

# Attention機構を用いた 文単位のニューラル評価極性分類

13173037 大森 光      指導教員 小町 守 准教授

平成 29 年 2 月 20 日

## 概要

近年、評価極性分類などの文書分類タスクにおいて、Long Short-Term Memory を用いた Recurrent Neural Network や、Convolutional Neural Network などのニューラルネットワークを用いた手法が優れた分類精度を示している。また機械翻訳や文圧縮などの系列生成タスクでは、入力の中の部分に焦点を当てるかを考慮できる Attention 機構を用いた手法が優れた性能を示している。本研究では、文脈によって変化する単語の重要度を考慮するために RNN や CNN などの手法に対し Attention 機構を適用し、文の評価極性分類タスクにおいて分類精度の向上に取り組んだ。

## 1 はじめに

近年、インターネットや SNS の普及によって、ユーザが商品やサービスなどの情報を自由に発信する機会が増大した。中でも、ユーザのレビュー文やツイッターのテキストはユーザの生の声を含むため、企業などにとっては貴重なものである。そのため大量にあるこれらのデータを解析し自動で評価極性を決定することは、より良い商品やサービスを提供する上で重要な技術の一つであると言える。

近年、評価極性分類などの文書分類タスクにおいて、長距離の依存関係を捉えることができる Long Short-Term Memory (LSTM) を用いた Recurrent Neural Network (RNN) [4] や、文の局所的な n-gram 素性を抽出することができる Convolutional Neural Network (CNN) [2, 3] などのニューラルネットワークを用いた手法が優れた分類精度を示している。また、機械翻訳 [1] や文圧縮 [7] などの系列生成タスクでは、入力の中の部分に焦点を当てるかを考慮できる Attention 機構が優れた性能を示している。

我々は文の評価極性を決める際に、単語の重要度が文脈によって変化すると考える。例えば、以下の Positive[P] な文と Negative[N] な文では文の評価極性を分類する際の単語“きれい”の重要度が異なる。

[P] 山頂では、とてもきれいな景色が楽しめる。

[N] 彼女はきれいだが、性格が悪い。

我々はこの入力文の文脈によって変化する単語の重要度を考慮するために、文の評価極性分類タスクに Attention 機構を適用する。

Sentiment Stanford Treebank (SST) [8] を用いた実験の結果、Attention 機構の適用によって、RNN による手法、CNN による手法において、一部のタスクで文単位の評価極性の分類精度を改善できた。

## 2 関連研究

### 2.1 RNN による評価極性分類

Li ら [4] は、LSTM や LSTM を双方向に結合した Bi-directional LSTM (BiLSTM) を用いた RNN による手法が、前後の文脈を考慮し、否定関係やサブテキストの接続関係を捉えることができることを示した。

これらの手法では末尾の隠れ層から得られたベクトルのみを分類に用いているが、我々の手法では全ての隠れ層のベクトルを分類に用いることで、より柔軟に各単語の重要度を考慮することができる。

### 2.2 CNN による評価極性分類

Kalchbrenner ら [2] や Kim ら [3] は、画像処理に活発に取り入れられていた CNN を自然言語処理の文書分類タスクに応用し、文から n-gram の素性を抽出するモデルを構築した。しかし、これらの単語系列を入力とする CNN では長距離の依存関係を扱えないため、Ma ら [6] は依存構造を用いて否定関係や従属関係を考慮した Dep-CNN を提案した。これらの手法では pooling 層において各 n-gram の重要度を考慮しないが、本研究では Attention 機構を用いて各 n-gram の重要度を考慮することができる。

### 2.3 RNN と CNN を併用した評価極性分類

Zhang ら [13] や Zhou ら [15] は LSTM から得られた各タイムステップごとの情報を持つ隠れ層のベクトルを CNN へ入力するモデルを構築した。Liang ら [5] や Wang ら [11] は CNN から得られた n-gram 表現ベクトルを BiLSTM に入力するモデルを構築した。これらのモデルでは句単位の素性を学習し、かつ長距離の依存関係を捉えた文表現を得ることができた。これらの手法は、pooling 層において各ウィンドウの外にあるグローバルな重要度は考慮されておらず、LSTM における各タイムステップの重要度も等しい。我々の手法ではこの 2 つの重要度も Attention 機構によって考慮することができる。

### 2.4 Attention 機構

機械翻訳 [1] や文圧縮 [7] では、Attention 機構によって入力文のある単語に注目しながら翻訳（要約）文を生成している。我々は評価極性分類も同じ特徴を持つタスクであると

考えており、入力文のある単語や句に注目しながら評価極性のラベルを出力することで分類精度の改善が期待できる。

Zhou ら [15] は文単位の評価極性分類タスクにおいて、BiLSTM に Attention 機構を適用することで、精度の向上が見られることを示した。我々は BiLSTM だけでなく、その他のネットワークモデルに対しても Attention 機構を適用する。

宮崎ら [17] は日本語評価極性分類で、TreeLSTM [16] に Attention 機構を導入した。TreeLSTM は LSTM を句構造木に適用したものである。この手法では、句構造木の葉には学習済みの単語分散表現が用いられており、それらを組み上げて句のベクトルを構成する。そのため単語や句のベクトルは文脈に依存せず常に一定である。本研究では文脈によって各単語の重要度を柔軟に変化させて考慮する手法を提案する。

## 3 Attention 機構を用いた文単位の評価極性分類

### 3.1 RNN

BiLSTM に Attention 機構を適用したモデルを図 1 に示す。BiLSTM に Attention 機構を適用したモデルを図 1 に示す。単語ベクトル  $x_i \in \mathcal{R}^d$  を RNN に入力し、各タイムステップ  $t$  ごとに文脈を考慮した隠れ層のベクトル  $h_t \in \mathcal{R}^{d^{(h)}}$  が得られる。ここで、 $i \in [1, 2, \dots, l]$  は単語のインデックス、 $l$  は入力文長を表し、 $d$  は単語分散表現の次元を、 $d^{(h)}$  は隠れ層ベクトルの次元を表す。

我々は RNN に Attention 機構を適用し、各単語の重要度を考慮した文の表現ベクトル  $a^{(R)} \in \mathcal{R}^{d^{(h)}}$  を得る。

$$a^{(R)} = \sum_i a_i^{(R)} \odot h_i \quad (1)$$

$$a_i^{(R)} = \frac{\exp(W^{(R2)} \tanh(W^{(R1)} h_i))}{\sum_k \exp(W^{(R2)} \tanh(W^{(R1)} h_k))} \quad (2)$$

ここで、 $\odot$  は要素積を表す。 $W^{(R1)} \in \mathcal{R}^{d^{(a)} \times d^{(h)}}$ 、 $W^{(R2)} \in \mathcal{R}^{1 \times d^{(a)}}$  はパラメータ行列であり、 $d^{(a)}$  は Attention ベクトルの次元を表す。ここでは簡略化のため、 $W^{(R1)}$  と  $W^{(R2)}$  に対応するバイアス項を記述していない。そして、 $a^{(R)}$  をソフトマックス分類器によって分類する。

### 3.2 CNN

CNN に Attention 機構を適用したモデルを図 2 に示す。単語ベクトル  $x_i \in \mathcal{R}^d$  から行列  $X = [x_1, x_2, \dots, x_l]$ 、 $X \in \mathcal{R}^{l \times d}$  を得る。そして、畳み込みフィルタ  $w \in \mathcal{R}^{d \times k}$  を用いて、単語 n-gram の特徴を抽出した特徴マップ  $c \in \mathcal{R}^{l-k+1}$  を得る。

$$c_i = f(w \cdot x_{i:i+k-1} + b) \quad (3)$$

$$c = [c_1, c_2, \dots, c_{l-k+1}] \quad (4)$$

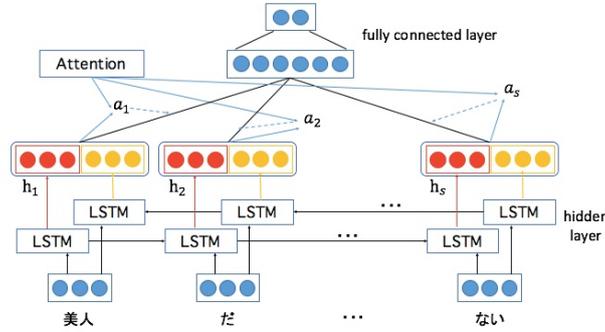


図 1: BiLSTM に Attention 機構を適用したネットワークモデル

ここで、 $k$  はウィンドウサイズを表し、 $f$  は非線形関数、 $b$  はバイアス項を表す。Kim ら [3] は特徴マップ  $c$  に max pooling を適用することで、そのフィルタが捉える素性を抽出している。我々は、max pooling の代わりに Attention 機構を用いることで、各ウィンドウから得られる  $n$ -gram 素性  $c_i$  の重要度を考慮し、特徴マップの pooling 素性  $p$  を得る。

$$p = \sum_i a_i^{(C)} \odot c_i \quad (5)$$

$$a_i^{(C)} = \frac{\exp(W^{(C2)} \tanh(W^{(C1)} c_i))}{\sum_k \exp(W^{(C2)} \tanh(W^{(C1)} c_k))} \quad (6)$$

ここで、 $\odot$  は要素積を表す。 $W^{(C1)} \in \mathcal{R}^{d^{(a')} \times 1}$ 、 $W^{(C2)} \in \mathcal{R}^{1 \times d^{(a'')}}$  はパラメータ行列であり、 $d^{(a')}$  は Attention ベクトルの次元を表す。ここでは簡略化のため、 $W^{(C1)}$  と  $W^{(C2)}$  に対応するバイアス項を記述していない。そして、全ての特徴マップから得られる  $p^j$ 、 $j \in [1, 2, \dots, m]$  を結合し、文の表現ベクトル  $a^{(C)}$  を得る。

$$a^{(C)} = p^1 \oplus p^2 \oplus \dots \oplus p^m \quad (7)$$

ここで、 $m$  は特徴マップの数を表す。この  $a^{(C)}$  をソフトマックス分類器によって分類する。

## 4 実験

### 4.1 実験データ

実験には、Stanford Sentiment Treebank (SST) [8] を利用した。SST は文に very negative, negative, neutral, positive, very positive の 5 つのラベルがついたデータセットであり、これらの 5 値分類タスク (SST-5) と、neutral の文を取り除いたポジティブまたはネガティブの 2 値分類タスク (SST-2) がある。データセットは、学習セットが 8,544 文、開発セットが 1,101 文、テストセットが 2,210 文に分割されている。

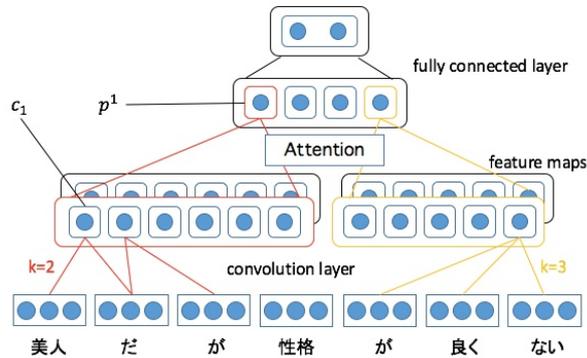


図 2: CNN に Attention 機構を適用したネットワークモデル

## 4.2 比較手法

我々は以下の手法に Attention 機構を適用する。単語分散表現は word2vec [10] による事前学習を行った 300 次元のものを利用し、未知語は  $[-0.25, 0.25]$  の一様乱数で初期化したベクトルとした。また LSTM の隠れ層の大きさは 300 次元とし、CNN のウィンドウサイズは 3, 4, 5 で、それぞれ 100 個の特徴マップを持つ。ミニバッチサイズは LSTM, LSTM+atten では 16 文でそれ以外では 50 文とした。最適化手法には AdaDelta [12] ( $\epsilon = 10^{-6}$ ,  $\rho = 0.95$ ) を適用し、勾配クリッピングは 3 に設定した。ソフトマックスで分類する前の最終層には Dropout を行い、確率は 50% に設定した。全ての手法は chainer [9] によって実装した。

**LSTM:** LSTM で得られた隠れ層のベクトルをソフトマックス分類器によって分類する。

**LSTM+atten:** LSTM に Attention 機構を適用し、得られたベクトルをソフトマックス分類器によって分類する。

**BiLSTM:** 双方向の LSTM で得られた隠れ層のベクトルをソフトマックス分類器によって分類する。

**BiLSTM+atten:** BiLSTM に Attention 機構を適用し、得られたベクトルをソフトマックス分類器によって分類する。

**CNN:** CNN の pooling 層に max pooling を適用し、得られたベクトルをソフトマックス分類器によって分類する。

**CNN+atten:** CNN の pooling 層に Attention 機構を適用し、得られたベクトルをソフトマックス分類器によって分類する。

**BiLSTM+CNN:** BiLSTM によって得られた隠れ層のベクトルを単語ベクトルとして CNN に入力する。CNN の pooling 層に max pooling を適用し、得られたベクトルをソフトマックス分類器によって分類する。

表 1: 各手法の正解率と標準偏差

Method	SST-2	SST-5
LSTM	86.2 (0.68)	48.3 (1.19)
LSTM+atten	86.3 (0.27)	48.6 (1.29)
BiLSTM	87.2 (0.49)	49.3 (1.50)
BiLSTM+atten	87.6 (0.35)	50.0 (0.29)
CNN	86.3 (0.27)	46.5 (1.13)
CNN+atten	86.0 (0.20)	47.2 (0.37)
BiLSTM+CNN	87.5 (0.67)	48.9 (1.44)
BiLSTM+CNN+atten	88.2 (0.45)	49.2 (0.70)

**BiLSTM+CNN+atten:** BiLSTM によって得られた隠れ層のベクトルを単語ベクトルとして CNN に入力する。CNN の pooling 層に Attention 機構を適用し、得られたベクトルをソフトマックス分類器によって分類する。

### 4.3 結果

実験の結果を表 1 に示す。5 回試行し得られた正解率の平均値を結果とし、丸括弧内の数値は標準偏差を表す。LSTM と LSTM+atten を比較すると、Attention 機構を組み合わせた LSTM+atten の方が SST-2 と SST-5 の両方のタスクにおいて高い分類精度を示すことがわかる。また、BiLSTM と BiLSTM+atten を比較しても、同様に Attention 機構を組み合わせた BiLSTM+atten の方が SST-2 と SST-5 の両方で高い分類精度を示した。次に、CNN と CNN+atten を比較すると、SST-2 においては、pooling 層に max pooling を用いた CNN の方がわずかに高い分類精度を示した。ところが SST-5 では、pooling 層に Attention 機構を用いた CNN+atten の方が優れた分類精度を示した。

我々は RNN と CNN を組み合わせた手法として BiLSTM+CNN も比較手法として取り入れた。CNN と比較するとどちらのタスクでも大きく精度が改善されている。また Attention 機構を加えた BiLSTM+CNN+atten は比較手法の中で SST-2 においては最も高い正解率を示した。

標準偏差に注目すると、BiLSTM+atten は BiLSTM より正解率の揺れが小さいことが確認でき、SST-5 においては BiLSTM より 1.21 ポイント小さい。CNN+atten に関しても同様に、SST-5 においては CNN より 0.76 ポイント小さい。このことから Attention 機構は正解率の揺れを抑える効果があることがわかる。

次に、SST-2 と SST-5 における各手法の分布を図 3,4 にそれぞれ示す。SST-5 における CNN の正解率の最大値は 48.3 であったのに対し、CNN+atten の正解率の最大値は 47.7 であった。同様に SST-5 における BiLSTM の正解率の最大値は 52.0 であったのに対し、BiLSTM+atten の最大値は 50.4 であった。どちらも Attention 機構を用いない手法の方が正解率の最大値は高かったが、Attention 機構を用いることで、正解率の揺れを抑えることができることがわかる。

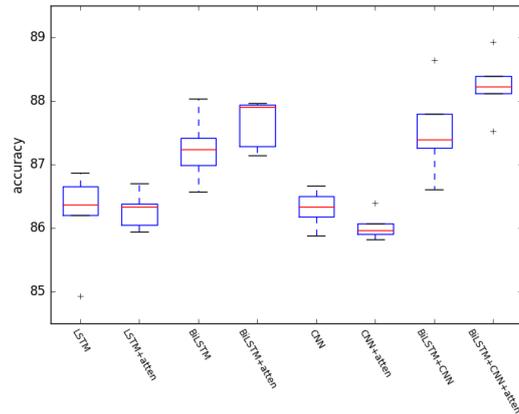


図 3: SST-2 における正解率の分布

## 5 議論

実験の結果、LSTM を適用した RNN の手法では Attention 機構が効果的に働いていることがわかる。これは Attention 機構により各単語の重要度を考慮することできるため、より表現能力の高い文ベクトルを作成できるからであると考えられる。

次に CNN と CNN+atten を比較する。CNN は max pooling で 1 つの特徴マップから 1 つの素性を抽出するのに対し、Attention 機構を pooling 層に適用した CNN+atten では、1 つの特徴マップから複数の素性を抽出する。SST-5 においては、Attention 機構により各 n-gram の重要度を考慮する CNN+atten が効果的であった。SST-2 では CNN の正解率が CNN+atten よりわずかに高かったが、RNN の手法においても Attention 機構による精度の向上は小さい。一方 RNN と CNN を組み合わせた手法においては Attention 機構を加えることで 0.7 ポイント精度が改善している。そのため 2 値の文単位評価極性分類タスクでは、Attention 機構の効果はネットワークの複雑さと関係があると考えられる。

BiLSTM+CNN の結果からは、CNN の入力として BiLSTM の隠れ層のベクトルが学習済みの単語分散表現よりも有効であることが分かった。これは BiLSTM によって文脈を考慮した表現能力の高い単語ベクトルが作られたからだと考えられる。

標準偏差や図 3,4 に注目すると、Attention 機構が正解率の揺れを抑える効果があることがわかる。これは通常の RNN の手法では末尾の隠れ層から得られたベクトルしか考慮せず、通常の CNN では max pooling のため 1 つの n-gram しか考慮しないが、Attention 機構によって文中の全ての単語や n-gram を考慮することができるようになるからである。

### 5.1 可視化

ここで、CNN の特徴マップと CNN+atten の特徴マップに attention がはられた様子を図 5,6 にそれぞれ示す。縦軸は各特徴マップ、横軸は各 n-gram である。“<PAD>” は入力の最大文長に合わせてパディングしたものである。図 5 では “admired this work” を max

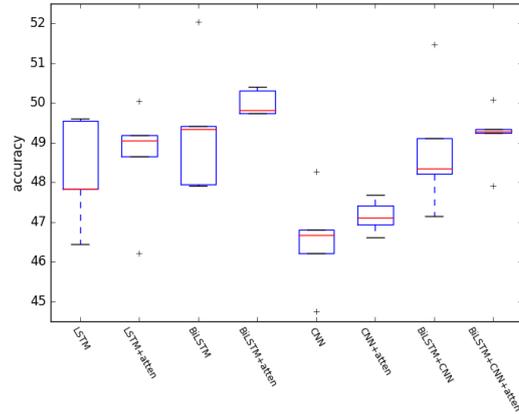


図 4: SST-5 における正解率の分布

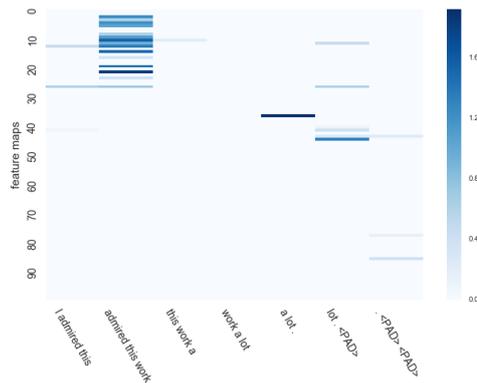


図 5: CNN の特徴マップ (予測:positive 正解:very positive)

pooling で抽出するような特徴マップが多く見られることがわかる。CNN では 1 つの特徴マップから 1 つの n-gram を取り出すため、“admired”と “admired”の極性を強調する意味を持つ “lot” を特徴マップ内で考慮することができない。そのため、このモデルでは “admired” の正解ラベルである positive を予測するものと考えられる。

一方、図 6 では図 5 と同様に、“admired this work” に attention を強くはる特徴マップが多く見られることがわかる。また、“admired this work” と同時に “a lot .” や “lot . <PAD>” を考慮している特徴マップもあり、CNN+atten では 1 つの特徴マップから複数の n-gram を抽出することができていることがわかる。そのため、このモデルでは “lot” を含む n-gram が “admired” を含む n-gram の極性を強調することを学習し、very positive を予測することができると考えられる。

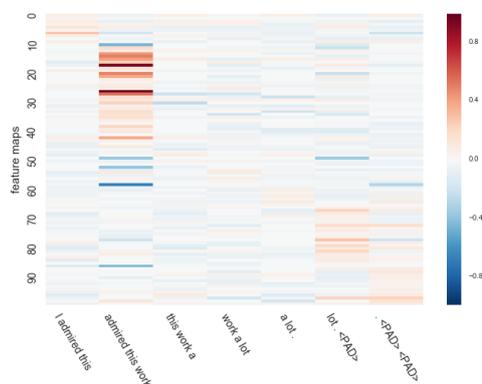


図 6: CNN+atten の attention がはられた特徴マップ (予測:very positive 正解:very positive)

## 5.2 エラー分析

pooling 層に max pooling を用いる CNN は Attention 機構を用いる CNN+atten より正解率のばらつきが大きかったことから、CNN と CNN+atten の出力結果を比較しエラー分析を行うことで Attention 機構の効果を調べる。SST-5 で 5 回実験を行い、テストセットの内 5 回全てで正解した文を“正解 A”、5 回中少なくとも 1 回は正解できた文を“正解 B”、5 回全て間違いそのどれも同じラベルを出力した文を“ミス A”、5 回全て間違い正解以外のラベルをバラバラに出力した文を“ミス B”と表記する。“ミス A”と“ミス B”は正解率に影響しない文であるが、出力結果をバラつかせる要素を含む文であることがいえるため区別した。集計結果を表 2 にまとめた。

CNN と CNN+atten の正解 A はそれぞれ 536 文、638 文あり、CNN+atten が 102 文多い結果となった。2 つの手法の正解率の差は単純にこの部分から生じているといえる。また CNN で正解 B、CNN+atten で正解 A となる文は max pooling では間違いやすいが、Attention 機構を用いることで安定して正解を予測できる文であり、これらは CNN と CNN+atten の正解率と標準偏差に重要な差を与える文であることがいえる。この 246 文に注目すると 246 文のうち約 6 割の 147 文が、ポジティブ (ネガティブ) な単語やフレーズを含むネガティブ (ポジティブ) な文であることがわかった。このことから文の極性と文を構成する単語や句の極性が一致しない場合、単語や句の重要度をより考慮をする必要があるため、Attention 機構の効果が高いことがいえる。

## 5.3 average pooling との比較

CNN では pooling 層に average pooling を用いる方法も一般的である。特徴マップから 1 つの素性のみ取り出す max pooling と違い、average pooling は特徴マップ全体の素性を平均して 1 つの素性として取り出す。そのため 1 つの特徴マップから複数の素性を考慮す

表 2: CNN と CNN+atten の出力結果を比較した

CNN \ CNN+atten	正解 A	正解 B	ミス A	ミス B	合計
正解 A	388	142	4	2	536
正解 B	246	507	157	41	951
ミス A	3	106	277	59	445
ミス B	1	37	115	125	278
合計	638	792	553	227	2210

るという点では CNN+atten と同様の働きを持っている。文書分類ではあるデータセットにおいて average pooling より max pooling が優れているという結果がすでに示されている [14]。しかし自然言語処理における CNN では、入力となる文の長さがそれぞれ異なることにより最大文長に合わせて入力文の長さを同一にするパディング処理が行われる。そのため実際には文長で平均をとっておらず、特徴マップの長さ（文長+パディング長）で平均をとっている。我々は文長で average pooling する “CNN+ave (sent len)” と特徴マップの長さで average pooling する “CNN+ave (map len)” を新たに比較手法として加え、各 pooling 手法との比較と average pooling の領域範囲による違いを調べる。結果を表 3 に示す。

CNN+ave (sent len) と CNN+ave (map len) を比較すると、各文長で平均をとるほうが正解率が高いことがわかる。SST には文長が短いものから 50 単語以上含む長いものまで様々な文長の文を含むため、この結果は当然であると言える。CNN+atten と CNN+ave (sent len) を比較すると、正解率は average pooling を用いるほうが正解率が高いことがわかる。

ここで CNN+ave (sent len) の特徴マップの様子を図 7 に示す。図 5 と比較すると、図 5 では、ある 1 つのウィンドウからは値の大きい素性を出力され、他のウィンドウでは 0 に近い、もしくは 0 が出力された特徴マップが多く見られたが、図 7 では、複数のウィンドウから値の大きい素性が出力された特徴マップが多く見られる。これは pooling 方法の違いによって特徴マップの学習の仕方が異なるということを表したものである。max pooling では最も大きい素性以外は考慮されないため、それ以外の素性は学習するにつれて小さくなっていくが、average pooling では 1 つの特徴マップ内でも重要な素性であれば、複数の素性の値が大きくなるように学習する。図 6 と比較すると、Attention 機構によって高い重要度を与えられていた “admired this work” と “lot . <PAD>” の 2 つを大きな素性として持つ特徴マップが図 7 でもみられるのが分かる。このことから、文長で平均をとる average pooling では各ウィンドウから得られる素性の重要度は等しいが、特徴マップ内で複数の重要な素性が大きくなるように学習され、結果的に CNN+atten と同様に特徴マップ内の素性の重要度を考慮したような 1 つの素性を抽出することができる。

## 6 おわりに

本研究では、LSTM を適用した RNN の手法と CNN の手法に対して、Attention 機構を適用し、文単位の評価極性分類において分類精度の向上に取り組んだ。今後は、CNN

表 3: 各 pooling 手法の正解率と標準偏差

Method	SST-2	SST-5
CNN	86.3 (0.27)	46.5 (1.13)
CNN+atten	86.0 (0.20)	47.2 (0.37)
CNN+ave (sent len)	86.6 (0.51)	47.3 (0.44)
CNN+ave (map len)	84.6 (0.38)	46.0 (0.69)

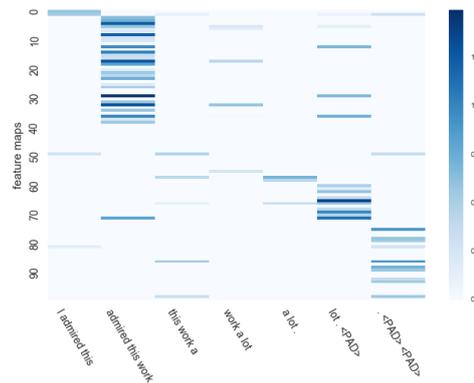


図 7: CNN+ave (sent len) の特徴マップ (予測:very positive 正解:very positive)

の畳み込み層に Attention 機構を適用したモデルに取り組みたい。

## 7 謝辞

本研究に際して、研究室に配属されてから 1 年間ご指導ご鞭撻を頂きました小町守先生に深く感謝いたします。

また、研究室の同期・先輩には日々の研究生活の中で、多くのアドバイスやご協力を頂きました。梶原智之さんには熱心なご指導を賜り、本研究やこれからの研究生活に関して様々なアドバイスを頂きました。宮崎亮輔さん、佐藤貴之さん、小平知範さんには、プログラムの実装において貴重な時間を割いてご協力頂きました。

皆様へ心から感謝の気持ちと御礼を申し上げたく、謝辞にかえさせていただきます。

## 参考文献

- [1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the 3rd International Conference on Learning Representations*, pp. 1–15, 2015.

- [2] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A Convolutional Neural Network for Modelling Sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 655–665, 2014.
- [3] Yoon Kim. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1746–1751, 2014.
- [4] Jiwei Li, Thang Luong, Dan Jurafsky, and Eduard Hovy. When Are Tree Structures Necessary for Deep Learning of Representations? In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 2304–2314, 2015.
- [5] Depeng Liang and Yongdong Zhang. AC-BLSTM: Asymmetric Convolutional Bidirectional LSTM Networks for Text Classification. In *arXiv preprint arXiv:1611.01884*, pp. 1–7, 2016.
- [6] Mingbo Ma, Liang Huang, Bowen Zhou, and Bing Xiang. Dependency-based convolutional neural networks for sentence embedding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 174–179, 2015.
- [7] Alexander M. Rush, Sumit Chopra, and Jason Weston. A Neural Attention Model for Abstractive Sentence Summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 379–389, 2015.
- [8] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, 2013.
- [9] Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. Chainer: a next-generation open source framework for deep learning. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*, 2015.
- [10] Greg Corrado, Tomas Mikolov, Kai Chen and Jeffrey Dean. Efficient estimation of word representations in vector space. In *In Proceedings of NIPS*, 2013.
- [11] Xingyou Wang, Weijie Jiang, and Zhiyong Luo. Combination of Convolutional and Recurrent Neural Network for Sentiment Analysis of Short Texts. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 2428–2437, 2016.
- [12] Matthew D Zeiler. Adadelata: An adaptive learning rate method. In *arXiv preprint arXiv:1212.5701*, 2012.

- [13] Rui Zhang, Honglak Lee, and Dragomir R. Radev. Dependency Sensitive Convolutional Neural Networks for Modeling Sentences and Documents. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1512–1521, 2016.
- [14] Ye Zhang and Byron Wallace. A sensitivity analysis of (and practitioners’ guide to) convolutional neural networks for sentence classification. In *arXiv preprint arXiv:1510.03820*, 2015.
- [15] Peng Zhou, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, and Bo Xu. Text Classification Improved by Integrating Bidirectional LSTM with Two-dimensional Max Pooling. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 3485–3495, 2016.
- [16] Xiaodan Zhu, Parinaz Sobihani, and Hongyu Guo. Long Short-Term Memory Over Recursive Structures. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1604–1612, 2015.
- [17] 宮崎亮輔, 小町守. 極性辞書を利用した句構造による注意型ニューラル評価極性分類. 第 227 回情報処理学会自然言語処理研究会, pp. 1–10, 2016.