# TransformerのSelf Attention の可視化を利用した英文読解支援アプリ

2020-01-24

# 概要

- 事前学習済みのTransformerモデルに英文を入力したときの Transformerのエンコーダー部分のSelf Attentionを可視化することで 英文の読解に役立てるアプリを作りました
- ・ついでに英文の形態素解析も行い、文中の単語を品詞毎に色分けして読みやすくする機能を追加しました
- (自分でTransformerモデルを学習させて可視化させてみたかったのですが時間の都合上できませんでした)

# 動機

- 英文を読むのが苦手なのをどうにかしたい
- ・人間並みの文章理解力を持つらしいBERTを使って英文の理解を助けるアプリが作れないか
- BERTのSelf Attentionを可視化して、文章に重ね合わせて表示すれば文章が理解しやすくなるのではないか
- ・ついでに文章を形態素解析して品詞毎に単語を色分けすると分かり やすいのではないか

# システム構成

- ・ブラウザ上で文章を入力すると、サーバー上でSelf Attentionの計算と形態素解析を行い、結果をブラウザ上で表示する、というWebアプリです
- Self Attention抽出用のモデルは huggingface/transformers の bertbase-uncasedを使用しました
- ・形態素解析にはspaCyのen\_core\_web\_smを使用しました

# 動作例: 品詞の色付け



文章を入力すると、サーバー上で形態素解析が行われ、品詞毎に色分けされた文章が表示されます。

(文章は、2020-01-22の BBC News の記事 https://www.bbc.com/ne ws/world-asia-china-51202000 から取得しまし た)

オレンジ: 形容詞

ピンク: 名詞

赤: 固有名詞

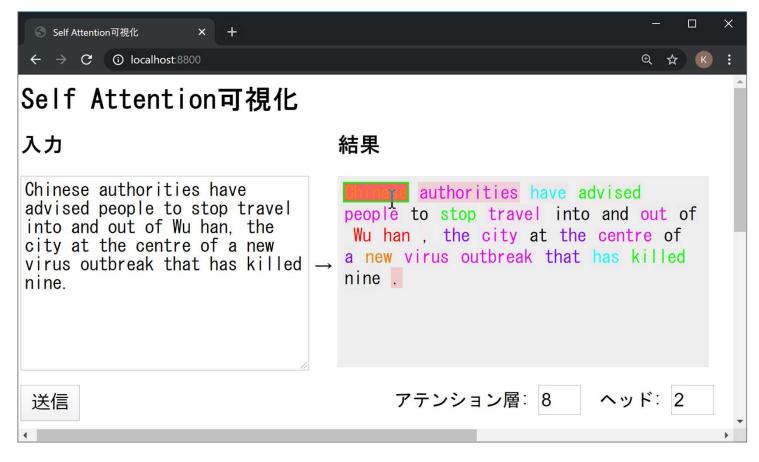
紫: 冠詞•関係代名詞

水色: 助動詞

緑:動詞

黒: その他

# 動作例: Self Attentionの可視化



単語にマウスカーソルをかざすと、その単語についてのSelf Attentionの値がヒートマップ表示されます。

アテンションの値が1に 近い単語ほど赤く表示 されます。

画像は8層目のヘッド2 の「Chinese」のアテン ションです。

# 備考:

Chinese自身の値が大きい authoritiesの値も若干大きい

## 備考:

haveの値が大きい advised、toの値も若干大きい

# 備考:

advisedの値が大きい people、to、stopの値も若干大きい

# 備考:

peopleの値が大きい to、stopの値も若干大きい

## 備考:

toの値が大きい

Chinese、have、advised、stopの値も若干大きい

備考:

stopの値が大きい

備考:

travelの値が大きい

#### 備考:

intoの値が大きい

Chinese、stop、and、out、ofの値も若干大きい

# 備考:

andの値が大きい ofの値も若干大きい

備考:

into、outの値が大きい

備考: ofの値が大きい

## 備考:

Chinese、and、Wu、han、theの値が若干大きい

#### 備考:

hanの値が大きい Chineseの値も若干大きい

# 備考:

,の値が大きい Chinese、theの値も若干大きい

備考:

cityの値が大きい

#### 備考:

at、the、centre、ofの値が大きい Chineseの値も若干大きい

## 備考:

the、centreの値が大きい

備考:

centreの値が大きい

備考: ofの値が大きい

備考: aの値が大きい

# 備考:

newの値が大きい ofの値も若干大きい

#### 備考:

Chinese、of、a、new、virus、outbreakの値が若干大きい

#### 備考:

Chinese、of、a、outbreakの値が若干大きい

## 備考:

Chinese、of、a、that、hasの値が若干大きい

## 備考:

has、killedの値が大きい

#### 備考:

Killed、nineの値が大きい

備考:

nineの値が大きい

備考:

Chineseの値が大きい

# 所感

- 品詞の色付けをすると文章が読みやすくなった気がしました
- Self Attentionの可視化は、読み解くのが大変で微妙でした
  - 単に読み解く人間がこじつけているだけのような気もします...
  - また、アテンション層数×マルチヘッド数の分だけアテンションが存在するので、そのなかから人間に理解できそうなアテンションを探すのも大変そうです
    - BERT(bert-base-uncased)はアテンション数は12、マルチヘッド数も12でした
  - なんとなく、浅い層や深い層より、中間層あたりのほうが直感的にしっくりくる値が得られるような気がしました(今回は8層目を使用)
    - ・ 浅い層(=入力に近い層)は、重みが文全体にまんべんなく分散しやすい?
    - ・深い層(=出力に近い層)は、重みが文の特定の単語に集中しやすい?
    - よく見ていないので気のせいかもしれません

# 今後の課題

- マルチヘッドの数を減らして学習させるとAttentionがどう変わるか確認したい
- ・モデルを変えるとAttentionがどう変わるか確認したい
- ・他の言語でも試してみたい
- Tensorflow.jsを用いるなどしてブラウザー上だけで完結するアプリにしたい
- ソースは以下にアップロード予定です
  - <a href="https://github.com/k-mitani/AISeminar2019AttentionVisualizer">https://github.com/k-mitani/AISeminar2019AttentionVisualizer</a>