E-commerceレビュー 分析

経済学部 高橋慶伍

内容の確認

[内容**]** **女性服**のオンラインレビューを以下の2つに分類するモデルを作成。

- **商品を推奨しているか　(recommended)**
- **商品を推奨していないか (not recommended)**

データの観察

[総サンプル数]　23,486
→レビューが空欄のとこが多々ある→前処理で削除
[column名]　Review Text　→レビュー文
Recommended IND →ラベル(0が非推奨、1が推奨)
[ラベルの内訳]　0の個数→**4101**
1の個数→**18540**　**比率=約1:4(recomendが8割を占める)**
→お勧め度が高い女性服のレビューである

[問題設定]モデルに用いる性能指標

＊分類問題において主要な性能指標は以下の4つ

accuracy(正答率)/recall(再現率)/precision(適合率)/F1(再現率と適合率の平均)

accuracyが一般的。だがこの場合ラベルの数に偏りがあり全て1に分けても

8割を超える

→正答率だけではモデルを適切に評価できない

[問題設定]モデルを用いたビジネス戦略

[ビジネス戦略]　お勧め度をさらに高める女性服を開発し売上を向上させる

仮定 お勧め度を高めればリピート客・新規客の確保ができ売上が向上する

→推奨派は勿論、約20%の非推奨派の原因を探る必要性

[重要]　少数派である **not recommended** の検出ができているモデル

→**Recall(再現率)(macro平均)**

[注意]　実用性の面から**Precision**の値の大きさにも注目

×**Recall**は大きいが**Precision**との差が大きい

[機械学習]自然言語処理

[機械学習を用いるメリット]

説明変数と目的変数の関係や結果の解釈がしやすい

[実行で注意した点]

1.パラメーターのチューニング

複数の評価指標を導出するため層化交差検証法とグリッドサーチを併用する

2.不均衡クラスを均等に評価する

0と1それぞれのクラスラベルに与える重みを調整する

[機械学習]単語の特徴量抽出

[TF-IDF]

それぞれの文書サンプルを際立たせる単語に重みつけをする手法

[仮定]　「ある文章に頻繁に出現し全体の文章にはあまり出現しない」単語

→その文章を特徴付ける重要な単語となり値も大きくなるようになる

[用いるメリット]

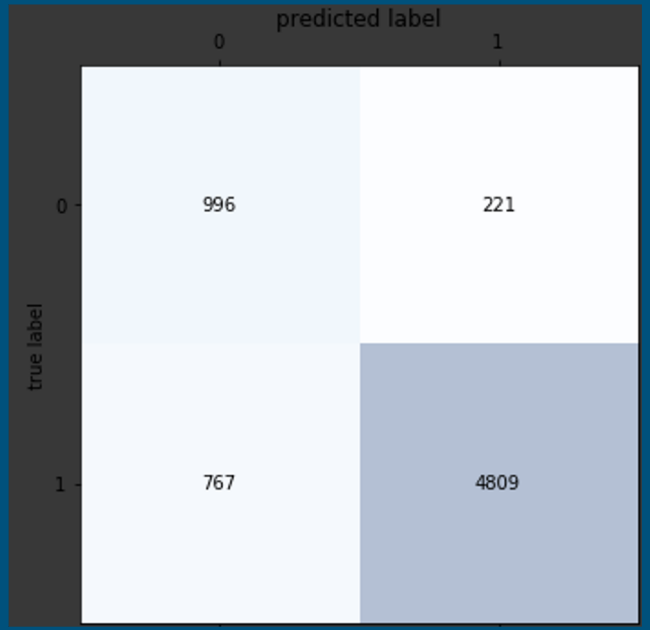
文章における単語の重要度を考慮できる

[機械学習]結果

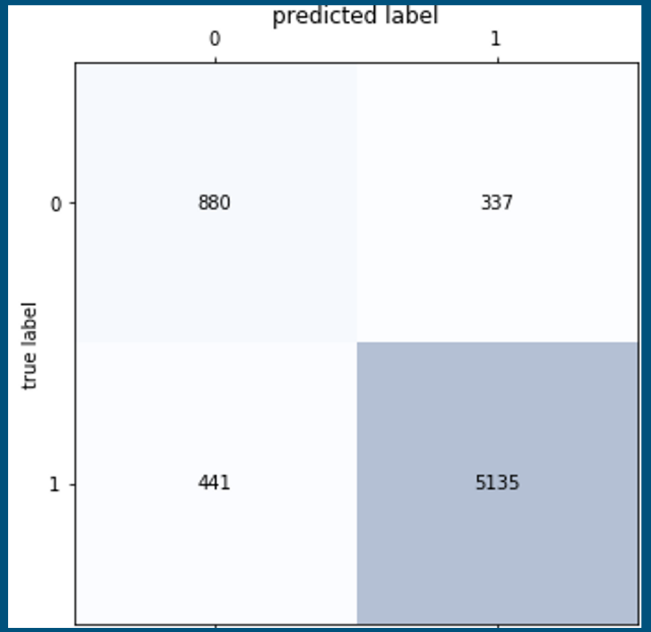
分類器	再現率 (recall)	適合率 (precision)	(参考：正答率)
SVM(kernel=rbf) C=1.0,gamma=1.0	0.810	0.806	0.885
ロジスティック回帰 C=1.0	0.840	0.761	0.855
SVM(kernel=linear) C=1.0	0.849	0.751	0.850
ランダムフォレスト n=10	0.678	0.758	0.852
決定木(depth=100)	0.671	0.648	0.782

[機械学習]混同行列

ロジスティック回帰

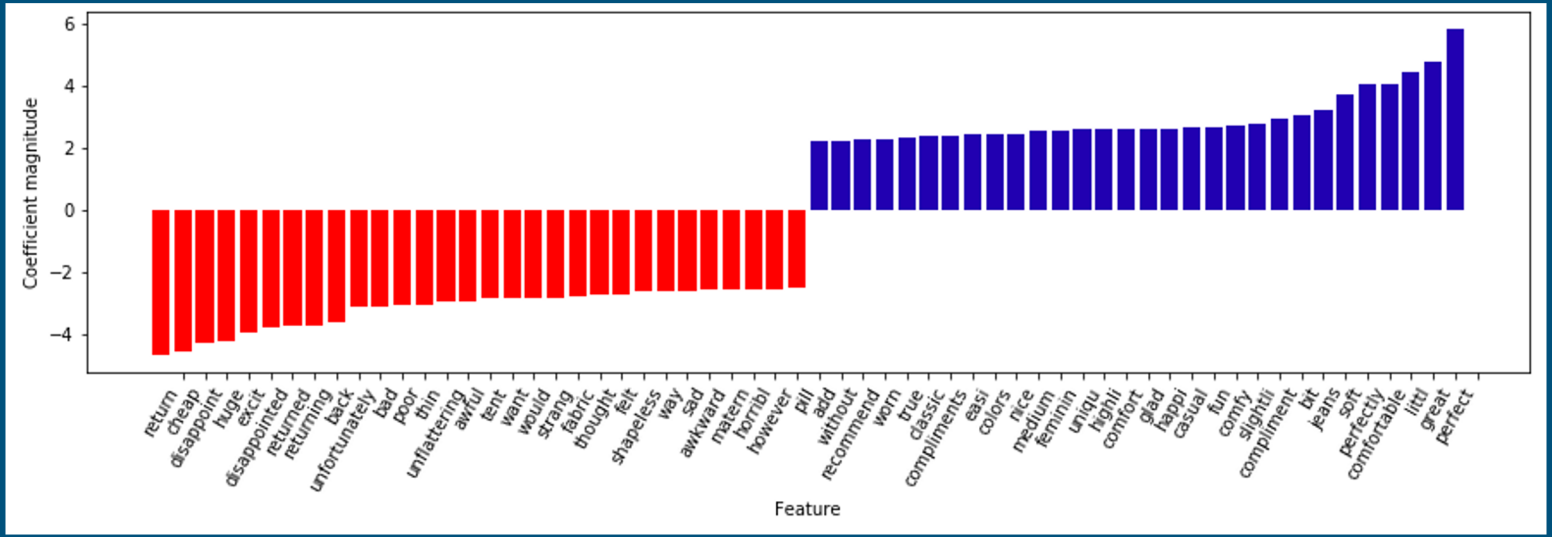


SVM(kernel=rbf)



実用性を踏まえるとSVM

[機械学習]説明変数



表は説明変数の係数が大きい・小さい単語best30までそれぞれ出した(ロジスティック回帰)

→説明変数の係数の大小は分類への重要度

肯定意見:comfortable,soft,worm, 否定意見:cheap,thin,awkward

[深層学習]導入

[Deeplearningを自然言語処理で用いるメリット]

・特徴量の自動設計が可能

→大量のパラメーターによって特徴量を自動的かつ客観的に選択・設計ができる

→特徴量の設計・抽出に手間がかかる自然言語処理に有効

[深層学習]分散表現(Word2vec)

単語を高次元の実数ベクトルに変換した表現

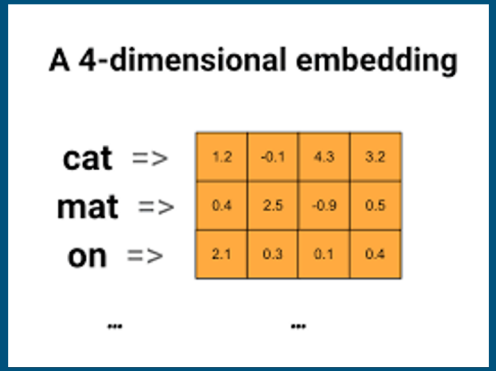
[分散表現を入力に用いるメリット]

＊単語の意味を捉えた表現→単語の意味を考慮して

　文章分類問題を解くことが可能

[分散表現・モデル選択]

今回は事前学習済みのWord2vecをファインチューニングする。



出典:http://tensorflow.classcat.com/category/embedding/

[深層学習]分散表現(ELMo)

[分散表現・ELMo]

Word2vecの欠点・・・単語の意味を文脈を考慮して判断できない

例：bank → 「銀行」「土手」word2vecでは意味の区別ができない

ELMoの利点

学習に双方向LSTMを用いており結果文脈を考慮した分散表現を作成できる

＊今回はtensorflow hub で公開されているELMoをファインチューニング

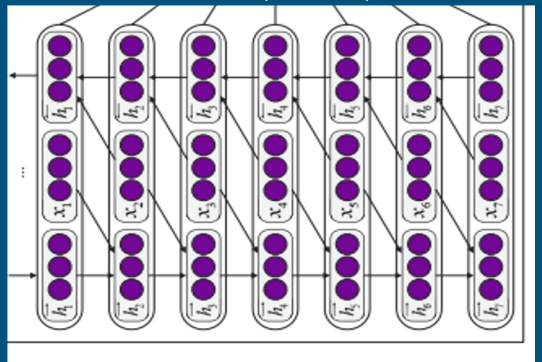
[深層学習]LSTM

[LSTMを用いる利点]

RNNより前の単語の情報を最後の出力に反映可能。

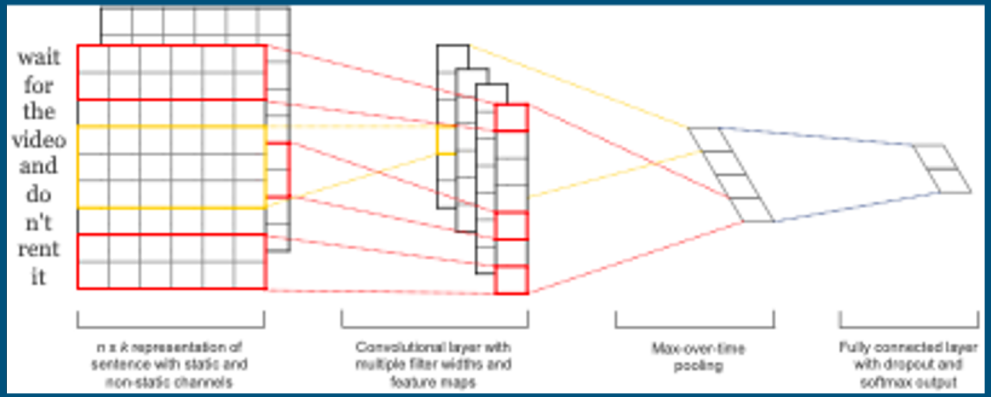
双方向LSTM(bidirectional LSTM)

過去の情報(単語)だけでなく、未来の情報(単語)も出力に反映可能。



[深層学習]CNN

[論文]Convolutional Neural Networks for Sentence Classification[Yoon Kim]



[概要]　分散表現を複数のカーネルで畳み込み

その後プーリング層を適用する

[メリット] 文のあらゆる情報を均一に入力する

処理速度がはやい

[モデル] 重みを更新する埋め込み層と重みを更新しない埋め込み層の出力それぞれに畳み込み・プーリング層を適用する

[深層学習]Bidirecrional LSTM と CNN の併用

[論文] Text Classification Improved by Integrating Bidirectional LSTM with Two-dimensional Max Pooling[Peng Zhonら]

[概要] 分散表現を学習したLSTMに1次元の畳み込み・プーリング層を用いる

[利点]

LSTM→前の情報を反映できるとはいえ最後の方の入力に重点がいく

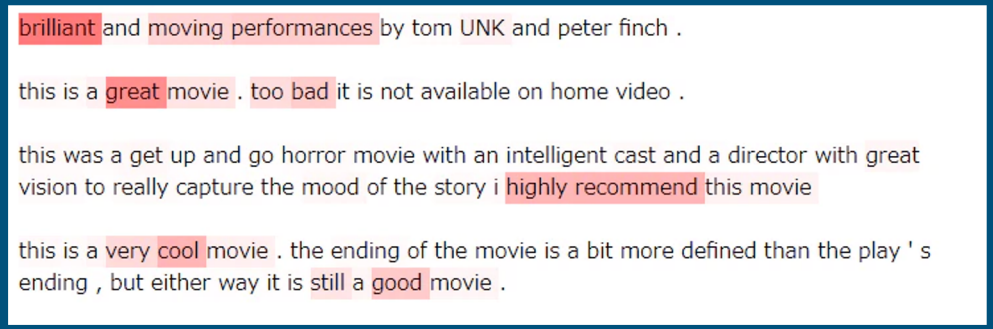
CNN →畳み込みで均等に文章の情報を抽出するが長い文脈関係の把握が不可

ある程度LSTMとCNNの欠点を補いあえる

[深層学習]Attention機構

・特定の情報量(今回は否定的or肯定的に関わる情報量)に注目するメカニズム

例



→出力層に否定・肯定に関わる注意部分を提示することで正答率の上昇の期待

→再現率・適合率でも上昇するか検証する

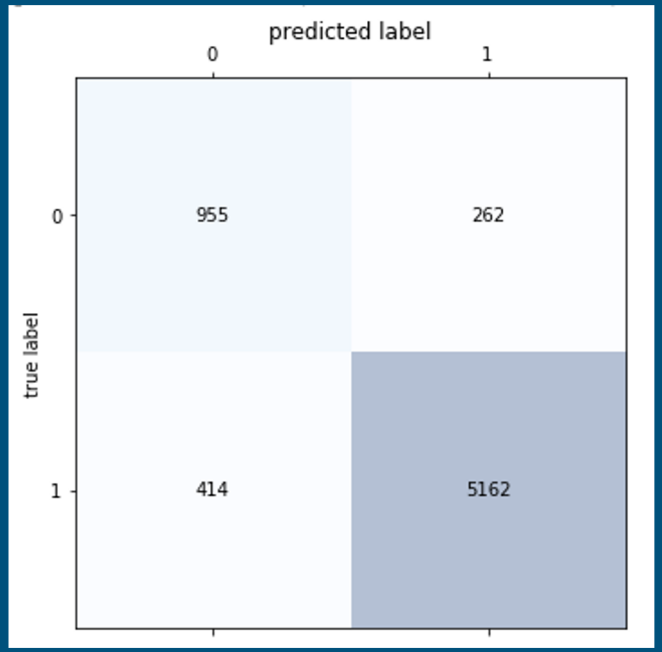
出典：https://qiita.com/itok_msi/items/ad95425b6773985ef959

[深層学習]結果

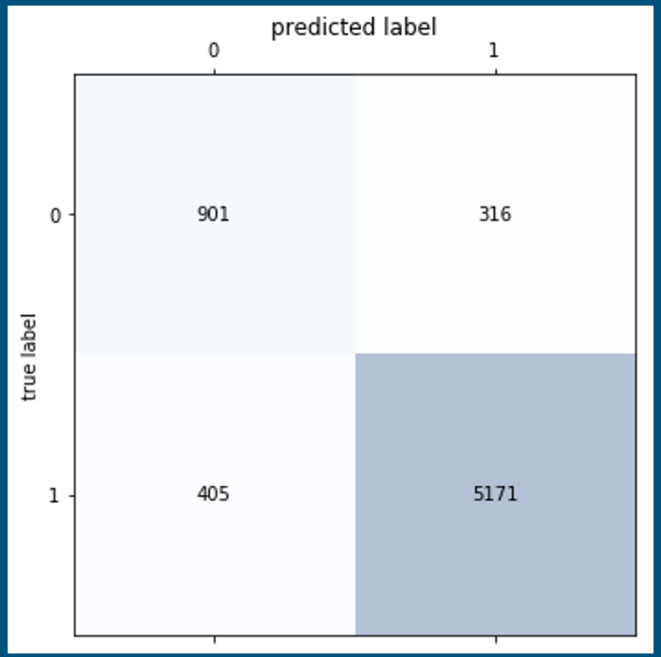
モデル	再現率(recall)	適合率(precision)	参考：正答率
LSTM	0.825	0.786	0.876
Bidirectional LSTM	0.850	0.790	0.878
Bidirectional LSTM +Attention	0.846	0.794	0.882
CNN	0.855	0.825	0.901
CNN+Attention	0.799	0.805	0.887
Bidirectional LSTM +CNN	0.834	0.816	0.894
ELMo+Dense	0.870	0.819	0.895

[深層学習]混同行列

CNN

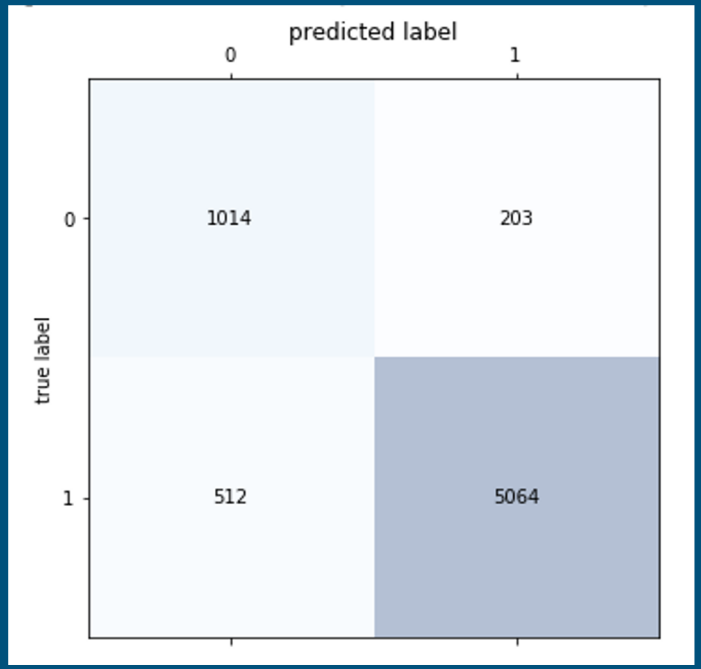


BidirectionalLSTM + CNN



[深層学習]混同行列・まとめ

ELMo



①CNN・BiLSTM + CNN ・ ELMo

→recallとprecisionが高い上に差がない

②混同行列をみてもnot recommended の検出数が多く誤分類の比率も同程度

自然言語処理・総括

SVMと深層学習3モデルの比較 (0に関するrecall,precision)

	Recall	Precision
SVM	0.72	0.66
CNN	0.78	0.69
Bidirectional LSTM CNN	0.74	0.68
ELMo+Dense	0.83	0.66

深層学習のモデル・共通

SVMのPrecisionの数値以上で
高いrecallの値を出している。

ふさわしいモデル

3つの深層学習のモデルのうち
今回の目的は0 not recommended
の検出ができるモデル
→ELMo+Dense

提案するモデル→ELMo+Dense

[課題]

機械学習・・・前処理でのチューニングを試し切れてない(bow変換前)
深層学習

- ハイパーパラメータのチューニング

  隠れ層の数,正則化,filterの数..etc またdropout層をはざまず実行してしまった

- 不均衡データの分類への対策

  今回はクラス毎に重みを指定したが、毎回指定していてはとても面倒

  損失関数 affinity loss を実装したかったができなかった

- ELMoでLSTMと結合しての実装

  テキスト処理(単語で分割しIDをつける)が課題

[今後の展望]

①損失関数・Affinity loss の理解と実装

[論文]Max-margin Class Imbalanced Learning with Gaussian Affinity

[Munawar Hayatら]

②BERT事前学習済みモデルをファインチューニングしての実装

③CNN自然言語モデル 文だけでなく文字においても単語ベクトルを作成する

[論文]Deep Convolutional Neural for Sentiment Analysis of Short Text[Cicero dos Santos]

参考論文・外部データ

Yoon Kim,(2014).*Convolutional Neural Networks for Sentence Classification,*

Zhang, Y., & Wallace, B. (2015). *A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification,*

Peng Zhou,Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, Bo Xu.*(2016).Text Classification Improved by Integrating Bidirectional LSTM with Two-dimensional Max Pooling*

M.Peters et al.(2018).*Deep contextualized word representations*

Depeng Liang,Yongdong Zhang.(2016).*AC-BLSTM:Asymmetric Convolutional Bidirectional LSTM NetWorks for Text Classification*

Ashish Vaswani et al. (2017). *Attention is All you Need*

参考論文・外部データ

外部データ

Google 事前学習済み Word2vec

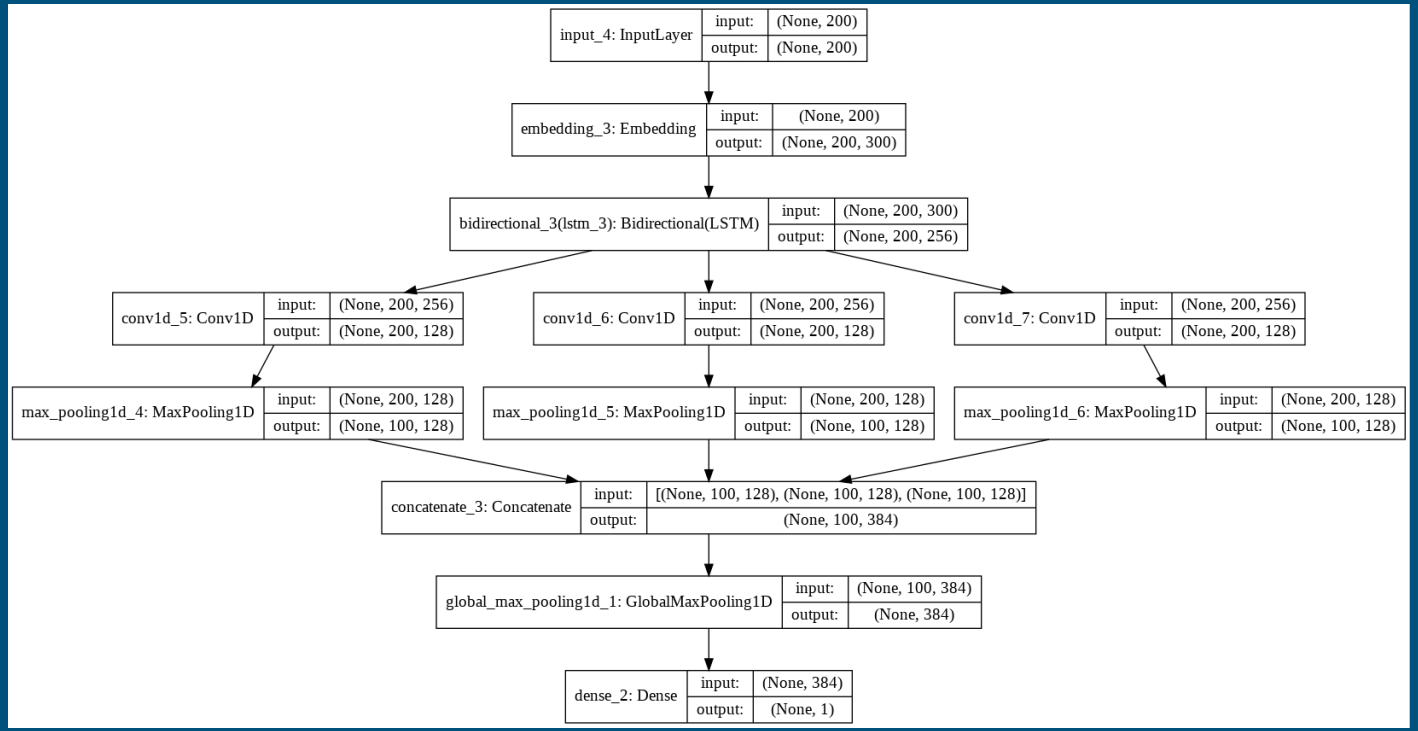
https://code.google.com/archive/p/word2vec/

Tensorflow hub 事前学習済み ELMo

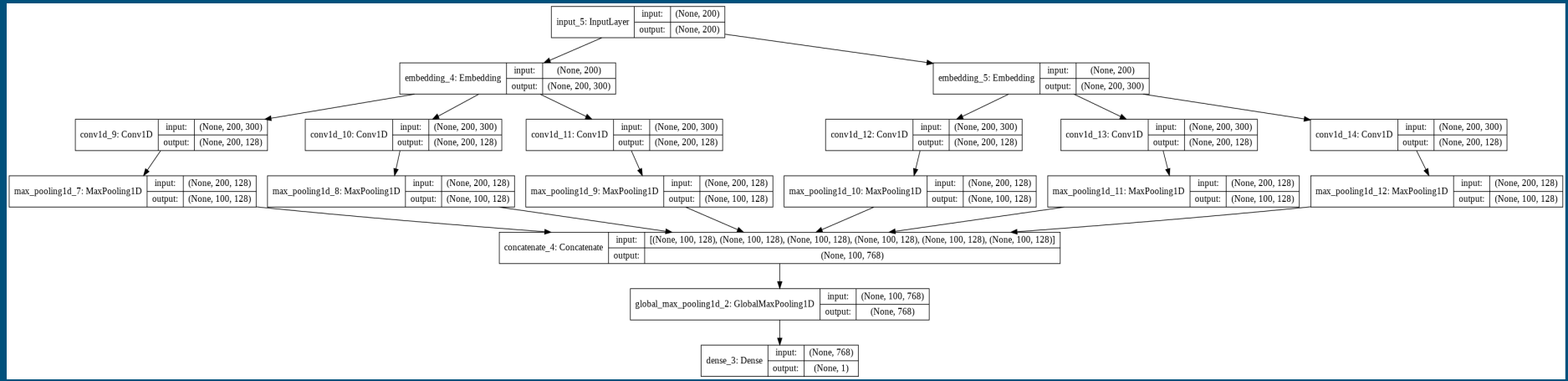
https://tfhub.dev/google/elmo/3

ご清聴ありがとうございました

[モデル]BiLSTM＋CNN



[モデル]CNN


input_5: InputLayer | input: (None, 200) | output: (None, 200)
embedding_4: Embedding | input: (None, 200) | output: (None, 200, 300)
embedding_5: Embedding | input: (None, 200) | output: (None, 200, 300)
conv1d_9: Conv1D | input: (None, 200, 300) | output: (None, 200, 128)
conv1d_10: Conv1D | input: (None, 200, 300) | output: (None, 200, 128)
conv1d_11: Conv1D | input: (None, 200, 300) | output: (None, 200, 128)
conv1d_12: Conv1D | input: (None, 200, 300) | output: (None, 200, 128)
conv1d_13: Conv1D | input: (None, 200, 300) | output: (None, 200, 128)
conv1d_14: Conv1D | input: (None, 200, 300) | output: (None, 200, 128)
max_pooling1d_7: MaxPooling1D | input: (None, 200, 128) | output: (None, 100, 128)
max_pooling1d_8: MaxPooling1D | input: (None, 200, 128) | output: (None, 100, 128)
max_pooling1d_9: MaxPooling1D | input: (None, 200, 128) | output: (None, 100, 128)
max_pooling1d_10: MaxPooling1D | input: (None, 200, 128) | output: (None, 100, 128)
max_pooling1d_11: MaxPooling1D | input: (None, 200, 128) | output: (None, 100, 128)
max_pooling1d_12: MaxPooling1D | input: (None, 200, 128) | output: (None, 100, 128)
concatenate_4: Concatenate | input: [(None, 100, 128), (None, 100, 128), (None, 100, 128), (None, 100, 128), (None, 100, 128), (None, 100, 128)] | output: (None, 100, 768)
global_max_pooling1d_2: GlobalMaxPooling1D | input: (None, 100, 768) | output: (None, 768)
dense_3: Dense | input: (None, 768) | output: (None, 1)

[モデル]ELMo+Dense