2025.04.29 Visualize Attention

前回の振り返り & 導入

The Annotated Transformerは難しかった

- 前回は"The Annotated Transformer"という有名な教材に従って、Transformerを実装しながら学ぼうとした
- 内容が難しすぎた。モデルの話は前提知識が無いと理解不能
- 70分ぐらいやって全体の5%ぐらいしか進めなかった
- 図やイラストが無く、文章だけで理解するのは難しい
- 聴講者の方々の反応も重かった

という反省を活かして・・・

今回は視覚的にわかる・面白い・軽めな内容

もっと視覚的に学ぶことを目指します
テーマ：Attention を可視化して直感的に理解する

今回の勉強会の内容目標

- Transformerの基礎（Attention）が感覚でわかる
- BERT内部の動きがイメージできる
- モデルを深く考える力がつく
- 深層機械学習という深い沼に片足を突っ込む

目次

"Attention" は結局何をしてるのか・何者なのかを体感する

具体的には、Attentionという仕組みを、可視化ツール[1]を使って実際に見ながら学んでいく
今日のゴールは、「Attentionってこういう風に働いているんだな」というイメージを持つこと

そもそも、Attention とは何か

概要

ざっくりいうと、入力系列(今回は文章)の中で、特に重要な部分に注目する仕組み
例えば、英文を読んでいるときに、「この単語は大事だな」とか、「この言葉に注意しよう」といった意識を持っているイメージ
下記では、"The cat sat on the mat"という文でどのように Attention が計算されているかを見る。
下のイメージ図では、"cat"という単語に注目している

```
【単語列】

The      ★cat★      sat       on       the      mat

【矢印列】

 ↓        ⇩          ⇩         ↓        ↓        ⇩

【注目度イメージ】

(弱い)    (中心)   (やや強い)   (弱い)     (弱い)    (やや強い)
```

上記の図は、catを中心(★)として、各単語のAttentionのイメージを表している。
"cat"から "sat" と "mat" にやや強いAttentionが飛んでいるが、他はほとんど無視しているイメージ
上記の図ではcatを中心としているが、実際の計算では、左から右(The→cat→sat→...)に中心をずらして、全単語それぞれが、全単語に対してAttentionを計算する
モデルも同じで、すべての単語を均等に見るのではなく、単語同士の関係を見ながら、どこに注目すべきかを決めている
少し混乱するかもしれないですが、上記のイメージ図は正確に言うと"Self-Attention"のイメージ図です。
普通の Attention では、

```
英語文： [I] [love] [you]
               ↓    ↓   ↓
日本語文： [私] [は] [あなたを]  ← 生成中に英語文の単語を参照する
```

のように、出力系列の各ステップが、入力系列に注目するもので、翻訳タスクなどの、seq2seqモデル特有のものです。
Transformer の主力部分は上記の Attention ではなく "Self Attention" で、

```
文： [CLS] [The] [cat] [sat] [on] [the] [mat] [SEP]
        ↑     ↑     ↑    ↑    ↑     ↑    ↑
       (それぞれが、他の単語や自分自身を注目 )
```

のように、同じ系列内での Attention を計算することで、単独の文や系列内の関係性を深く理解するためのものです。

仕組み

モデルの中では、各単語が「特徴ベクトル」という形で表現され、このベクトルを使って、「今注目すべき単語はどれか？」を判断する
具体的には、クエリ（Query） と キー（Key） というベクトルの内積を計算する
つまり、内積が大きい、つまり似ている単語ほど、強く注目する(cos類似度?)
次に、注目度に応じて"バリュー（Value）ベクトル"を重み付けして、情報をまとめる
この一連の流れが、Dot-product Attentionと呼ばれる仕組み
ここまでで、「Attentionは似ている単語に注目するんだな」というイメージが持てればOK
数式で表すと、以下になる

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

Transformer とは

概要

ここまで Attention について学んできました。ここからは、それがどう実際のモデルに使われているのかを見ていきます。
代表的なモデルとして、 Transformer があります。前回紹介してぐだった奴です。
前回も言いましたが、 Transformer は、もともと機械翻訳のために設計されていて、エンコーダ（入力を理解する）とデコーダ（出力を生成する）の２つの部分から成り立っています

```
英語 → Encoder(文章→ベクトル表現) → Decoder(ベクトル表現→文章) → 日本語
```

この中で、エンコーダ部分だけを取り出したものを Transformer Encoder と呼びます。
エンコーダ・デコーダ構造のものでも、エンコーダのみでも、デコーダのみでも Transformer と呼ぶみたいです。
Transformer Encoderは、入力された文章を読み取り、単語同士の関係性をSelf-Attentionを使って捉え、文章全体の意味をうまく内部に表現する役割を持っています。

```
文章 → Encoder(文章→ベクトル表現) → 文章全体の構造をコンピュータが把握
```

BERT について

BERT というものを知っていますか？
今日の可視化では、BERT というモデルを扱います。
これは、Transformer Encoderだけを積み重ねたもので、たくさんの文を左右両方向から読み取りながら、単語同士の関係を細かく捉えることができる言語モデルです。

つまり、BERTの内部にはたくさんのAttentionが走っていて、どの単語にどれくらい注目しているかが刻々と変わっています。
これを可視化することで、"モデルがどう意味を理解しているか"が見えるようになります

可視化ツール"BERTviz"について

BERTvizは、BERTモデル内部のAttentionを可視化するためのツールです。
Webブラウザ上で動作し、

- どの単語がどの単語に注目しているか
- その注目の強さはどれくらいか
  を、視覚的に確認することができます。
  BERTvizには大きく3つのモードがあります。
- Head view
  これは、モデルの中の各Attentionヘッドごとに、 「どの単語がどの単語を見ているか」を、線でまとめて見るモードです。
  今日のメインはこれを使います！
- Neuron view
  これは、特定の単語同士のAttentionだけに注目して、 層やヘッドを細かく絞って見られるモードです。
  「この関係を詳しく見たい！」というときに便利です。
- Model view
  これは、入力と出力のAttentionのつながりを見るモードですが、 BERTはエンコーダだけなので、今回はあまり使いません。
  今日の勉強会では、 まずHead viewを使ってざっくり可視化を体験して、 もし興味があればNeuron viewも少し触ってみようと思います！

可視化してみよう

実際にBERTvizを動かして、 Attention の可視化をしていきます。
https://colab.research.google.com/drive/1hXIQ77A4TYS4y3UthWF-Ci7V7vVUoxmQ?usp=sharing
のリンクから可視化ツールにアクセスしてください。
まず、Head viewからやってみようと思います
初期の状態だと、

```
sentence_a = "The cat sat on the mat"
sentence_b = "The cat lay on the rug"
```

という二つの文が設定されていると思います
BERTは２つの文を入力するのが標準みたいなのですが、いきなり２文のAttentionを行うのは難しいので、

```
sentence_a = "The cat sat on the mat"
sentence_b = ""
```

の状態にして、１文だけのAttentionを計算・可視化するようにしてください。
以下では、可視化GUIの各機能の説明をします

Attentionの表示オプションについて

GUIの上部分では、LayerとAttentionの設定と表示するヘッドの選択を行うことができます。
Layerについて説明をする前に、Attentionの設定について、最初に説明を行います。
上記の通り、BERTの入力は２つの文が標準になっています
例えば、

```
A文："The cat sat."
B文："The dog barked."
```

みたいに、2つの文を並べて入力して、文と文の関係（Next Sentence Prediction）を学習している。
ただし、

```
A→A:    A文の単語がA文の単語を見る Self Attention だけを表示
A→B:    A文の単語がB文の単語を見る Self Attention だけを表示
B→A:    B文の単語がA文の単語を見る SElf Attention だけを表示
B→B:    B文の単語がB文の単語を見る Self Attention だけを表示
All:    上記全部を表示
```

のように、表示範囲を絞り Self Attention の結果を見ることもできる。

Layerについて

BertVizでは、Layerを0~11の範囲で指定でき、Layerの数値によって結果が変わります。
ここでは、Layerについて軽く説明します。

BERTは、入力された文を、 Self-AttentionとFeedForwardを組み合わせた層（Layer）を何段も通して処理していきます。

```
Input 文
  ↓
Layer 1
  ↓
Layer 2
  ↓
Layer 3
  ↓
...
  ↓
Layer 12（最終層） → 出力（CLSベクトルなど）
```

BERT-baseなら、層の数は12です。
通常、タスクに使うのは、最後（12層目）の出力です。

なぜなら、深い層に行くほど、意味のまとまりや文全体の特徴をしっかり捉えているからです。
ただし、タスクによっては少し工夫することもあります。
たとえば、9層〜12層までを平均して使うとか、 中間層（たとえば8層目）をわざと使うこともあります。

Layerのイメージ

- 浅い層（Layer 1〜3）
  単語同士の近い関係や局所的な特徴を見る
  モデルはまだあまり文脈を理解していないため、 文全体の代表である[CLS]に多くのAttentionを向けがち。
  手探りで「どこが大事なのか」を探している段階です。
- 中間層（Layer 4〜8）
  文の構造を意識し始め、文脈を意識したAttentionが増える
  主語と動詞の関係、目的語とのつながりなど、文法的な意味を理解しようとする
- 深い層（Layer 9〜12）
  文全体の意味を抽象的に理解する
  ただし、短くシンプルな文だと、意味的ターゲットがなくなり[SEP]にAttentionが流れることがある
  → 意味理解が完了して、Attentionが漂い始めることも
  ただし、今回のように短いシンプルな文だと、 意味的にこれ以上掘り下げるものがないため、
  AttentionがSEPなどに流れてしまう現象が起こることがあります。
  BERTvizでAttentionを見るときは、"層ごとの役割の違い"も意識しながら観察すると、 モデル内部の動きがもっと深く見えてきます！

CLS と SEP

Layerのイメージの所で、 CLS と SEP が出てきました。

- [CLS]とは？
  Classification（分類） の略で、文全体の代表情報をまとめるために、文の先頭に挿入される特別なトークン
  BERTでは、例えば、①文分類、②質問応答、③文と文の関係判定（Next Sentence Prediction） みたいなタスクのときに、[CLS]の出力ベクトルを使って最終予測をする
  入力の一番最初に必ず置かれる
  ざっくり言うと、「この文のまとめ役」
- [SEP]とは？
  Separator（区切り） の略で、文章の終わりや、2つの文章の間に挿入するトークン
  BERTは1文だけじゃなく、2文を同時に入力できるから、どこで1文目が終わって2文目が始まるかを示す必要がある
  文の終わりに必ず置かれる（1文だけのときも置く）

ざっくり言うと、 「ここで文章が区切れてますよ」というしるし

- 入力のイメージ
  一文を入れる場合

```
[CLS] The cat sat on the mat [SEP]
```

二文を入れる場合

```
[CLS] The cat sat. [SEP] The dog barked. [SEP]
```

CLSに Attention が飛んでいる場合は、文全体の意味をまとめている時
SEPに Attention が飛んでいる場合は、文の終わりを意識している時

ヘッドの種類について

現在、 Head Model を可視化しています。
BERTでは、Attentionヘッドというものが複数存在し、それぞれのヘッドごとに役割が違います。

```
色: 役割・機能の例
緑: 文全体をまとめる（CLSに情報を集める）
ピンク:　文の区切り（SEP）を意識する
紫: 主語と動詞の対応を見る（文法構造意識）
オレンジ: 名詞句（the catなど）のまとまりを検出する
青: 単語間の係り受け（依存関係）を追う
赤: 動詞と目的語の関係（eat → appleなど）を捉える
黄緑:　特定単語の自己保持（自分自身に注目する）
水色:　代名詞（it, he, sheなど）の参照先を追う
灰色:　周辺単語に均等にAttentionを向ける（位置意識）
```

BERTの各Attentionヘッドは、学習を通して、だんだん得意な機能・役割を持つようになります。
例えば、あるヘッドは文全体をまとめる[CLS]に注目しやすくなったり、 またあるヘッドは、主語と動詞のつながりを意識するようになったり・・・
このスライドは、それらを色ごとにわかりやすくまとめた例です。

ただし注意してほしいのは、 この色と役割は固定ではありません。

別のモデルや別の文では、色と機能の対応が変わることもあります。

ここで大事なのは、

- 各ヘッドが違う視点で文を見ている！
- だからこそTransformerは高い表現力を持てる！

ということです！

ミニワーク

ここからはミニワークです！

https://colab.research.google.com/drive/1hXIQ77A4TYS4y3UthWF-Ci7V7vVUoxmQ?usp=sharing
を開いてください。

次に、自分で短い英文を1つ考えて入力してみましょう。

あまり長い文章だとAttentionがぐちゃぐちゃになりやすいので、 今回は5〜10語くらいのシンプルな文をオススメします！

文章を入力できたら、BERTvizを使ってAttentionを可視化します。

最初は「Allヘッド表示」で全体をざっくり見てみましょう。 そのあと、興味が湧いたら特定のヘッドだけ選んで、

- 特定の単語に注目する
- 層を切り替えてみる

など、自由に触ってみてください！

観察するときのポイントは、

- 単語同士のつながり
- CLSやSEPへの注目
- 浅い層と深い層での違い
- ヘッドごとの違い

等に注目してみてください。

面白い気づきがあったら、あとでみんなで共有しましょう！

例文

「何を書いていいかわからない！」という人向けに、 いくつかおすすめの英文を用意しました！

最初は、

```
"The cat sat on the mat."
"She loves programming."
```

のようなシンプルな文から始めてOKです。

もっと試してみたくなったら、 代名詞を含んだ文（"it"がどこを指してるか？）とか、 少しだけ複雑な文にチャレンジしてみてください！

もし自分でオリジナルな文を考えたい人は、 好きな英文でも大丈夫です！

どの単語がどこを見ているか、 どの層・どのヘッドで何が起きているか、 自由に観察して楽しんでみてください！

まとめ

今日の勉強会では、

- Attentionとは何か
- BERTとはどういうモデルか
- BERTvizを使って実際にAttentionを見てみる体験

をしてきました！

特に大事なことは、

- Attentionは「誰が誰を見ているか」を表す仕組みであること
- 層が深くなると単語間のつながりが意味的に深まること
- Attentionヘッドは、それぞれ違った視点で文を見ていること

です！

これから、自然言語処理のもっと高度なモデル（例えばT5やGPT系）を学んでいくときも、 Attention の動きを意識できると、モデルの理解がぐっと深まります！

今日学んだことを、ぜひ今後の勉強や研究にも活かしてください！

This material benefited from the assistance of ChatGPT.

Kazuma Aoyama(kazuma-a@lsnl.jp)

参考文献

[1] http://github.com/jessevig/bertviz