Vol.2021-NL-251 No.29
Vol.2021-SLP-139 No.29
2021/12/3

テキストのみを用いたドメイン適応のための Intermediate-CTC コンフォーマーモデルに関する検討

佐藤 裕明[1,a)]　小森 智康[1]　三島 剛[†1]　河合 吉彦[1]　望月 貴裕[1]　佐藤 庄衛[1]　小川 哲司[2]

概要：本研究では，Connectionist Temporal Classification（CTC）音声認識モデルにおいて，テキストデータのみで音声認識モデルを学習し，推論対象の話題のドメインに適応する手法を提案する．音声認識モデルは，学習した発音と記号列の対応に従って認識結果を出力するため，学習データの話題と異なる話題の音声を認識させると，話題に適さない記号列を出力し，認識誤りを引き起こす．推論する話題の音声-テキスト対のデータが入手できれば，音声認識モデルを事前に学習し，出力記号列を話題のドメインに適応させることが可能だが，推論する話題の音声が入手できるとは限らず，入手が可能でも音声に対するアノテーション作業はコストがかかる．そこで，テキストデータを疑似的に CTC 記号列に変換し，音声認識モデルが出力する潜在ベクトルにマッピングすることで，音声認識モデルをテキストデータで学習できる技術を開発した．日本語話し言葉コーパス（CSJ）において学習させた CTC 音声認識モデルを天気予報のテキストデータを用いて学習し，天気予報の話題のドメインに適応することで，天気予報の評価音声に対して文字誤り率（CER）が 18.7 %から 13.7 %に改善した．

Text-only domain adaptation of intermediate CTC-based conformer networks

Hiroaki Sato[1,a)]　Tomoyasu Komori[1]　Takeshi Mishima[†1]　Yoshihiko Kawai[1]
Takahiro Mochizuki[1]　Shoei Sato[1]　Tetsuji Ogawa[2]

Abstract: We present a text only domain adaptation method for connectionist temporal classification (CTC) speech recognition model. Automatic speech recognition (ASR) model outputs recognition results according to the correspondence between the learned pronunciation and the symbol sequence. When ASR model recognizes speech on a topic different from the topic of the training data, it outputs a symbol sequence unsuitable for the topic, causing recognition errors. If the speech and text pair data of the topic to be inferred are available, it is possible to train the ASR model in advance and adapt the output symbol sequence to the domain of the topic, but the speech of the topic to be inferred is not always available, and even if it is available, annotating the speech is costly. Therefore, we have developed a technique that can train a speech recognition model on text data by pseudo-transforming the text data into CTC symbol sequences and mapping them to latent vectors output by ASR model. After training the CTC speech recognition model with Corpus of Spontaneous Japanese (CSJ), we trained it with text data from a weather forecast and adapted it to the topic domain of the weather forecast, which improved the character error rate (CER) from 18.7% to 13.7% for the evaluation speech of the weather forecast.

1. はじめに

音声認識において，発音記号を介さずに直接音声をテキストに変換する End-to-end 音声認識が盛んに研究されており，様々なモデル構造が提案されている [1], [2], [3]. End-to-end 音声認識モデルは，自己回帰構造を有する Au-

[1] 日本放送協会
NHK Science & Technology Research Laboratories
[2] 早稲田大学
Waseda University
[†1] 現在，一般財団法人 NHK エンジニアリングシステム
Presently with NHK Engineering System
[a)] satou.h-fk@nhk.or.jp

toregressive(AR) 型のモデル [1], [2] と，自己回帰構造が存在しない Non-autoregressive(NAR) 型のモデル [3] に大別され，NAR 型である CTC 音声認識モデルは，AR 型モデルと比較して推論速度に優れており [4]，早期に認識結果を出力したいアプリケーションとの相性が良い．

NAR 型，AR 型モデルのいずれも，学習データと異なる話題のドメインの音声を認識させたとき，話題に沿わない単語や文字を出力して認識誤りを引き起こすという課題がある [5], [6]．推論対象の話題の音声-テキスト対のデータを事前に入手することができれば，それらを学習させることで，推論対象の話題のドメインに適応させることが可能であるが，音声に対するアノテーション作業は時間とコストがかかる．一方，テキストデータは，音声-テキスト対のデータよりも入手が比較的容易である．上記の背景から，テキストデータのみを用いて，推論対象の話題のドメインに音声認識モデルを適応する技術が求められる．

テキストデータのみを用いたドメイン適応の手法として，音声認識モデルに内在する言語モデルの出力確率分布を推定し，推論対象のドメインのテキストで学習させた外部言語モデルの出力確率分布に置き換える手法が提案されている [7]．この手法は，モデルの自己回帰構造を内部的な言語モデルと仮定している．一方，Deep Fusion[8]，Cold Fusion[9] は，予め学習した外部の言語モデルを，音声認識モデルと融合して学習する手法であり，音声認識モデルと外部言語モデルの潜在変数をゲーティング機構を用いて結合して学習する．上記のいずれの手法も，自己回帰構造をもたず，文字列長と音響特徴量の系列長の不一致から潜在変数の結合に工夫を要する CTC 音声認識モデルに対しては適用が困難である．

そこで本稿では，CTC 音声認識モデルにおいて，テキスト文字列を擬似的に CTC 記号列に変換して音声認識モデルを学習し，推論対象の話題にドメイン適応する手法を提案する．具体的には，Intermediate-CTC[10] モデルの仕組みを利用して中間層から CTC 記号列を出力し，CTC 記号列を音声認識モデルの中間層が出力する潜在ベクトルに変換するようなニューラルネットワーク（Adapter）を学習する．その後，推論対象の話題のテキスト文字列を疑似的に CTC 記号列に変換し，学習済みの Adapter により潜在ベクトルに変換した後，CTC 音声認識モデルの後段を再学習する．なお，モデルの各層は音声認識のベンチマークである LibriSpeech において最高性能を達成した Conformer[3] を用いる．

日本語話し言葉コーパス（CSJ）にて Intermediate-CTC を学習し，NHK が過去に放送した天気予報のテキストデータでドメイン適応した結果，天気予報の評価セットに対する文字誤り率（CER）がドメイン適応の前後で 18.7 % から 13.7 %に改善した．また，ドメイン適応に用いた天気予報のテキストデータで学習した言語モデルを Shallow Fusion[11] することにより，相補的に CER が改善することが確認できた．

後述する第 2 章では，本手法の基本技術となる Intermediate-CTC モデルについて述べ，第 3 章では，提案手法であるテキストのみを用いたドメイン適応の手法について述べる．第 4 章では，行った実験と結果について述べ，第 5 章でまとめを述べる．

2. 基本技術

End-to-end 音声認識は，T の長さの音響特徴量系列 $\boldsymbol{x} = \{\boldsymbol{x}[t] \in \mathbb{R}^D | t = 1, ..., T\}$ を L の長さの出力記号 $\boldsymbol{y} = \{y[l] \in \mathcal{V} | l = 1, ..., L\}$ に変換するモデルである．ただし，$\mathcal{V}$ を出力記号の集合，D を $\boldsymbol{x}[t]$ の次元とする．本手法では CTC 音声認識において，中間層にも CTC 損失関数を設ける Intermediate-CTC と同様の枠組みにより，ソースモデルの学習を行う．

2.1 CTC 音声認識

CTC 音声認識モデルは, 特徴量 1 フレームごとに 1 記号を出力する．音響特徴量の系列長と出力記号の系列長の差を，ブランク記号<blank>と，出力文字を連続させることで吸収して学習する．推論時は, ブランク記号を消去し, 連続した文字を 1 文字に短縮することで認識結果を得る．CTC 音声認識モデルにおけるニューラルネットワークの層から出力された潜在変数 $\boldsymbol{h} = \{\boldsymbol{h}[t] | t = 1, \ldots, T_{\mathrm{h}}\}$ において，出力記号列とのアライメント $\boldsymbol{a} = \{a[t] \in \mathcal{V} \cup \{\text{<blank>}\} | t = 1, \ldots, T_{\mathrm{h}}\}$ の確率 $P(\boldsymbol{a}|\boldsymbol{h})$ は次式で計算される．

$$P(\boldsymbol{a}|\boldsymbol{h}) = \prod_t P(a[t]|\boldsymbol{h}[t]) \tag{1}$$

出力記号 $\boldsymbol{y}$ とのあり得るすべてのアライメントにおける確率の総和は次式で与えられる．

$$P(\boldsymbol{y}|\boldsymbol{h}) = \sum_{\boldsymbol{a} \in \beta^{-1}(\boldsymbol{y})} P(\boldsymbol{a}|\boldsymbol{h}) \tag{2}$$

ただし，$\beta^{-1}(\boldsymbol{y})$ は，$\boldsymbol{y}$ と潜在変数 $\boldsymbol{h}$ とのとりうるすべてのアライメントの集合である．CTC 損失関数は，対数をとることで次式で計算される．

$$L = -\log P(\boldsymbol{y}|\boldsymbol{h}) \tag{3}$$

L を最小化するようにモデルのパラメータを更新することで CTC 音声認識モデルは学習される．

2.2 Intermediate-CTC

Intermediate-CTC では，モデルの中間層でも CTC 損失関数を計算し，重み付けした和を最終的な損失関数として学習する．本手法においては，ソースモデルを $\mathrm{Encoder_l}$, $\mathrm{Encoder_m}$，Decoder のサブモデルに分割し，$\mathrm{Encoder_l}$,

Vol.2021-NL-251 No.29
Vol.2021-SLP-139 No.29
2021/12/3

Encoder_m，Decoder の出力における，CTC 損失関数の平均をとるものとする．なお，音響特徴量 $\boldsymbol{x}$ に対して，畳み込みニューラルネットワーク（CNN）によってサブサンプリングを行うものとし，サブサンプリング後の出力を $\boldsymbol{h_0} \in \mathbb{R}^{T_\text{h} \times D_\text{h}}$ とする．ただし，$\boldsymbol{h_0}$ の系列長を T_h，次元を D_h とする．Encoder_l の出力 $\boldsymbol{h_\text{l}} \in \mathbb{R}^{T_\text{h} \times D_\text{h}}$，$\text{Encoder}_\text{m}$ の出力 $\boldsymbol{h_\text{m}} \in \mathbb{R}^{T_\text{h} \times D_\text{h}}$，Decoder の出力 $\boldsymbol{h_\text{d}} \in \mathbb{R}^{T_\text{h} \times D_\text{h}}$ は次式で与えられる．

$$\boldsymbol{h_\text{l}} = \text{Encoder}_\text{l}(\boldsymbol{h_0}) \tag{4}$$

$$\boldsymbol{h_\text{m}} = \text{Encoder}_\text{m}(\boldsymbol{h_\text{l}}) \tag{5}$$

$$\boldsymbol{h_\text{d}} = \text{Decoder}(\boldsymbol{h_\text{m}}) \tag{6}$$

Encoder_l，Encoder_m，Decoder の出力に対する損失関数 L_l，L_m，L_d は次式で与えられる．

$$L_\text{l} = -\log P_\text{l}(\boldsymbol{y}|\boldsymbol{h_\text{l}}) \tag{7}$$

$$L_\text{m} = -\log P_\text{m}(\boldsymbol{y}|\boldsymbol{h_\text{m}}) \tag{8}$$

$$L_\text{d} = -\log P_\text{d}(\boldsymbol{y}|\boldsymbol{h_\text{d}}) \tag{9}$$

ただし，本手法において，P_l，P_m，P_d の確率算出時に用いる，潜在ベクトルの次元を語彙数の次元に変換する線形変換層はそれぞれ異なるパラメータであるとする．ソースモデルの目的関数は，L_l，L_m，L_d の平均として次式で計算する．

$$Loss_1 = \frac{1}{3}(L_\text{l} + L_\text{m} + L_\text{d}) \tag{10}$$

$Loss_1$ を最小化するように，各種モデルパラメータを更新し，ソースモデルを学習する．推論時は，最終層の出力 Decoder に対してのみ最尤アライメントを求め，認識結果を得る．

3. テキストのみを用いたドメイン適応

本手法では，テキストを疑似 CTC 記号列に変換した後，CTC 記号列をソースモデルの潜在変数に変換する Adapter を介して出力段の Decoder を学習する．**図 1** に概要を示す．

3.1 Adapter の学習

Adapter の学習の概要を**図 2** に示す．学習済みのサブサンプリング層に音響特徴量を入力して $\boldsymbol{h_0}$ を算出し，次式により $\boldsymbol{h_\text{m}}$ を算出する．

$$\boldsymbol{h_\text{l}} = \text{Encoder}_\text{l}(\boldsymbol{h_0}) \tag{11}$$

$$\boldsymbol{h_\text{m}} = \text{Encoder}_\text{m}(\boldsymbol{h_\text{l}}) \tag{12}$$

また，$\boldsymbol{h_\text{l}}$ に対応する最尤アライメント $\widehat{\boldsymbol{a}}_\text{l}$ を次式で算出する．

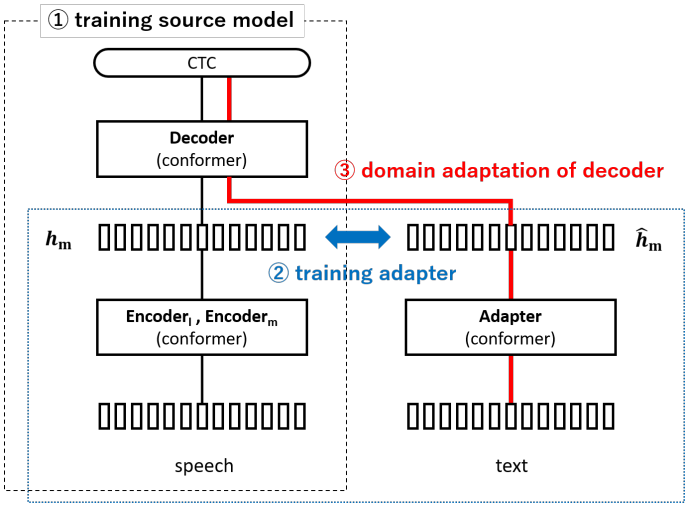

図 1 提案するドメイン適応の概要．①音声とテキストのペアデータを用いたソースモデルの学習，②テキスト入力に基づく潜在ベクトル（$\widehat{\boldsymbol{h}}_\text{m}$）を音声入力に基づく潜在ベクトル（$\boldsymbol{h_\text{m}}$）と一致させるようにアダプターを学習，③テキストデータのみを用いたデコーダの適応の 3 つの処理から成る．

$$\widehat{\boldsymbol{a}}_\text{l} = \underset{\boldsymbol{a}_\text{l}}{\text{argmax}} P_\text{l}(\boldsymbol{a}_\text{l}|\boldsymbol{h_\text{l}}) \tag{13}$$

CTC アライメント系列を Encoder_m の出力 $\boldsymbol{h_\text{m}}$ に近づけるように Adapter を学習する．Adapter の出力 $\widehat{\boldsymbol{h}}_\text{m}$ は次式で与えられる．

$$\widehat{\boldsymbol{h}}_\text{m} = \text{Adapter}(\widehat{\boldsymbol{a}}_\text{l}) \tag{14}$$

$\widehat{\boldsymbol{h}}_m$ が，$\boldsymbol{h}_m$ の潜在空間に近づくように次式の損失関数を計算する．

$$L_\text{m} = -\log P_\text{m}(\boldsymbol{y}|\widehat{\boldsymbol{h}}_\text{m}) \tag{15}$$

$$L_\text{mse} = \frac{1}{D_\text{h} T_\text{h}}(\boldsymbol{h_\text{m}} - \widehat{\boldsymbol{h}}_\text{m})^2 \tag{16}$$

Adapter の学習における目的関数は，次式で計算する．

$$Loss_2 = L_\text{m} + \alpha L_\text{mse} \tag{17}$$

ただし，α はハイパーパラメータとする．$Loss_2$ を最小化するように，Adapter のパラメータを更新する．なお，Adapter のパラメータは Adapter の最終層以外の層について学習するものとする．学習が完了した Adapter は，CTC 記号列を $\boldsymbol{h_\text{m}}$ の潜在変数に写像するモデルになると期待される．

3.2 Decoder の再学習

Decoder の再学習の概要を**図 3** に示す．Decoder の再学習では，推論対象の話題が含まれるテキスト文字列 $\mathbf{c} = \{c[j] \in \mathcal{V} | j = 1, \ldots, J\}$ のみを学習データとして用いる．テキスト文字列 1 文から後述する疑似 CTC 記号列変換手法により，N 個の疑似的な CTC 記号列 $\widehat{\boldsymbol{a}}_{\boldsymbol{i}} (i = 1, \ldots, N)$

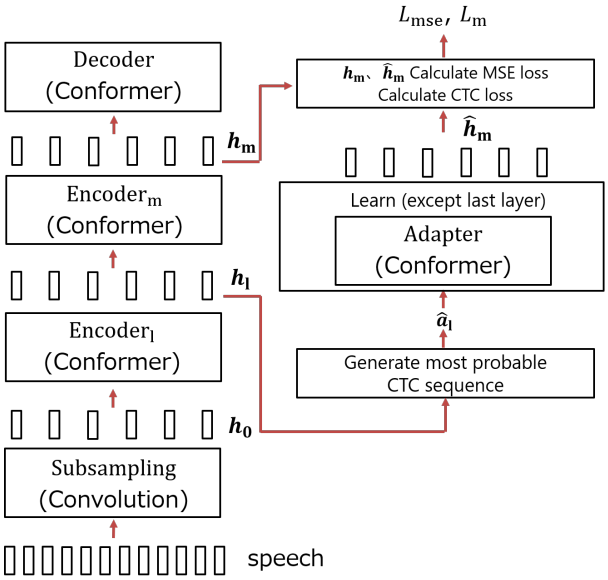

図 2 Adapter の学習の概要

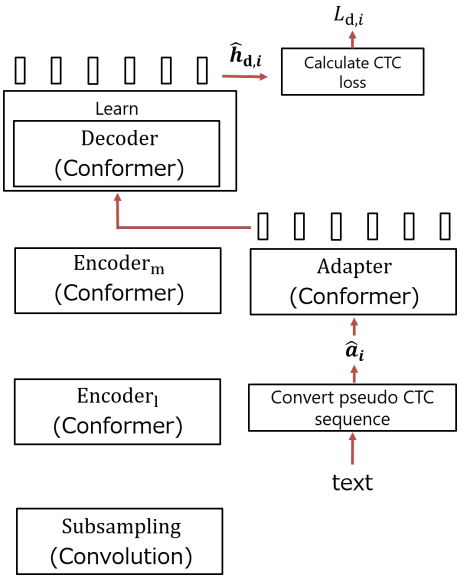

図 3 Decoder の再学習の概要

を生成し，Adapter で潜在変数に変換した後，Decoder を再学習する．疑似 CTC 記号列変換手法と Decoder の再学習について述べる．

3.2.1 疑似 CTC 記号列変換手法

音声対が存在しないテキスト文字列を発音したと仮定して，疑似的に CTC 記号列を生成することを考える．例えば，「いい天気」を CTC 記号列の長さ 10 で発音したと仮定した場合，膨大な数のパターンの CTC 記号列が存在し，すべてのパターンの CTC 記号列に変換すると，計算に多大な時間を要する．しかし，「い_ _ _ _ _ い天気」のように，「い」と「い」に間がある発音や，「い_い天気気気気気」のように特定の記号の発音区間が長くなる発音は現実的には考えにくい．ただし，簡略化のため<blank>を「_」と表記した．また，音声区間検出と音声認識を併用した場合，発音の長さにも制約があると考えられる．したがって，現実的な CTC 記号列に変換するために，学習済みのソースモデル Encoder$_{\mathrm{l}}$ から出力される $\mathbf{h}_{\mathrm{l}}$ のアライメント，すなわち CTC 記号列 $\widehat{\mathbf{a}}_{\mathbf{l}}$ の<blank>の連続数と文字の連続数の傾向と類似した CTC 記号列にテキストを変換する．ソースモデルを学習した 1epoch 分の音声データに対して $\widehat{\mathbf{a}}_{\mathbf{l}}$ を出力し，<blank>の連続数と文字の連続数の統計から生成する確率密度関数に則り，疑似的に CTC 記号列に変換する．1epoch 分の $\widehat{\mathbf{a}}_{\mathbf{l}}$ のうち，n 回<blank>が連続して出現した回数を $N_{\mathrm{b}}(n)$，n 回文字が連続して出現した回数を $N_{\mathrm{c}}(n)$ とする．<blank>の連続数の確率密度関数 $p_{\mathrm{b}}(n)$，文字の連続数の確率密度関数 $p_{\mathrm{c}}(n)$ を次式で定義する．

$$p_{\mathrm{b}}(n) = \frac{N_{\mathrm{b}}(n)}{\sum_{\mathrm{n}} N_{\mathrm{b}}(n)} \tag{18}$$

$$p_{\mathrm{c}}(n) = \frac{N_{\mathrm{c}}(n)}{\sum_{\mathrm{n}} N_{\mathrm{c}}(n)} \tag{19}$$

Algorithm 1 疑似 CTC 記号列変換手法

1: $\widehat{\boldsymbol{a}}_{\boldsymbol{i}} = \phi$，$j = 1$ に初期化
2: $p_{\mathrm{b}}(n)$ に従い，確率的に<blank>の連続数 N_{b} を決定
3: $j \neq 1 \wedge c[j-1] == c[j] \wedge N_{\mathrm{b}} == 0$ を満たす場合，2 に戻る
4: N_{b} 個の<blank>を $\widehat{\boldsymbol{a}}_{\boldsymbol{i}}$ に結合
5: $p_{\mathrm{c}}(n)$ に従い，確率的に文字の連続数 N_{c} を決定
6: N_{c} 個の c[j] を $\widehat{\boldsymbol{a}}_{\boldsymbol{i}}$ に結合
7: $j = j + 1$
8: $j \neq J$ を満たす場合，2 に戻る
9: $p_{\mathrm{b}}(n)$ に従い，確率的に<blank>の連続数 N_{b} を決定
10: N_{b} 個の<blank>を $\widehat{\boldsymbol{a}}_{\boldsymbol{i}}$ に結合

Algorithm 1 に，疑似 CTC 記号列変換手法を示す．例えば，テキスト文字列 $\mathbf{c}$ が，「明日はいい天気」であったとする．始めに，<blank>の連続数の確率密度関数 $p_{\mathrm{b}}(n)$ の確率に従い，N_{b} を決定する．$N_{\mathrm{b}} = 3$ であった場合，疑似 CTC 記号列 $\widehat{\boldsymbol{a}}_{\boldsymbol{i}}$ は，「 _ _ _ 」となる．次に，文字の連続数の確率密度関数 $p_{\mathrm{c}}(n)$ の確率に従い，N_{c} を決定する．$N_{\mathrm{c}} = 2$ であった場合，疑似 CTC 記号列 $\widehat{\boldsymbol{a}}_{\boldsymbol{i}}$ は，「 _ _ _ 明 明」となる．この処理を繰り返すことで，テキスト文字列を疑似 CTC 記号列に変換する．ただし，「いい」のように前後の文字が連続するような文の場合，CTC では必ず<blank>を挿入する必要がある．したがって，前後の文字が連続する間の<blank>連続数 N_{b} が 0 に決定した場合，再び N_{b} を決定する操作を繰り返す．

3.2.2 Decoder の再学習

テキスト文字列 $\mathbf{c}$ に対し，変換した N 個の疑似 CTC 記号列 $\widehat{\boldsymbol{a}}_{\boldsymbol{i}}$ に対して次式で損失関数を計算する．

$$\widehat{\boldsymbol{h}}_{\mathbf{d},\boldsymbol{i}} = \mathrm{Decoder}\left(\mathrm{Adapter}\left(\widehat{\boldsymbol{a}}_{\boldsymbol{i}}\right)\right) \tag{20}$$

$$L_{\mathrm{d},i} = -\log P_{\mathrm{d}}(\boldsymbol{y}|\widehat{\boldsymbol{h}}_{\mathbf{d},\boldsymbol{i}}) \tag{21}$$

Decoder の再学習の目的関数は，N 個の損失関数の平均として次式で計算する．

$$Loss_3 = \frac{1}{N}\sum_{i} L_{\mathrm{d},i} \tag{22}$$

$Loss_3$ を最小化するように，各種パラメータを更新して学習する．ただし，Adapter のパラメータは学習せず，固定とする．学習が完了した Decoder は，テキスト文字列 $\boldsymbol{c}$ の話題のドメインに適応したモデルになる事が期待される．

4. 実験・評価

4.1 実験諸元

CSJ(約 520 時間)を用いてソースモデルの学習と Adapter の学習を行い，NHK が過去放送した天気予報 1000 文でドメイン適応を行った．評価データは，CSJ の評価セット eval 1-3 と NHK が過去放送した天気予報の音声 2 時間分を用いた．ただし，学習データに用いた天気予報 1000 文に対応する音声は評価データに含めていない．音響特徴量は Kaldi ツールキット [12] を用いて抽出し，80 次元のログメルスペクトログラムに 3 次元のピッチ，Δ，ΔΔ 特徴量を加え，249 次元とした．過学習を緩和するため，入力音響特徴量に対し，SpecAugment[13] の手法を適用した．また，音響特徴量系列は 2 層の CNN により，サブサンプリングした．出力記号は CSJ で用いられている 3260 文字を使用した．$\mathrm{Encoder_l}$，$\mathrm{Encoder_m}$，Decoder，Adapter はそれぞれ，6 層，3 層，3 層，6 層の Conformer で構成し，ヘッドの数を 8，アテンションの次元を 512，線形変換層の次元を 2048 に設定した．学習率については，ESPnet ツールキット [14] が提供するスケジューラに則り学習を行った．ソースモデルは 100epoch 学習し，最も validation loss が低い epoch におけるパラメータを使用した．学習済みのソースモデルを用い，CSJ 学習データ 1epoch 分から，<blank>の連続数と文字の連続数の統計を取り，確率密度関数を生成した．Adapter は 50epoch 学習し，最も validation loss が低い epoch におけるパラメータを使用した．Decoder の再学習は，20epoch のうち，最も天気予報評価セットの CER の改善がみられた epoch のパラメータを使用した．

4.2 実験結果

表 1 は，CSJ 評価セット（eval1, eval2, eval3）と天気予報評価セット（weather）における，ソースモデル（InterCTC）と，提案手法であるテキストのみを用いたドメイン適応を行った後のターゲットモデル（InterCTC+TODA）の CER の比較である．なお，Adapter 学習時における L_{mse} の係数 α は 1.0 とし，greedy search により認識結果を出力した．天気予報の評価セットにおいて，ドメイン適応を行う前後で CER が 18.7 %から 13.7 %に改善した．**表 2** は，認識結果の改善例である．ソースモデルは，「最低気温」という単語に対し，発音が類似した「最適音」という単語に誤って認識結果を出力した．これは，音声学に関する学会発表の音声が収録されている CSJ コーパスでソースモデルを学習しているため，「最低気」や「温」の発音に対し，「最適」や「音」などといった単語を出力するよう学習されていたためと考えられる．一方，天気予報のテキストで学習した後のモデルは，「最低気温」という単語に対して，正しい認識結果を出力した．Decoder の再学習に用いた学習テキストには「最低気温」という単語が含まれており，音声認識モデルが天気予報のドメインに適応したと考えられる．なお，ドメイン適応後は，CSJ 評価セットに対する認識精度が劣化した．これは，音声認識モデルの出力分布が，天気予報のテキストに出現する単語の分布にシフトしたためであると考えられる．

4.3 言語モデルとビームサーチの併用の効果

表 3 は，言語モデルを Shallow Fusion した場合と，ビーム幅を 20 にしてビームサーチを行った場合のドメイン適応前後における天気予報評価セットの CER である．言語モデルは隠れ層の次元数が 2048 である 2 層の LSTM を用い，Decoder の再学習に用いた天気予報テキスト 1000 文を用いて 50epoch 学習し，最も validation loss が低い epoch におけるパラメータを使用した．なお Shallow Fusion における言語モデルの重みは，0.1 から 0.6 まで 0.1 刻みで評価を行い，最も認識率が改善した重みにおける CER を表 3 に示した．ソースモデルに Shallow Fusion を適用した場合（InterCTC+SF），CER は 16.1 %となり，ドメイン適応後（InterCTC+TODA）の CER13.7 %の方が認識精度で上回った．また，ドメイン適応と Shallow Fusion を併用（InterCTC+TODA+SF）することで CER が 13.7 %から 12.8 %に改善した．このことから，本手法は言語モデルと併用することで認識精度をより改善できることが確認できた．また，ドメイン適応と Shallow Fusion を併用した場合，ソースモデルに Shallow Fusion とビーム幅 20 のビームサーチを併用（InterCTC+SF+beam20）した場合の CER13.0 %を上回り，計算に時間を要するビームサーチをしなくても十分な認識精度の改善が可能であることが示唆された．ドメイン適応後にビーム幅 20 のビームサーチを併用（InterCTC+TODA+beam20）したところ，CER が 13.7 %から 13.7 %と改善はみられなかった．このことから，本手法のドメイン適応は，学習テキストに含まれる単語を認識結果候補の最上位に出力させる手法であることがわかる．ドメイン適応，Shallow Fusion，ビーム幅 20 のビームサーチを併用（InterCTC+TODA+SF+beam20）した場合，最も認識精度が改善し，CER11.3 %となり，ソースモデルから 7.4 %の CER 改善がみられた．

表 1 ドメイン適応（TODA）前後の CER

Model	eval1	eval2	eval3	weather
InterCTC	7.3	5.2	6.7	18.7
InterCTC+TODA	17.8	15.3	14.9	13.7

表 2 ドメイン適応（TODA）前後の認識結果の改善例
下線は認識誤り 太字は認識改善を表す

Model	Sentence
Reference	予想最低気温です
InterCTC	予想最適音です
InterCTC+TODA	予想最**低気温**です
Reference	あす午前九時の予想天気図です
InterCTC	え明日午前九***の予想研究図です
InterCTC+TODA	え**あす**午前九**時**の予想**天気**図です

表 3 Shallow Fusion（SF）とビームサーチの効果

Model	weather
InterCTC	18.7
InterCTC+SF	16.1
InterCTC+beam20	18.4
InterCTC+SF+beam20	13.0
InterCTC+TODA	13.7
InterCTC+TODA+SF	12.8
InterCTC+TODA+beam20	13.7
InterCTC+TODA+SF+beam20	11.3

表 4 Adapter 学習時における L_{mse} の係数 α の効果

Model α	weather
InterCTC	18.7
InterCTC+TODA α=0.0	13.8
InterCTC+TODA α=1.0	13.7
InterCTC+TODA α=10.0	14.0

4.4 Adapter 学習時における $L_{\mathbf{mse}}$ の係数 α の効果

表 4 は，Adapter 学習時における L_{mse} の係数 α を 0.0, 1.0, 10.0 とした場合のドメイン適応後の天気予報評価セットにおける CER の比較である．α が 0.0 の場合は，L_{mse} の項を Adapter 学習時に加えず，L_{m} のみで学習を行うことを意味する．結果，α が 0.0 のときでも，認識精度の改善がみられた．P_{m} の確率算出時に用いる線形変換層は，ソースモデルの学習完了後のパラメータを用いており，Adapter 学習時は固定としている．このため，L_{m} のみでも，ソースモデルの潜在変数 $\boldsymbol{h}_{\mathbf{m}}$ に Adapter の出力が近づくように学習することが可能であることがわかる．α を 1.0 にすることで，α が 0.0 のときよりもわずかに認識精度は上回った．しかし，α が 10.0 のときは，0.0 のときよりも認識精度が下回った．このことから，α の値を適切に調整したうえで，より強い制約 L_{mse} を入れるとソースモデルの潜在変数 $\boldsymbol{h}_{\mathbf{m}}$ に Adapter の出力がより近づくように学習されることが推察される．

5. まとめ

本稿では，推論対象の話題のテキストデータのみで音声認識モデルをドメイン適応することで，推論対象の話題の音声の認識精度を改善する手法を提案した．CSJ コーパスでソースモデルを学習させた後，NHK が過去に放送した天気予報のテキストでドメイン適応させた結果，天気予報の音声に対する CER が 18.7 %から 13.7 %に改善した．また，言語モデルを Shallow Fusion させることで相補的に認識精度の改善が可能であることを示した．また，提案手法と Shallow Fusion，ビームサーチを併用することで，CER が 11.3 %まで改善した．今回は天気予報にドメインを絞ったが，スポーツや政治などより広範なドメインに対しても適用可能か，今後検討をすすめていきたい．

参考文献

[1] Chung-Cheng Chiu et al. “State-of-the-Art Speech Recognition with Sequence-to-Sequence Models,” in *Proc. of ICASSP*, 2018.
[2] Kanishka Rao et al. “Exploring architectures, data and units for streaming end-to-end speech recognition with RNN-transducer,” in *Proc. of ASRU*, 2017.
[3] Anmol Gulati et al. “Conformer: Convolution-augmented Transformer for Speech Recognition,” in *Proc. of Interspeech*, 2020.
[4] Yosuke Higuchi et al. “A Comparative Study on Non-Autoregressive Modelings for Speech-to-Text Generation,” in *Proc. of ASRU*, 2021.
[5] Erik McDermott et al. “A Density Ratio Approach to Language Model Fusion in End-To-End Automatic Speech Recognition,” in *Proc. of ASRU*, 2019.
[6] Cal Peyser et al. “Improving Tail Performance of a Deliberation E2E ASR Model Using a Large Text Corpus,” in *Proc. of Interspeech*, 2020.
[7] Zhong Meng et al. “Internal Language Model Estimation for Domain-Adaptive End-to-End Speech Recognition,” in *Proc. of ICASSP*, 2021.
[8] Anjuli Kannan et al. “An analysis of incorporating an external language model into a sequence-to-sequence model,” in *Proc. of IEEE*, 2018.
[9] Anuroop Sriram et al. “Cold fusion: Training seq2seq models together with language models,” *arXiv:1708.06426*, 2017.
[10] Jaesong Lee et al. “Intermediate Loss Regularization for CTC-based Speech Recognition,” in *Proc. of ICASSP*, 2021.
[11] Caglar Gulcehre et al. “On Using Monolingual Corpora in Neural Machine Translation,” in *arXiv:1503.03535v2*, 2015.
[12] Daniel Povey et al. “The Kaldi speech recognition toolkit,” in *Proc. of ASRU*, 2011.
[13] Daniel S Park et al. “SpecAugment: A simple data augmentation method for automatic speech recognition,” in *Proc. of Interspeech*, 2019.
[14] Shinji Watanabe et al. “ESPnet: End-to-end speech processing toolkit,” in *Proc. of Interspeech*, 2018.