

中の人も作ってみた【ハンズフリー音声翻訳機】

●はじめに

こんにちは。情報通信研究機構（NICT）の木村です。

多言語音声翻訳コンテストを開催し、音声翻訳の API が利用できるサンドボックスサーバを無料公開しています。サーバの動作検証用に音声認識、翻訳、音声合成の API を使っているうちに、つい楽しくなってきたいくつかのアプリを作ってみました。



●何を作ったか

ご紹介するのは、ハンズフリー音声翻訳機です。一方通行ですが、相手に伝えたいことを話すだけで翻訳して伝えることが出来る装置です。仕組みとしては単純なものです。

1. 音声をマイクから録音する
2. 話している間の音声だけを切り出す（発話区間検出 API に含まれていません。※1）
3. 切り出した音声を文字にする（音声認識）
4. 音声認識の結果を翻訳する（機械翻訳）
5. 翻訳結果を音声にする（音声合成）
6. 音声をスピーカーから再生する

※1：本アプリで使用したのとは異なりますが、記事の最後で NICT 製の発話区間検出をご紹介します。

マイクはスマートフォン内蔵のマイクを使えば良いですし、スピーカーもスマートフォンにあります。ただ、発話区間検出をしようとする、スマートフォンのスピーカーから鳴った音を発話として認識してしまいます。マイクとスピーカーの位置を遠ざければ解決するので、スピーカーは外部接続にしました。こうすると、翻訳結果の合成音を大きく再生し相手に伝えるためにも有効です。

スマートフォンは手に持たず胸のポケットに入れることで、マイクと口の距離を近くして S/N 比（音声周囲の音よりどれだけ大きい）をかせぐようにしました。

●システムイメージ

以下の写真のように、装置の装着も簡単に出来るようにイメージして開発しました。

実際このイメージ通りに使えます。

静かな状態が0.4秒間あると、発話終了とみなすように設定してあるので、続けて話す場合は、文の切れ目で0.4秒以上の間を取れば複数の文章として翻訳されます。

長すぎる文章は翻訳誤りとなりやすいので、適度に文章を切って使うことをお勧めします。



●API とサンプルコードのご紹介

サンドボックスサーバでは、API だけではなくサンプルアプリも公開されています。

録音、音声認識、翻訳、音声合成、音声の再生もこのサンプルを流用すれば一から作らなくて良いので楽です。

多言語
音声翻訳
コンテスト

TOP
トップ

IDEA
CONTEST
アイデアコンテスト

PoC
CONTEST
試作品コンテスト

NEWS
ニュース

FAQ
よくある質問

INQUIRY
お問い合わせ

NICT

世界大喜利
World Oogiri

■ サンドボックスサーバーで使えるAPI

多言語音声翻訳は以下の1~3の機能の組み合わせで実現しています。
ご利用いただく際は、これらを自由に組み合わせて（もちろん単品でも）利用いただくことができます。

 **APIとは**

Application Programming Interfaceの略で、アプリを開発する人が自分の作っているプログラムの中で、他の人が作ったプログラムの機能を使うための窓口です。

SHARE



1 **音声認識API**

音声データを送信すると、その音声をテキストに変換したデータを取得できる機能です。

2 **機械翻訳API**

ある言語で書かれたテキストと翻訳したい言語を指定すると、翻訳されたテキストを取得できる機能です。

3 **音声合成API**

テキストデータを送信するとそのテキストを読み上げる音声データを取得できる機能です。

音声翻訳のリクエストは、STML(Speech Translation Markup Language)という記述言語を使って書くのですが、細かい事を気にしなければ、サンプルコードに埋め込まれている STML の断片に含まれる、音声翻訳したい言語の文字列の部分だけを書き換えれば済みます。

【多言語音声翻訳コンテスト サンドボックスサーバ】

<https://mimi.readme.io/page/tagengo>

Java、Android で日本語から英語へ翻訳をする時のサンプルコード

```
String inputLanguage = "ja";//翻訳元言語の指定
String targetLanguage = "en";//翻訳先言語の指定
String text = srOutputView.getText().toString();
String MTRequestTemplate = "<?xml version=¥"1.0¥" encoding=¥"UTF-8¥"?">¥n" +
    "<STML UtteranceID=¥"0¥" Version=¥"1.0¥">¥n" +
    "<User ID=¥"N/A¥"/>¥n" +
    "<MT_IN SourceLanguage=¥"%s¥" TargetLanguage=¥"%s¥">¥n" +
    "<InputTextFormat Form=¥"SurfaceForm¥"/>¥n" +
    "<OutputTextFormat Form=¥"SurfaceForm¥"/>¥n" +
    "<s>%s</s>¥n" +
    "</MT_IN>¥n" +
    "</STML>¥n";//翻訳リクエストのテンプレート

String requestXML = String.format(MTRequestTemplate, inputLanguage, targetLanguage, text);
ClientComCtrl client = new ClientComCtrl(accessToken);
responseData response = client.request(SSURL, requestXML);//翻訳のリクエスト送信
```

こんな風なコードがあるので、真似して（コピペして）書けばいいです。
ソースは [GitHub](#) に上がっていますので、ダウンロードして使います。

【サンプルコード】

<https://github.com/FairyDevicesRD/mimi.tagengo.examples>

● サンプルコードを使ってみる (Android 編)

私は Android 用に作ったので、Android Studio を使って開発しました。

1、Android Studio をインストールする

こちらのサイトからダウンロードできます (無料)

<https://developer.android.com/studio/?hl=ja>

2、サンプルアプリをダウンロードする

ここに Android 用のコードが置いてあります。

<https://github.com/FairyDevicesRD/mimi.tagengo.examples/tree/master/src/Android>

3、コンパイルする

ビルド方法については、サンプルアプリと同じところに説明があります。

4、実機にインストールする

Google Play に公開するのでなければ、Google のデベロッパー登録は不要です。

Android Studio を使えば、Android のスマートフォン、タブレットに USB 接続でインストール出来ます。

5. ソースの参照場所

以下のソースを見れば、録音、再生方法、音声認識、翻訳、音声合成の書き方が分かります。

場所 : `src/Android/TagengoExampleApp/app/src/main/java/jp/fairydevices/mimi/example/`

ソース : `PrismClient.java`

これを参考にしながら、独自のシステムを開発してください。

●今回作ったアプリケーションの仕様

画面仕様

マイクボタンを押すと録音を開始します。

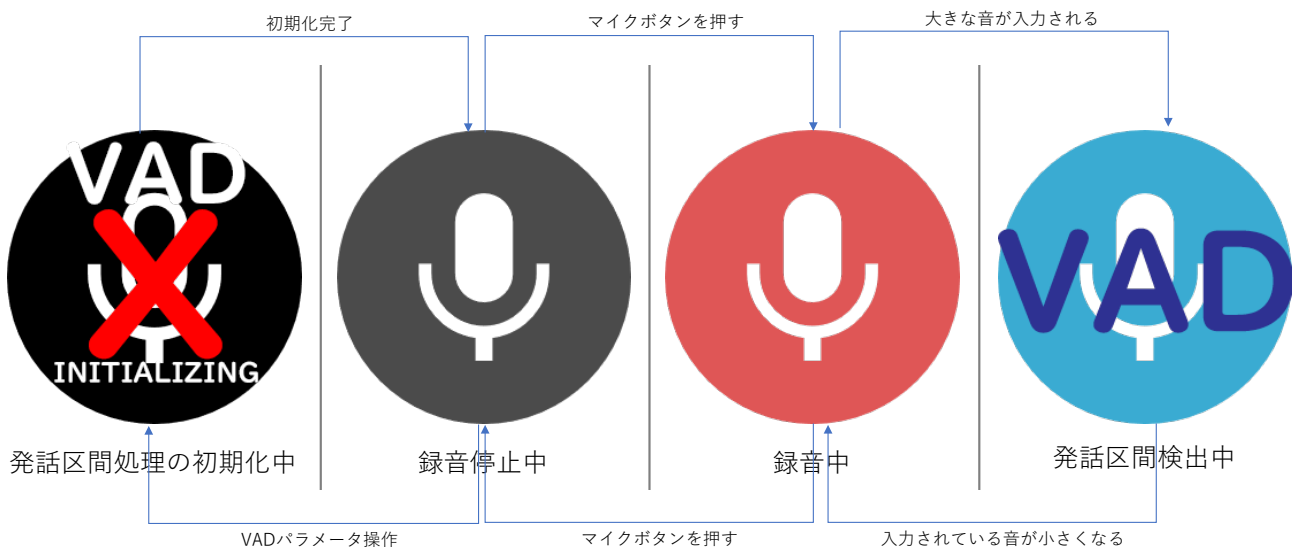
発話区間検出のパラメータ以上に大きい音が入力されたら、音声認識を始めます。



マイクボタンアイコンの遷移

マイクボタンの絵で状態を表示しています。

実際に使っている時は、スマートフォンを胸に入れているので見えませんが。



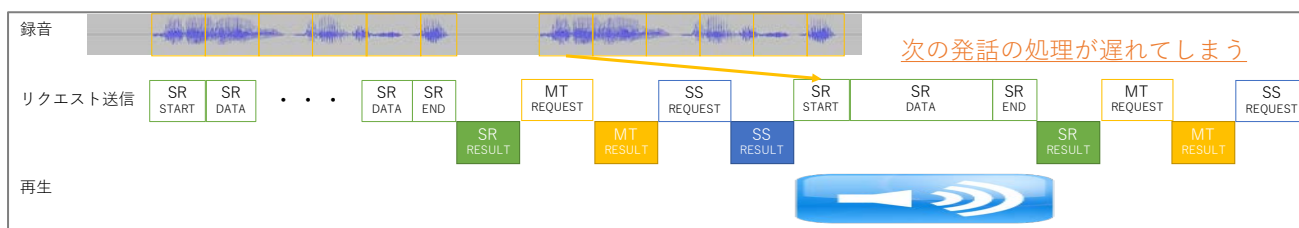
連続音声翻訳の仕組み

単純に音声認識(SR)、翻訳(MT)、音声合成(SS)、(逆翻訳※2)を実行すると、1回分の発話処理については、問題ありませんが、発話の翻訳・合成の処理が終わるまでは、次の発話の音声認識を行えないため、発話するのを待たなければならず、不便です。

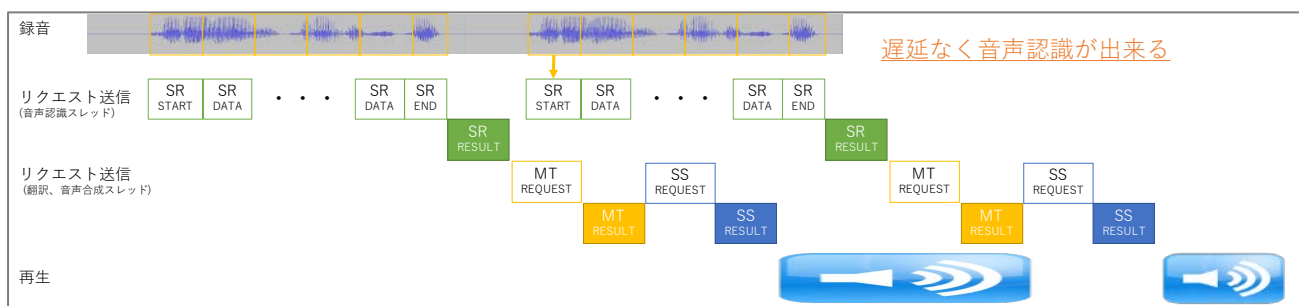
そこで、音声認識と、翻訳・音声合成の処理を別のスレッドに分けて行う事で、翻訳・合成の処理を気にせず次々に発話出来るようにしました。

※2：逆翻訳は翻訳結果の翻訳結果確認用なので、下図の説明には記載していません。

シングルスレッドで処理した場合



音声認識と、翻訳・音声合成にスレッドを分けて処理した場合



翻訳言語の切り替え方

「〇〇語に翻訳して」

というフレーズで、翻訳先の言語を〇〇に変更する仕組みを追加で作成しました。

仕組みは単純で音声認識結果に、「に翻訳」が入っていればその前にある文字が言語名なら翻訳言語を切り替えます。

これで、画面操作をすることなく翻訳言語を切り替えられます。

翻訳言語の設定は、スマートフォン保存しておき、次に起動したときはその言語で翻訳出来るようにしてあります。

●まとめ

中の人が言うのもなんですが、想像以上に簡単に作れます。

私は Java で Android 用を開発しましたが、iOS 用の API もあるので iPhone や iPad 用のアプリも開発できます。

是非皆さんもサンドボックスサーバを使って、試作品コンテストに応募してください。

(中の人が作って見たその他のアプリ)

【語学学習用アプリ (Android 版)】

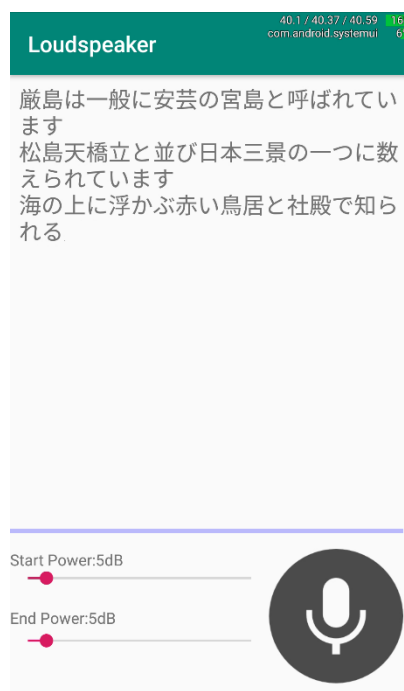


上の枠で翻訳したい文を話して、下の枠で発音練習が出来ます。ちゃんと発音できると音声認識結果が翻訳結果と一致するので「ぴんぽーん」と音が鳴ります。

黄色のマイクボタンで翻訳結果の合成音を何度も聞いて、耳を鳴らして発話練習をします。

自分の発話がどんな風になってるかは、青いマイクボタンを押せば確認できます。

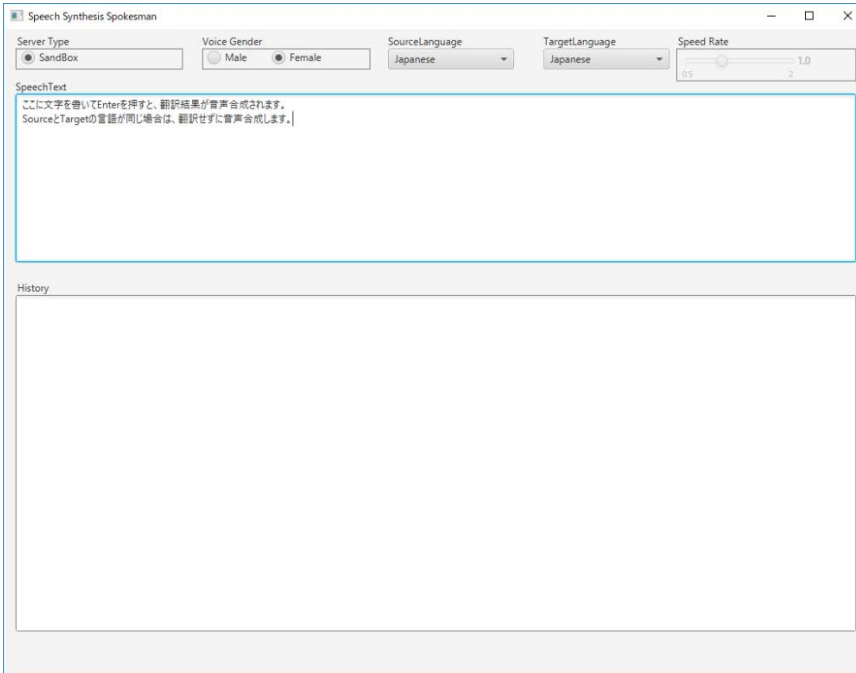
【書き起こし機能付き拡声器 (Android 版)】



スマートフォンのマイクに話した音が、外部スピーカーから拡声されて聞こえるという拡声器アプリです。

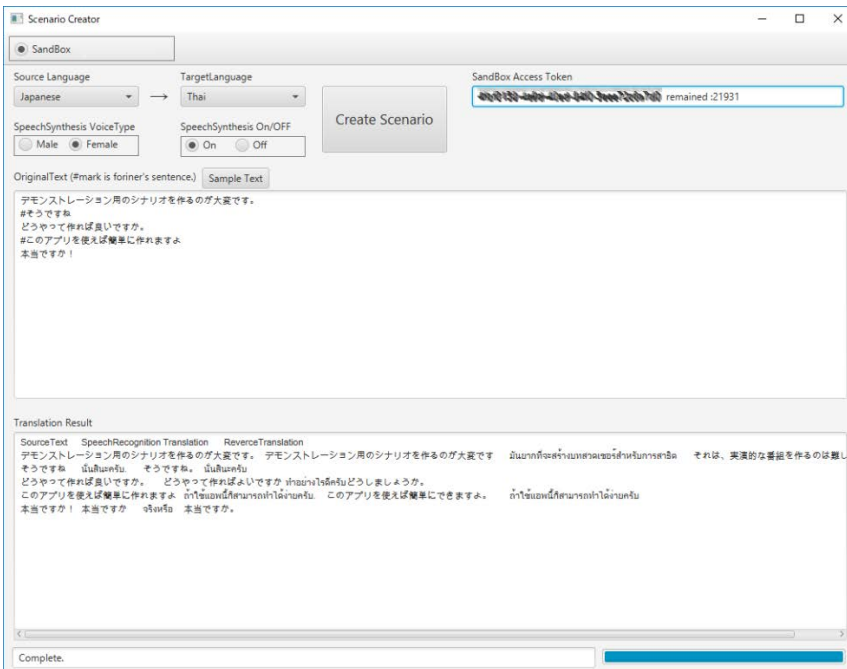
その時話した言葉は、音声認識して文章として書き起こします。

【翻訳機能付き音声合成アプリ(Windows版とAndroid版)】



文字を書いて **Enter** キーを押すと、音声合成してスピーカーから再生します。
 キーボードを打つのが速い人は、話すかのようにこれで会話も出来ます。翻訳する事も出来ます。

【デモンストレーション用シナリオ作成アプリ (Windows版)】



書いた文字を、音声合成→音声認識→翻訳→逆翻訳と実行して上手く音声認識と翻訳が正しく行われるかを確認できます。
 外国語は、文の先頭に#をつけると翻訳→音声合成→音声認識→翻訳→逆翻訳と実行して、外国語のチェックも出来ます。
 これで、デモンストレーションのシナリオを簡単に作れます。

●参考

NICTの発話区間検出モジュール

<https://github.com/ASTL-NICT/VAD>

●著者紹介

著者プロフィール

木村 法幸 (きむら のりゆき)

NICT ASTREC企画室所属

音声翻訳アプリVoiceTraの元開発プロジェクトリーダー

趣味：料理、OSのインストール

撮影：藤田智子

