テーマ番号	1EP6		
プロジェクトテーマ	和文 Whisper を用いた音素アライメントツールの開発	指導教員	鷹合 大輔 准教授
	英文 Development of a phoneme alignment tool using Whisper		
プロジェクトメンバー	4EP2-20 熊崎　拓人 (TAKUTO Kumazaki)		

Abstract　In recent years, the rapid growth of music distribution services has increased the demand for subtitles for music that synchronize time and lyrics.Therefore, we decided to develop a system that automatically creates subtitles for music.Although Whisper can be used to create subtitles for music, simply inputting music as it is is not accurate enough.Therefore, we constructed a system for sound source separation, a recognition system for silent sections, and a replacement system with correct lyrics. As a result, although not perfect, we were able to handle Whisper with better accuracy and automatically create subtitles for music.

Keywords　Whisper，Spleeter，音素アライメント，音源分離，編集距離

1.　はじめに

近年，音楽配信サービスの急速な増加により，時間と歌詞を同期付けした音楽用の字幕の需要が高まっている．一方，多くの大手カラオケ会社では，歌詞字幕の作成が手動で行われており，非常に時間と労力を必要とする業務となっている．これらの背景から，本プロジェクトでは音楽と正解の歌詞データを入力値として，文字起こし AI である Whisper を用いて自動で時間同期付きの歌詞データを出力するシステムを開発した．本稿では，歌詞と時間の同期情報が記されているデータのことを「字幕データ」，音声ファイル中の特定の音の始まりと終わりを自動で見つけ出す技術のことを「音素アライメント」と呼ぶ．

2.　システム概要

本システムの全体の流れを図 1 に示す．入力値を音楽データと正解歌詞データとし，最初に入力された音楽データから Spleeter を用いて音源分離を行う．その後，音源分離された音楽データから有声区間の抽出を行い，抽出された音源から Whisper を用いて音楽の文字起こしと時間の同期を行う．最後に精度を上げるために正解歌詞データと比較し，間違っている部分を置換し時間と歌詞が同期されたデータが出力されることで終了となる．

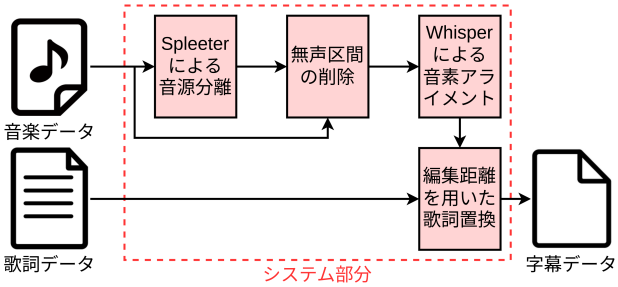

図 1 システムの流れ
Fig.4　Overrall system flow

3.　システム詳細

3.1　Whisper とは

Whisper とは，OpenAI が文字起こしサービスとして公開した音声認識モデルである．音楽データを Whisper に入力することで自動で字幕データを作成できるが，音声ファイルをそのまま入力するだけでは精度が不十分であるため，他のシステムを実装することで精度を向上させる．Whisper の具体的な問題点を以下に記す．

- ボーカルがない部分における誤認識
- 時間同期の精度が不十分
- 文字起こしに所々間違いがある

3.2　Spleeter を用いた音源分離

音源分離とは様々な音が混ざった状態からひとつひとつの音を取り出す技術のことである．今回は音楽の中からボーカルのみを取り出す．Spleeter という Python ライブラリを用いることで簡単に音源分離を行うことができる．

3.3　無声区間の削除

ボーカルがない部分で誤認識をするという問題点と時間同期の精度が不十分という問題点を解決するために無声区間を削除する必要がある．librosa というライブラリと音源分離で作成したボーカルのみの音楽データを用いて時間と音量の対応を示した csv ファイルを作成する．次に無声区間を特定するために閾値を設定し，音量の値が閾値より小さい場合は csv ファイルの 3 列目に 0 を，閾値以上の場合は 0.1 を追加する．現在の csv ファイルをグラフで示したものが図 2 である．無声区間が細かく設定されすぎているため，設定した時間以上無声区間が持続しなかった場合はその無声区間を有声区間に変更する．変更を加えたグラフが図 3 である．この作成した csv ファイルを使用して無声区間の削除を行う．音源分離後の音楽データに比べ，音源分離前の音楽データのほうが Whisper の文字起こし精度が高いため，音源分離前の音楽データの無声区間を削除する．

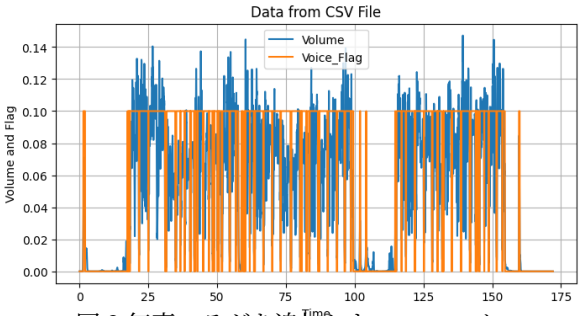

図 2 無声フラグを追加した csv ファイル
Fig.4　Csv file with silent flag added

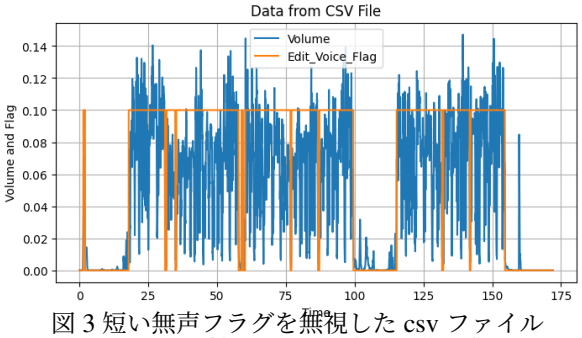

図 3 短い無声フラグを無視した csv ファイル
Fig.4　Csv file ignoring short silent flag

3.4 Whisper による音素アライメント

無声区間を削除した音楽データを Whisper に入力し音素アライメントを行う．Whisper には 5 つの学習済みモデルがある．本稿では実行時間を考慮して 2 番目に文字起こし精度が高い medium モデルを使用する．Whisper に音楽を入力することで文字起こしされた文章とその文章が出現する時間が出力される．この 2 つの情報を図 4 のような形式の srt ファイルという字幕データにする．

00:00:15,000 --> 00:00:27,000 ←開始時刻と終了時刻
夕焼け小焼の赤とんぼ ←文字起こしされた文章

00:00:28,000 --> 00:00:40,000
覚えていたのは いつの日か

図 4 srt ファイルの形式
Fig.4 Srt file format

3.5 編集距離を用いた歌詞置換

編集距離とは，2 ☒つの文字列がどの程度異なっているかを示す数値であり，語の挿入・削除・置換によって，一方の文字列をもう一方の文字列に変形するのに必要な手順の最小回数で求めることができる．Whisper の文字起こしに所々間違いがあるという問題点を解決するために，正解歌詞と出力された歌詞を比較し，編集距離が最小になるように置換を行う．最初に，入力した正解歌詞データと Whisper で出力された字幕データのそれぞれを MeCab を用いて形態素解析を行い細かい単語や語にする．次に編集距離が最小になるような単語と語の組み合わせを求め，正解歌詞データと字幕データを対応付ける．字幕データの歌詞部分を正解歌詞で置換することによって，より正確な字幕データを作成することができる．

4. 評価実験

4.1 実験方法

テンポの違う音楽を用いて時間同期の精度がどのように変化するかを実験する．歌詞置換が終了した後の srt ファイルから文章を取り出し，手動で正解の開始時刻と終了時刻を入力した新しい srt ファイルを作成する．もとの srt ファイルと新しく作成した srt ファイルを比較して違いがどの程度発生するかを調査する．なお，時間の許容範囲は 0.1 秒とし，許容範囲内のズレは問題ないものとする．

4.2 実験結果とその考察

5. まとめ

参考文献

[1] 織田 信長, 明智 光秀, ”JPEG2000 画像符号化システムにおける係数ビットモデリングと適応算術符号化，”Journal of signal processing(基礎シリーズ), vol.7, no.4, pp.257-266, July 2003.
[2] Parrot, ”AR.Drone Developer Guide SDK 2.0”
[3] ” 金沢の暮らし”, http://www.kanazawa-it.ac.jp
[4] 山田 太郎, ” 金沢の一人暮らし”, トンチンカン出版, 2016.