

# 안드로이드 OS 기반 한국어 TTS 서비스의 설계 및 구현

## Implementation of Korean TTS Service on Android OS

김태권\*, 김봉완\*, 최대림\*, 이용주\*\*

원광대학교 음성정보기술산업지원센터\*, 원광대학교 컴퓨터공학부\*\*

Tae-Guon Kim(yoplle@wku.ac.kr)\*, Bong-Wan Kim(tacanemo@wku.ac.kr)\*,  
Dae-Lim Choi(dlchoi@wku.ac.kr)\*, Yong-Ju Lee(yjlee@wku.ac.kr)\*\*

### 요약

국내에서 출시된 안드로이드 기반의 스마트폰은 한국어 TTS 엔진이 내장되어 있지 않고, 구글에서도 공식적인 한국어 TTS 기술 개발을 발표하지 않고 있는 상황이다. 따라서 안드로이드 스마트폰을 사용하는 어플리케이션 개발자 및 사용자들의 불편이 갈수록 심해져 가고 있다. 본 논문은 안드로이드 기반의 스마트폰에서 서비스할 수 있는 TTS시스템의 설계 및 구현에 대해 기술하였다. 신속·명료한 TTS를 위해 안드로이드 NDK를 이용하여 텍스트 전처리와 합성을 생성 라이브러리를 구현하였다. 또한, 자바의 스레드 기법과 스트림을 적용한 AudioTrack 클래스 객체를 사용하여 TTS 응답시간을 최소화 하였다. 구현된 한국어 TTS 서비스를 테스트하기 위해 수신된 문자메시지를 읽어주는 어플리케이션을 설계 및 개발하였다. 평가 결과, 임의의 문장에 대해 자연스러운 합성음을 생성하였으며, 실시간 청취가 가능하였다. 또한, 어플리케이션 개발자들은 구현된 한국어 TTS 서비스를 이용하여 음성을 통한 정보 전달을 손쉽게 적용할 수 있다. 본 논문에서 구현한 한국어 TTS 서비스는 기존 제한적 음성합성 방식의 어플리케이션의 단점을 개선하였으며, 음성을 통한 정보전달 어플리케이션 개발자 및 사용자들에게 사용성과 편의성을 제공할 수 있다.

■ 중심어 : | HMM 기반 음성합성 | 안드로이드 서비스 |

### Abstract

Though Android-based smart phones are being released in Korea, Korean TTS engine is not built on them and Google has not announced service or software developer's kit related to Korean TTS officially. Thus, application developers who want to include Korean TTS capability in their application have difficulties. In this paper, we design and implement Android OS-based Korean TTS system and service. For speed, text preprocessing and synthesis libraries are implemented using Android NDK. By using Java's thread mechanism and the AudioTrack class, the response time of TTS is minimized. For the test of implemented service, an application that reads incoming SMS is developed. The test shows that synthesized speech are generated in real-time for random sentences.

By using the implemented Korean TTS service, Android application developers can transmit information easily through voice. Korean TTS service proposed and implemented in this paper overcomes shortcomings of the existing restrictive synthesis methods and provides the benefit for application developers and users.

■ keyword : | HMM based Speech Synthesis | Android Service |

\* 본 연구는 2009학년도 원광대학교 교비 지원에 의해서 연구 되었습니다.

접수번호 : #111209-001

접수일자 : 2011년 12월 09일

심사완료일 : 2012년 01월 02일

교신저자 : 이용주, e-mail : yjlee@wku.ac.kr

## I. 서론

최근 무선통신기술의 발달로 인해 인터넷을 통한 생활 여러 분야의 정보 활용뿐만 아니라 업무 수행 역시 인터넷에 의존하는 사람들이 많아지고 있다[1]. 이러한 사람들의 인터넷 의존성은 PDA를 거쳐 오늘날 스마트폰을 등장하게 했다. 안드로이드 스마트폰은 국내에서 급속도로 성장하여 2010년 8월 초 100만, 1년만인 2011년 6월 말에는 900만 명 가입자 수를 기록함으로써 국내 스마트폰 시장의 70%를 점유하고 있으며 앞으로도 다양한 모델의 안드로이드 스마트폰이 출시되면서 전세계적으로 점점 더 빠르게 확산될 것으로 기대되고 있다[2][3]. 그러나 국내에서 출시한 안드로이드 스마트폰은 한국어 음성합성(TTS, Text-To-Speech) 엔진이 내장되어 있지 않고, 구글에서도 공식적인 한국어 TTS 기술 개발을 발표하지 않고 있는 상황이기에 안드로이드 어플리케이션을 개발하거나 이용하는 국내 사용자들의 불편은 계속되고 있다[4][5]. 이에 안드로이드 마켓과 국내 마켓에 음절편집합성 방식의 어플리케이션들이 속속 개발되어져 나오고는 있지만 이러한 어플리케이션들은 사용 가능한 음절 및 저장 공간의 한계를 가진다는 단점이 있다. 따라서 다양한 어플리케이션을 통한 효율적인 정보전달과 개발자 및 사용자 편의를 위한 한국어 TTS 서비스의 개발이 반드시 필요하다고 사료된다.

이에 본 논문에서는 기존의 음절편집합성 방식의 어플리케이션의 단점을 개선하며, 안드로이드 어플리케이션 사용자들에게 사용성과 편의성을 제공할 수 있도록 안드로이드 OS 기반의 한국어 TTS 서비스를 설계하고 구현하였다.

본 논문의 구성은 다음과 같다. 2장에서는 TTS 기술과 분석, HMM 기반 TTS에 대해 알아보며, 3장에서는 안드로이드 OS 기반의 TTS 시스템을 설계하고 이를 구현한다. 4장에서는 문자 메시지를 읽어주는 테스트 어플리케이션을 설계 및 개발하고, 5장에서는 제안된 한국어 TTS 서비스를 평가하는 과정을 통해 기존의 안드로이드 스마트폰의 문제점이 보완되었는지 검증한다. 마지막으로 6장에서는 본 논문에 기술한 연구에 대

한 결론과 향후 연구내용을 기술한다.

## II. 관련 연구

### 1. TTS 개요 및 분석

TTS란 글자, 문장, 숫자, 기호 등을 사람이 일반적으로 발성하는 음성으로 변환하는 것을 말한다. TTS 기술은 시각 의존적인 작업과 병행이 가능하다는 장점이 있다. 또한, 수시로 변할 수 있는 정보를 전달하기에 가장 쉽고 간편한 방법이며, 시각 장애인을 위한 매우 효율적인 정보 제공수단이기도 하다[6].

TTS는 실제 응용 방식에 따라 제한적 TTS 또는 자동 음성응답 시스템(Automatic Response System)과 무제한 TTS 시스템으로 구분된다. 최근 PC와 통신기술이 급속하게 발전함에 따라 실생활에 무제한 TTS가 적용되는 사례가 점차 늘게 되었다. PDA, 스마트폰 또한 빠른 속도로 보급되면서 단말기에 내장되어 TTS를 수행하는 다양한 서비스가 개발되고 있다. 그러나 이러한 단말기들은 PC나 서버에 비해 매우 제한된 하드웨어 환경에서 동작하므로 고품질의 합성음을 실시간으로 생성하기란 쉬운 일이 아니다.

스마트폰에서 TTS 기술을 사용하기 위한 요소로는 고품질 합성음을 생성하기 위한 음성 DB 용량과 신속한 TTS를 위한 CPU 성능을 꼽을 수 있다. 기존의 PC 및 서버 환경에서 사용할 수 있는 TTS 기술은 종류와 방식이 다양하지만, 안드로이드 기반의 스마트폰에 한국어 TTS 기술을 적용하는 것은 위와 같은 이유 때문에 국내에서 그 적용 사례를 찾을 수 없고, 공식 배포된 어플리케이션 또한 찾아보기 힘들다.

### 2. HMM 기반 TTS

최근 많이 연구되고 있는 HMM(Hidden Markov Model) 기반의 TTS 방법은 음성 신호를 특징 파라미터 형태로 변환하여 통계적으로 모델링해 각 음소의 대표값을 산출하여 합성음을 생성하는 방식으로, 기존의 코퍼스(Corpus) 방식의 TTS에 비해 시스템의 용량이 작아 개인 단말기에서도 TTS 시스템을 탑재시킬 수

있다[7]. 또한, 합성음의 의미과약도 전혀 문제가 없을 정도의 합성음 품질을 보여준다. HMM 기반의 TTS는 훈련과 합성 부분으로 구분된다. 아래 [그림 1]은 HMM 훈련 과정을 나타낸다.

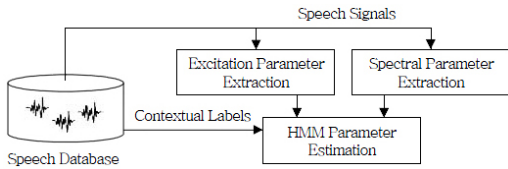


그림 1. HMM 훈련

HMM 훈련의 첫 번째 작업은 음성 DB로부터 MFCC(Mel-Frequency Cepstral Coefficient) 특징을 추출하는 것이다. MFCC 특징 추출 과정은 다음과 같은 순서로 이루어진다. 음성신호는 Anti-Aliasing 필터 및 A/D변환을 거쳐서 디지털 신호  $x(n)$ 로 변환된다. 디지털 음성신호는 고대역 통과 특성을 갖는 디지털 프리엠퍼시스 필터를 거치며 해밍 윈도우(Hamming Window)를 씌워 블록 단위의 프레임으로 나누어진다. 이후부터의 처리는 모두 프레임 단위로 이루어지는데, 프레임의 크기는 보통 20-30ms이며 프레임 이동은 10ms가 흔히 사용된다. 한 프레임의 음성신호는 FFT(Fast Fourier Transform)를 이용하여 주파수 영역으로 변환되며, 주파수 대역을 여러 개의 필터뱅크(Filter Bank)로 나누고 각 बैं크에서의 에너지를 구한다. 구해진 에너지에 로그를 취한 후 DCT(Discrete Cosine Transform)를 하면 최종적인 MFCC가 얻어진다. MFCC 계수는  $C_1 \sim C_{12}$ 까지의 12개를 사용하며 이와는 별도로 구한 프레임 로그 에너지가 추가적으로 사용되어 13차 벡터를 음성인식 분야에서 주로 사용한다. 필터뱅크를 통과한 신호를  $S[k]$ 라고 할 경우, 각각의 MFCC 특징벡터  $C_{[n]}$ 은 아래의 수식 1과 같이 표현된다. 여기서  $M$ 은 필터뱅크 개수이고,  $L$ 은 MFCC 차수를 의미한다.

$$C_{[n]} = \sum_{k=1}^M \log(S[k]) \cos \left[ (k-0.5) \frac{n\pi}{M} \right], n = 1, \dots, L \quad (1)$$

두 번째 과정에서는 텍스트분석기 등을 통해 미리 분석된 음성 DB의 레이블 정보를 이용하여 문맥중속 HMM 모델을 만든다.

아래 [그림 2]는 HMM 합성 과정을 나타낸다. 우선 주어진 텍스트를 문맥중속 레이블 열로 변환하고 레이블 열에 따라 대응하는 문맥중속 HMM 모델을 연결함으로써 문장 HMM 모델을 만든다. 그리고 문장 HMM에서 상태지속시간 모델을 기반으로 각 상태의 지속시간을 결정하고, HMM의 관측확률이 최대가 되는 스펙트럼과 특징 파라미터들을 결정한다. 마지막으로 생성된 스펙트럼과 특징 파라미터들로부터 MLSA(Mel Log Spectrum Approximation) 필터를 통해 합성음을 만들어 낸다.

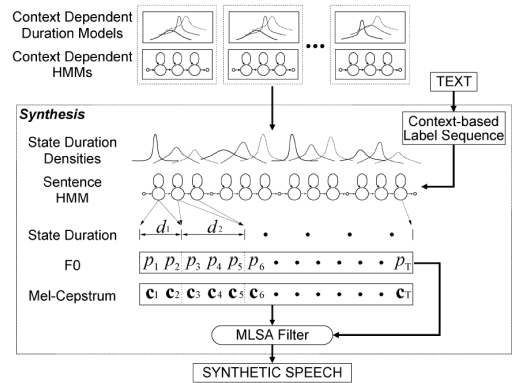


그림 2. HMM 합성

아래 수식 2는 HMM의 관측확률이 최대 값을 갖는 단어열을 찾는 수식을 나타낸다.

$$W' = \underset{W}{\operatorname{argmax}} P(W|O) \quad (2)$$

여기서  $P(W|O)$ 는 Bayesian 규칙에 의하여 다음과 같이 음향모델 확률  $P(O|\lambda)$ , 발음모델 확률  $P(\lambda|W)$ 과 언어모델 확률  $P(W)$ 의 곱으로 분리된다.

$$\begin{aligned} P(W|O) &= \frac{P(O|W)P(W)}{P(O)} \propto P(O|W)P(W) \quad (3) \\ &= P(O|\lambda)P(\lambda|W)P(W) \end{aligned}$$

음향모델은 HMM으로 표현되고, 발음모델은 발음사전으로 주어지며, 언어모델은 n-1차 Markov chain인 n-gram이 사용된다. 비터비 알고리즘에 의한 음성합성은 모든 가능한 상태열 중에서 최고의 확률을 나타내는 경로를 찾는 것이다.  $X_\lambda$ 가 음소열  $\lambda$ 에 대한 상태열의 집합을 의미한다면 최종 확률은 아래 수식 4를 통해 구할 수 있다.

$$W' = \underset{W}{\operatorname{argmax}} P(W) \max_{X \in X_\lambda} P(X)P(O|X) \quad (4)$$

### III. 안드로이드 OS 기반의 한국어 TTS 서비스 설계 및 구현

#### 1. HMM 훈련을 위한 환경

HMM 문맥중속 음향모델을 만들기 위해 원광대학교 음성정보기술산업지원센터에서 제작한 합성용 음성 DB를 사용하였다[8]. 훈련에 사용된 문장 수는 1,000개이며, 음성신호는 16kHz 샘플링에 16bits로 양자화되었다. 벨 캡스트림과 로그 기본주파수( $\log F0$ ), 그리고 이들의 차분값(Delta coefficients), 차분-차분값(Acceleration coefficients) 등 75개의 값을 산출하여 특징벡터의 스펙트럼 파라미터로 사용하였다. HMM에 의한 음성모델링에서의 음성 신호는 단순 좌우 천이(Left-to-Right topology)만을 갖는 HMM으로 모델링되며, 각 상태에서는 특징벡터가 출력된다. 훈련에 사용한 음소 모델은 [표 1]과 같다.

표 1. 훈련에 사용한 음소 모델

자음	ㄱ	ㄲ	ㅋ	ㄴ	ㄷ	ㄸ	ㅌ	ㄹ		ㅁ
	g	G	k	n	d	D	t	l	r	m
	ㅂ	ㅃ	ㅍ	ㅅ	ㅆ	ㅇ	ㅈ	ㅊ	ㅅ	ㅎ
	b	B	p	s	S	ŋ	j	J	c	h
모음	ㅏ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ
	a	ya	v	yv	o	yo	u	yu	U	i
	ㅘ	ㅙ	ㅚ	ㅜ	ㅝ	ㅞ				
	Ui	ye	wa	wv	we	wi	wE			

#### 2. 안드로이드 OS 기반의 한국어 TTS 서비스 제안

본 논문에서 제안하는 한국어 TTS 시스템의 구조는 [그림 3]과 같이 TTS 요청을 수행하는 어플리케이션들과 이러한 요청을 받아 텍스트 정보를 음성으로 합성해 재생시켜 주는 서비스로 이루어진다.

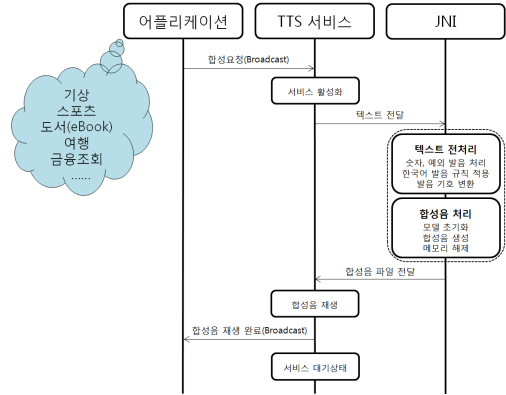


그림 3. 제안한 TTS 시스템 구조

#### 3. 한국어 TTS 라이브러리 구현

안드로이드 어플리케이션은 JNI(Java Native Interface)와 안드로이드 NDK (Native Development Kit)를 통해 자바가 아닌 C, C++언어로 만들어진 함수 및 라이브러리와도 상호 작용할 수 있다. 이를 통해 자바로 실행하기에 처리속도가 느릴 수 있는 부분이나 함수의 속도성능을 향상시킬 수 있다[9]. 또한, 이러한 개발 툴은 안드로이드 개발자 웹페이지에서 무료로 배포되고 있다. 따라서 신속한 TTS 처리를 위해 아래 [그림 4]의 순서에 따라 안드로이드 NDK를 이용하여 두 개의 라이브러리를 구현하였다.

##### 3.1 텍스트 전처리 라이브러리

음성 합성을 위한 텍스트의 전처리 성능의 고도화를 위해서는 한국어 형태소 해석기와 대규모의 사전이 필요하나 본 논문에서는 안드로이드 OS 단말기에서의 동작 속도와 저장 공간을 고려하여 형태소 해석이 필요 없는 한국어 읽기 규칙만을 적용하였다[10]. 본 논문에서 구현된 텍스트 전처리 라이브러리에 적용된 규칙은

총 21개이며, 읽기 규칙에 의해 해결되지 않는 예외 발음은 예외 발음 사전을 이용하여 해결하였다. 본 논문에서 사용된 예외 발음은 약 5,400개 정도이다. 따라서 본 논문에서 사용된 텍스트 전처리기의 성능은 형태소 해석기를 사용하는 서버 기반의 전처리기에 비하여 제한된 성능을 갖는다.

### 3.2 합성음 생성 라이브러리

오픈 소스인 HTS 엔진을 이용하여 합성음 생성 라이브러리 모듈을 구현하였다[11]. 입력된 문장에 대하여 HMM 훈련 과정을 통해 산출된 문맥중속 HMM 모델들의 관측 확률값이 최대가 되는 스펙트럼과 특징 파라미터, 상태지속시간 모델들을 결정하고 연결하여 MLSA 필터를 통해 합성음을 만들어 낸다.

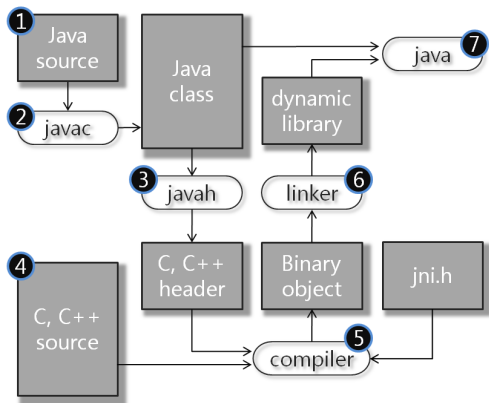


그림 4. JNI 및 NDK를 이용한 라이브러리 구현 절차

### 4. 서비스와 어플리케이션간의 상호 연동

한국어 TTS 서비스는 임의의 어플리케이션으로부터 TTS 요청(브로드캐스트, Broadcast)이 수신되면 활성화되도록 구현했다. 이 요청을 수행할 객체는 [그림 5]와 같이 4가지 정보를 설정해 주면 된다.

어플리케이션의 요청은 TTS 서비스의 리시버가 수신하며, TTS 요청이 수신되면 텍스트 전처리부와 합성음 생성부를 거쳐 합성음을 출력하고 해당 어플리케이션에게 합성음 재생 완료 통보를 보낸다. 한국어 TTS 서비스 구현을 위한 개발 환경은 [표 2]와 같다.

```
Intent intent = new Intent();
intent.setAction("Service_Action_Read");
intent.putExtra("TEXT", message);
intent.putExtra("GENDER", female);
intent.putExtra("SPEAKER", kor_01);
sendBroadcast(intent);
```

**Service\_Action\_Read** : TTS 서비스 요청 액션  
**message** : TTS를 수행할 String 타입의 문자열  
**female** : 화자 성별  
**kor\_01** : 화자 ID

그림 5. TTS 요청 코드

표 2. 개발 환경

O S	Windows Vista
Virtual Environment	Cywin(v2.693)
Java JDK	version 1.6.0_18
Android SDK	Android 2.3
Android NDK	ndk-r7-windows
Development Tools	Eclipse(Galileo)
	Visual Studio 2008
	Ultra Editor(v16.0)

### 5. 실시간 TTS 처리를 위한 설계 및 구현

사용자의 요청에 의해 생성되는 합성음의 실시간 재생을 위해서 한국어 TTS 서비스에 자바 스트림 버퍼(Java Stream Buffer)를 이용한 오디오 트랙(AudioTrack) 객체를 적용하였다. 또한, 합성음 생성과 합성음 재생의 병렬화를 위해 스레드(Java Thread) 기법을 적용하였다. 앞서 설명한 합성음 생성 라이브러리의 소스코드를 수정하여 네이티브 함수(Native Function)로부터 만들어진 합성음을 프레임 단위로 자바 스트림 버퍼에 누적시키며, 합성음 재생을 위한 조건이 만족되면 바로 합성음 청취가 가능하게 됨으로써 합성음 청취 소요시간을 단축시킬 수 있다. 합성음 생성과 재생을 모두 끝나치면 한국어 TTS 서비스는 사용자의 다음 명령을 수행하기 위해 최소 메모리를 유지한 채로 백그라운드에 상주하게 된다. 아래 [그림 6]은 합성음 생성 라이브러리의 작업 순서도를 나타낸다. 여기서 #Frames는 합성 문장의 프레임 개수를 말하며 FrameData는 80Bytes의 원시 합성음을 나타낸다. 또한, OverallBytes는 누적된 합성음의 크기이며 PlayOnce는 합성음이 재

생되었는지 체크하기 위한 변수로 사용되었으며 BufferSize는 재생을 위한 최소 버퍼 크기를 나타낸다.

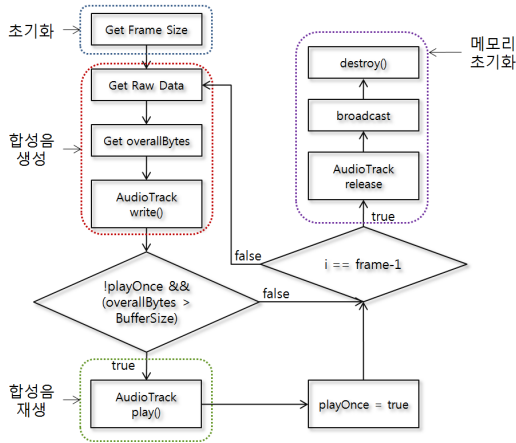


그림 6. 합성음 생성 라이브러리 작업 순서도

#### IV. 테스트 어플리케이션의 설계 및 구현

##### 1. 테스트 어플리케이션의 설계

본 논문에서 구현된 한국어 TTS 서비스를 테스트하기 위해 문자 메시지를 읽어주는 어플리케이션(SMS Reader)을 설계하였다. 이 어플리케이션은 사용자에게 문자 메시지가 수신되면 자동으로 활성화되어 실행할 수 있게 설계하였다. 이 어플리케이션의 리시버 클래스는 문자 메시지의 수신을 감지하고, 문자 메시지 발신자, 전송 시각, 메시지 내용을 얻어와 SMS Reader 메인 클래스에 이 값들을 넘겨주며, 구현된 한국어 TTS 서비스에 TTS를 요청하는 역할을 한다.

##### 2. 테스트 어플리케이션의 구현

제안된 SMS Reader 어플리케이션의 레이아웃은 아래 [그림 7]과 같이 문자 읽기, 합성음 생성, 소개, 설정 탭으로 구성하였다. 문자 읽기 화면은 사용자에게 수신된 문자 메시지의 수신시간 정보 및 발신자, 문자 메시지 내용을 보여준다. “문자 다시읽기”와 “문자 저장하기” 버튼을 통해 합성음을 재생 및 저장할 수 있다. 합

성을 생성 화면에서는 사용자가 입력한 텍스트 정보를 한국어 TTS 서비스를 이용하여 합성음으로 재생하거나 저장할 수 있다. 소개 화면에서는 SMS Reader 어플리케이션에 대한 소개와 사용법에 대한 간략한 정보가 담겨 있으며, 설정 화면은 두 개의 체크 박스와 두 개의 버튼으로 구현 되었는데, 문자 메시지가 수신되었을 때 체크박스 옵션에 따라 문자 메시지 내용을 바로 읽어주게 하거나, 문자 메시지가 도착했을 때 메시지 내용을 안드로이드 스마트폰 액정 화면에 바로 보여주게 할 수 있는 기능을 수행한다.



그림 7. SMS Reader 어플리케이션 레이아웃

#### V. 평가 및 고찰

삼성전자에서 출시한 “갤럭시 S” 단말기에서 80Bytes

의 문자 정보를 합성음으로 생성한 결과 약 2초 만에 합성음 청취가 가능하였다. 테스트를 수행한 단말기의 제원을 고려했을 때 개발 초기의 우려와는 다르게 신속한 TTS 처리가 가능했음을 알 수 있었다. 또한, 수신된 문자 메시지 텍스트 정보는 한국어 TTS 서비스에 내장된 텍스트 전처리 라이브러리를 통해 듣기 자연스러운 합성음으로 재생되었다. 아래 [그림 8]은 텍스트 전처리 라이브러리를 통해 입력된 문자를 통해 레이블을 생성하는 과정을 나타낸다.

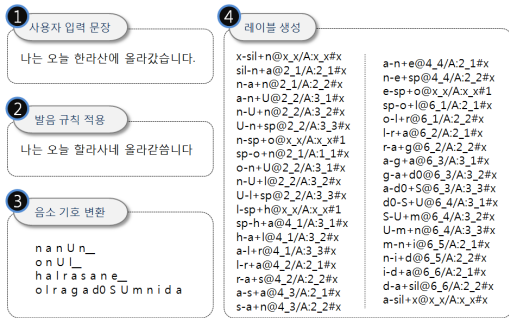


그림 8. 텍스트 전처리 라이브러리 작업 과정

네 개의 인수(TTS 요청 식별자, 합성할 텍스트, 화자 성별, 화자 ID)를 갖는 인텐트 객체를 선언하여 이를 브로드캐스트하면 원하는 문장을 실시간으로 합성음을 재생시켜 주기 때문에 음성을 통한 정보전달 어플리케이션을 개발하고자 하는 개발자들은 본 논문에서 구현된 한국어 TTS 서비스를 이용하면 큰 어려움 없이 이를 구현할 수 있다.

구현된 한국어 TTS 서비스는 HMM 훈련을 마친 파라미터를 저장하기 위한 5MB 정도의 저장 공간만으로 텍스트의 크기나 내용에 상관없이 무제한 합성이 가능하기 때문에 기존 편집합성 방식의 어플리케이션들의 단점을 여러 부분 보완할 수 있다.

안드로이드 스마트폰 기반의 다양한 어플리케이션(기상, 증권, 뉴스, 이메일, eBook, 지식, 쇼핑 등)에 구현된 한국어 TTS 서비스를 적용한다면 더욱 효율적이고 사용자 편의적인 정보 전달이 가능할 것이다.

## VI. 결론 및 향후 연구

기존의 안드로이드 OS 기반의 스마트폰은 한국어 TTS 엔진이 내장되어 있지 않아 어플리케이션 개발자 및 사용자들의 불편을 초래했다.

본 논문에서는 안드로이드 OS 기반의 TTS 시스템을 제안하고 한국어 TTS 서비스를 구현하였다. 구현된 한국어 TTS 서비스를 테스트하기 위해 SMS Reader 어플리케이션을 설계 및 개발, 구현하여 제안된 한국어 TTS 서비스의 효용을 검증하였다.

구현된 한국어 TTS 서비스의 텍스트 전처리 라이브러리는 한국어 발음 규칙에 따라 합성음을 듣는 사용자가 자연스럽게 이해하기 쉽게끔 입력된 텍스트 정보의 음소 기호 처리가 이루어지도록 구현하였다. 합성음 생성부에서는 신속한 무제한 합성이 가능하도록 구현하기 위해 HMM 기반 음성 합성 방식의 라이브러리를 생성하였으며, 훈련된 HMM 파라미터들을 사용하였다. 또한, 합성음의 생성과 재생을 동시에 수행하기 위해 스트림이 적용된 AudioTrack 클래스 객체와 자바의 스레드 기법을 사용하여 사용자의 TTS 요청에 대한 응답시간을 최소화하였다.

언제 어디서든 TTS 요청을 수행할 수 있는 서비스로 탑재되었기 때문에 안드로이드 스마트폰 어플리케이션 개발자 및 사용자들에게 생활의 유용한 정보를 음성을 통해 쉽고 빠르고 편리하게 전달할 수 있도록 하였다. 따라서 기존의 안드로이드 마켓과 국내 오픈마켓에 배포된 음절편집합성 방식 어플리케이션들의 여러 단점이 보완 및 개선되었다고 판단된다.

향후 연구 과제로는 텍스트 전처리 수행 시 비문법적 표현에 대한 예외처리 등을 보완하여 텍스트 전처리 라이브러리의 성능을 개선하는 것과, 합성음 품질 개선 및 목소리의 추가이다.

구현된 한국어 TTS 서비스를 통해 안드로이드 OS 기반의 스마트폰 어플리케이션 개발자들은 음성을 통한 정보전달 어플리케이션의 개발을 쉽고 편리하게 구현할 수 있으리라고 판단된다.



참고문헌

- [1] 대한상공회의소, *직장인의 인터넷 활용실태 보고서*, 2006.
- [2] <http://www.seoul.co.kr/news/newsView.php?id=20110623019012>
- [3] 고석훈, “안드로이드 플랫폼 동향”, 한국콘텐츠학회지, 제8권, 제2호, pp.11-122, 2010.
- [4] 한국소비자원 정보교육국 소비자 정보팀, *스마트폰 이용 만족도 비교정보 조사 결과*, 2010.
- [5] 한국정보통신기술협회(TTA) 웹 표준화 프로젝트 그룹(PG605), *모바일 접근성*, 2010.
- [6] 김종진, 김정세, 김상훈, 박준, *내장형 TTS 기술 동향 및 사례*, 전자통신동향분석, 제23권, 제1호, 2008.
- [7] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, “Simultaneous Modeling of Spectrum, Pitch and Duration in HMM-Based Speech Synthesis,” Proc. of EUROSPEECH, Vol.5, pp.2347-2350, 1999.
- [8] <http://www.sitec.or.kr>
- [9] <http://developer.android.com>
- [10] 국립국어원, *표준어 규정*, 문교부 고시 제88-2호, 1988.
- [11] <http://hts.sp.nitech.ac.jp>

저자소개

김태권(Tae-Guon Kim)

정회원



- 2009년 2월 : 원광대학교 컴퓨터 공학과(공학사)
- 2011년 2월 : 원광대학교 컴퓨터 공학과(공학석사)
- 2011년 3월 ~ 현재 : 원광대학교 음성정보기술산업지원센터

연구원

<관심분야> : 음성합성, 안드로이드

김봉완(Bong-Wan Kim)

정회원



- 1995년 2월 : 원광대학교 컴퓨터공학과(공학사)
  - 1997년 2월 : 원광대학교 컴퓨터공학과(공학석사)
  - 2002년 2월 : 원광대학교 컴퓨터공학과(공학박사)
  - 2001년 5월 ~ 현재 : 원광대학교 음성정보기술산업지원센터 연구 기획실장
- <관심분야> : 음성인식, 오디오 분류

최대림(Dae-Lim Choi)

정회원



- 2000년 2월 : 전북대학교 전자공학과 전자공학(공학사)
  - 2002년 2월 : 전북대학교 전자공학과 전자공학(공학석사)
  - 2009년 8월 : 전북대학교 전자공학과 전자공학(공학박사)
  - 2002년 5월 ~ 현재 : 원광대학교 음성정보기술산업지원센터 책임 연구원
- <관심분야> : 음성인식, 음성합성

이용주(Yong-Ju Lee)

정회원



- 1976년 2월 : 고려대학교 전자공학과(공학사)
  - 1987년 8월 : 고려대학교 전자공학과(공학석사)
  - 1992년 8월 : 고려대학교 전자공학과(공학박사)
  - 1980년 8월 ~ 1994년 2월 : 한국전자통신연구원 책임 연구원
  - 1994년 3월 ~ 현재 : 원광대학교 공과대학 컴퓨터공학부 교수
- <관심분야> : 음성정보처리, 멀티미디어응용, HCI