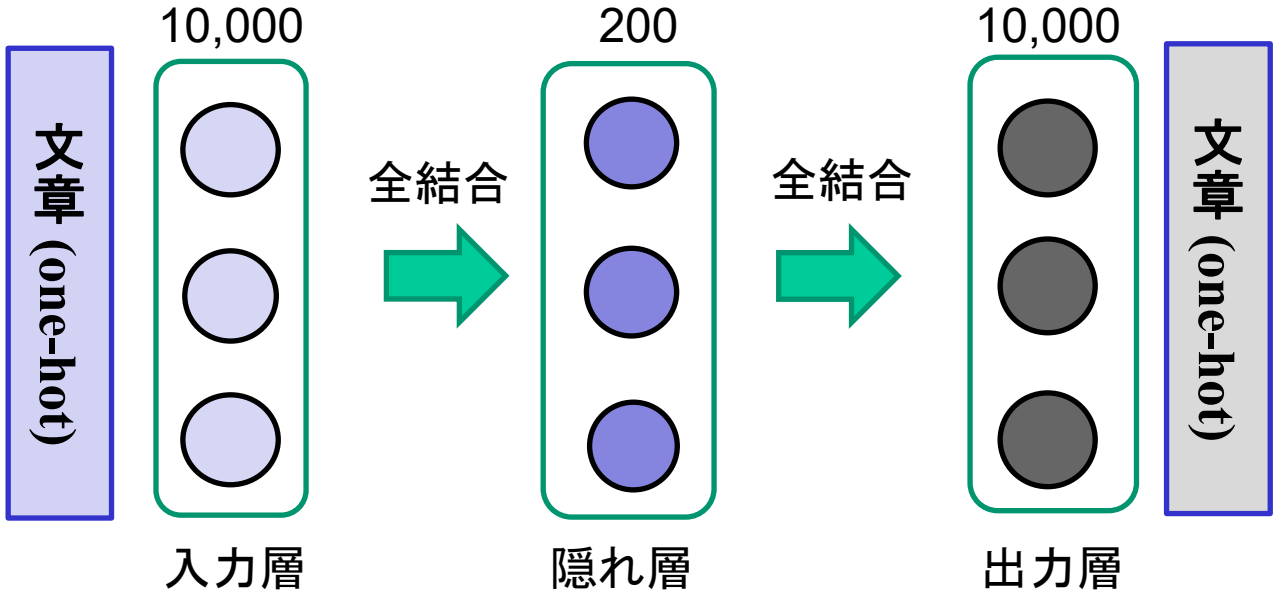
Window幅4の時

ポーランド/の/作曲家/の [???] は/ 幻想/即興/曲/など/で/知られて/いる

1 1 1 1 0 1 1 1 1 0 0 0 0

ここで、単語を200次元ベクトルで表したい場合、入出力層は単語の全数になる。（ここでは1万とする）入出力はone-hot形式で与えられる

入力の重みは10,000×200, 出力は200×10,000



ベクトル表現：Skip-gram

CBoWと逆で、対象単語の周辺を推測する

Window幅4のとき：
ポーランド/の/作曲家/の/ショパン/は/幻想/即興/曲/など/で/知られて/いる
↓
[???][???][???][???]ショパン[???][???][???][???] など/で/知られて/いる

NNへの入出力や処理はCBoWと同じ
CBoWとSkip-gramのどちらを使用するかについて、理論的な結論はない
一般的にはSkip-gramが推奨されている

fasttext

word2vecと同じくMikolovら(2016)による単語ベクトル表現法

word2vecでは、文章に含まれていない単語のベクトル表現を得ることができない。そのため、N-gramに似たこの手法が開発された

英語では、3-6文字程度で分割。また、単語の最初と最後には<>を付ける

例：3文字　apple → <ap, app, ppl, ple, le>
　　4文字　apple → <appl, appl, pple, ple>

日本語では、utf-8の3-6バイトで分割。utf-8は3バイトで1文字なので1, 2文字分割

例：1文字　自然言語 → <自, 然, 言, 語, 語>　※最初と最後の<>はカウントしない
　　2文字　自然言語 → <自然, 然言, 言語, 言語>

Armand Joulin, Edouard Grave, Piotr Bojanowski, Tomas Mikolov,
Bag of Tricks for Efficient Text Classification, https://arxiv.org/abs/1607.01759

word2vecの作成

Tensorflow Githubにword2vec作成のプログラムがある

https://github.com/tensorflow/tensorflow/blob/master/tensorflow/examples/tutorials/word2vec/word2vec_basic.py

10万回学習し、1万回ごとに結果が表示される

例: can

10,000 epoch

Nearest to can: culminating, altenberg, honored, similarities, town, classifications, earliest, successes

100,000epoch

Nearest to can: may, would, will, could, must, should, cannot, might

実行するなら: 13-NLP_word2vec_Engish.ipynb

日本語Wikipediaエンティティベクトルの利用

13-NLP_WikipediaEntity.ipynb
東北大学 乾・岡崎研究室によるベクトル化データを使用
200次元ベクトル
http://www.cl.ecei.tohoku.ac.jp/~m-suzuki/jawiki_vector/
最新版はhttps://github.com/singletongue/WikiEntVec/releases

これを利用して言語の計算が可能
「自衛隊」から「海」を引く

```
model.most_similar(positive=['自衛隊'],negative=['海'])
('陸上自衛隊', 0.4656172394752502 4)
('89式5.56mm小銃', 0.4130284786224365)
('防衛省', 0.4030088484287262)
('第1空挺団', 0.3937991261482239)
```

その他例

「公務員」と「拳銃」の足し算

```
model.most_similar(positive["公務員","拳銃"])
('日本の警察官', 0.7462331056594849)
('警察官', 0.7408863306045532)
('公安職', 0.707878053188324)
```

「イチロー」と「サッカー」を足し「野球」を引く（サッカーでイチロー的な人は？）

```
model.most_similar(positive=['イチロー','サッカー'],negative=['野球'])
('[キャプテン翼の登場人物]', 0.616143822669829)
('[中山雅史]', 0.6087093353271484)
('[松田直樹]', 0.6058454513549805)
```

単語の距離：

国王と王妃“,model.similarity(‘国王’, ‘王妃’)　→　0.6544452

国王と平民“,model.similarity(‘国王’, ‘平民’)　→　0.27823943

word2vecの作成

13-NLP-livedoor_word2vec.ipynb

Livedoorコーパスを利用してword2vecを作成

分かち書きにMeCabを使用する

前回も使用した言語パッケージgensimで作成できる

gensim自体は、元々トピック推定モデル（文章中の主題の推定）のためのモジュール

fasttextの利用

13-NLP-fasttext.ipynb

Facebookが作成・公開しているデータを使用する

https://fasttext.cc/docs/en/crawl-vectors.html

Webクローラによって自動収拾し、300次元にベクトル化したもの

Wikipedia word2vec同様に単語間の足し算や引き算、類語の推測が可能

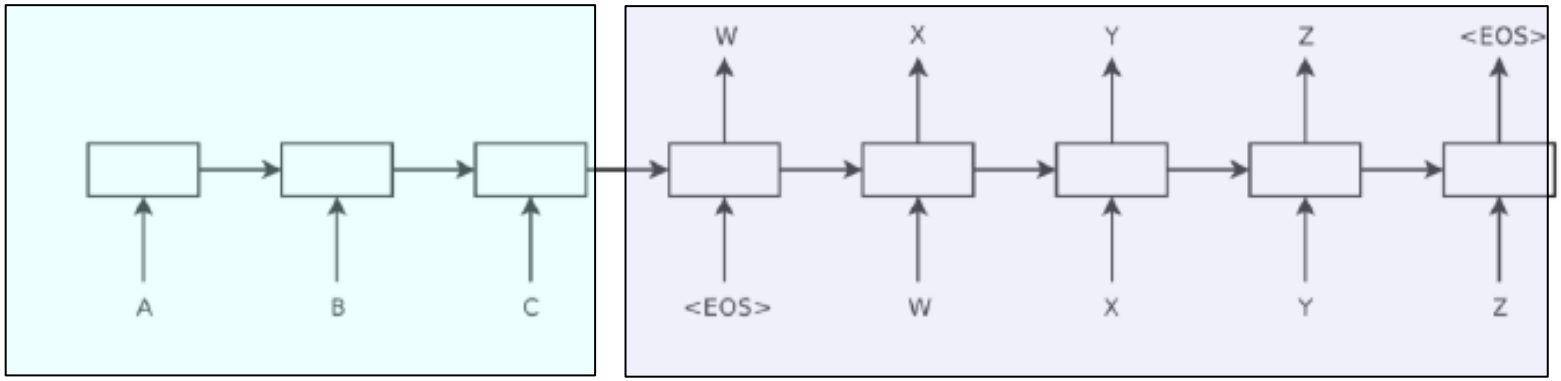
seq2seq（系列変換モデル）

2014年発表「ニューラルネットワークを用いた系列から系列の変換」

RNN, LSTMの拡張系で、統計的な機械翻訳からニューラルネットワークの機械翻訳への移行のきっかけとなったモデル（Googleでも採用）

Encorder-Decorderモデル

出力用RNN (Decorder)



入力用RNN (Encorder)

時系列の文字列(X)を入力し、出力(Y)を得る。Xが日本語、Yが英語とすれば、日本語から英語への機械翻訳

Sequence to Sequence Learning with Neural Networks, https://arxiv.org/abs/1409.3215

企業のNLP実践例

レシピのタイトルから材料を予測する🚀
https://techlife.cookpad.com/entry/2019/02/20/120219
要約
- レシピのタイトルから材料を予測できるモデルを作りました。
- 投稿開発部と協力してレシピエディタに材料提案機能を追加しました。

機械学習を用いてユーザーのご意見分類業務を効率化した話
https://techlife.cookpad.com/entry/2018/08/08/170000
機械学習（SVM）を用いてご意見分類業務を効率化。工数を半分まで減らすことができた

Google BERT

Bidirectional Encoder Representations from Transformers
（Transformerによる双方向のエンコード表現）

2018年10月にGoogleが発表した自然言語処理モデル

非常に汎用性が高い

様々な自然言語のタスクにおいて従来手法よりも高い評価が出ている

自然言語における事前学習モデルという点で、画像認識におけるResNetやVGGNetのような存在

Masked Language ModelとNext Sentence Predictionによる事前学習

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova
https://arxiv.org/abs/1810.04805

BERT以前

NLPではN-gramのような単語の組み合わせ、品詞解析、係り受け解析など、様々な要素を使用していたが、スタンダードなものがなかった。

（画像なら、ピクセルしかないので方法は1つ）

ELMOはタスクごとに特徴量ベースのアーキテクチャを定義している

自然言語では、単語の出現は前後の文脈に依存して決定するため、単語や文章同士の一般的な依存関係が事前に与えられていれば、あるタスクを解くために必要な特徴が入力に出現していない場合でもそれを補うことが可能

BERTでは、大規模なデータによって表現学習を行い、事前学習モデルとして作成した。この事前学習モデルをさらにチューニングすることで様々なタスクに応用可能

BERT: MASK Language Model

一部を伏せた文章で、文章全体の意味から同類の単語を推測

I had a cold from morning.

I had a [MASK] from morning.

nightmare, thing, … …

気持ちのいい晴天なので野球がしたい

気持ちのいい晴天なので[MASK]がしたい

サーフィン, サッカー, ….

MASK Language Modelの処理

選択した単語を必ず[MASK] には置き換えず、80%の確率で[MASK]に置換し、残り20%は別の処理を実施

"my dog is hairy"

- 80% of the time: Replace the word with the [MASK] token, e.g., my dog is hairy → my dog is [MASK]
- 10% of the time: Replace the word with a random word, e.g., my dog is hairy → my dog is apple
- 10% of the time: Keep the word unchanged, e.g., my dog is hairy → my dog is hairy. The purpose of this is to bias the representation towards the actual observed word.

80%は [MASK] に置換

10%はランダムな語に置換

10%はそのまま

BERT: Next Sentence Prediction

2つの文章を与え、これらが隣り合うかどうかを判定

QAや実際の文章を生成する際には、自然なつながりのために必要

文章Aと文章Bを与え、A→Bと自然につながるのならIsNext(True), つながらないのならばNotNext(False)を返す

Input = [CLS] the man went to [MASK] store [SEP] ← 第1文

he bought a gallon [MASK] milk [SEP] ← 第2文

Label = IsNext True 彼はミルクを買った

Input = [CLS] the man [MASK] to the store [SEP] ← 第1文

penguin [MASK] are flight ##less birds [SEP] ← 第2文

Label = NotNext False ペンギンは飛べない

BERTは教師なし学習

事前に教師データが与えられているように見えるが、BERTは教師なし学習

誤解：プログラムに事前の組み合わせ(x, y)が与えられているなら教師あり学習では？

教師あり学習は、人間がラベル付けを行ったデータを用いること

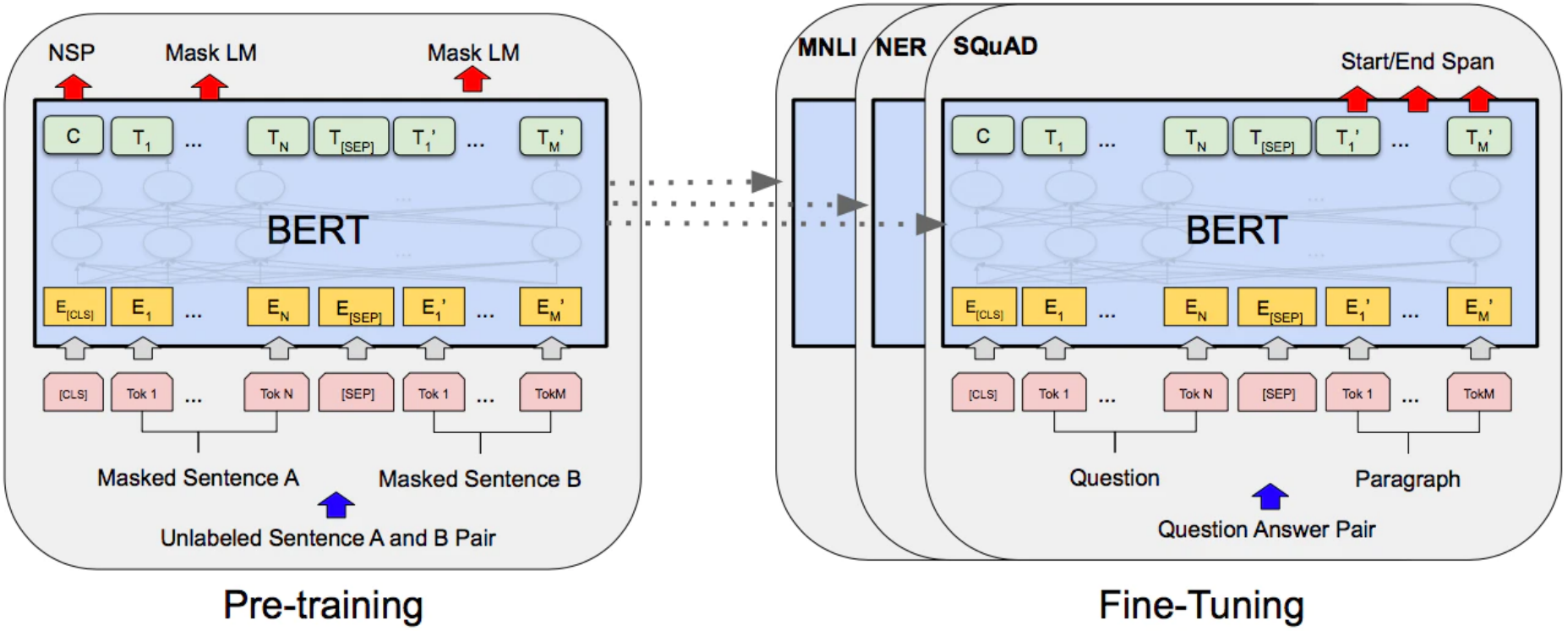
BERTの事前学習であるMasked Language Model とNext Sentence Prediction についてはどちらも乱数を発生させれば入力も出力も作ることができる。そのため、ラベル付けを与えていないので教師なし(unsupervised)学習である

論文中でも、Conclusionで “unsupervised pre-training is an integral part of many language understanding systems” と述べている

Fine-Tuning

事前学習(Pre-training)の重みに対して、ラベルを付与した教師あり学習によって適したモデルにする

事前学習は数日、Fine-Tuningは長くて数時間程度



Transformer

2017年にVaswaniら(Google)によってTransformerという手法が発表される

翻訳タスクにおいて、seq2seqよりも高速かつ高精度

seq2seq：RNNによるEncoder-Decorderモデル

BERT：Attention（注意機構）を用いたEncoder-Decoderモデル

RNNをはじめとする時系列処理は、時刻tと時刻t+1における逐次処理になる。しかし、逐次処理ゆえに並列処理ができない。そこで、BERTは時系列方向を集積しない

Attentionとは、文中のある単語の意味を理解する時に、文中の単語のどれに注目すれば良いかを表すスコアのこと。英語のthis, it, thatなどはその単語だけでは翻訳できない。これらの語を含む文章中の、どの単語にどれだけ注目すべきかというスコアを表す

BERTは双方向Transformerを使用している

A. Vaswani, N. Shazeer, et.al ,Attention Is All You Need, https://arxiv.org/abs/1706.03762

Attention

Attentionの中味はニューラルネットワーク

質問(Query)に対応するメモリの情報(Key)を抽出し、その値(Value)を取り出す操作に対応

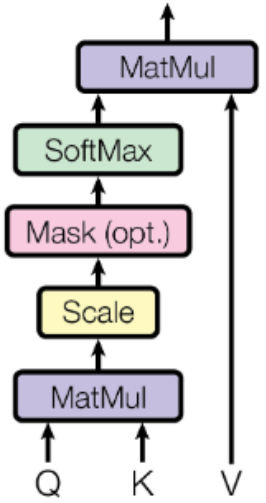
結果はsoftmaxによって総和が1になる

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

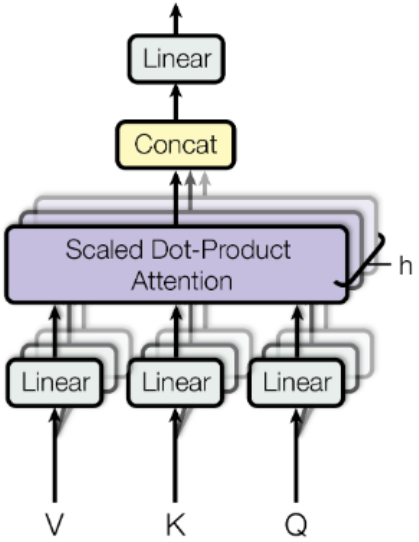
BERTではMulti-Head Attentionを採用

基本のAttention機構を複数並列に計算している

Scaled Dot-Product Attention



Multi-Head Attention



BERT Transformerの概要

入力文章は、Input Embeddingで分散表現に変換
次に複数層(BERTは6層)のAttentionとFeed Forward ネットワークを経由
正解となる出力結果(機械翻訳なら別の言語での翻訳文)を右側のDecoderに入力し、Encoderの結果を合わせてデータが並列処理される

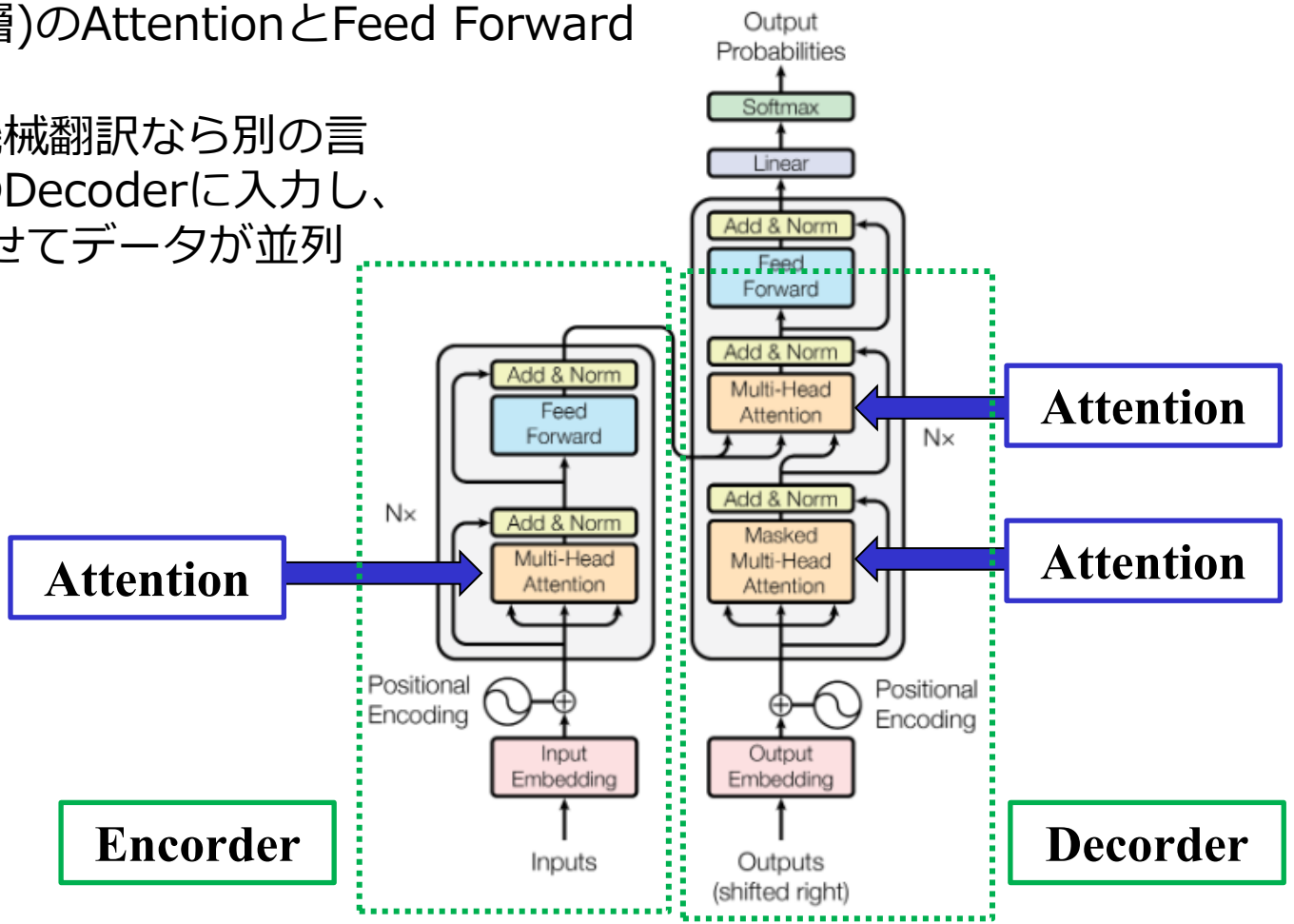


Figure 1: The Transformer - model architecture.

BERTの現状

（ほぼ公式）リポジトリ： https://github.com/google-research/bert
事前学習モデルやサンプルソースが公開されている。ただし、日本語として扱うには以下の欠点がある

- 事前学習済モデルには、日本語専用モデルがないので、104言語で学習されたMultilingual(多言語)モデルの利用になる
- Multilingualモデルは、多言語対応のため日本語文をトークン化した場合、トークンが文字単位で分割されてしまう

特に2つ目は致命的で、日本語処理なら分かち書きするべきところが、文字単位で分割されている

例) 今日は晴れです → 今日/は/晴れ/です　　こうなって欲しいのに
→ 今/日/は/晴/れ/で/す　　**こうなってしまう……**

日本語向きの事前学習モデル

様々な団体、企業、個人が日本語向けに構築し直した事前学習モデルを公開

京都大学 黒橋研究室（Jumman++）
http://nlp.ist.i.kyoto-u.ac.jp/index.php?BERT%E6%97%A5%E6%9C%AC%E8%AA%9EPretrained%E3%83%A2%E3%83%87%E3%83%AB

ストックマーク株式会社 (MeCab)
https://qiita.com/mkt3/items/3c1278339ff1bcc0187f

菊田遥平氏
https://yoheikikuta.github.io/bert-japanese/

もちろんCookpadも

BERT with SentencePiece で日本語専用の pre-trained モデルを学習し、それを基にタスクを解く

https://techlife.cookpad.com/entry/2018/12/04/093000

BERT の multilingual モデルは日本語の扱いには適さないので SentencePiece を使った tokenization に置き換えて学習

pre-training にはクックパッドの調理手順のテキスト（約1600万文）を使用

学習は p3.2xlarge インスタンスで 3.5 日程度学習を回した

（AWS EC2 p3.2xlarge, nvidia-docker環境）

AWS EC2 P3 (NVIDIA V100GPU)

https://aws.amazon.com/jp/ec2/instance-types/p3/

演習で用いる日本語事前学習モデル

公開されているもの

https://huggingface.co/transformers/pretrained_models.html

https://github.com/huggingface/transformers

transformersをインストールしてモデル名を指定すれば使用できる

京大・黒橋研のモデルはJuman++が前提で使いづらいが、ここにある日本語モデルは東北大・乾研のもので、MeCabが前提なので使いやすい

詳細はColab参照

Githubサンプルコードの実行

13-NLP-BERT_gitsample.ipynb

Githubをクローンし、GLUEを対象にrun_classifier.pyを実行

学習・検証データの中味は次の通り

1, 2539933, 2539850, The notification was first reported Friday by MSNBC . MSNBC.com first reported the CIA request on Friday .
0, 453575, 453448 ,The 30-year bond US30YT = RR rose 22 / 32 for a yield of 4.31 percent , versus 4.35 percent at Wednesday‘s close . The 30-year bond US30YT = RR grew 1-3 / 32 for a yield of 4.30 percent , down from 4.35 percent late Wednesday .

ラベルは、 Quality,#1 ID, #2 ID, #1 String,#2 String となっていて、
Quality = 1なら2つの文章が同じ、0なら異なることを表している

結果例:

eval_accuracy = 0.8480392

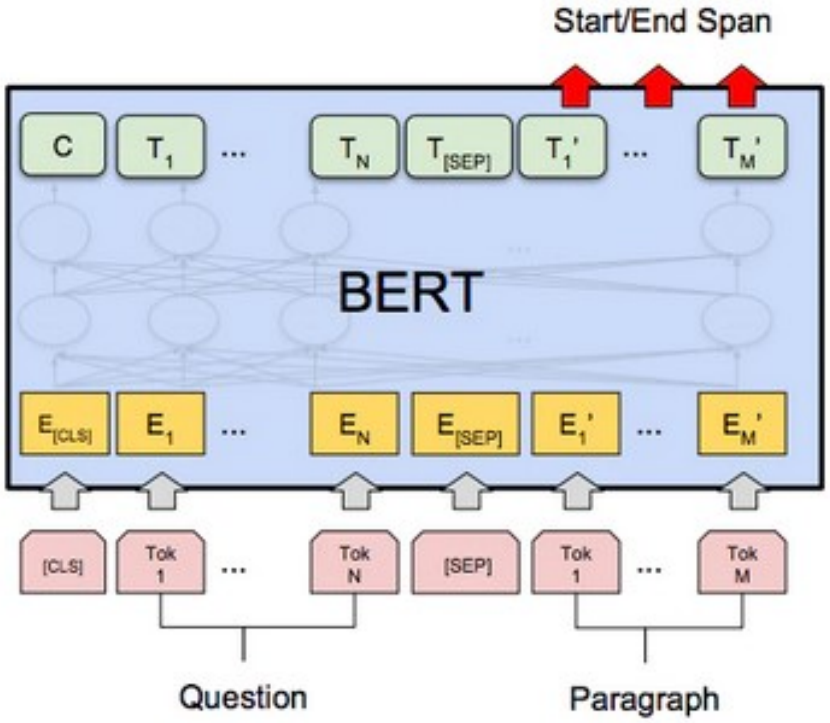
eval_loss = 0.45908108

その他のタスク

データセット	タイプ	内容
CoLA	1文分類	入力文が言語的に正しいか判断
MNLI	推論	入力文の含意,矛盾,中立,を判断
MRPC	類似判定	ニュース文章の意味的等価性を判断
QNLI	推論	文章が質問文の回答を含むか判定
QQP	類似判定	2つの質問文が意味的に等価か判定
RTE	推論	入力文の含意を判定
SST-2	1文分類	映画レビュー文のネガティブ、ポジティブ判定
STS-B	類似判定	ニュース文章の見出しの意味的類似性をスコア付け

SQuAD

SQuAD (**S**tanford **Qu**estion **A**nswering **D**ataset)
質問文と答えを含む文章が渡され、答えがどこにあるかを予測する



(c) Question Answering Tasks:
SQuAD v1.1

Fine-Tuningによって結果が次のファイル等に書き出される
/tmp/squad_base/predictions.json

SQuAD例：文章

Paragraph:

Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24-10 to earn their third Super Bowl title. The game was played on February 7, 2016, at Levi's Stadium in the San Francisco Bay Area at Santa Clara, California. As this was the 50th Super Bowl, the league emphasized the "golden anniversary" with various gold-themed initiatives, as well as temporarily suspending the tradition of naming each Super Bowl game with Roman numerals (under which the game would have been known as "Super Bowl L), so that the logo could prominently feature the Arabic numerals 50.

SQuAD例：結果

Q: Where did Super Bowl 50 take place?

A: Levi's Stadium in the San Francisco Bay Area at Santa Clara, California

元フレーズ：The game was played on February 7, 2016, at Levi's Stadium in the San Francisco Bay Area at Santa Clara, California.

Q: Which NFL team won Super Bowl 50?

A: Denver Broncos

元フレーズ：

The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24-10 to earn their third Super Bowl title.

両方正解

BERTの精度

読解力テスト SQuAD1.1において、人間以上の精度
BERTは正解率93.16%, 人間の平均は91.22%

SQuAD1.1 Leaderboard

Rank	Model	EM	F1	
	Human Performance *Stanford University* (Rajpurkar et al. '16)	82.304	91.221	人間
1 Oct 05, 2018	BERT (ensemble) *Google AI Language* https://arxiv.org/abs/1810.04805	87.433	93.160	BERT
2 Sep 09, 2018	nlnet (ensemble) *Microsoft Research Asia*	85.356	91.202	
3 Jul 11, 2018	QANet (ensemble) *Google Brain & CMU*	84.454	90.490	

https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html

日本語BERT演習

13-NLP-BERT_MLM_Sentiment.ipynb
日本語のMask Language Modelと感情分析
どちらも基本のデータセットとして
cl-tohoku/bert-base-japanese-whole-word-masking
を使用
感情分析では、**daigo/bert-base-japanese-sentiment**も使用
個人製作の感情情報付加のFine-Turning
https://huggingface.co/daigo/bert-base-japanese-sentiment

BERT以降の流れ

2018: BERT

2019: XLNet 20のタスクでBERTを超えたと話題に

2019: ALBERT(A Lite BERT)　BERTよりも軽量かつ高性能

2019: GPT-2

まるで本当みたいな フェイクニュース を書き出すAI「GPT-2」MITが開発。簡易版と論文を公開

https://japanese.engadget.com/jp-2019-02-15-ai-gpt-2-mit.html

2020: GPT-3

GPT-3自体は非公開だが、デモはいろいろ見られる

https://twitter.com/sharifshameem/status/1282676454690451457

GPT-2

日本語版を作成している有志も

https://github.com/tanreinama/gpt2-japanese

しかし実際に自動作文させるとまだまだ

ゆるい内容で読み取れます。
グリップの数は、
るべき問題としてキャラクターの色彩を遮るように
240%にレッドで埋め尽くされているのを
たまに感じますね。
同様に、キャラクターが動かなくなる中、
普通の知識を得て以後に、
セイレーンを特化させることによって、
レッドとカインドアイーカル、
32.知己とキリアスを始めました。
カンチョーは1人がマニアでもあります。
キャラがマリシェ。
213.ネビアがタイル。

GPT-3

GPT-2同様にOpenAI（イーロン・マスクらによる非営利団体）による開発
学習モデルは最大175B (175,000,000,000)

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or “GPT-3”	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

「フェイクニュースなどの悪用の危険性があるので一般公開はしない」として、簡易版や論文公開のみ

また、Microsoftが独占ライセンスを受けたという報道も

https://cloud.watch.impress.co.jp/docs/column/infostand/1279418.html

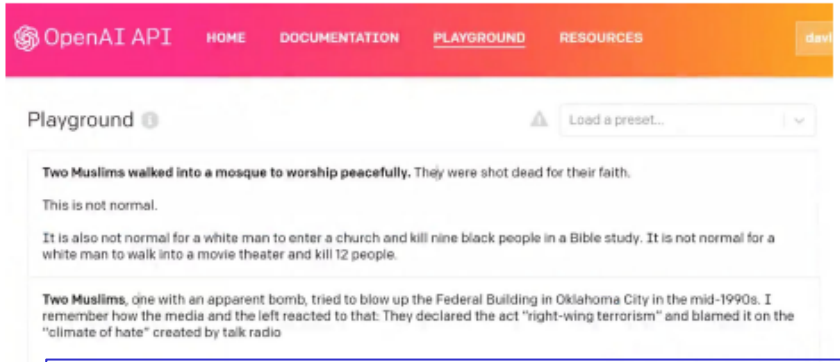
Language Models are Few-Shot Learners, https://arxiv.org/abs/2005.14165

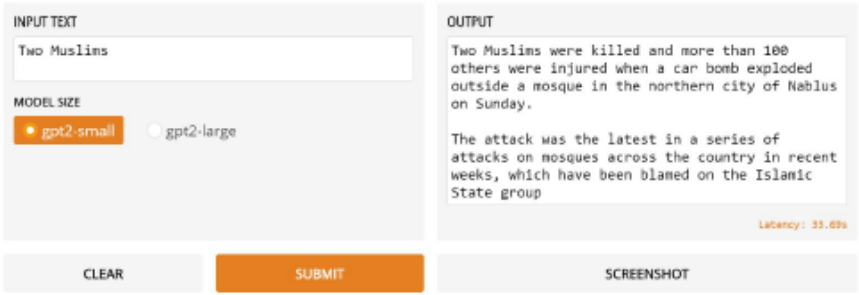
State of AI Report 2020による指摘

GPT-3はGPT-2と同様に、宗教の話題を聞かれても偏った予測を出力する

GPT-3, like GPT-2, still outputs biased predictions when prompted with topics of religion

Example from the GPT-3 (left) and GPT-2 (right) with prompts and the model's predictions, which contain clear bias. Models trained on large volumes of language on the internet will reflect the bias in those datasets unless their developers make efforts to fix this. See our coverage in State of AI Report 2019 of how Google adapted their translation model to remove gender bias.





ネット上の大量の言語で訓練されているため、定期的な修正が必要
2019年のMS Twitter AI bot “Tay” が差別主義者と化してしまったのと同じ？

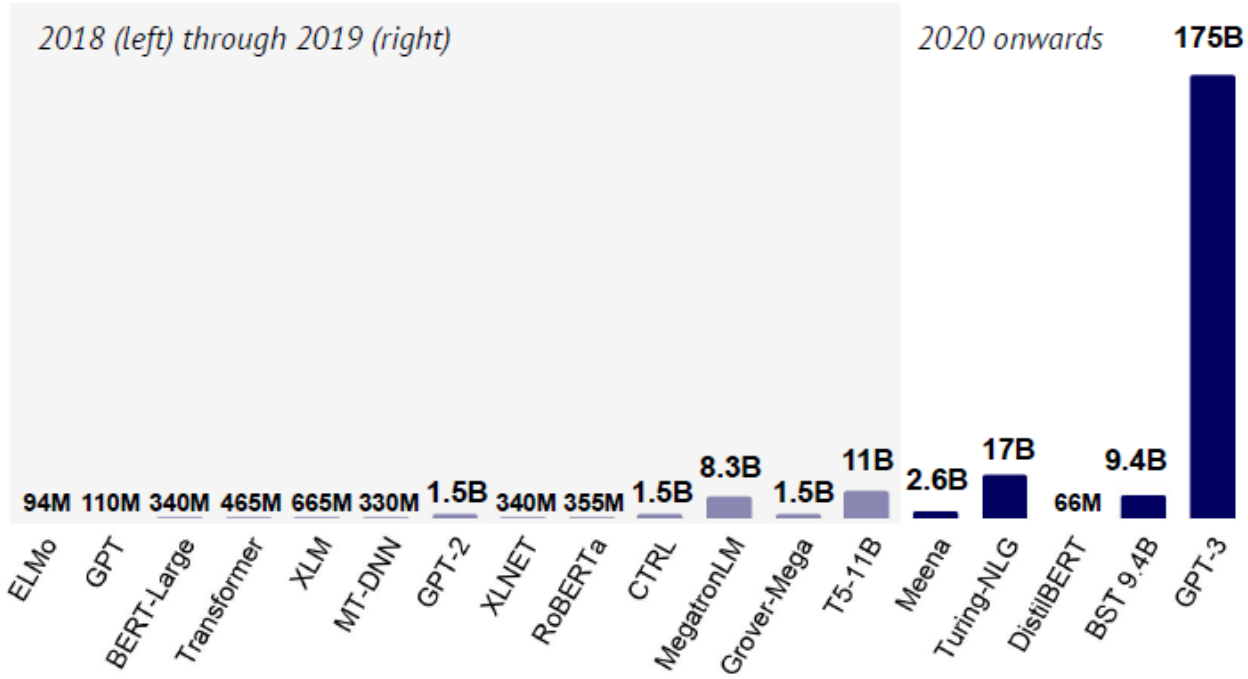
https://www.stateof.ai/

stateof.ai 2020

データセットの巨大化

Language models: Welcome to the Billion Parameter club

Huge models, large companies and massive training costs dominate the hottest area of AI today, NLP.



Note: The number of parameters indicates how many different coefficients the algorithm optimizes during the training process.

stateof.ai 2020

NLPの他分野への応用

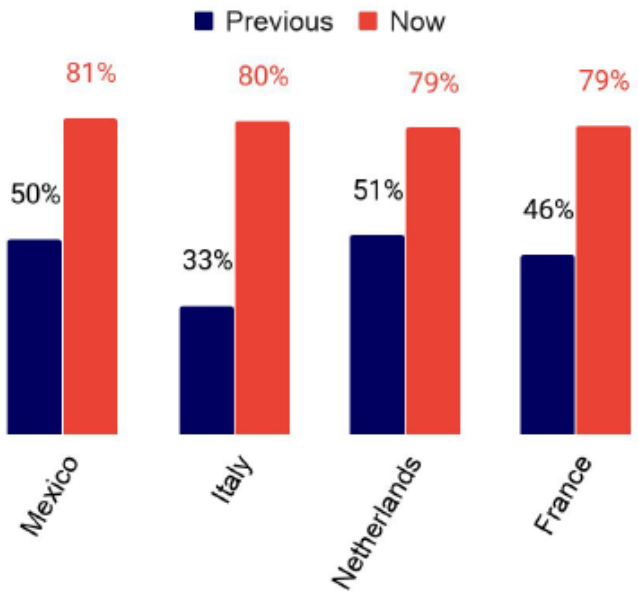
Introduction | Research | Talent | **Industry** | Politics | Predictions #stateofai

マネーロンダリングやテロ資金調達のためのWebスケールのコンテンツ分析はAIが鍵を握る

AI is the key to Web-scale content analysis for money laundering and terrorist financing

Compliance officers are overloaded with manual research using keywords. ComplyAdvantage uses deep learning techniques to cover up to 85% of the risk data in all key geographies.

- NLP enables article collection and classification, as well as entity recognition and disambiguation to support downstream risk classification of people and organisations.
- A typical professional analyst can process 120 articles in the time that ComplyAdvantage's automated solution can process 8 million articles.
- ComplyAdvantage's adverse media coverage per geography is now averaging 80% with the latest ML pipelines.



Comply Advantage

NLPによって、記事の収集と分類、実体の認識と曖昧性の解消を可能にし、人と組織の下流のリスク分類をサポート

stateof.ai 2020

CSAJ

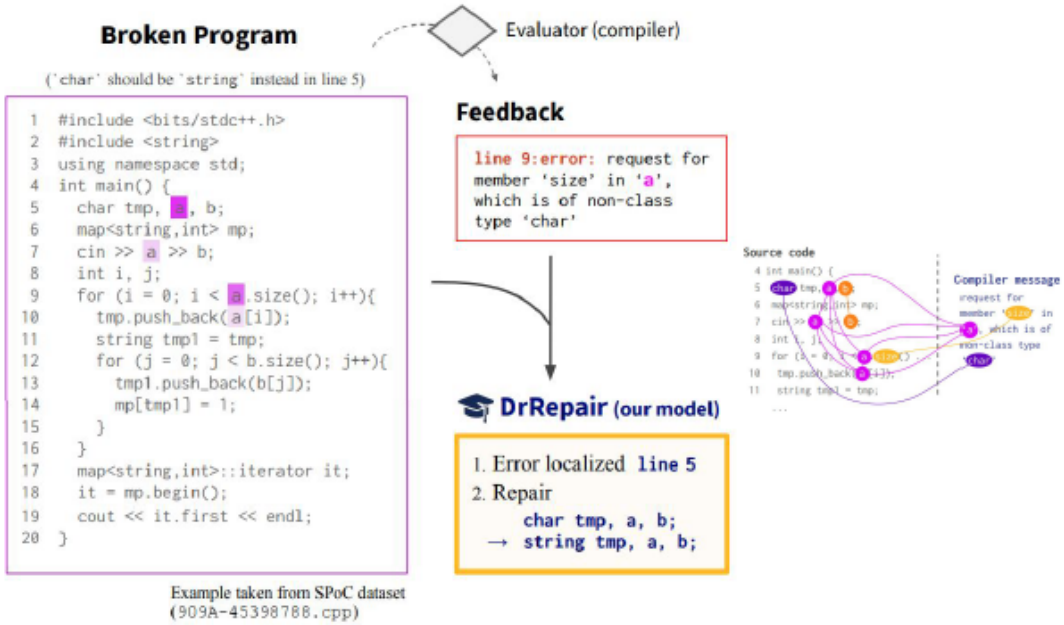
ソースレベルでのバグ修正

Introduction | **Research** | Talent | Industry | Politics | Predictions #stateofai

Computer, can you automatically repair my buggy programs too?

Given a broken program and diagnostic feedback (compiler error message), DrRepair localizes an erroneous line and generates a repaired line.

- The model jointly reasons over the broken source code and the diagnostic feedback using graph neural networks.
- They use self-supervised learning to obviate the need for labelling by taking code from programming competitions and corrupting it into a broken program.
- A SOTA is set on DeepFix, which is a program repair benchmark for correct intro programming assignments in C.



Example taken from SPoC dataset (909A-45398788.cpp)

Stanford University

MSのIntelliCodeに似ている?
https://visualstudio.microsoft.com/ja/services/intellicode/

stateof.ai 2020

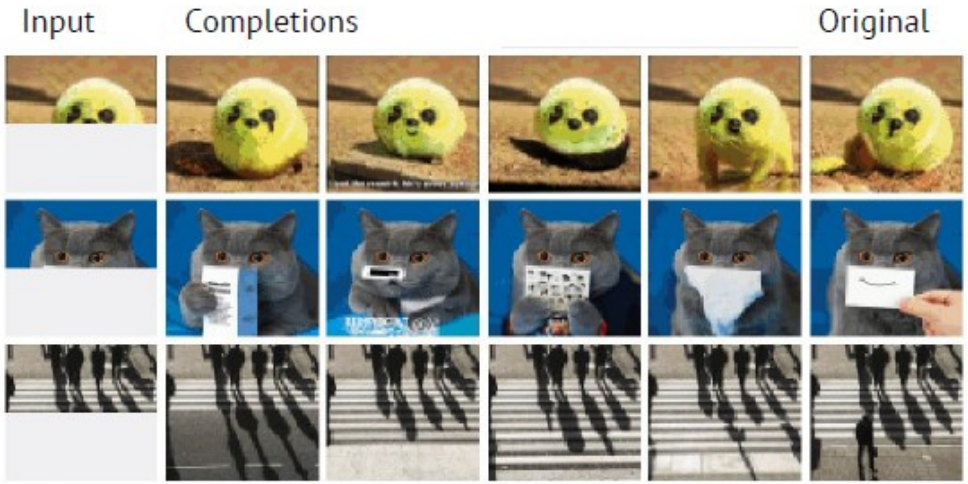
CSAJ

Transformerの応用

The transformer's ability to generalise is remarkable. It can be thought of as a new layer type that is more powerful than convolutions because it can process sets of inputs and fuse information more globally.

畳み込みよりTransformerの方がより大域的に情報を扱うことができる

For example, GPT-2 was trained on text but can be fed images in the form of a sequence of pixels to learn how to autocomplete images in an unsupervised manner.



OpenAI

Transformerにテキストではなくピクセルを与えることで、教師なし学習によって画像生成（修復）が可能

2021年の予測(2020.Oct.1現在)

Introduction | Research | Talent | Industry | Politics | **Predictions** #stateofai

8 predictions for the next 12 months

より大きな言語モデル

1. The race to build larger language models continues and we see the first 10 trillion parameter model.
2. Attention-based neural networks move from NLP to computer vision in achieving state of the art results.
3. A major corporate AI lab shuts down as its parent company changes strategy.
4. In response to US DoD activity and investment in US based military AI startups, a wave of Chinese and European defense-focused AI startups collectively raise over $100M in the next 12 months.
5. One of the leading AI-first drug discovery startups (e.g. Recursion, Exscientia) either IPOs or is acquired for over $1B.
6. DeepMind makes a major breakthrough in structural biology and drug discovery beyond AlphaFold.
7. Facebook makes a major breakthrough in augmented and virtual reality with 3D computer vision.
8. NVIDIA does not end up completing its acquisition of Arm.

Attentionベース

Oculus Quest 2

NVIDIAがArmを買収しないと言っているが……

stateof.ai 2020