AqKanji2Koe-A Android　マニュアル

株式会社 アクエスト
www.a-quest.com

概要

本文書は、言語処理ライブラリ AqKanji2Koe-A Android をアプリケーションに組み込んで使用するためのプログラミングの方法、注意点を示したものです。

AqKanji2Koe-A は、漢字かな混じり文のテキスト情報を AquesTalk 用のアクセント付きの音声記号列に変換するライブラリです。
このライブラリと音声合成ライブラリ AquesTalk（1/2/10）を使えば、動的に変化する様々なテキストから、リアルタイムに音声メッセージを生成できるようになります。

特長
・簡単に組み込み可能
　- テキスト文字列を入力すると音声記号列を返す、シンプルな API
・高速な変換処理
　- 約 20 万字/秒の高速な変換処理
・登録単語の追加が可能
　- ユーザ辞書ライブリを用いて、アプリで単語の追加や、編集、削除が可能
・高精度な読み・アクセント付与
　- 約 50 万語の単語辞書とアクセントルールにより、正確な読みとアクセントを生成

本ライブラリを使ったプログラミングに先立って動作を確認される場合は、弊社サイトのオンラインデモや、言語処理エンジンのページからデモアプリ AqK2kDemo(Windows 版)をダウンロードしてお試しください。漢字を含んだ文を音声記号列に変換や、ユーザ辞書を編集する機能を確認できます。

オンラインデモ	https://www.a-quest.com/demo/
デモアプリ ダウンロード	https://www.a-quest.com/products/aqkanji2koe.html

本ライブラリを使用するには開発ライセンスキーの設定が必要です。このライセンスキーを設定しない場合は評価版として動作し、以下の制限があります。

評価版の制限	**「ナ行、マ行」は、すべて「ヌ」と出力されます**

本ライブラリをアプリケーションに組み込んで使用する際には**使用ライセンス**、配布には**頒布ライセンス**が必要です。ライセンスの種類や購入方法は、弊社サイトのライセンスのページを参照してください。

仕様

ライブラリ形式	so 形式 共有ライブラリ（JNI 実装） 各種 ABI に対応　armeabi-v7a, arm64-v8a, x86, x86_64
対応 OS	Android 4.1 (API 16) 以降
入力データ形式	漢字かな混じり文テキスト（Java String）
出力データ形式	かな表記音声記号列（Java String）
関数 I/F	java static 関数呼び出し
ライブラリサイズ	約 200KByte
辞書サイズ	約 10MB（約50万語）
処理速度	約 20 万文字/秒（Windows XP, AthlonX2 2.7GHz 環境で計測）
ビルド環境	Android NDK r 22.0 NDK_TOOLCHAIN_VERSION := clang APP_PLATFORM := android-14 APP_STL := c++_static

ビルド・実行

必要なファイル

ビルド及び実行時に下記のファイルを使用します。

- libAqKanji2Koe.so ： AqKanji2Koe-A ライブラリの実体（ネイティブコード）
- AqKanji2Koe.java ： libAqKanji2Koe.so を Java からアクセスするためのラッパークラス
- aqdic.bin ： 辞書ファイル（システム辞書）
- aq_user.dic ： 辞書ファイル（ユーザ辞書）

「AqKanji2Koe-A Android」の実体はネイティブコードで書いてあり、JNI 経由で呼び出しています。そのため、使用する環境（ABI)に応じたライブラリを使用する必要があります。たとえば、使用端末が ARM CPU の場合は armeabi-v7a、arm64-v8a を、Intel CPU の場合は x86、x86_64 のライブラリを使用します。

任意のアプリケーションを開発するときは、libAqKanji2Koe.so と AqKanji2Koe.java ファイルを、アプリケーションのプロジェクトに追加します。Android Studio の環境であれば、プロジェクトディレクトリに、上のファイルを下記のようなディレクトリ構造上に(ディレクトリがなければ作成する)コピーすればよいでしょう。

Andorid では FAT-Binary(シングル APK)というもので、複数の ABI のライブラリを 1 つの apk に含めて各種動作環境に応じたアプリを作成できます。例えば、下記のように jniLibs に armeabi-v7a と x86 の 2 つのライブラリを配置してビルドするだけで ARM と Intel CPU の両方に対応したアプリが作成できます。

```
/<app/src/main>
    /java
        /<アプリのソース>
        /aqkanji2koe
            /AqKanji2Koe.java
    /jniLibs
        /armeabi-v7a
            /libAqKanji2Koe.so
        /x86        <必要に応じて
            /libAqKanji2Koe.so
    /res
    /assets
      .
      .
```

なお、呼び出しの関係は、次のようになります。

* アプリ -> AqKanji2Koe.java -> libAqKanji2Koe.so

実行時の辞書ファイルの配置

本ライブラリは、実行時に辞書ファイルにアクセスします。
辞書ファイルは本 SDK の aq_dic ディレクトリの下にある 3 つのファイルから構成されています。

- aqdic.bin：システム辞書ファイル
- aq_user.dic：ユーザ辞書ファイル（ユーザ固有の辞書 AqUsrDic ライブラリでカスタマイズ可能）
- CREDITS: クレジット記載ファイル

これら辞書ファイルは、Android OS の制限により apk の中(例えば assets）に置いた状態のままでは使用できないので、実行時にはローカルファイル領域に配置しなおす必要があります。

辞書ファイルの実装方法としては、次の 2 つが考えられます。

1．ZIP ファイルを展開して使用する方法
辞書ファイルを 1 つの ZIP ファイルに圧縮して、apk ファイルの assets ディレクトリに配置しておく。アプリの初回の起動時に、この ZIP ファイルを files フォルダの下に展開する（サンプルアプリはこの方法を用いています）。

2．辞書ファイルをダウンロードする方法
ネットワーク上のサーバーに辞書ファイルをおき、アプリの初回起動時にネットワーク経由で辞書ファイルをダウンロードしてローカルファイル領域に配置する。

アプリのデバッグ時には、辞書ファイルの配置の確認や削除が必要になります。これには、adb や Android Studio の Device File Explorer を利用すると良いでしょう。

ユーザ辞書ライブラリ　AqUsrDic

本 SDK は、ユーザ辞書ライブラリ AqUsrDic を含んでいます。これを用いると、エンドユーザがアプリの操作でユーザ辞書に任意の単語を登録・編集できます。

AqUsrDic は、AqKanji2Koe と同様に、java のラッパーと JNI ライブラリから構成されています。

付属のサンプルアプリで、ユーザ辞書ファイル aq_user.dic 中の単語を追加・編集・削除する機能を実装しています。AqUsrDic の使用方法はこのサンプルアプリを参照してください。

関数 API

AqKanji2Koe

convert　　package aqkanji2koe

説明	漢字かな交じりのテキストを音声記号列に変換
構文	static String **convert** (String *dirDic*, String *kanjiText*)
引数	
dirDic	辞書のディレクトリを指定。 例）"/data/data/<app dir>/files"
kanjiText	入力漢字かな混じり文テキスト文字列
戻り値	音声記号列文字列。最大文字数 4096byte。 処理エラーの時は"[ERR]<エラーメッセージ>" という文字列を返す。

setDevKey　　package aqkanji2koe

説明	開発ライセンスキーを設定。アプリ起動後、他の関数を呼び出す前に呼び出すことで、以降、製品版とし動作し、評価版の制限がなくなる。
構文	static int **setDevKey**(String *key*)
引数	
key	開発ライセンスキー文字列(半角英数)
戻り値	ライセンスキーが正しければ 0、正しくなければ 1 が返る。 不正なキーでも 0 を返す場合がある。このとき制限は解除されない。

AqUsrDic（ユーザ辞書ライブラリ）

AqUsrDic ライブラリは、個々の単語の登録・編集を行うインターフェースはありません。アプリでユーザ辞書を変更するときは、都度、CSV 形式の単語リストファイルを介して行います。

importCsv　　package aqusrdic

説明　CSV 形式の単語リストからユーザ辞書(aq_usr.dic)を生成
生成する aq_usr.dic と同じディレクトリに、システム辞書（aqdic.bin）が必要。
aq_usr.dic がすでにある場合は上書きする(Append しない)。

構文　static int **importCsv**(String *pathUserDic*, String *pathDicCsv*)

引数

pathUserDic　出力するユーザ辞書(aq_user.dic)ファイル名を指定
例）"/data/data/<app dir>/files/aq_user.dic"

pathDicCsv　CSV 単語リストファイルファイルを指定(入力データ)
例）"/data/data/<app dir>/files/aq_user.csv"

戻り値　0:正常終了　それ以外：エラーコード(詳細は getLastError()で取得)

exportCsv　　package aqusrdic

説明　ユーザ辞書(aq_usr.dic)から CSV 形式の単語リストを生成
aq_usr.dic と同じディレクトリに、システム辞書（aqdic.bin）が必要。

構文　static int **exportCsv**(String *pathUserDic*, String *pathDicCsv*)

引数

pathUserDic　ユーザ辞書(aq_user.dic)ファイルを指定
例）"/data/data/<app dir>/files/aq_user.dic"

pathDicCsv　出力する CSV 単語リストファイルファイル名を指定
例）"/data/data/<app dir>/files/aq_user.csv"

戻り値　0:正常終了　それ以外：エラーコード(詳細は getLastError()で取得)

checkEntry　　package aqusrdic

説明　単語表記、読みの書式や品詞コードが正しいかチェックする

構文　static int **checkEntry**(String *surface*, String *yomi*, int *posCode*)

引数	
surface	単語の表記の文字列を指定
yomi	単語の読み(アクセント付き音声記号列)の文字列を指定
posCode	単語の品詞コード(下記参照)を指定
戻り値	0:チェック OK それ以外:NG(詳細は getLastError()で取得)

getLastError package aqusrdic

説明	最後のエラーの詳細メッセージを返す
構文	static String **getLastError**()
引数	なし
戻り値	エラーメッセージ文字列(日本語)

エラーコード表

AqKanji2Koe ライブラリの関数が返すエラーコードの内容は、次の通りです。
AqUsrDic ライブラリに関しては、getLastError()関数で内容を取得してください。

値	内容
100	その他のエラー
101	関数呼び出し時の引数が NULL になっている。
104	初期化されていない(初期化ルーチンが呼ばれていない)
105	入力テキストが長すぎる
106	システム辞書データが指定されていない
107	変換できない文字コードが含まれている
200 番台	システム辞書(aqdic.bin)が不正
300 番台	ユーザ辞書(aq_user.dic)が不正

CSV 単語リストファイル

ユーザ辞書の編集で使用する CSV 単語リストファイルの例を下に示します。
1 列目：単語の表記を記述します。文字コードは UTF8 です。すべて全角で記述します。空白は指定できません。
2 列目：単語の読みを音声記号列で指定します(UTF8)。アクセントも ' 記号で指定できます。詳細は音声記号列仕様書の読み記号とアクセント記号の項を参照ください。区切記号は指定できません。
3 列目：単語の品詞コード(下記参照)を半角数値で指定します。

CSV 単語リストファイルの例

```
大江戸線,オーエドセン,5
ＧＤＧＤ,グ'ダグダ,1
天王洲,テンノーズ,7
Ｊアラート,ジェイアラ'ート,5
就活,シューカツ,1
魔理沙,マ'リサ,4
どや顔,ドヤガオ,1
アラフォー,アラフォー,0
```

品詞コード一覧

0	名詞
1	名詞(サ変)
2	人名
3	人名(姓)
4	人名(名)
5	固有名詞
6	固有名詞(組織)
7	固有名詞(地域)
8	固有名詞(国)
9	代名詞
10	代名詞(縮約)
11	名詞(副詞可能)
12	名詞(接続詞的)
13	名詞(形容動詞語幹)
14	名詞(ナイ形容詞語幹)
15	形容詞
16	副詞
17	副詞(助詞類接続)
18	接頭詞(名詞接続)
19	接頭詞(動詞接続)
20	接頭詞(数接続)
21	接頭詞(形容詞接続)
22	接続詞
23	連体詞
24	記号
25	記号(アルファベット)
26	感動詞
27	間投詞

サンプルアプリ

本 SDK には、サンプルアプリ AqK2kDemo の Android Studio 用プロジェクト一式が含まれています。このアプリは、以下の機能を実装しており、アプリ開発の参考にしてください。サンプルアプリのソースコードは、ご自由に改変してご使用いただけます。なお、評価版の制限があります。この制限を外すには、ソースコード中の setDevKey 関数の引数にライセンスキーの設定が必要です。

機能

- ZIP 化した辞書ファイルを初回起動時にローカルファイル領域に展開
- EditText からテキストを取得し AqKanji2Koe にて音声記号列に変換・出力
- 音声記号列を AquesTalk10 ライブラリにて音声データに変換・出力
- AqUsrDic ライブラリにて、ユーザ辞書の登録単語一覧の表示
- ユーザ辞書の登録単語の編集、単語の追加・削除

使用方法

so ライブラリや java ラッパークラスは、プロジェクト内に既にコピーしてあるので、Android Studio でサンプルアプリ AqK2kDemo を Open し、Run するだけです。
初回起動時に、assets/aq_dic.zip ファイルをユーザアプリ領域に展開します。
[変換]ボタンで漢字仮名混じり文のテキストを音声記号列に変換します。
音声合成エンジン AquesTalk10 ライブラリを含んでいるので、[再生]ボタンで音声を出力できます。
[辞書編集]ボタンでユーザ辞書ファイルを編集できます。

なお、独自のアプリを作成するときは、SDK 内の libAqKanji2Koe.so, libAqUsrDic.so, AqKanji2Koe.java, AqUsrDic.java, aqdic.bin, aq_user.dic ファイルを既定の場所に手動でコピーする必要があります。コピー場所等は、上記の「ビルド・実行」の章を参照ください。

アプリ開発ガイドライン

アプリケーションの開発(評価での使用を除く)は、以下のガイドラインに従ってください。

ライセンスキー

本ライブラリは、そのままでは評価版として制限付きで動作し、開発ライセンスキーを設定することで、製品版として動作し、評価版の制限がなくなります。

具体的には、AqKanji2Koe.setDevKey（）をアプリケーションの起動初期に一度呼び出します。引数には開発ライセンスキーを指定します。

開発ライセンスキーは、開発ライセンス購入時に発行されるライセンス証に記載されています。

CREDITS

本ライブラリは、BSD ライセンスに基づいてライセンスされている下記のオープンソースソフトウェアを使用しています。このライセンスの表示は SDK 付属の CREDITS ファイルを参照ください。また、本ライブラリを含んだアプリを配布する場合は、CREDITS ファイルまたはその内容が含まれるようにしてください。

- MARISA: Matching Algorithm with Recursively Implemented StorAge
- NAIST Japanese Dictionary：形態素解析用辞書

文書履歴

日付	版	変更箇所	更新内容	更新者
2011/01/17	1.0	新規作成		N.Y
2013/06/26	2.0		Ver.2 用に書き換え	N.Y
2016/03/11	2.1		Ver.2.1、IDE を AndroidStudio に変更	N.Y
2017/11/17	3.0		Ver.3 用に書き換え	N.Y
2020/11/14	4.1			N.Y