Yooi: An Indonesian Short Message Dictation

1Suyanto, 2Jeffry Adityatama
1The faculty of Informatics, Telkom Institute of Technology, suy@ittelkom.ac.id
2The faculty of Informatics, Telkom Institute of Technology, jeffry.adityatama@yahoo.co.id

Abstract
This paper focuses on developing Yooi, a mobile application to dictate a short message for Android smartphones using Indonesian and three major local languages. It is developed using Pocketsphinx and Java. Testing to Indonesian and local languages shows that the word error rates are quite similar among those languages. Testing to three environments with different noise levels shows that Yooi is quite feasible for low noise environments, but it is not feasible for noisy environment since it uses rule-based grammar where a miss-recognized word affects the whole recognized sentence.

Keywords: Short Message Dictation, Indonesian, Local Language, Android Smartphones

1. Introduction

In Indonesia, there are 260 billions SMS transactions in 2001 [1]. The users write or reply an SMS while meeting, being in a public area, or even driving a car or a motorcycle. Hence, an important application for Indonesian users is writing and replying SMS using speech. This paper discusses developing such application called “Yooi”. The term of Yooi comes from an Indonesian slang word that means alright. Yooi is a new version of IndoVMS as described in [3]. It is developed using automatic speech recognition (ASR).

ASR can be applied in a smartphone using two different approaches, i.e. online system and offline system. For the first approach, the ASR is in a server and the users should connect to the server whenever they use the ASR. For the second one, the ASR is injected into a smartphone so that the users do not need an internet connection [3]. Yooi is designed to be an offline system. It is developed using Pocketsphinx and Java. Both are open sources. The Indonesian speech corpus used here is one that developed by Telkom research and development center (Telkom RDC) as described in [10]. The text corpus is developed by collecting SMS messages that are frequently used by users.

In ASR, there are three major components, i.e. feature extraction as preprocessor, acoustic model, and language model. Many researchers use Mel Frequency Cepstrum Coefficient (MFCC) as feature extraction since it contains both full simulation of the human auditory system and better recognition performance as well as anti-noise capacity [4], [7], [9]. The ASR usually uses Hidden Markov Model (HMM) to develop acoustic models. The language model can be developed using either rule-based grammar or statistical language model that usually use trigram to predict each target word in the sequence given its two predecessors [5]. The first one is usually addressed for specific domain ASR [6] and the second one is for general domain. But, researchers in [8] uses voice search approach based on template matching. The voice search proceeds in two steps: an utterance is first converted into text, which is then used as a search query to match the most similar items of an index using information retrieval techniques [8].

In [3], IndoVMS has been investigated for both rule-based grammar and the 3-gram language model and also the response times of three different smartphone devices for the three text corpora. IndoVMS is reliable for rule-based grammar since it gives low word error rate (WER) and the response time is varying, between 1 and 9 seconds, based on the devices. Hence, Yooi is developed using rule-based grammar. This paper focuses on investigating Yooi to see the WER for Indonesian and three local languages and to see the WER for three environments with different noise levels.

2. Android and Sphinx

Android is an operating system developed for mobile devices such as smartphones or tablets. Android is based on kernel linux 2.6 which is modified for mobile devices. Android is an open source. The software development kit (SDK) for Android could be downloaded for free from
Android applications could be implemented using Java, C, HTML + CSS, etc. But, the most popular one is Java.

Android allows developers to develop an innovative application. The developers have full access to the APIs framework that is also used by core applications. Android architecture is developed so flexible that the components are reusable. Some different applications could access a component and change it, such as add contact list, SMS, or set reminder alarm.

All applications are basically set of services and systems that consist of: 1) View System, to develop user interface such as button, list, grid, text boxes, etc.; 2) Content Provider, to allow application to access data from other application (e.g. contact), or to share data to other application; 3) Resource Manager, to provide access to non-code resources such as graphics and files layout; 4) Notification Manager, to allow application to make a notification in the notification bar; and 5) Activity Manager, to manage activity life cycle of an application.

Sphinx is an open source toolkit for speech recognition. It is developed by Carnegie Mellon University (CMU), USA. There are some versions of Sphinx. One of them is Pocketsphinx, which is specially designed for embedded platforms or micro devices such as mobile phone. The latest Pocketsphinx is version 0.7, which is developed using C language. Pocketsphinx uses Hidden Markov Model (HMM) to develop an acoustic model since HMM gives high performance and so robust for speech recognition.

CMU Sphinx toolkit provides some tools to develop both acoustic model and language model that are customizable by developer. But, the common tools used by developer are sphinxbase-0.7, Pocketsphinx-0.7, CMU Sphinx4, cmuclmtk and sphinxtrain.

Sphinxbase provides some libraries that will be exploited by Pocketsphinx, CMU Sphinx and sphinxtrain. Both Pocketsphinx and CMU Sphinx function as speech recognition engine to convert speech input into text. The cmuclmtk and sphinxtrain are used to train language model and acoustic model.

3. Design

Yooi is specially designed for Indonesian people using national and local languages. An important issues should be considered is the accuracy in both silent and noisy environments. The tools used here are Pocketsphinx and Java. Pocketsphinx is developed for embedded platforms. It provides HMM to train acoustic model and also some functions to develop language model. Pocketsphinx decoder is quite fast since it uses semi continuous HMM.

3.1. Graphical user interfaces

The design of Yooi is similar to IndoVMS, but there are three additional menus, i.e. utterance speed setting, language setting, and abbreviation and symbol setting. Yooi is designed using simple graphical user interfaces (GUI), as illustrated by figure 1 to 7.

- Main menu;
- Find contact;
- Message dictation;
- Noise level setting;
- Utterance speed setting;
- Language setting; and
- Abbreviation and symbol setting.

The GUI of Yooi is designed to use Indonesian language since it is for smartphone users in Indonesia. In the main menu, two microphone symbols are used. The first one located on the right above is to find contact on the phonebook. The second one, on the bottom middle, is to dictate a message sentence. This GUI also provides a keyboard to find contact and to write a message. So, a user could use either speech or keyboard to interact with the Yooi.

The GUI for find contact is deliberately designed to produce some contacts as illustrated by figure 2. The user could select a desired contact from the list. This design has two reasons, i.e. the contact names are usually similar for some people and it is quite hard for a user to remember a contact name exactly. As illustrated by figure 2, a contact name “Alifya” have some similar names.
The GUI for message dictation, as illustrated by figure 3, provides a wide space for many message sentences and two buttons, the one on the left is to clear a last sentence and another one on the right is to send the message. This GUI gives a full control for user to either add or delete message sentences.

Figure 4 illustrates the GUI for setting the noise level, which is designed using slide bar to set the percentage of the noise level as the threshold of signal energy rate. The default of noise level is set to 5. It means signal segments below 15% (5 x 3%) of the signal energy rate is considered as noise. The maximum threshold is defined to 60%.

Figure 1. The GUI for main menu.

Figure 2. The GUI for find contact in the phonebook.
Alima Tasnim 08119999999
Alifya Fatimah 02270707070


List of found contacts.
List of resulted sentences.
Green $\rightarrow$ Recording
Gray $\rightarrow$ Done
Tap – Please speak when microphone is green (recording) – If there is no speech with energy greater than a defined threshold in a particular millisecond then the microphone is gray (done) – The Pocketsphinx will process the input speech in real time – Then it will output one sentence or blank if there is no result.

Figure 3. The GUI for message dictation.

Slide the bar to the left to decrease or to the right to increase the noise level (“tingkat kebisingan”). The Default is 7.

Tap to set the noise level
Tap to cancel (“batal”) the setting.

Figure 4. The GUI for setting “Tingkat kebisingan” or noise level.

The GUI for setting utterance speed, as you can see in figure 5, is also designed using slide bar to set slow utterance (1) to fast utterance (10). The GUI for setting language, as in figure 6, is designed using radio button so that user can select one of the provided languages: Indonesian, Batak, Javanese, or Sundanese. The GUI for setting abbreviation and symbol, as in figure 7, is designed using check box so that user can check or uncheck this feature.
3.2. Speech and text corpora

The Indonesian speech corpus used here is one that developed by Telkom research and development center (Telkom RDC) as described in [10]. The corpus actually contains 44,000 .wav files uttered by 400 speakers. But, 220 .wav files have very low quality and could not be used. So, the rest 43,780 .wav files uttered by 398 speakers, 199 males and 199 females, are used for training acoustic model. The speakers are range from teenager to elderly. They come from four major dialects in Indonesia, i.e. Batak, Betawi, Javanese and Sundanese.

The text corpus is developed by collecting messages that frequently used by users when they are in public areas or while driving, such as “Saya sedang di jalan (I’m on the way)”, “maaf, saya sedang nyetir (Sorry, I’m driving)”, “Baiklah, aku segera sampai sana (OK, I will be right there)” “Nanti aku telepon kamu (I will call you later)”, “Mohon tunggu 5 menit (Please wait 5 minutes)”, etc. Those Indonesia messages will be translated into Batak, Javanese, and Sundanese.

4. Results

Yooi is examined by two Javanese speakers uttering four different languages: Indonesian, Javanese, Sundanese, and Batak. It is examined using 100 sentences containing 401 to 442 words. The examination is performed in a room with level noise of around 15 db. The results are illustrated by figure 8. The lowest WER is achieved for Sundanese and the highest one is achieved for Javanese. In Sundanese, a sentence contains less words than other language. But in Javanese, a sentence contains more words than other language and each word contains more phonemes. However, in general the WER is quite similar for the four languages.

To see its performance in noisy environment, Yooi is investigated in three different environments, i.e. room (noise level ≥ 15 db), public area (noise level ≥ 5 and < 15 db), and street (noise level < 5 db). The results are shown by figure 9. The WER is very low, less than 3%, when it is tested in room with noise level of more than 15 db. The WER is more than 15%, if it is tested in street with noise level of less than 5 db. This result shows that Yooi is feasible for silent environment, but it is not feasible for noisy environment since it uses rule-based grammar where a miss-recognized word affects the whole recognized sentence.
5. Conclusions

The mobile application to dictate a short message called Yooi has similar word error rate for Indonesian and three major local languages. Yooi is quite feasible for low noise environments, but it is not feasible for noisy environment since it uses rule-based grammar where a miss-recognized word affects the whole recognized sentence. In the future, it will be better if Yooi use a voice search approach for the language model.
6. References


