深層学習を用いた応答文に適したモーションとボイスの推定手法

西 佑真[1]　梶岡 慎輔[1]　山本 大介[1]　高橋 直久[1]

概要：近年，Siri や Alexa の登場により，ユーザがエージェントに対して発話し，エージェントの応答を聞くことにより情報を得ることが一般的になっている．それに伴い，本研究室では，一般ユーザ向け音声対話コンテンツ作成システムである，MMDAgent EDIT の研究，開発が行われている．MMDAgent EDIT では，エージェントが応答する文章である応答文と，その応答文を呼び出すためのユーザが発話する，認識キーワードを Web サイトを通じて登録することで，簡単に一問一答式の音声対話コンテンツを作成可能である．その際に，エージェントが行うモーションや応答文を読み上げる声色を設定することが可能となっているが，一般のユーザにとって応答文に適したモーションや声色を設定することは難しく面倒であると考えられるため，設定されないことが多い．そこで応答文とモーション，応答文と声色の組を学習しそれぞれのモデルを作成することでモーションと声色を推定する手法を考えた．応答文を入力することで，推定されたモーションと声色が得られるためユーザにモーションと声色の設定を促すことが可能であると予想し，テストデータを用いて評価を行った。

Estimation Method of Motion and Voice by Deep Learning

Yuma NISHI[1]　Shinsuke KAJIOKA[1]　Daisuke YAMAMOTO[1]　Naohisa TAKAHASHI[1]

1. はじめに

近年，Siri や Alexa などのユーザがエージェントと対話することで情報を得る音声対話システムが普及している．本研究室で研究されている MMDAgent EDIT[1] は音声インタラクションシステム構築ツールキット MMDAgent[2] を基に，一般のユーザでも簡単に一問一答式の音声対話コンテンツを作成できるシステムである．MMDAgent EDIT ではエージェントとして 3D のキャラクタが使用されているため，ガッツポーズやお辞儀といったモーションをすることが可能となっている．さらにエージェントが発話する際の声色も幸せな調子や悲しい調子のように設定することが可能である．しかし，一般のユーザにより生成された，MMDAgent EDIT に投稿された音声対話データを見ると，ユーザによる設定がないため、デフォルトのモーションなしや声色は普通の調子に設定されているものがとても多くなっている．ここで一般ユーザにとっては応答文に適したモーションや声色を考えることは難しく，面倒なことなのではないかと考えた．そこで深層学習を用いたモーションと声色の推定が実現できれば，一般ユーザがモーションや声色を設定する際の参考になり，設定を促すことができると考えた．しかし，モデルを作成する上で用いる学習用データは MMDAgent EDIT に投稿された音声対話データとなるため，モーションなしや声色が普通というようなデータが多く，お辞儀や悲しい声色のようなデータが少ない不均衡データとなっている．そのため，学習をそのまま行ってしまうと多数派クラスを予測すればほとんど正解すると学習され，少数派クラスを予測しなくなるという問題があった．

[1] 名古屋工業大学大学院工学研究科
Graduate School of Engineering, Nagoya Institute of Technology

2. 実現上の課題とアプローチ

2.1 データの前処理

1 つ目の課題として，学習データとして用いるユーザ生成の MMDAgent EDIT のデータには，テストデータのような学習に適さないデータや，学習に役に立たないと考え

られる単語が多く含まれていることが挙げられる．

この課題は学習データに対して除去処理や整形処理といった前処理を行うことで対応した．

2.2 不均衡データへの対応

2 つ目の課題として学習データがラベル数に大きなばらつきのある，不均衡データとなっている点が挙げられる．

この課題は，不均衡データに対して，アンダーサンプリングと損失関数の調整を行うことで対応した．

3. 関連研究

本研究室では音声対話インタラクションシステム構築ツールキット MMDAgent を用いて研究を行っている．FST と呼ばれる対話スクリプトを編集することで，画面上の 3D キャラクタと対話が可能である．また，応答する内容だけでなく，応答する際のモーションや応答文を読み上げる声色の設定も可能となっている．図 1 に，MMDAgent の動作画面を示す．

また，MMDAgent で構築可能なコンテンツを web 上で作成可能なシステムである MMDAgent EDIT がある．FST ファイルを編集する必要がなく，エージェントがユーザに対して応答する文章である応答文，応答文をエージェントから引き出すためにユーザが発話する認識キーワードの組を登録するだけで対話が作成できるため，一般のユーザでも簡単に一問一答式の音声対話コンテンツを作成することができる．このシステムでも MMDAgent と同様に 3D キャラクタと対話する形式をとっており，ユーザがモーションや声色を設定することが可能となっている．MMDAgent EDIT の動作画面を図 2 に，設定画面を図 3 に示す．

関連する論文として，固有表現抽出を用いた音声対話コンテンツ向け認識キーワードの推定手法 [3] がある．一般ユーザにとって，応答文に適した認識キーワードを設定することは難しいという問題を解決している．固有表現抽出器を作成することで，入力された応答文から音声対話コンテンツ向け認識キーワードを自動推定することを可能とした．

次に，テキストに現れる感情，コミュニケーション，動作タイプの推定に基づく顔文字の推薦 [4] が挙げられる．この論文では電子メールなどでよく使われる顔文字を，ユーザの入力したテキストに現れる感情，コミュニケーション，動作タイプの推定を行うことで推薦する手法を提案している．感情推定には k-NN を利用し，入力されたテキストに辞書内の語が含まれているか調べ，学習データとの類似度を計算することで推定し，その結果を用いて顔文字データベースから適切な顔文字を取り出して推薦している．

最後に，実践 自然言語処理 実世界 NLP アプリケーション開発のベストプラクティス [5] が参考図書として挙げられる．この図書では、自然言語処理の幅広い問題へのアプローチやテクニックが示されている．前処理の知識や，ストップワード，学習のアルゴリズムなどの情報を得ることができる．

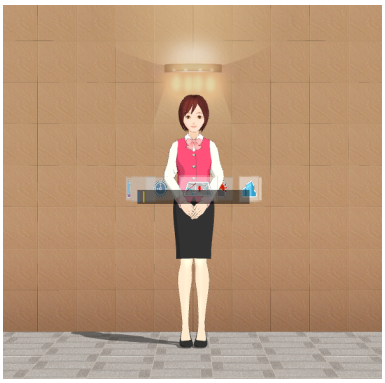

図 1 MMDAgent の動作画面

図 2 MMDAgent EDIT の動作画面

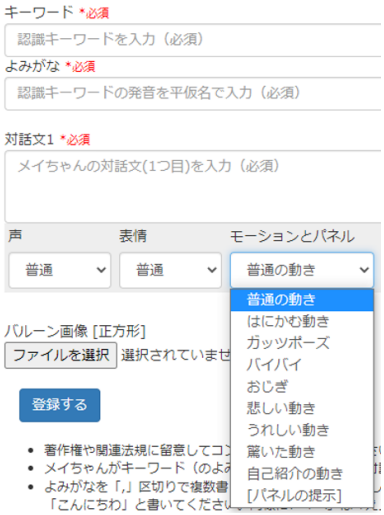

図 3 MMDAgent EDIT の設定画面

4. 提案手法

4.1 提案手法の構成

提案手法の構成図を図 4 に示す．提案手法では，はじめに MMDAgent EDIT のデータに対して応答文中の数字を 0 に置き換え，形態素解析を行い単語を原型化し，ストップワードと呼ばれる一般的で役に立たないと考えられる単語を除去するというような前処理を行い学習データを作成する．次に学習コーパスのラベル数の差を小さくするために，アンダーサンプリングを行う．最後に損失関数を少数派クラスを重視するように変更し，少数派クラスの推定が行われるように変更することで多クラス分類モデルが作成される．この多クラス分類モデルにユーザが作成した応答文を入力として渡すことで，モーション，声色の推定が行われる．

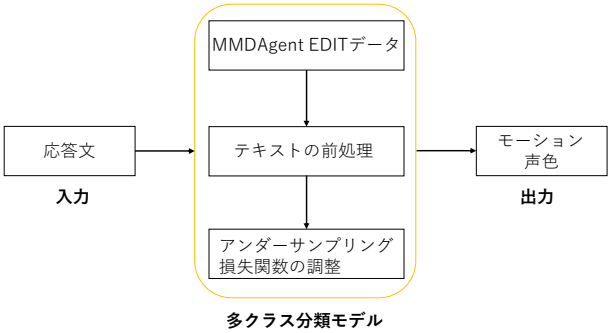

図 4 システムの構成図

4.2 提案手法の機能

4.2.1 学習データの前処理

MMDAgent EDIT に投稿された，ユーザ生成の音声対話データに対して前処理を行う．このデータは認識キーワード，応答文，モーション，声色など様々なデータが含まれているため，応答文とモーション，応答文と声色の 2 組を抽出した．抽出した応答文には「これはテストです」というような，ユーザがテストのために作成したと考えられる，学習には不必要なデータが含まれている．また意味のないひらがなの羅列や，3 文字以下のようなテキストも学習には適していないと考えられるため除去する必要がある．

不適切な応答文が除去されたら，学習が行われやすい形にデータを整形する必要がある．今回は数字の置き換え，形態素解析，単語の原型化，ストップワードの除去を行った．

4.2.2 多クラス分類モデルの作成

多クラス分類モデル作成には前処理が行われた学習データを入力として，モーション，声色のそれぞれを出力とするように作成した．学習には LSTM と CNN という手法を用いてそれぞれの精度を比較した．

5. 提案手法の実現法

5.1 学習データの前処理

5.1.1 学習データの作成

まず，MMDAgent EDIT に投稿された音声対話データから，応答文とモーション，応答文と声色の組を抽出する．応答文とモーション，応答文と声色の組について行う処理は同様であるため，応答文とモーションの組について説明する．MMDAgent EDIT に投稿された音声対話データはデータベースの形式をとっており，表 1 のように学習データを作成した．ここで speech は応答文を表し，motion はモーションを表している

表 1 学習データの構成

属性名	speech	motion
型	text	ファイル名

5.1.2 除去するデータ

学習データに含まれている除去すべきだと考えられるデータを表 2 に示す．データ 1 ではユーザがテストのために作成したと考えられるデータであるため削除を行う必要がある．データ 2 では文章ではなく，単語が登録されているという状況であり，3 文字以下のような短いテキストは削除を行った．

表 2 学習データの構成

属性名	speech	motion
データ 1	これはテストです	動作なし
データ 2	あなた	動作なし

5.1.3 データの整形

学習に不適切だと考えられるデータの削除は完了したが，学習しやすいデータに整形を行わなければ良い精度が得られない．そこで本研究では数字の正規化，形態素解析と単語の原型化，ストップワードの除去を行った．まず，数字の正規化の例を表 3 に示す．例えば日付が含まれる応答文では，数字をすべて 0 で置き換えた．数値表現は多様で出現頻度が高いにもかかわらず，学習にあまり役に立たないとされているので，置き換え処理を行った．形態素解析には janome[6] を利用した．

次に形態素解析と単語の原型化を行った例を表 4 に示す．形態素解析を行うことで単語が形態素ごとに分割され，「いえ」が「いう」のように原型化されている．

最後にストップワードの除去について説明する．ストップワードは学習を行う際に，一般的で役に立たないなどの理由で処理対象外とする単語のことである．出現頻度が高いにもかかわらず，学習に役に立たないだけでなく，計算量や性能に悪影響を及ぼすと考えられるため除去を行った．本研究では定義済みの SlothLib[7] というデータを利

用した．SlothLib の例を表 5 に示す．このような単語を除去して学習データを作成した．

表 3 数字の正規化の例

	テキスト
処理前	2020 年にオリンピックが開かれる
処理後	0000 年にオリンピックが開かれる

表 4 形態素解析と単語の原型化の例

	テキスト
処理前	うどんといえば，味噌煮込みうどんです
処理後	うどん と いう ば ， 味噌 煮込み うどん です

表 5 SlothLib の例

ストップワード
あそこ
あたり
あちら
あっち
あと
あなた
いくつ

5.2 不均衡データへの対応

5.2.1 アンダーサンプリング

不適切なデータの除去，データの整形を行い作成した学習データのモーションのクラス数を表 6，声色のクラス数を表 7 に示す．ここで MMDAgent EDIT では他にもさまざまなモーションや声色の設定が可能であるが，データの件数があまりにも少なく，学習がうまくできないと考えられたものは除外している．まず，モーションの件数を見ると，もっとも件数の少ないお辞儀が 42 件であるのに対して，普通の件数が 1252 件となっている．このデータで学習を行うと，約 66%が普通であるため，すべて普通と推定するだけで高い正解率が得られるため，学習がうまく行われなかった．そこでアンダーサンプリングを行うことで，モーションのクラス数の最大値が 500，声色の件数の最大値が 600 になるように，モーションのクラス数を表 8，声色のクラス数を表 9 に示すように変更した．本研究では，作成したモデルのうちマクロ平均の数値がもっとも良かった 500 件と 600 件を選択した．

表 6 モーションの件数

モーション	件数
ガッツ	201
お辞儀	42
幸せ	324
悲しい	66
普通	1252

表 7 声色の件数

声色	件数
幸せ	411
普通	1385
悲しい	76

表 8 モーションの件数

モーション	件数
ガッツ	201
お辞儀	42
幸せ	324
悲しい	66
普通	500

表 9 声色の件数

声色	件数
幸せ	411
普通	600
悲しい	76

5.3 損失関数の調整

多クラス分類モデルを作成する際によく使われる損失関数として，式 1 のような交差エントロピー誤差がある．p_i は正解ラベルを表し，q_i は推定された確率を表している．交差エントロピー誤差では 1 件ごとの誤差がすべて同等に計算されるため，今回の学習データの場合，すべて同等に扱ってしまうと，件数が多いクラスを重視するようになり少数派クラスが軽視されてしまう．そこで式 2 のように，クラスごとの件数で割るように変更することで，すべてのクラスが同等に計算されるように変更した．

$$-\frac{1}{n}\sum_{i=1}^{m} p_i \log\left(q_i\right) \tag{1}$$

$$-\sum_{i=1}^{m} \frac{1}{n_i} p_i \log\left(q_i\right) \tag{2}$$

5.4 多クラス分類モデルの作成

MMDAgent EDIT に投稿された音声対話データに対して前処理を行い，作成した学習データを用いて多クラス分類モデルを作成する．多クラス分類モデルの作成には，Long Short Term Memory(LSTM) と Convolutional Neural Network(CNN) という手法を用いる．

5.4.1 Long Short Term Memory(LSTM)

Long Short Term Memory(LSTM) はリカレントニューラルネットワークの一種である．リカレントニューラルネットワークは，可変長の入力を扱うことができることに加えて，入力系列の要素間に存在する依存性を扱うことができる．リカレントニューラルネットワークはループを持ち，後続のネットワークへ情報を渡すことで，予測する際にそれまでに入力された系列を考慮して予測することができる．LSTM はリカレントニューラルネットワークよりも長期の時系列を考慮することができるモデルである．リカレントニューラルネットワークでは入力を等しく記憶しようとするのに対し，LSTM では重要でないことを忘れてから重要なことを記憶するという仕組みになっているため，より長期の時系列を考慮できるようになっている．

5.4.2 Convolutional Neural Network(CNN)

畳み込みニューラルネットワーク (Convolutional Neural Network(CNN)) はニューラルネットワークの一種であり，主に画像認識の分野で使われているネットワークである．CNN では畳み込み層とプーリング層を繰り返し適用し，

最後に全結合層を使って分類を行う．畳み込み層は複数のフィルタから構成され，それらのフィルタを用いて畳み込みを行う．畳み込みは入力データの対応する要素とフィルタの要素を乗算し足し合わせることで，特徴量を圧縮する．プーリング層ではある領域を一つの要素に縮約する演算を行い，入力のサイズを小さくする．このようにして作成されたフィルタを重ねて活性化関数でつないでいくことでネットワークを構築し，分類を行う．

5.4.3 層化 k 分割交差検証

交差検証はモデルの汎化性能を評価するための手法である．データを k 個に分割し，そのうちの 1 つをテストデータ，残りを学習データとして用いて k 個のモデルを作成し，それぞれの評価の平均をとることにより，モデルの性能を評価することができる．ここで，k 分割交差検証を行うとデータ中にラベルの偏りがあった場合，分割ごとに等しい割合でデータを分割することができない．また，今回のような不均衡データの場合偏りが起こりやすい状況であるため，層化 k 分割交差検証を適用することで，各分割内のラベル比率が全体と同じになるように実験を行った．本研究では、テストデータとして 20%のデータを取り分けたのち、k=5 として実験を行った．

5.4.4 fastText

fastText[8] は，Facebook により Word2vec を基に作成された単語をベクトル化するライブラリである．本研究では fastText を利用することにより，単語のベクトル表現を取得し，実験を行った．

6. プロトタイプシステム

6.1 プロトタイプシステム実装環境

システムの開発は, Windows10 上で Eclipse の環境の下で行い，プログラミング言語として Python を利用した．また学習データとして，MMDAgent EDIT に投稿された音声対話データを利用した．モデルの作成には keras/Tensorflow を利用した．

6.2 プロトタイプシステムの構成

6.2.1 LSTM を利用した構成

図 5 に本研究で使用した LSTM を利用した構成について示す．

入力層 入力を受け取る層であり，本研究では前処理が行われた学習データに対してシーケンス番号によるベクトル化を行い，長さを揃えるためにパディングを施したものを入力している．

埋め込み層 複数の文章を受け取り，文中の単語をベクトル表現にした文章を返す層である．入力層から受け取ったベクトルを埋め込み層に通すことで，fastText によるベクトル表現を得ることができる．

LSTM 層 前章で紹介した LSTM の処理を行う層である．

Dropout 層 学習を行う際に学習データに過剰に適合することで，汎化性能が落ち，未知のデータに対応できないモデルが出来てしまう過学習という問題がある．その過学習を抑制する方法として，Dropout がある．Dropout は学習時の更新でランダムな入力ユニットを 0 とすることで，一部のデータが欠損してても正しく認識されるように学習を行う．本研究でも過学習を防止するため Dropout を適用した．

全結合層 全結合層ではその層内のすべてのニューロンと接続する層で，入力とレイヤーの重み行列を掛け合わせ，バイアス項を足したものに活性化関数を適用することにより計算される．本研究では活性化関数に softmax 関数を用いることで，出力を確率値とし，その確率値が入力に対する各ラベルの推定された確率となる．

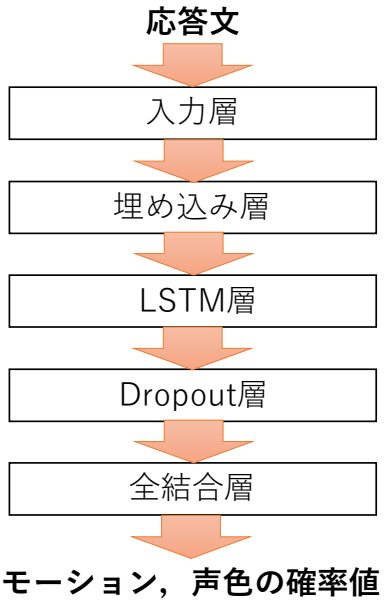

図 5 LSTM を利用した構成図

6.2.2 CNN を利用した構成

図 6 に本研究で使用した CNN を利用した構成について示す．

入力層 LSTM を利用した構成と同様に，シーケンス番号によるベクトル化とパディングを施したベクトルを受け取る．

埋め込み層 LSTM を利用した構成と同様に，入力層から受け取ったベクトルを埋め込み層に通すことで，単語の関係性を考慮した文章のベクトル表現を得ることができる．

畳み込み層 前章で紹介した CNN の処理を行う層である．Conv1D を利用して 1 次元の畳み込みを行った．活性化関数には ReLU を使用した．

Pooling 層 本研究では GlobalMaxPooling を利用したため，畳み込み層から得られた値の最大値を抽出した

ベクトルを作成する．これにより，学習時のパラメータを大きく減らし，過学習を抑制する効果も期待できる．

全結合層 LSTM を利用した構成と同様に，活性化関数に softmax 関数を用いることで，出力を確率値とし，その確率値が入力に対する各ラベルの推定された確率となるように作成した．

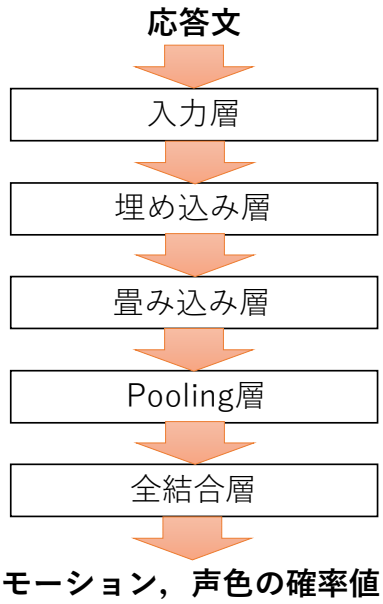

図 6　CNN を利用した構成図

7.　提案手法の評価

本章ではプロトタイプシステムを作成する際に実験を行った結果と，実際に作成したモデルに対して行った実験の結果について述べる．

7.1　モデル作成に関する実験

7.1.1　実験の目的

提案手法で述べた LSTM，CNN を用いて実験を行うにあたり，よい精度を得るためにはパラメータなどの調整をする必要がある．また，不均衡データへの対応を行ったことでどのように性能の改善が見られたかを検証することが本実験の目的である．

7.1.2　実験方法

MMDAgent EDIT のデータに前処理を行い作成した学習データに対して，不均衡データへの対応を行わなかった場合，アンダーサンプリングのみを行った場合，アンダーサンプリングと損失関数の調整を行った場合の 3 通りでモデルを作成する．この際 LSTM と CNN の 2 通りでモデルを作成するため，6 種のモデルを作成して実験を行う．実験は k=5 として層化 k 分割交差検証を行った．評価として式 3 で表される F 値と，式 5.4 で表されるマクロ平均を重視した．F 値は式 4 に示す適合率と，式 5 に示す再現率の調和平均である．またマクロ平均は各ラベルを同等に扱うため評価指標として選択した．

$$\text{F 値} = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$\text{マクロ平均} = \frac{F_{\text{ラベル 1}} + \ldots + F_{\text{ラベル N}}}{N} \tag{6}$$

7.2　モーションについての結果と考察

6 種類のモデルについて F 値をグラフにしたものを図 7 に，具体的な数値を表 10 に示す．ここで，アンダーサンプリングを US，損失関数の調整を LF と表記した．まず，悲しいの項目を見ると，不均衡データへの対応を行わなかった場合，ほとんど推測が行われておらず 0 に近い数値か 0 になっていることがわかる．一方でアンダーサンプリングや損失関数の調整を行った場合，推測が行われるようになり，最大で 0.4 を超える数値が得られたため，不均衡データへの対応は有効であったと観察できた．普通の項目を見ると，不均衡データへの対応を行わない方が高い数値となっている．これは，アンダーサンプリングを行わなければ多数のデータが動作なしとなっているため，ほとんどの応答文に対して動作なしと推定していたためだと考えられた．

図 8 にそれぞれの手法の F 値のマクロ平均，表 11 に具体的な数値を示す．マクロ平均のグラフからも不均衡データへの対応を行うことで，モデルの性能が向上していると観察された．また，今回の実験では LSTM よりも CNN を用いたほうが良い結果が得られた．

最後に問題点として F 値は向上したがそれでも低い数値であるという問題点が挙げられる．悲しいの項目の F 値は向上したものの，0.45 程度であり，普通の数値などとは大きな差が見られる．ガッツの項目を見ると，アンダーサンプリングを行うだけである程度数値が向上していることがわかるため，さらに F 値を向上させるには少数派クラスのデータを増やしたり，オーバーサンプリングなどの別の手法を組み合わせるなどして，データの件数の差を小さくする必要があると考えられた．

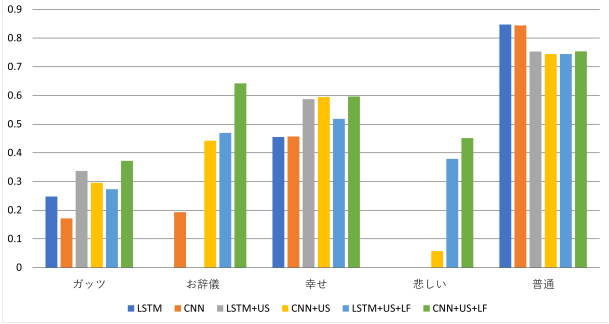

図 7　モーションの実験結果 (F 値)

表 10　モーションの実験結果 (F 値)

	ガッツ	お辞儀	幸せ	悲しい	普通
LSTM	0.247	0	0.455	0	0.847
CNN	0.171	0.193	0.457	0	0.844
LSTM+US	0.337	0	0.587	0	0.753
CNN+US	0.295	0.442	0.595	0.057	0.745
LSTM+US+LF	0.273	0.469	0.518	0.379	0.744
CNN+US+LF	0.372	0.642	0.596	0.452	0.754

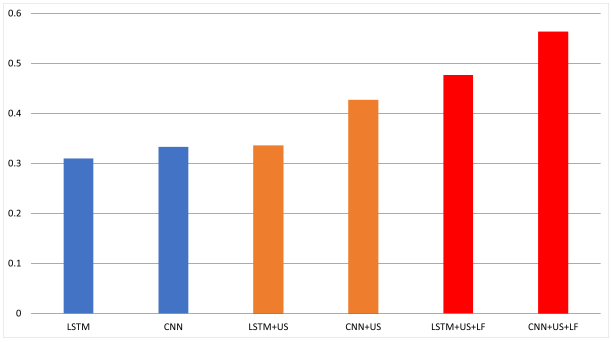

図 8　モーションの実験結果 (マクロ平均)

表 11　モーションの実験結果 (マクロ平均)

	マクロ平均
LSTM	0.310
CNN	0.333
LSTM+US	0.335
CNN+US	0.427
LSTM+US+LF	0.477
CNN+US+LF	0.563

7.3　声色についての結果と考察

声色について，F 値を比較したグラフを図 9 に，具体的な数値を表 12 に示す．不均衡データへの対応を行っていなかった場合，件数が多い普通のラベルを多く推定しているため，普通のラベルの F 値が高く，悲しいのラベルの F 値は 0 もしくは 0 に近い値となっていた．アンダーサンプリングを行うことで幸せの件数と普通の件数が近づいたためか，幸せのラベルの F 値は向上したが，まだ悲しいのラベルはほとんど推定されておらず，数値はあまり向上しなかった．次に損失関数の調整を行うと悲しいのラベルの F 値は 0.5 ほどになり大きな改善が見られた．また，幸せや普通の項目に関してもわずかではあるが，F 値が向上していることが観察された．

次に，図 10 にそれぞれの手法の F 値のマクロ平均，表 13 に具体的な数値を示す．アンダーサンプリングを行うだけでは大きな変化はなかったが，損失関数の調整を行うことで，悲しいのラベルが改善したためマクロ平均が向上した．また，全体としてはわずかではあるが LSTM よりも CNN のほうが良い結果が得られた．

声色に関しても，モーションの結果と同様に少数派クラスの悲しいの項目の F 値は向上したが，低いと感じられた．声色でも悲しいの項目のデータを増やすことができればさらなる性能の向上が期待できるのではないかと観察された．

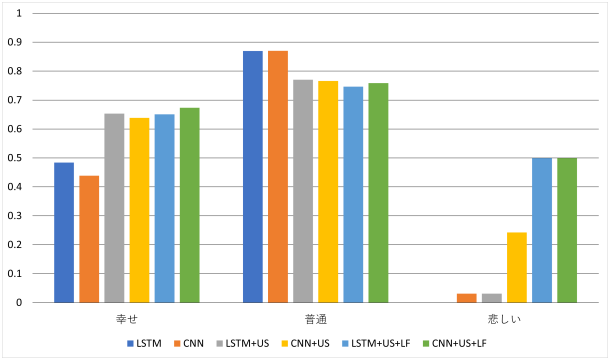

図 9　声色の実験結果 (F 値)

表 12　声色の実験結果 (F 値)

	幸せ	普通	悲しい
LSTM	0.484	0.870	0
CNN	0.438	0.870	0.031
LSTM+US	0.653	0.770	0.031
CNN+US	0.638	0.766	0.243
LSTM+US+LF	0.650	0.746	0.499
CNN+US+LF	0.673	0.759	0.499

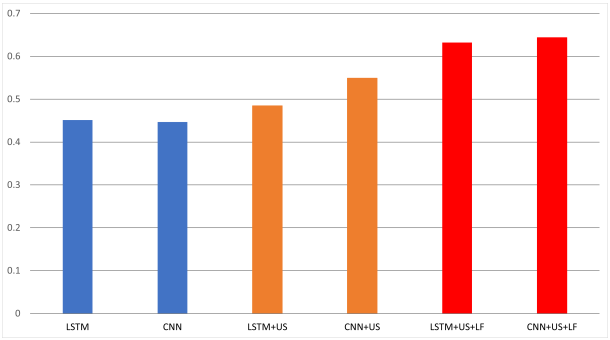

図 10　声色の実験結果 (マクロ平均)

表 13　声色の実験結果 (マクロ平均)

	マクロ平均
LSTM	0.451
CNN	0.447
LSTM+US	0.485
CNN+US	0.549
LSTM+US+LF	0.632
CNN+US+LF	0.644

8. おわりに

本論文では，一般ユーザ向け音声対話コンテンツ作成システム，MMDAgent EDIT の応答文に対するモーション，声色の設定を支援するために応答文とモーション，声色の組を学習した，多クラス分類モデルを提案した．まず，学習データとして利用した，MMDAgent EDIT に投稿された音声対話データは学習に不適切なデータが多く含まれていたため，それらを取り除いた．また，学習の精度を向上させるためにデータの整形を行った．次に学習データはラベルの件数に大きな差がある不均衡データであったため，その対応として，アンダーサンプリングと損失関数の調整を行った．そして，不均衡データへの対応を行っていないもの，アンダーサンプリングのみ行ったもの，アンダーサンプリングと損失関数の調整を行ったものの 3 種類を，LSTM と CNN の 2 種類の手法に基づいて学習させた計 6 種類の構成を比較し，F 値やマクロ平均を指標としてモデルの評価を行った．

モーションのモデルの評価では，件数が少ない悲しいの項目が不均衡データへの対応を行わなかった場合には推定されず F 値が 0 となっていたが，アンダーサンプリングと損失関数の調整を行うことで最大 0.452 の値まで向上した．そのためアンダーサンプリングと損失関数の調整が学習において有効であり，性能が向上することが分かった．

今後の課題として，少数派クラスの F 値は他の項目と比べると小さい値となっているという問題点が挙げられる．パラメータのチューニングによる性能向上の余地はまだ存在すると考えられるが，少数派クラスのデータを増やすなど追加の工夫を行うことで性能が向上するのではないかと考えられる．また，未知語への対応も今後の課題として挙げられる．柔軟に入力に対応するために解決すべき問題だと考えられた．

参考文献

[1] Ryota Nishimura, Daisuke Yamamoto, Takahiro Uchiya and Ichi Takumi, Web-based environment for user generation of spoken dialog for virtual assistants, EURASIP Journal on Audio, Speech, and Music Processing 2018, Article number17, 2018.

[2] A.Lee, K.Oura, and K.Tokuda, MMDAgent - A fully open-source toolkit for voice in-teraction systems, Proceedings of the ICASSP 2013, pp. 8382-8385, 2013.

[3] 前田一樹，固有表現抽出を用いた音声対話コンテンツ向け認識キーワードの推定手法，名古屋工業大学修士論文，2020.

[4] 江村 優花，関 洋平，テキストに現れる感情，コミュニケーション，動作タイプに基づく顔文字の推薦，IPSJ SIG Technical Report，pp.1-7，2012.

[5] Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta, Harshit Surana 著，中山光樹 訳，実践 自然言語処理 実世界 NLP アプリケーション開発のベストプラクティス，オライリー・ジャパン，2022.

[6] Janome, https://mocobeta.github.io/janome/ (2022.1.31 参照)

[7] SlothLib, http://svn.sourceforge.jp/svnroot/slothlib/CSharp/Version1/SlothLib/NLP/Filter/StopWord/word/Japanese.txt (2022.1.17 参照)

[8] fastText https://fasttext.cc/ (2022.4.18 参照)

[9] 中山光樹，機械学習・深層学習による自然言語処理入門 scikit-learn と TensorFlow を使った実践プログラミング，マイナビ出版，2020.