

# Aspect-Based Sentiment Analysis における Neural Attention の効率的な 利用方法

朝倉 遼<sup>†</sup> 新妻 弘崇<sup>†</sup> 太田 学<sup>†</sup>

<sup>†</sup> 岡山大学大学院自然科学研究科 〒700-8530 岡山県岡山市北区津島中 3-1-1

E-mail: †{asakura,niitsuma,ohta}@de.cs.okayama-u.ac.jp

あらまし 文書には書き手の様々な感情が含まれており、それらの極性を特定するのが文書分析のタスクの一つである。一般に、これは文書中に含まれる観点を予測し、それに対する感情の極性を予測するというタスクであり、Aspect-Based Sentiment Analysis と呼ばれる。本稿では、Neural Attention と呼ばれる、ある状態を生成する際に、ある別の情報を選択的に参照する処理を Neural Network に学習させる仕組みの Aspect-Based Sentiment Analysis への応用方法、またその半教師あり学習への応用方法を提案する。

キーワード Aspect-Based Sentiment Analysis, Neural Attention, Semi-Supervised Learning

## 1. はじめに

文書の感情極性推定は自然言語処理における基本的なタスクの一つである。それは文書に positive, negative などの感情極性ラベルを付与することでなされる。しかし、文書全体についての感情極性推定は有用でない場合もある。なぜなら、文書には複数の異なる感情極性が含まれており、その対象も様々であることが多いためである。Aspect-Based Sentiment Analysis (ABSA) [10] はこの問題に対処するためのフレームワークである。ABSA では文書から様々な観点を抽出し、その観点到感情極性を付与する。以下の飲食店のレビューの例文を使って ABSA について、より詳細に説明する。

“Food is good, but attitude of clerk is so rude.”

この飲食店のレビューには“料理の質”，“サービスの質”という2つのレビューの観点が存在する。前者の“料理の質”については FOOD#QUALITY，後者の“サービスの質”については SERVICE#QUALITY といった、特性の組合せを # で区切った文字列のラベルで表わすことにする。これらのラベルは Aspect Category と呼ばれる。また、FOOD#QUALITY についての上記の例文中の対象は“Food”，同様に SERVICE#QUALITY については“attitude of clerk”である。これらのような意見の対象となる表現は Opinion Target Expression (OTE) と呼ばれる。そして、(Aspect Category, OTE) のペアに対して付与されるべき、positive, negative, neutral のいずれかの感情極性を推定する。この推定問題が ABSA である。

SemEval 2016 Task 5 Subtask 1 (SE16T5S1)<sup>(注1)</sup> は飲食店や商品の英語のレビューのデータセットを利用して ABSA を行うタスクであり、本稿では SE16T5S1 の飲食店のレビューを扱うレストランメインのデータセットをベンチマークとして利用する。データセットとタスクの詳細については 2. 節で述べる。

ABSA では文書中の観点を考慮する必要があるが、本稿では Neural Attention (NA) を用いてそれに対処する。NA はある状態を生成する際に、ある別の情報を選択的に参照する処理を Neural Network に学習させる仕組みである。NA は、画像キャプション生成 [18]、文書要約 [14]、ニューラル機械翻訳 (Neural Machine Translation; NMT) [1] など、Encoder-Decoder モデルでの利用が多い。特に NMT ではターゲット言語側の単語を生成する際にソース言語中の特定の単語を参照することが有効であることが知られており [1]、Word-by-word attention と呼ばれる。これはデコーダの出力単語から、エンコーダが出力したある単語への参照の強さを学習可能なパラメータを用いて計算し、ターゲット言語の生成に用いることでなされる。本稿では、この NMT での利用に基づいて、ABSA における NA の効率的な利用方法を提案する。

また、本稿ではその NA を用いたモデルに半教師あり学習を適用する方法も提案する。一般に教師あり学習における、外部データを用いた事前学習は様々なタスクにおいて有効とされる。ここで外部データとは、ドメインは異なるが問題の解決に有効と考えられるデータを指す。SE16T5S1 で提供される学習用データは 2000 レビューにも満たないため、外部データでの事前学習を行った先行研究は多い。しかし、利用できる外部データのほとんどには Aspect Category を考慮しない全体的な感情極性などのラベルしかついておらず、感情極性の他に、Aspect Category, OTE など多くの教師ラベルを必要とする ABSA のようにタスクを行うには不十分である。例えば、Yelp Academic Dataset<sup>(注2)</sup> のレビューデータなどがこれに該当する。また、解決したい問題によってはラベル付きデータを全く用意できないこともあるため、教師なしデータを効果的に利用する手段として、半教師あり学習は重要であるといえる。近年、Neural Network を用いることが一般的になっているタスクのほとんどは教師ありタスクであるが、半教師

(注1): <http://alt.qcri.org/semeval2016/task5/>

(注2): [https://www.yelp.com/dataset\\_challenge/](https://www.yelp.com/dataset_challenge/)

あり学習を行うための手法もいくつか提案されている。例えば、生成モデルをベースとしたもの [8], Virtual Adversarial Training (VAT) [12], Ladder Network [13] などがある。[8] は Variational Autoencoder [9] をベースとした手法であり、パラメタ化されたガウス分布からサンプルした潜在変数を用いて画像の分布、ラベルと画像の同時分布を生成し、半教師あり学習に用いる。VAT は近い距離にあるデータ点の事後分布はおおよそ同じであるべきであるという考えが元になっている手法である。VAT を実現するためには、入力データに対して微小なノイズを加え、その前後の事後分布を比較すればよいが、学習においてより効果的なノイズを利用することがよいモデルの実現につながる。そこで VAT は Adversarial Example [4] を作成する際に加えらる敵対摂動 (Adversarial Perturbation; AP) を擬似的に作成し、それを加える前後で事後分布を比較し、その誤差を教師なし誤差として教師あり誤差に加えて最小化する。ここで、Adversarial Example とは、識別モデルが最も識別エラーを起こしやすい方向の摂動が加えられたデータ点のことである。

Ladder Network も同様に微小なノイズを用いて半教師あり学習を行うための手法であるが、VAT とは異なり、ノイズを加えた後の事後分布からノイズを加える前の入力を復元するように学習する。そしてその復元誤差を教師なし誤差とし、教師あり誤差と同時に最小化することで入力データ点の摂動に頑健になり、汎化性能が向上する。ノイズを打ち消すようにモデルを学習することが汎化性能向上に寄与することは denoising autoencoder (dAE) [16] などによってすでに示されており、Ladder Network も dAE に似た構造となっている。本稿では感情極性推定タスクにおいて、Ladder Network で用いられている手法を応用し、半教師あり学習を行う。

## 2. タスク概要

SemEval 2016 Task 5 Subtask 1 (SE16T5S1)<sup>(注3)</sup> は飲食店や商品の英語のレビューのデータセットを利用して ABSA を行うタスクである。本稿では SE16T5S1 の飲食店のレビュー集であるレストランドメインのデータセットをベンチマークとして利用する。データセットは英語のレストランレビューであり、学習用が 2000 レビュー、テスト用が 676 レビューとなっている。以下はこのデータセットからの抜粋である。

“The wine list has many good values.”

```
<Opinion category="DRINKS#PRICES"
  polarity=" positive" />
```

<Opinion> は Aspect Category である category と感情極性である polarity 属性を持つ。<Opinion> は 1 レビュー内に複数存在する場合もある。1. 節でも述べた通り、category は Entity#Attribute の形で表現され、全部で 12 定義されている。また、polarity は positive, negative, neutral の 3 種類がある。SE16T5S1 は以下の 3 つのサブタスクから構成されるが、本稿では Slot1, Slot3 に取り組む。

Slot1 文中に 0 個以上含まれる Aspect Category を抽出する。抽出の精度は F 値で評価する

Slot2 文中に 0 個以上含まれる OTE を抽出する。抽出の精度は F 値で評価する。OTE は単語かフレーズである。

Slot3 与えられたレビュー文と Aspect Category のペアに対して Positive, Negative, Neutral のいずれかの感情極性を付与する。結果はテストデータ中のすべてのペアに対し、正しく付与できた割合で評価する。

## 3. Neural Attention

本稿では ABSA のタスクである Aspect Category の抽出 (Slot1), それを踏まえての感情極性の予測 (Slot3) の 2 つのタスクに取り組む。初めに、Neural Attention を用いて Slot3 を行うモデルについて説明する。そして、そのモデルの一部を拡張し、Slot1 を行う方法について説明する。Slot3 を行うモデルはソース系列のエンコード方法などを除いて先行研究 [17], [19] と同一のモデルである。

### 3.1 定式化

ABSA では感情極性を予測する際に与えられる Aspect Category を考慮する必要があるが、ここで Neural Attention (NA) を用いる。具体的には、与えられた Aspect Category に対し、エンコーダの出力の中から重要であると思われるベクトルを選び出す。例えば FOOD#QUALITY が与えられた場合、入力される文章の単語の列から “yummy” を選び出すと言った具合である。初めに ABSA における NA を定式化する。ソース系列  $X$ , エンコーダの出力系列  $H$ , ある Aspect Category  $cat_i$  (例: FOOD#QUALITY) を表すベクトル  $c_i$  を並べた系列  $C$ , 感情極性分布の系列  $Y$  を以下のように定義する。

$$\begin{aligned} X &= (x_1, \dots, x_L), & x_i &\in \mathbb{R}^{d_x} \\ H &= (h_1, \dots, h_L), & h_i &\in \mathbb{R}^{d_h} \\ C &= (c_1, \dots, c_N), & c_i &\in \mathbb{R}^{d_c} \\ Y &= (y_1, \dots, y_N), & y_i &\in \mathbb{R}^{d_p} \end{aligned} \quad (1)$$

ソース系列は文を表す単語ベクトル列、 $L$  はその長さである。 $N$  は Aspect Category の種類数を表す。本稿の実験では  $N = 12$  である。 $H$  は以下のように  $\theta$  でパラメタ化された関数  $\psi_\theta$  によって生成される文章のエンコード結果である。

$$H = \psi_\theta(X) \quad (2)$$

$X, \theta$  は学習によって最適化されているとする。 $C$  は NMT におけるデコーダの各ステップの出力系列に対応するが、ABSA ではこれを Aspect Category のベクトルの並びとする。仮にエンコーダの出力系列中の  $h_i$  が  $y_j$  に強く影響を与えるとする場合、 $y_j$  を生成する際、直接  $h_i$  を参照することを考える。これを  $h_i$  を  $\{h_1, \dots, h_L\}$  から選択する問題と考えると、 $h_i$  の期待値  $\bar{h}_j$  は次式で計算される。

$$\bar{h}_j = \sum_{i=1}^L \text{align}(h_i, c_j) h_i$$

(注3): <http://alt.qcri.org/semeval2016/task5/>

ここで,  $\text{align}(\mathbf{h}_i, \mathbf{c}_j)$  は,  $\mathbf{c}_j$  を考慮した上で  $\mathbf{y}_j$  を生成する際の  $\mathbf{h}_i$  の影響力を表すスコア関数  $\text{score}(\mathbf{h}_i, \mathbf{c}_j)$  の値を正規化した値であり, 次式で定義される.

$$\text{align}(\mathbf{h}_i, \mathbf{c}_j) = \frac{\exp(\text{score}(\mathbf{h}_i, \mathbf{c}_j))}{\sum_{k=1}^L \exp(\text{score}(\mathbf{h}_k, \mathbf{c}_j))}$$

score 関数は様々なものが考えられ, Luong ら [11] は NMT において, 以下の式 (3) の 3 種類の比較を行っている.

$$\text{score}(\mathbf{h}_i, \mathbf{c}_j) = \begin{cases} \mathbf{h}_i \cdot \mathbf{c}_j, & \text{dot} \\ \mathbf{h}_i \cdot \mathbf{W} \mathbf{c}_j, & \text{general} \\ \mathbf{w} \cdot \tanh(\mathbf{W}[\mathbf{h}_i, \mathbf{c}_j]) & \text{concat} \end{cases} \quad (3)$$

NMT の場合はここで  $\mathbf{c}_j$  と  $\bar{\mathbf{h}}_j$  を用いて単語の予測を行う. しかし ABSA の場合は,  $\bar{\mathbf{h}}_j$  と学習済みのパラメタ  $\mathbf{W}_p \in \mathbb{R}^{d_p \times d_h}$  を用いることで  $\mathbf{c}_j$  を観点とした場合の入力系列に対する感情極性の予測分布  $\mathbf{y}_j$  を生成する.

$$\mathbf{y}_j = \text{softmax}(\mathbf{W}_p \bar{\mathbf{h}}_j) \quad (4)$$

$\mathbf{y}_j$  の推論を行うモデルを M0 と呼ぶことにし, また,  $(L \times N)$  の行列 Alignment,  $\bar{\mathbf{h}}$  の系列  $\bar{\mathbf{H}}$  をそれぞれ式 (5, 6) で定義する.

$$\bar{\mathbf{H}} = (\bar{\mathbf{h}}_1, \dots, \bar{\mathbf{h}}_L) \quad (5)$$

$$\text{Alignment} = \begin{pmatrix} \text{align}(\mathbf{h}_1, \mathbf{c}_1) & \dots & \text{align}(\mathbf{h}_1, \mathbf{c}_N) \\ \vdots & \ddots & \vdots \\ \text{align}(\mathbf{h}_L, \mathbf{c}_1) & \dots & \text{align}(\mathbf{h}_L, \mathbf{c}_N) \end{pmatrix} \quad (6)$$

### 3.2 ソース系列のエンコード方法

NA においては, 系列のエンコードとしては多くの場合 RNN が用いられる. 特に NMT, 文章要約などの系列変換タスクの場合は, 係り受けなどの複雑な構造を扱う必要があるため, LSTM などが用いられることが多い. しかし, ABSA の場合は感情極性さえ推定できればよいため問題が単純に成る. 例えば, FOOD#QUALITY の観点でレビュー文書の感情極性を推定する場合, “yummy” の 1 単語で感情極性が決定してしまうといったことも少なくない. よって, 本稿では RNN ではなく Sequential Convolution [3] によって系列をエンコードする. Sequential Convolution は画像の特徴抽出に用いられる畳み込みを単語ベクトルに対して適用するものであり, フィルタと呼ばれる時系列方向の少数のベクトルをカバーするパラメタが局所領域の特徴を抽出する機構である. これが文書分類に適していることは [3], [6] によって報告されている. 例えば, 3 つのベクトルをカバーする  $(3 \times d_x \times d_c)$  のテンソルのフィルタ  $\mathbf{W}_c$  が入力系列中の  $\mathbf{x}_i$  を  $\mathbf{h}_i$  にエンコードした場合,  $\mathbf{h}_i$  は  $\{\mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1}\}$  を要約したベクトルとなる. 本稿では感情極性分布  $\mathbf{y}_j$  の生成において, エンコーダ  $\psi_\theta$  を恒等関数, つまり  $\mathbf{h}_i = \mathbf{x}_i$  とするモデルを M0, Bi-LSTM とするモデルを M1, Sequential Convolution とするモデルを M2 と呼ぶ.

### 3.3 感情極性推定モデルの学習

M0, M1, M2 は最終出力として,  $\mathbf{c}_j$  を観点とした場合の入力系列に対する感情極性の予測分布  $\mathbf{y}_j$  を出力する. よって,  $\mathbf{y}_j$  とその教師分布  $\mathbf{y}_j^{\text{true}}$  の間のクロスエントロピーである式 (7) で定義される  $\text{cost}^{\text{pol}}$  を目的関数に勾配降下法を行えば, M0, M1, M2 は感情極性推定モデルとして最適化される.

$$\text{cost}^{\text{pol}} = \sum_{i=1}^{d_p} -\mathbf{y}_j^{\text{true}}(i) \log \mathbf{y}_j(i) \quad (7)$$

その結果, Alignment のある要素  $\text{align}(\mathbf{h}_i, \mathbf{c}_j)$  は,  $\mathbf{c}_j$  を考慮した上で文章の感情極性を推定する場合の  $\mathbf{h}_i$  の重要度となる.

### 3.4 Aspect Category の抽出モデルとその学習

$\mathbf{y}_j$  の推論モデルの一部を拡張し Aspect Category を抽出する方法について説明する. Aspect Category の抽出は SE16T5S1 では Slot1 と呼ばれており, 2. 節で説明した. Slot1 では与えられた文から 0 個以上の Aspect Category を抽出するが, これを以下の手順で行う. 1) 与えられた文における各 Aspect Category のスコアを並べたベクトル Alignment を計算する 2) 各スコアを平均と分散を用いて正規化する 3) 正規化した各スコアがある閾値  $\text{th}$  以上になっているカテゴリ群を, 抽出結果のカテゴリ群とする.

まずベクトル Alignment を計算する. Alignment は式 (6) で与えられる Alignment を単語ベクトルの並んでいる次元方向に足し合わせ, そして各要素を sigmoid 関数で正規化することで計算される.

$$s_i = \sum_{k=1}^L \text{align}(\mathbf{h}_k, \mathbf{c}_i) \quad (8)$$

$$\hat{s}_i = \text{sigmoid}(s_i)$$

$$\text{Alignment} = (\hat{s}_1, \hat{s}_2, \dots, \hat{s}_N)$$

次の 3.5 節で定義する目的関数  $\text{cost}^{\text{cat}}$  が最小化されている場合,  $\hat{s}_i \in [0, 1]$  は与えられた文に Aspect Category  $\text{cat}_i$  が含まれている確率となる. 感情極性推定の場合と同様, Aspect Category のスコア列 Alignment の生成において, エンコーダ  $\psi_\theta$  を恒等関数とするモデルを M0<sup>cat</sup>, Bi-LSTM とするモデルを M1<sup>cat</sup>, Sequential Convolution とするモデルを M2<sup>cat</sup> と呼ぶ.

### 3.5 目的関数

ここでは Aspect Category の抽出のための式 (9) で定義される目的関数  $\text{cost}^{\text{cat}}$  について説明する.  $s_i^{\text{true}}$  を, ある Aspect Category  $\text{cat}_i$  がレビューに含まれている場合に 1, そうでない場合に 0 をとる値であり, 教師データとして与えられているものとする. 各 Aspect Category のスコア  $\hat{s}_i$  は sigmoid 関数に通してあるため, その Aspect Category をとる確率と考えることができる. よって, 各カテゴリについて  $\hat{s}_i$  と  $s_i^{\text{true}}$  の間のクロスエントロピーを計算することができ, それを  $\text{cost}^{\text{cat}}$  とする.

$$\text{cost}^{\text{cat}} = -s_i^{\text{true}} \log \hat{s}_i - (1 - s_i^{\text{true}}) \log(1 - \hat{s}_i) \quad (9)$$

$\text{cost}^{\text{cat}}$  を目的関数に勾配降下法を行うことで, モデルは最適な

Alignment を生成することができるようになる。また, Aspect Category を決定するための閾値  $th$  はハイパパラメタであるため, 実験によって決める。

### 3.6 実験

#### 3.6.1 感情極性の推定

ここでは感情極性の推定を行なうモデルである  $M0$ ,  $M1$ ,  $M2$  の評価をそれぞれ行う。  $M1$  は系列のエンコードに LSTM を,  $M2$  は Convolution を用いる。さらに比較手法として CNN, Bidirectional-LSTM (Bi-LSTM), Attention-based LSTM with Aspect Embedding (ATAE-LSTM) [17] の評価も行う。ATAE-LSTM を含むこれらすべての手法は独自に実装したものを用いる。CNN は [6] の手法を簡略化したものを実装したものであり, Convolution, ReLU, Max-over-time pooling [3] を順に適用するが, フィルタ数は固定する。Max-over-time pooling は, Sequential Convolution により生成された特徴ベクトル列全体の, 系列の並びの次元で最大値を取ることである。モデルの勾配はすべて backpropagation で計算し, 最適化は Adam [7] によって行う。Adam のパラメータは [7] で推奨されているものを用いる。単語ベクトルの初期値は Google News Corpus の 300 次元の単語ベクトル<sup>(注4)</sup>を用い, 学習中にさらに最適化されるようにした。ハイパパラメタは 10 分割交差検定により決定した。  $M0$  にはハイパパラメタは存在せず,  $M1$  には LSTM の出力次元が,  $M2$  には畳み込みフィルタがカバーする単語ベクトルの数, フィルタ数などがある。最終的に,  $M1$  は各 LSTM の出力次元を 128,  $M2$  は畳み込みフィルタのカバー範囲を 3 ベクトル, フィルタ数は 128 とした。また, 比較手法として実装した, Aspect Category を考慮しないモデルである Bi-LSTM, CNN のハイパパラメタも交差検定で決定した。Bi-LSTM のそれぞれの LSTM の出力次元数は 128 とし, CNN のフィルタ数は 128, フィルタのカバー範囲を 3 ベクトルとした。また, epoch は  $M0$ ,  $M1$ ,  $M2$ , CNN, Bi-LSTM, ATAE-LSTM それぞれで 5, 5, 4, 3, 4, 5 とした。

#### 3.6.2 感情極性推定モデルの評価

次に,  $M0$ ,  $M1$ ,  $M2$  それぞれのハイパパラメタを交差検定によって最適値にした場合の, テストデータにおける感情極性推定精度を算出した (表 1)。SE16T5S1 では外部データによる事前学習を許しているが, この表 1 中には事前学習をしていないチームの成績のみを記載している。一番下の baseline は Support Vector Machine によるクラス分類である。  $M0$ ,  $M1$ ,  $M2$  の中ではソース系列のエンコーダが恒等関数である  $M0$  が最も推定精度が良く, Bi-LSTM で系列全体を考慮する  $M1$  が最も悪い。これより, ABSA の感情極性推定においては局所領域への注目が重要であることが分かる。また, Aspect Category を全く考慮しない CNN は考慮する  $M1$ ,  $M2$  よりも良い精度を達成していることから, Convolution, Max-over-time pooling の感情極性タスクへの適性の高さが読み取れる。また, SE16T5S1 Slot3 の 1 位のチーム (1st ranked) は構文解析

や様々な特徴量, 分類器を組み合わせた手法を用いている [2]。SE16T5S1 は提供される学習用データが少量であるため, このようなデータの静的解析や事前知識の利用はやはり有効であると考えられる。

表 1: SE16T5S1 Slot3 のテストデータにおける各モデルの評価結果。

モデル	テストデータにおける感情極性推定精度
$M0$	<b>0.831</b>
$M1$	0.818
$M2$	0.822
CNN	0.828
Bi-LSTM	0.811
ATAE-LSTM [17]	0.817
1st ranked	<b>0.881</b>
2nd ranked	0.818
baseline	0.599

#### 3.6.3 Aspect Category 抽出モデルの評価

SE16T5S1 Slot1 をベンチマークに,  $M0^{cat}$ ,  $M1^{cat}$ ,  $M2^{cat}$  をそれぞれ評価する。勾配の計算方法, パラメタの最適化手法は  $M0$ ,  $M1$ ,  $M2$  の場合と同様であり, 単語ベクトルの初期値も Google News Corpus のものを用いる。ハイパパラメタのバリエーションも同様であるが,  $\hat{s}_i$  (式 (8)) の正規化値に適用するしきい値  $th$  が存在する。これらのハイパパラメタは SE16T5S1 の学習用データにおける 10 分割交差検定によって決定した。 $\hat{s}_i$  の正規化値に適用するしきい値  $th$  は最終的に  $M0^{cat}$ ,  $M1^{cat}$ ,  $M2^{cat}$  それぞれについて 1, 1.2, 1.3 とした。また,  $M1^{cat}$  については順, 逆方向の LSTM とともに, 128 ベクトルを出力するようにし, それらを連結するようにした。 $M2^{cat}$  の畳み込みフィルタの数は 128, カバーする単語ベクトル数は 1 とした。学習回数である epoch は  $M0^{cat}$ ,  $M1^{cat}$ ,  $M2^{cat}$  それぞれについて 10, 11, 10 としている。

$M0^{cat}$ ,  $M1^{cat}$ ,  $M2^{cat}$  それぞれのハイパパラメタを交差検定で最適値にした場合の, Slot1 のテストデータにおける F 値の micro 平均は表 2 のようになった。Convolution を用いた  $M2^{cat}$  が最もよく, 次いで,  $M0^{cat}$ ,  $M1^{cat}$  となった。これより, Aspect Category の抽出ににおいては文脈の考慮はさほど重要ではなく, また, フィルタがカバーする単語数も 1 が最も良いことから, 単純に文中の単語それぞれにスコアを計算するのが効果的であることが読み取れる。また, SE16T5S1 の参加チームのモデルの精度も併記している。SE16T5S1 Slot1 では Slot3 同様, 外部データを用いた事前学習を行うことができる。事前学習を行ったチームも含めると  $M2$  が 7 位相当, 含めない場合は 2 位相当となった。 $M0^{cat}$ ,  $M1^{cat}$ ,  $M2^{cat}$  はいずれも生の単語ベクトル列しか特徴量として用いていないため, 事前知識や構文情報を入力ベクトルに付加することで, さらなる精度向上が見込めると考えられる。

#### 3.6.4 アライメントの可視化

最後に, 学習済みの  $M0^{cat}$  において, 式 (6) の Alignment を可視化する。可視化に用いるレビュー文は以下である。

(注4): <https://code.google.com/p/word2vec/>

表 2: SE16T5S1 Slot1 のテストデータにおける各モデルの評価結果．ここでは事前学習を行わなかったチーム，手法の成績のみを記載している．提案手法の中では 1 単語ベクトルをカバーする Convolution を用いる M2<sup>cat</sup> が F 値 0.695 で最も良いという結果となった．baseline は Support Vector Machine によるものである．

model	F value
M0 <sup>cat</sup>	0.678
M1 <sup>cat</sup>	0.670
M2 <sup>cat</sup>	<b>0.695</b>
1st ranked	<b>0.715</b>
2nd ranked	0.687
3rd ranked	0.678
baseline	0.599

“best restaurant in the world , great decor , great customer service , friendly manager.”

また，このレビューには実際に RESTAURANT#GENERAL , AMBIENCE#GENERAL , SERVICE#GENERAL の 3 つの Aspect Category が含まれており，感情極性はすべて positive である．可視化結果は 図 1 の通りとなり，ある程度正確なアライメントが計算できているのが確認できる．

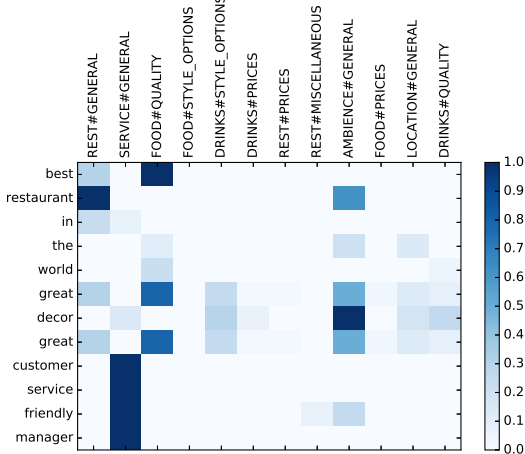


図 1: Slot1 における M0<sup>cat</sup> の Alignment の可視化例．与えられた RESTAURANT#GENERAL , AMBIENCE#GENERAL , SERVICE#GENERAL に対して，ある程度適切な単語のスコアが大きくなっている．しかし，M0<sup>cat</sup> は係り受けは全く考慮しない．例えば “best” と “restaurant” の関係性を考慮していない．

## 4. Attention-based モデルの半教師あり学習

### 4.1 モデルの概要

次に 3.1 節で定義した M2 で半教師あり学習を行うモデルについて述べる．このモデルを M3 と呼ぶことにする．その概観を図 2 に示す．このモデルは Ladder Network [13] が元になっており，その特徴を以下に挙げる．

- $X$  にガウシアンノイズを加えて  $X^{\text{cor}}$  を生成し，その事後分布  $Y^{\text{cor}}$  から  $X$  の復元である  $X^{\text{rec}}$  を生成する．なお cor は破壊されている (corrupted) ことを，rec は復元 (reconstruction) を表す．

- エンコーダとデコーダの間に関数  $g$  によるスキップコネクションを作ることによって，デコードの際にエンコーダの出力を直接利用する

- デコーダは  $X^{\text{rec}}$  だけではなくエンコードの各ステップで生成されるものも復元する

図 2 中の関数  $g$ , dec1, dec2, dec3 については 4.3 節で詳しく述べる．初めに，エンコーダの出力  $Z$ ，ガウシアンノイズに

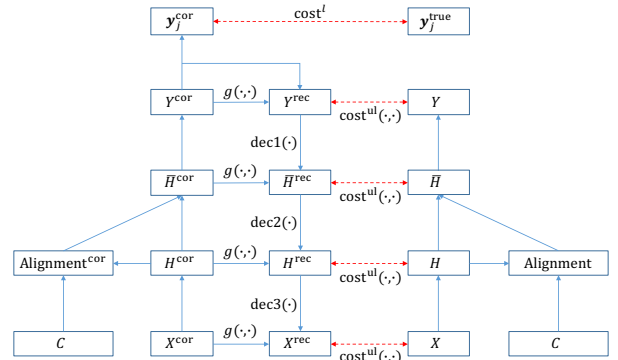


図 2: NA を用いた半教師あり学習モデル M3 の概観．右側の経路はエンコーダによって生成されるものであり，3.1 節で述べた NA を用いた感情極性の推定を表す．また，左側の経路も同様にエンコーダによるものであるが，これはガウシアンノイズによる破壊を伴っており，cor は破壊されていることを表す．これら左右の経路のパラメタはすべて共有される．rec を伴う経路はデコーダによる，ガウシアンノイズによる破壊からの復元経路を表す．パラメタは全て左側の経路と共有される．また， $y_j^{\text{true}}$  は  $y_j$  (式 (4)) の教師分布である． $\text{cost}^l$ ,  $\text{cost}^ul$  はそれぞれ教師あり誤差，教師なし誤差を表すが，詳しくは 4.5 節で述べる．

より破壊されたエンコーダの出力  $Z^{\text{cor}}$ ，デコーダの出力  $Z^{\text{rec}}$  を以下のように定義する．

$$Z = (z_1, z_2, z_3, z_4) = (X, H, \bar{H}, Y)$$

$$Z^{\text{cor}} = (z_1^{\text{cor}}, z_2^{\text{cor}}, z_3^{\text{cor}}, z_4^{\text{cor}}) = (X^{\text{cor}}, H^{\text{cor}}, \bar{H}^{\text{cor}}, Y^{\text{cor}})$$

$$Z^{\text{rec}} = (z_1^{\text{rec}}, z_2^{\text{rec}}, z_3^{\text{rec}}, z_4^{\text{rec}}) = (X^{\text{rec}}, H^{\text{rec}}, \bar{H}^{\text{rec}}, Y^{\text{rec}})$$

$X, H, \bar{H}, Y$  は式 (1), (5) で述べた変数であり，rec, cor はそれぞれ復元，ガウシアンノイズによる破壊を表す．教師なしデータには感情極性ラベルだけでなく Aspect Category ラベルも付いていないため，エンコーダはすべての Aspect Category についての事後分布である  $Y^{\text{cor}}$  を生成し，デコーダはそれを元に  $Z^{\text{rec}}$  を生成することになる．

### 4.2 エンコーダ

エンコーダは 3.1 節で述べた方法で  $Z, Z^{\text{cor}}$  を生成するが， $z_2 = H$  を生成する際 [13] のやり方にならないバッチ正規化

(Batch Normalization) [5] を行う .

$$\begin{aligned}\mathbf{H}_{\text{BN}} &= \text{BN}(\psi_{\theta}(\mathbf{X})) \\ \mathbf{H} &= \phi(\gamma \mathbf{H}_{\text{BN}} + \beta)\end{aligned}$$

ここで  $\beta, \gamma$  は学習済みのパラメタである . BN はバッチ正規化を表し , 次式で定義される .

$$\text{BN}(\mathbf{x}) = \frac{\mathbf{x} - \mu}{\sigma} \quad (10)$$

$\mu, \sigma$  はそれぞれ  $\mathbf{x}$  のミニバッチ次元での平均 , 標準偏差を表す . バッチ正規化は主に画像認識で用いられる手法であり , 自然言語処理における利用例は少ないが , M3 と後述する M4 においては有効に働くことを確認している .  $\phi$  は活性化関数 ,  $\psi_{\theta}$  は式 (2) で述べたソース系列を実際にエンコードするパラメタ化された関数を表す . M3 では  $\psi_{\theta}$  を Sequential Convolution とする . また ,  $\mathbf{X}^{\text{cor}}, \mathbf{H}^{\text{cor}}$  の生成においては , 次式のようにバッチ正規化に加えてガウシアンノイズによる破壊を行う .

$$\begin{aligned}\mathbf{X}^{\text{cor}} &= \mathbf{X} + \mathbf{n} \\ \mathbf{H}_{\text{BN}}^{\text{cor}} &= \text{BN}(\psi_{\theta}(\mathbf{X}^{\text{cor}})) + \mathbf{n} \\ \mathbf{H}^{\text{cor}} &= \phi(\gamma(\mathbf{H}_{\text{BN}}^{\text{cor}}) + \beta)\end{aligned}$$

$\mathbf{n}$  はガウシアンノイズを表し ,  $\beta, \gamma, \phi, \psi_{\theta}$  は破壊されていない  $\mathbf{H}$  を生成するときのものと共有する . それ以外は 3.1 節と同じ方法でエンコードを行う . エンコーダ全体において ,  $\mathbf{Z}$  の生成時に用いるパラメタは  $\mathbf{Z}^{\text{cor}}$  を生成するときに再利用する .

### 4.3 デコーダ

デコーダは破壊された  $\mathbf{Z}^{\text{cor}}$  から復元  $\mathbf{Z}^{\text{rec}} = \{\mathbf{z}_1^{\text{rec}}, \mathbf{z}_2^{\text{rec}}, \mathbf{z}_3^{\text{rec}}, \mathbf{z}_4^{\text{rec}}\}$  を生成する . デコーダによる  $\mathbf{z}_i^{\text{rec}}$  の生成について述べる .  $\mathbf{z}_i^{\text{rec}}$  は次式により計算される .

$$\begin{aligned}\mathbf{z}_4^{\text{rec}} &= \mathbf{Y}^{\text{rec}} = g(\mathbf{Y}^{\text{cor}}, \text{BN}(\mathbf{Y}^{\text{cor}})) \\ \mathbf{z}_3^{\text{rec}} &= \bar{\mathbf{H}}^{\text{rec}} = g(\bar{\mathbf{H}}^{\text{cor}}, \text{BN}(\text{dec1}(\mathbf{Y}^{\text{rec}}))) \\ \mathbf{z}_2^{\text{rec}} &= \mathbf{H}^{\text{rec}} = g(\mathbf{H}_{\text{BN}}^{\text{cor}}, \text{BN}(\text{dec2}(\bar{\mathbf{H}}^{\text{rec}}))) \\ \mathbf{z}_1^{\text{rec}} &= \mathbf{X}^{\text{rec}} = g(\mathbf{X}^{\text{cor}}, \text{BN}(\text{dec3}(\mathbf{H}^{\text{rec}})))\end{aligned}$$

$\mathbf{z}_4^{\text{rec}}$  はエンコーダが出力した破壊された感情極性の予測分布  $\mathbf{Y}^{\text{cor}}$  を元に , また ,  $\mathbf{z}_i^{\text{rec}} \in \{\mathbf{z}_1^{\text{rec}}, \dots, \mathbf{z}_3^{\text{rec}}\}$  はそれぞれの上位層である  $\mathbf{z}_{i+1}^{\text{rec}}$  を元に計算される . dec1, dec2, dec3 はパラメタによる入力の変換を行う関数を表すが , これについては後述する . また , BN はバッチ正規化 (式 (10)) を表す . これは  $g$  のノイズ除去の対象がバッチ正規化適用前のものであることを表すが , その理由は 4.5 節で述べる .

図 2 の関数  $g$  は [13] で提案されているノイズ除去関数である . これは破壊されたエンコード結果  $\mathbf{z}_i^{\text{cor}}$  と デコーダによる 1 つ上位層の出力である  $\mathbf{u}$  の重み付き和で与えられる (式 (11)) . ここで ,  $g$  の目的は  $\mathbf{z}_i^{\text{cor}}$  からガウシアンノイズを除去することである . 一般に , 独立ガウス分布に従う潜在変数  $\mathbf{z}_i$  にガウシアンノイズを加えてできた  $\mathbf{z}_i^{\text{cor}}$  からノイズを除去する場合 , 除去関数は  $\mathbf{z}_i$  に対して線形になることが知られている [15] , [13] .

よって ,  $\mathbf{z}_i^{\text{cor}}$  に対して線形なノイズ除去関数が適用され , 正しく除去できた場合 ,  $\mathbf{z}_i$  はガウス分布に従うことになり , 任意のデータ点を生成することができるようになる [13] .  $g$  も実際に  $\mathbf{z}_i^{\text{cor}}$  に対して線形になっており , また ,  $v(\mathbf{u}) * \mathbf{z}_i^{\text{cor}}$  はそれぞれの要素同士の積であるため  $\mathbf{z}_i^{\text{cor}}$  の各要素の独立性が保たれる (式 (11)) .

$$\begin{aligned}g(\mathbf{z}_i^{\text{cor}}, \mathbf{u}) &= v(\mathbf{u}) * \mathbf{z}_i^{\text{cor}} + (1 - v(\mathbf{u})) * \mu(\mathbf{u}) \\ &= (\mathbf{z}_i^{\text{cor}} - \mu(\mathbf{u})) * v(\mathbf{u}) + \mu(\mathbf{u})\end{aligned} \quad (11)$$

$\mu(\mathbf{u}), v(\mathbf{u})$  は学習済みのパラメタ  $\{\mathbf{a}_1 \dots \mathbf{a}_{10}\}$  を用いて計算される .

$$\begin{aligned}\mu(\mathbf{u}) &= \mathbf{a}_1 \text{sigmoid}(\mathbf{a}_2 \mathbf{u} + \mathbf{a}_3) + \mathbf{a}_4 \mathbf{u} + \mathbf{a}_5 \\ v(\mathbf{u}) &= \mathbf{a}_6 \text{sigmoid}(\mathbf{a}_7 \mathbf{u} + \mathbf{a}_8) + \mathbf{a}_9 \mathbf{u} + \mathbf{a}_{10}\end{aligned}$$

次に , デコード時のパラメタ変換である dec1, dec2, dec3 を定義する .

$$\begin{aligned}\text{dec1}(\mathbf{Y}^{\text{rec}}) &= \mathbf{Y}^{\text{rec}} \mathbf{W}_1, \mathbf{W}_1 \in \mathbb{R}^{N \times d_p} \\ \text{dec2}(\bar{\mathbf{H}}^{\text{rec}}) &= \bar{\mathbf{H}}^{\text{rec}} \otimes \mathbf{e}_L \\ \text{dec3}(\mathbf{H}^{\text{rec}}) &= \mathbf{H}^{\text{rec}} \oplus \mathbf{W}_3, \mathbf{W}_3 \in \mathbb{R}^{k \times d_x \times d_c}\end{aligned}$$

dec1 は  $\mathbf{Y}^{\text{rec}}$  に対する学習済みパラメタ  $\mathbf{W}_1$  によるアフィン変換の適用である . dec2 の  $\bar{\mathbf{H}}^{\text{rec}} \otimes \mathbf{e}_L$  は ,  $\bar{\mathbf{H}}^{\text{rec}}$  と  $\mathbf{e}_L \in \mathbb{R}^L$  の内積によって  $\bar{\mathbf{H}}^{\text{rec}}$  を  $L$  回繰り返すことを表す . これにより , dec2( $\bar{\mathbf{H}}^{\text{rec}}$ ) は  $\mathbf{H}^{\text{cor}}$  と同じ形状となる .  $\bar{\mathbf{H}}^{\text{rec}}$  は  $\mathbf{H}^{\text{rec}}$  の期待値になるべきであるため ,  $\bar{\mathbf{H}}^{\text{rec}}$  は  $\mathbf{H}^{\text{rec}}$  の復元に有用であるといえる . dec3 の  $\mathbf{H}^{\text{rec}} \oplus \mathbf{W}_3$  は  $\mathbf{H}^{\text{rec}}$  の  $\mathbf{W}_3$  による転置畳み込み (transposed convolution [20]) を表す . dec3 に対応するエンコードは入力系列  $\mathbf{X}^{\text{cor}}$  を要約する畳み込みであるため , 転置畳み込みによって ,  $\mathbf{H}^{\text{rec}}$  から要約前の  $\mathbf{X}$  を復元する .

### 4.4 デコーダの拡張

M3 は Ladder Network を自然に ABSA に適用したものであるが , M3 のデコーダを拡張した M4 を提案する . Ladder Network はパラメタの事後分布からデコーダが  $\mathbf{Z}^{\text{rec}}$  を生成する . しかし事後分布としては各 Aspect Category に対する感情極性分布に加え , Alignment (式 (6)) も存在する . よって , M4 では Alignment から以下  $\mathbf{Z}^{\text{rec}'}$  を生成する .

$$\mathbf{Z}^{\text{rec}'} = (\mathbf{z}_5^{\text{rec}'}, \mathbf{z}_6^{\text{rec}'}, \mathbf{z}_7^{\text{rec}'}) = (\mathbf{X}^{\text{rec}'}, \mathbf{H}^{\text{rec}'}, \text{Alignment}^{\text{rec}'})$$

$\mathbf{z}_i^{\text{rec}} \in \{\mathbf{z}_5^{\text{rec}'}, \dots, \mathbf{z}_7^{\text{rec}'}\}$  は  $\mathbf{z}_i^{\text{rec}} \in \{\mathbf{z}_1^{\text{rec}}, \dots, \mathbf{z}_4^{\text{rec}}\}$  と同様に次式により計算される .

$$\begin{aligned}\mathbf{z}_7^{\text{rec}'} &= \text{Alignment}^{\text{rec}'} = g(\text{Alignment}^{\text{cor}'}, \text{BN}(\text{Alignment}^{\text{cor}'})) \\ \mathbf{z}_6^{\text{rec}'} &= \mathbf{H}^{\text{rec}'} = g(\mathbf{H}_{\text{BN}}^{\text{cor}'}, \text{BN}(\text{dec4}(\text{Alignment}^{\text{rec}'}))) \\ \mathbf{z}_5^{\text{rec}'} &= \mathbf{X}^{\text{rec}'} = g(\mathbf{X}^{\text{cor}'}, \text{BN}(\text{dec5}(\mathbf{H}^{\text{rec}'})))\end{aligned}$$

ここで , dec4, dec5 は次式で表される .

$$\begin{aligned}\text{dec4}(\text{Alignment}^{\text{rec}'}) &= \text{Alignment}^{\text{rec}'} \mathbf{W}_4, \mathbf{W}_4 \in \mathbb{R}^{N \times d_h} \\ \text{dec5}(\mathbf{H}^{\text{rec}'}) &= \mathbf{H}^{\text{rec}'} \oplus \mathbf{W}_5, \mathbf{W}_5 \in \mathbb{R}^{k \times d_x \times d_c}\end{aligned}$$

dec4 は上層から来た  $\text{Alignment}^{\text{rec}}$  の、学習済みパラメタ  $W_4$  によるアフィン変換である。dec5 は dec3 と同様に、上層から来た  $H^{\text{rec}}$  に対する学習済みパラメタ  $W_5$  による転置畳み込みである。M3 の拡張である M4 の概観を図 3 に示す。

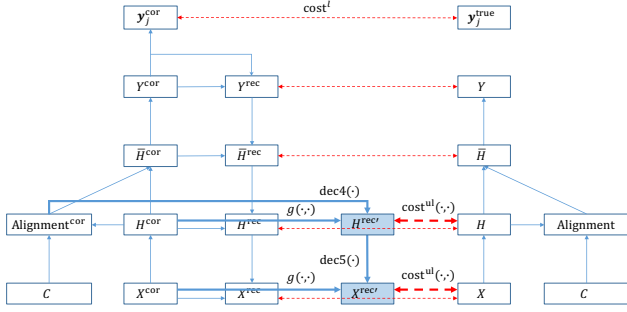


図 3: M3 の拡張である M4 の概観。拡張前の M3 (図 2) に対して dec4, dec5,  $g$ ,  $\text{cost}^{\text{ul}}$  が追加されており、追加分の経路は太字で示す。実際に経路が増えた分モデルが大きくなり、計算量が増えてしまうが、比率としては大きくないので問題ではない。詳しくは 4.5 節で述べるが、誤差関数  $\text{cost}^{\text{ul}}$  は M3 と同様に、破壊されていないエンコーダの出力との間で計算される (太い赤の矢印)。

#### 4.5 目的関数

M3, M4 の目的関数について述べる。教師あり誤差を、破壊された感情極性の予測分布  $y_j^{\text{cor}}$  と、教師分布の間の交差エントロピーとする。  $y_j^{\text{true}}$  を  $y_j^{\text{cor}}$  の教師分布とすると、それは次式で表される。

$$\text{cost}^l = \sum_{i=1}^{d_p} -y_j^{\text{true}}(i) \log y_j^{\text{cor}}(i) \quad (12)$$

また、 $Z^{\text{rec}}$  の復元誤差  $\sum_i \text{cost}^{\text{ul}}(z_i, z_i^{\text{rec}})$  を教師なし誤差とする。  $z_i, z_i^{\text{rec}}$  は実数のベクトルであるため、復元誤差はそれぞれの間の二乗誤差とする (式 (13))。しかし  $z_i$  の生成にバッチ正規化が用いられた場合、単純な二乗誤差では不十分である [13]。本稿の場合、それは  $i = 2$  の場合である。通常ミニバッチの大きさは有限であるため、バッチ正規化の適用によってある程度のノイズが発生する。そして、  $z_i$  と  $z_i^{\text{cor}}$  は同じミニバッチから生成されるため、生じるノイズはかなり似たものとなると考えられる。その場合、ノイズ除去関数  $g$  は  $z_i^{\text{cor}}$  をそのままコピーしてそれを  $z_i$  の復元としてしまうことでバッチ正規化によるノイズのほとんどを除去できてしまい、エンコーダが加えたガウシアンノイズの除去を妥協した場合でも復元誤差が最小化されてしまう可能性がある。この問題を解決するため [13] では、二乗誤差を計算する前に、復元  $z_i^{\text{rec}}$  に、  $z_i$  生成時のバッチ正規化に用いられた  $\mu$  と  $\sigma$  を前もって適用しておく。これにより、  $g$  はエンコーダで生じたガウシアンノイズの除去だけに専念できるようになるため、本稿もそれにならう。それを踏まえると、  $z_i$  の生成にバッチ正規化が用いられた場合の復元誤差は式 (14) となる。ここで、  $\mu, \sigma$  はそれぞれエンコーダによる  $z_i$  生成時のバッチ正規化に用いた平均、標準偏差である。

$$\text{cost}^{\text{ul}}(z_i, z_i^{\text{rec}}) = \begin{cases} \|z_i - z_i^{\text{rec}}\|^2 & (13) \\ \|z_i - \frac{z_i^{\text{rec}} - \mu}{\sigma}\|^2 & (14) \end{cases}$$

実際の M3, M4 の学習時は、  $\text{cost}^l + \frac{\lambda_i}{m_i} \sum_i \text{cost}^{\text{ul}}(z_i, z_i^{\text{rec}})$  を目的関数に勾配降下法を行ってモデルを最適化することで、教師あり誤差と教師なし誤差が同時に最小化される。ここで  $m_i$  はベクトル  $z_i^{\text{rec}}$  の大きさ ( $z_i$  の大きさに等しい)、  $\lambda_i$  は各復元誤差の重みを表すハイパラメタを表す。

#### 4.6 実験

M3, M4 の評価を行う。モデルの評価は SE16T5S1 の感情極性推定タスクで行うが、ハイパラメタは SemEval 2015 Task 12 レストランメインのデータセットを用いて決定する。また、教師なしデータは Yelp Academic Dataset (注5) のレビューデータから 20000 文を用いた。単語ベクトルの初期値は教師あり学習モデルの場合と同様、Google News Corpus のものを用いるが、学習中に最適化されないようにした。なぜならば M3, M4 では入力を復元するように学習するため、入力である単語ベクトルは固定されている方が良いためである。実際に、単語ベクトルが学習によって最適化されないほうが精度が高くなることを確認している。また、パラメタの更新は教師あり学習の場合と同様に Adam によって行うが [13] にならって手動で学習率  $\alpha$  の減衰を行う。具体的には 5 epoch 以降は学習するサンプル数に対して  $\alpha$  を線形に減衰するようにする。  $\alpha$  の初期値は [7] で推奨されている 0.001 とした。M3, M4 とともに畳み込みフィルタがカバーする単語数は 2 とした。また、M3, M4 のハイパラメタである  $\lambda_i$  の最適値は、M3 については  $(\lambda_1, \dots, \lambda_4)$  が順に  $(0, 0.01, 0.01, 0)$ , M4 については  $(\lambda_1, \dots, \lambda_7)$  が順に  $(0, 0.01, 0.1, 0, 0, 0.01, 0.05)$  となった。M3, M4 の epoch はそれぞれ 15, 21 とした。次に、M3, M4 の Slot3 のテストデータにおける感情極性の推定精度を表 3 に示す。M4 が拡張前の M3 よりも良い精度を達成していることから、Alignment を事後分布と考え、そこから入力を復元するのが有効であることが分かる。なお、このタスクでは外部データ不使用のチームが推定精度 0.881 最高精度を達成しているが、全体的には外部データを用いたチームが上位を占めている。また、M3, M4 とともにラベルなしデータを約 20,000 件以上に増やすと推定精度が低下することを確認している。これはラベル付きデータのラベルの分布に偏りがあることが、ラベルなしデータの増加により顕現したか、あるいはラベルなしデータ自体のノイズの影響が大きくなりすぎることなどが原因であると考えられる。

#### 5. 関連研究

Convolutional Neural Network (CNN) は文書分類において最も成功した手法である。Collobert ら [3] は単語ベクトル列である文章に対する Sequential Convolution が様々なベンチマークで良い精度を達成したことを報告した。Kim [6] は [3] の手法を拡張し、文書分類の様々なベンチマークで既存手法を上回

(注5): [https://www.yelp.com/dataset\\_challenge/](https://www.yelp.com/dataset_challenge/)

表 3: SE16T5S1 Slot3 のテストデータにおける各モデルの評価結果。1-3rd ranked は外部データの教師ラベルを利用しているが, M3, M4 では利用していない。M3, M4 の順位はそれぞれ 3 位, 2 位相当である。また, 参考のために表 1 にも記載した, 事前学習をしていない, 総合 1 位のチームの成績も一番下に記載している。

model	accuracy
M3	<b>0.846</b>
M4	<b>0.854</b>
1st ranked	<b>0.867</b>
2nd ranked	0.854
3rd ranked	0.839
baseline	0.599
1st ranked (事前学習なし)	0.881

る精度を達成した。具体的には異なる属性を持つ単語ベクトルをチャンネル方向に連結したものを CNN への入力とし, また, 異なるフィルタサイズの畳み込み層を複数重ねることで様々な大きさの局所領域を要約できるというものである。

観点付き感情極性推定については, Wang ら [17], Yanase ら [19] が NA を ABSA の感情極性抽出タスクに用いた。これらは本稿同様, NMT での利用が元になっているが, score 関数 (式 (3)) を含めた NA の利用方法, Aspect Category のベクトル表現の方法が異なる。特に [19] は Aspect Category が Entity#Attribute で表されることから, Entity, Attribute をそれぞれ別のベクトルとして表現し, 選ばれたベクトルの期待値  $\bar{y}$  (式 (5)) もそれぞれ別に計算し, 連結したものを感情極性の予測に用いている。[17], [19] はともに単語ベクトルを LSTM でエンコードしているが, 本稿では ABSA では局所領域への注目がより重要であると考え, Sequential Convolution を用いている。また [19] は Yelp Academic Dataset を用いた事前学習を行っている。本稿でも同データセットを用いるが, 半教師あり学習を行うため, 教師データは用いなかった。

## 6. おわりに

本研究では, ABSA における Neural Attention の利用について提案した。Aspect Category の抽出モデルは単語それぞれの, 各 Aspect Category への関連の強さも同時に算出できるため, トピックモデリングへの応用も考えられる。また, 感情極性推定タスクにおいて半教師あり学習を行うモデル M3, M4 を提案した。これらは半教師ありの画像認識で高い精度を達成している Ladder Network をベースとしており, SemEval 2016 の ABSA タスクのベンチマークで実験を行った結果, 上位チームに迫る感情極性推定精度を達成した。

## 文 献

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[2] Caroline Brun, Julien Perez, and Claude Roux. Xrce at semeval-2016 task 5: Feedbacked ensemble modeling on syntactico-semantic knowledge for aspect based sentiment

analysis. In *SemEval@NAACL-HLT*, 2016.

[3] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, Vol. 999888, pp. 2493–2537, November 2011.

[4] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[5] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis R. Bach and David M. Blei, editors, *ICML*, Vol. 37 of *JMLR Workshop and Conference Proceedings*, pp. 448–456. JMLR.org, 2015.

[6] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, Vol. abs/1412.6980, , 2014.

[8] Diederik P. Kingma, et al. Semi-supervised learning with deep generative models. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *NIPS*, pp. 3581–3589, 2014.

[9] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[10] Bing Liu. Sentiment analysis and opinion mining, 2012.

[11] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, 2015*, pp. 1412–1421, 2015.

[12] Takeru Miyato, Shin ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *CoRR*, Vol. abs/1704.03976, , 2017.

[13] Antti Rasmus, Harri Valpola, Mikko Honkala, Mathias Berglund, and Tapani Raiko. Semi-supervised learning with ladder network. *CoRR*, Vol. abs/1507.02672, , 2015.

[14] Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*, 2015.

[15] Harri Valpola. From neural pca to deep unsupervised learning. *CoRR*, Vol. abs/1411.7783, , 2014.

[16] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning, ICML '08*, pp. 1096–1103, New York, NY, USA, 2008. ACM.

[17] Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 606–615. Association for Computational Linguistics, 2016.

[18] Kelvin Xu, et al. Show, attend and tell: Neural image caption generation with visual attention. *CoRR*, Vol. abs/1502.03044, , 2015.

[19] Toshihiko Yanase, et al. bunji at semeval-2016 task 5: Neural and syntactic models of entity-attribute relationship for aspect-based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pp. 289–295, San Diego, California, June 2016. Association for Computational Linguistics.

[20] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus. Deconvolutional networks. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2528–2535, June 2010.