部門

競技部門

No.1 登録番号

30028

予定開発期間:6カ月
予定開発人数:3人

No.2

	4月		5月			6月			7月			8月			9月		10月		1		
問題分析									¥												
設計					4	Ŧ							Á								
実装								\												\	
試用・トレーニング													7								

実現方法

1) 音声の解析アルゴリズム

音声の解析には以下の 2 つの解法のプログラムを使用する。それぞれのプログラムで答えとして出力されたもののうち、答えが一致した札を問題データに含まれているかるたのデータであると考え、問題データからその音声データを取り除き、小さな問題へと帰着させていき、答えを求める。

解法(1)

すべての読みデータ、問題データの音声データに対して音声波形を 1/30 秒ほどの区間に分け、その時点での波 形に高速フーリエ変換を行い、切り取られた区間の時間での音声の振幅の大きさを周波数ごとに分析する。複数の 音声のそれぞれの振幅の大きさの和が、複数の音声を合成した音声の振幅の大きさと近似できるということを利 用して解析を行う。

問題データの音声の解析は、山登り法というアルゴリズムを用いて行う。具体的には以下の方法で行う。 現在一番答えに近いものであると推測している札の数列を T とし、T の初期値は読み合せるデータ数だけの札を 重複しないようにすべてランダムに選んだものとする。以下の操作(1),(2)をランダムに選択し実行する。

- (1): Tの中から一つ選び、選んだ要素の札の音声の再生位置や長さをすべてランダムに変更する。
- (2): T の中から一つ選び、それを T の要素と重複しないように選んだ新しい札に更新し、その札の音声の再生位置、長さをランダムに決める。

Tに含まれる札の音声を合成した音声データと問題データとの一致度が前回より近いと判断された場合、現在答えに一番近い札の数列として更新する。そうでない場合、Tを前回の状態に戻す。一致度は、推測したデータの振幅と問題データの振幅の差の総和の大きさで計算する。これを時間の許す限り繰り返し行う。

解法(2)

No.3

解法(2)では Python3 のモジュールである PyRoomAcoustics を使用し、音源分離手法の一つである FastMMF 法を用いて行う。音源の開始時間や終了時間がずれている音源を分離し、事前に受け取った合成されていない音声と比べて答えを出す。また、分離した音源が重なっていないか実際に聞いて確認する。重なっていた場合は解法(1)を使って音源を検出する。

2) その他(独創的なところ)

読み札の音声データから問題データ、分割データを自動生成するプログラムを作成し、試用・トレーニングで現在のプログラムがどのくらいの正答率であるのかを効率的に調べられるようにする。

また、分割データの取得や、音声の解析プログラムの実行、スコア計算、取り札の出力を行う対話形式の GUI を作成し、回答提出までの効率化や札の変更に対応できるようにする。

ほかにも、問題データから、問題データに含まれていると推測された札の読みデータを取り除くプログラムを作成し、人間がその音声を聞くことで確認・訂正できるようにする予定である。

開発環境

使用言語: C++, Python3

No.4 開発環境: Visual Studio, Visual Studio Code ライブラリ: OpenSiv3D, PyRoomAcoustics

第33回 全国高等専門学校 プログラミングコンテスト: 群馬大会