Vol.2018-EC-50 No.32
2018/12/22

機械学習を用いた音声処理によるオンラインゲーム上の交流ツールの検討

安孫子智樹, 加保貴奈

2018 年 11 月 29 日

1　概要

近年，インターネットに接続し，異なるユーザー同士が一緒に対戦・協力して遊べる，いわゆるオンラインゲームの数は多く存在する．ソフトウェアの種類によっては．ユーザーはパソコンやスマートフォン，家庭用ゲーム機などで，インターネットを介した通信プレイをすることができる．ユーザーが，画面の向こうにいる別のユーザーに，リアルタイムでコミュニケーションを取る手段はいくつかあるものの，方法によってはユーザーに操作を要求し，結果としてプレイに支障をきたす恐れがある．本研究では，コミュニケーションをとる手段として，SNS（Social Networking Service）などで用いられるスタンプ機能を，音声処理で利用する方法を提案した．ユーザーの声を入力とし，音声処理には機械学習を用いる方法を検討する．

2　はじめに

近年，インターネットの普及により，デジタルゲームには様々な形態が登場している．ユーザー同士が離れたところにいても，インターネットを介して，レースゲームや格闘ゲーム，シューティングゲームなど，同じ場面でプレイするオンラインゲームは広くなされている．パソコンやスマートフォンなどには，テキスト入力や，通話をするような感覚の声でチャットを行う機能を持つアプリケーションが存在し，それらをプレイヤーがゲームをしながら使うことが多い．また，ソフトウェアによっては，ゲームを共にプレイしながら，お互いが意思疎通を行えるような機能が提供される場合がある．その機能の一つに，ソフトウェア側で用意される「スタンプ機能」がある (アイコンと呼ばれることもあるが，ここでは一貫してスタンプと呼ぶ).

「スタンプ機能」は，既存の SNS などでも一般化されている他，実際に一部のゲームタイトルでも，ユーザー間のコミュニケーション用に使われている．基本的には，ゲームに登場するキャラクターが，ユーザーの声を代弁する形で使用される [1]. ただし，スタンプ機能を使うためには画面を切り替える操作や，プレイ中とは別に画面を選択する必要がある．常に場面が変化するゲームにおいては，画面の切り替えはプレイに支障を来たしてしまう恐れがある．そのため，スタンプを送るタイミングは少なく，それに伴い使われるスタンプの種類は，実際に用意されるごく一部に制限されている印象がある．スタンプ機能を十分に活かすためには，画面の切り替えや操作を行わずにできる必要がある．音声を使って実現することを考えた．本研究の実装のプログラミング言語には，Python3 を用いた．また，実験の順序としては，こちらの書籍 [2] を用いた．

3　特徴量

音声処理の技術では，分類処理が多用される．分類処理は，自動的な分類であるクラスタリング，認識の他にも，変換や生成，合成など，様々な局面で使われる．分類を行う際は，グループを区別するための手がかりとなるものを抽出する処理が必要となる．この手がかりのことを，特徴量と呼ぶ．本研究における音声処理においては，「短時間エネルギー」

と「零交差」の 2 つの特徴量を用いる．

3.1 短時間エネルギー

一般に録音を行うと，音がある部分と無音である部分ができる．音がある部分のみを抽出される処理は，多くの場面で用いられる．次の図 1 は，「南（みなみ）」という単語を録音したデータのうち，「南」の最初の部分を拡大して表示したものである．

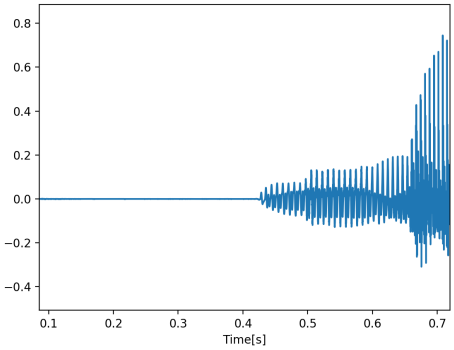

図 1 「南」の最初の部分

図 1 の 0.42 秒あたりから，“/m/” の音韻が始まっているのが分かるように，音のある部分には波の変位が大きくなる．変位が大きいことを反映する特徴量には，フレームごとのエネルギーである「短時間エネルギー」というものがある．入力音声 x の第 n フレームの値を F[n] とし，F[n] の短時間エネルギーを E[n] としたとき，F[n] および E[n] は以下の式で表される．

$$F[n] = x[nS], x[nS+1], \cdots, x[nS+N-1]$$

$$E[n] = \sum_{m=nS}^{nS+N-1} (x[m])^2$$

上の式において，S はフレームシフト，N はフレーム長を指す．サンプリング周波数 16kHz の wav ファイルに対して，フレーム長 512 点，フレームシフト 256 点（N=512, S=256）で短時間エネルギーを計算してプロットしたのが，下の図 2 である．

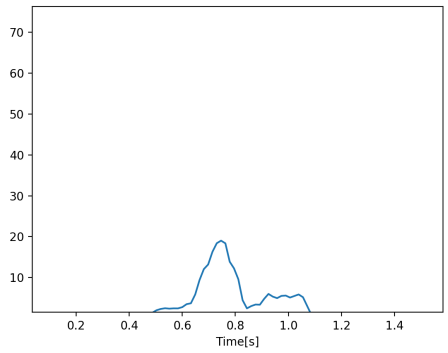

図 2 「南」の短時間エネルギー

3.2 零交差

音韻はいくつもの種類があり，それに伴い様々な方法で発声される．発声方法によっては，短時間エネルギーだけで十分に特徴をとらえられない音韻がある．図 3 は，「北（きた）」という単語の最初の部分を拡大したものである．

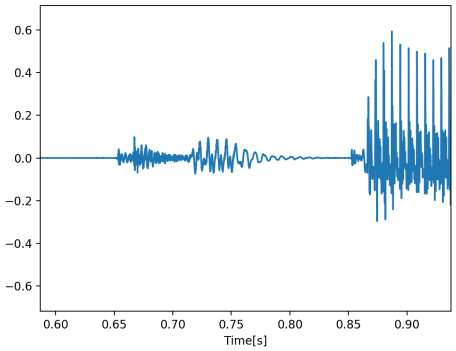

図 3 「北」の最初の部分

図 3 の 0.66 秒あたりから，“/k/” の音韻が始まっていることが読み取れる．0.86 秒から始まるの “/t/” の音韻と比較しても分かる通り，“/k/” の音韻のエネルギーはさほど高いものではない．ただし，無音の区間と比較すると，小刻みな振動が幾度も起こっていることが読み取れる．この特徴をフレームごとに容易にとらえるために使われる特徴量が，「零交差」である．

零交差は，一定時間内に波形がどれだけ零を通

るか，言い換えれば，波形の中で符号が何回変化するかを表す [3]．例えば，正弦波は 1 周期の間に 2 回符号が変化するので，1000Hz の正弦波の場合，1.0ms では 2 回符号が変化する．零交差は，短時間エネルギーに並び，波形を理解するのに役立つ特徴量である．

4　k 最近傍分類による機械学習

本研究では，音声を分類する際に，短時間エネルギーと零交差を用いて，音声ファイルから音声区間と無音区間に分類する．はじめに，先ほどの 2 つの特徴量から，2 次元平面に「南（みなみ）」の音声データ中にある，音声フレームとその前後の無音フレームを散布図にして表示した図を図 4 に示す．なお，短時間エネルギーが小さい場面が多くなるため，y 軸の短時間エネルギーを対数にして表示する．

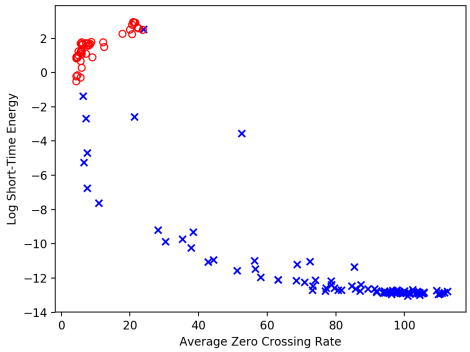

図 4　音声フレーム（◯）と無音フレーム（×）

多少は被っているものの，◯と×の領域は重ならないため，分類は行えそうであることがわかる．

この場面では，データ分類の方法として広く使われている，教師付き分類を使用する．教師付き分類は，手動で別々に分類したいグループ（クラスと呼ぶ）を分けたデータを用いて分類する手法である．教師付き分類の方法のうち，アルゴリズムが比較的単純な最近傍分類がある．最近傍分類とは，分類したい入力に対して，学習データから近いものを探し，それらのデータのグループであると分類する手法である．分類するとき，学習データから近いものを k 個探索し，もっとも多いクラスに分類するのが，k 最近傍分類である [4]．

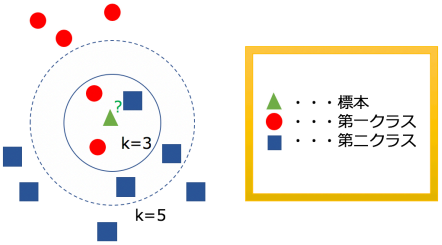

図 5　k 最近傍分類の説明図

上の図 5 を例として説明を行う．標本である緑の三角は，第一クラス（赤い丸）と第二クラス（青の四角）のいずれかに分類される．その際，探索する個数 k を 3 にすると，内側の円（実線）の内側が近傍であり，丸の数の方が多いため，第一クラスに分類される．しかしながら，探索する個数 k を 5 にすると，外側の円（破線）の内側が近傍であり，四角の数の方が多いため，第二クラスに分類される．k の取り方によって分類の結果が変わるため，注意が必要である．

5　実験評価・考察

先ほど説明した特徴量を用いて k 再近傍分類を行い，ゲームのプレイ中におけるプレイヤーの声から，プレイヤーがどの言葉を発したかを分類する．教師付き分類を用いるため，分類の際に使用する学習用の音声データが必要である．本実験では，ゲームのプレイ中に，プレイヤーの口から出る言葉の中で，出てくる頻度が高いと考えられる 3 種類の言葉を用いた．

- 強い敵を倒したときやステージをクリアしたときの「やった」
- 味方が活躍しているときや，強力な必殺技が発動したときの「いけ〜」
- 大ダメージを受けてしまったときや，敗北が決まったときなどの「うわ〜」

ゲームをする環境や，その場にいる他のプレイヤーと遊んでいるときなど，出てくる言葉は状況次第で

はいくらでも考えられるが，ここでは，どのような場面でも比較的出やすく，スタンプでの表示も容易な短い一言の言葉を用いている．

5.1 実験環境

はじめに，静かな環境の中で，著者の一人である安孫子智樹が自ら声を発して録音・保存することでサンプル音声を作成した．録音は，MacBook Pro 上のソフトウェア “Audacity” で用いて，wav ファイルで保存した．実験は，Python3 を用いて実装したプログラムを用いた．録音した音声サンプルの波形を以下に載せる（図 6，図 7，図 8）．

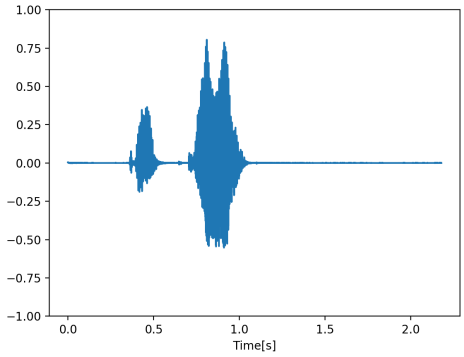

図 6 「やった」の波形

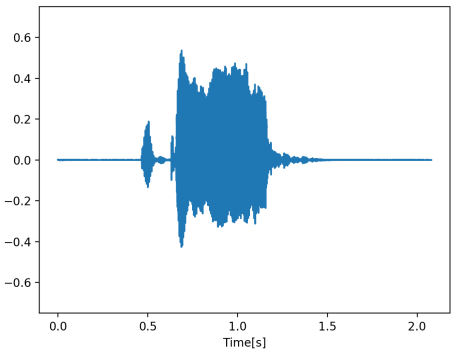

図 7 「いけ〜」の波形

また，k 再近傍分類を行ってプロットした散布図を以下に載せる（図 9，図 10，図 11）．図 4 と同様に，◯が音声フレーム，×が無音フレームである．

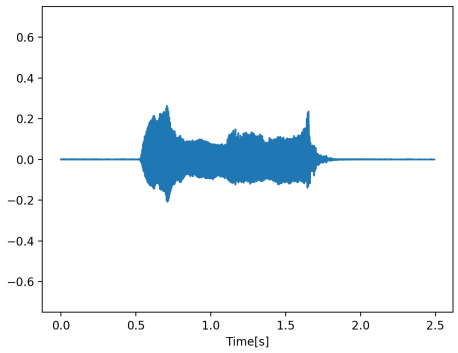

図 8 「うわ〜」の波形

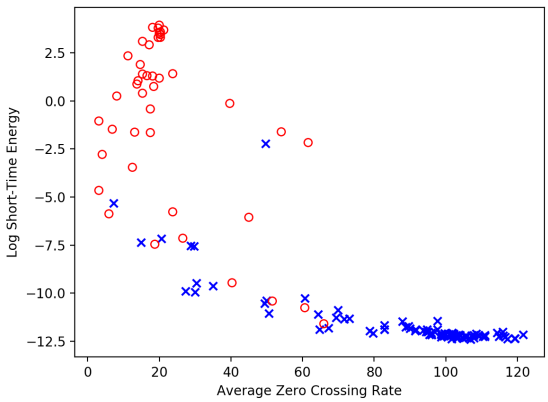

図 9 「やった」の分類図

それぞれのサンプル音声から，短時間エネルギーや零交差を用いて，「分類器」を作成する．分類器を作成するにあたって，先ほど説明した学習データ，つまり探索する k の個数は 5 としている．各分類器を使い，元の 3 種類の声データに対する誤り率を測定する．そして，各音声データに対応したスタンプを表示するようにした (図 12) ．スタンプ機能は SNS でみられるように，文字だけではなく送り手の意図や感情などを相手に伝える，情報伝達の手段として用いられており，コミュニケーションを活性化することに繋がると考えられている [5].

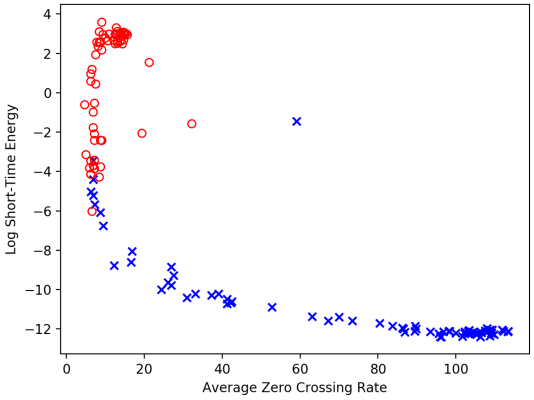

図 10　「いけ～」の分類図

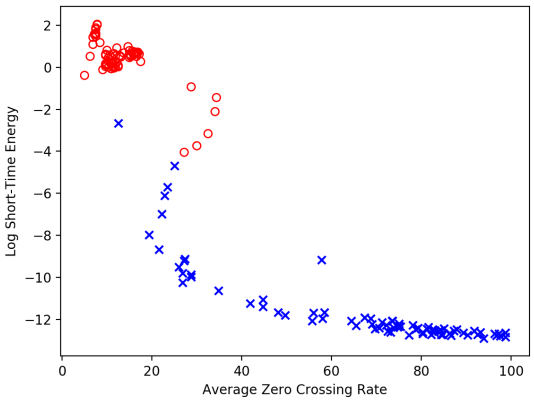

図 11　「うわ～」の分類図

図 12　3 種類のスタンプ

5.2　評価

それぞれ作成した分類器を用いて，元の 3 種類のサンプル音声に対する誤り率を測定する．誤り率が低いほど，サンプル音声が分類器の言葉に近いことを表している．また，各言葉に対応するスタンプも同時に表示されるようにする．

各分類器を用いた，サンプル音声に対する誤り率は以下の表 1 になった．

表 1　分類する言葉と各言葉の分類器の誤り率

分類する言葉	「やった」	「いけ～」	「うわ～」
「やった」	0.05926	0.08889	0.08148
「いけ～」	0.09375	0.03125	0.06250
「うわ～」	0.07143	0.04545	0.006494

(いずれの値も，有効数字を 4 桁としている)

今回の評価では，どの言葉も元の言葉の分類において最も誤り率が低くなり，分類は成功した．また，表示されたスタンプも，分類する言葉に対応したものとなった．一般に，学習データ自身に対する分類性能は，学習データとは異なるデータに対する性能よりも高くなるので，本実験では期待した通りの結果となったことが言える．

表 1 を見ると，「うわ～」の分類については，他の 2 種類の言葉と比較すると，元の言葉に対する誤り率が明らかに低い結果となった．

図 8 に示す，「うわ～」という言葉の波形を見ると，音声を発している間は，無音区間に近い特徴がなく，図 6 に示す「やった」および図 7 に示す「いけ～」と比較すると，学習が行いやすかったのではないかということが考えられる．

なお，今回の結果の前に，「やった」，「いけいけ」，「やられた」で評価をとっていた．その際，「やった」「いけいけ」の学習データについては，元の音声に対する誤り率が低くなり，分類は成功した．しかしながら，「やられた」の学習データでは，「やった」に対して最も誤り率が小さくなってしまい，分類は失敗した．これについては，「やった」と「やられた」では，最初の「や」の音と，最後の「た」の音が同じで共通性があったことが考えられる．そもそも「やった」の「っ」が無音区間と似た特徴であり，前後の無音が無音区間と判断されているのに対し，間の無音部分を音声区間とみなしているため，図 9 で見られるように，音声フレームがかなり分散し，正確な分類が困難だった可能性があげられる．

6 まとめ

音の特徴量としてあげられる，「短時間エネルギー」と「零交差」を用いて，k 最近傍分類による機械学習を行い，ゲームのプレイ中にユーザから発せられると思われる，短い言葉を判別することができた．それにより，オンラインゲーム上でのスタンプ機能の表示は可能であることが示された．小さい「っ」など，音によっては学習を効果的に行いにくいものもあるので，機械学習を用いた音声入力によるスタンプ機能を実装する際は，自然に出る言葉の中でも学習を行いやすく，また互いの共通性の低い言葉，言い換えれば，人間が聞き間違いにくいであろう言葉を選ぶことが，正しいスタンプを表示する効果的な方法であることが考えられる．

今後の展望を以下に述べる．まず，今回の音声区間，無音区間の決定は，作成済みの音声ファイルから，著者が手動で行っていたのだが，入力された音声から，自動でどこからどこまでの部分を音声と判断することが必要と思われる（オンラインプレイ中のため）．また，場面やユーザによっては，一緒にプレイする相手にとって不適切な言葉を発する可能性もあるので，一部の言葉の表現が入力された際，その言葉を解釈した上で適切な形に変換してスタンプを表示させることも考えている．

7 謝辞

研究場所の提供や，研究内容の議論をしていただき，日頃からお世話になっている，大阪大学 大学院情報科学研究科 情報ネットワーク学専攻 渡辺研究室の，渡辺尚教授，猿渡俊介准教授をはじめとした皆様に心から感謝の意を表する．

参考文献

[1] “https://www.nintendo.co.jp/3ds/ea3j/gameplay/index.html”.

[2] 伊藤克亘, 花泉弘, 小泉悠馬. Python で学ぶ実装画像・音声処理入門. コロナ社, 2018.

[3] David Gerhard. *Pitch extraction and fundamental frequency: History and current techniques*. Department of Computer Science, University of Regina Regina, 2003.

[4] Nicolás García-Pedrajas, Juan A Romero del Castillo, and Gonzalo Cerruela-García. A proposal for local k values for k-nearest neighbor rule. *IEEE transactions on neural networks and learning systems*, Vol. 28, No. 2, pp. 470–475, 2017.

[5] 須田康之, 大関達也, 菊地康介, 高山美畝, 山我拓也, SHI San, TEI Zengetsu. Line スタンプを用いたコミュニケーションの特質. 兵庫教育大学研究紀要第 49 巻, 2016.