Evaluation of Text-to-Speech Synthesizer for Indonesian Language Using Semantically Unpredictable Sentences Test: IndoTTS, eSpeak, and Google Translate TTS

Nur Aziza Azis, Rose Maulidiyatul Hikmah, Teresa Vania Tjahja and Anto Satriyo Nugroho
Center for Information and Communication Technology (PTIK) Agency for the Assessment and Application of Technology (BPPT), Jalan M.H. Thamrin No. 8, Jakarta 10340, Indonesia
Email: nurazizaazis@gmail.com, rose.maulidya@gmail.com, teresa.vania@gmail.com, asnugroho@ieee.org

***Abstract*—A text-to-speech (TTS) synthesizer aims to produce speech from the corresponding text input. For Indonesian language, IndoTTS, eSpeak, and text-to-speech on Google Translate are available and widely used among the Indonesian. In this study, we evaluated those systems and conducted semantically unpredictable sentences (SUS) test. SUS test results showed that the intelligibility of those systems needs further improvement.**

I. Introduction

A text-to-speech (TTS) synthesizer produces speech based on the corresponding text input. It has been utilized to provide easier means of communication and to improve accessibility for people with visual impairment to textual information. This language dependent system has been widely developed for various languages with continuous research to improve the quality of the produced speech. However, only a few TTS systems were developed for Indonesian language.

Among the existing TTS system, IndoTTS [1], eSpeak [2], and text-to-speech on Google Translate [3] are available and widely used for Indonesian language. This paper provides assessment of the three systems and evaluates their intelligibility based on semantically unpredictable sentences (SUS) test [4]. Also, this research aims to address problems that need to be fixed in TTS synthesizer for Indonesian language in order to encourage further researches in this field.

II. TTS System for Indonesian Language

A. Review on Text-to-Speech Synthesis

Text-to-speech is the production of speech by machines, by way of automatic phonetization of the sentence to utter [5]. TTS synthesizer is different from other talking machine such as cassette player or voice response systems. Cassette player plays sentences that have been previously recorded, while TTS synthesizer produces new sentences from the text input. It also differs from voice response system (for example machine for announcing arrival in train station). Voice response system simply concatenates recorded words from a limited vocabulary made for specific purposes, while the expected vocabulary for TTS system is all words from the focus language, which are impossible to be recorded entirely.

There are two main modules in the TTS synthesizer, namely Natural Language Processing (NLP) and Digital Signal Processing (DSP), as shown in Fig.1. Natural language processing module is responsible for conversion of text input into phonetic transcription and prosody information. Prosody information, which includes melody (intonation) and rhythm, is necessary to make the resulting speech sounds natural (not flat/robot-like). The DSP module then transforms the resulting phonetic transcription and prosody information into corresponding speech.

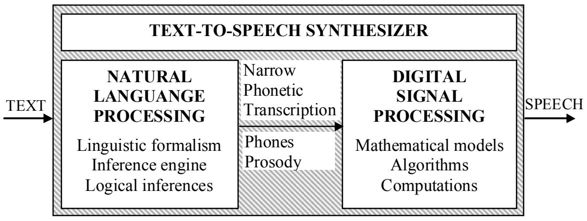

Fig. 1. General Functional Diagram of a Text-to-Speech System [5].

The NLP module mainly consists of text-analyzer, letter-to-sound, and prosody generator module which is depicted in Fig.2. Text-analyzer aims to give syntactic information to the subsequent modules. Text-analyzer itself comprises [5]:

1. Pre-processor: in this stage, all words in a sentence including numbers, abbreviations, and acronyms are stored and transformed into full-text form.
2. Morphological analyzer: aims to propose all

possible part of speech (POS) categories for each individual word.

3. Contextual analyzer: considers a word in its context and therefore makes it possible to reduce the list of possible part of speech categories by removing those do not match with the context.
4. Syntactic-prosodic parser: finds the text structure, defined as its organization into clause and phrase-like constituents, which is more closely related to its expected prosodic realization.

Meanwhile, letter-to-sound module is responsible to convert input letters into phonetic code that will be added with prosody information by the prosody generator module.

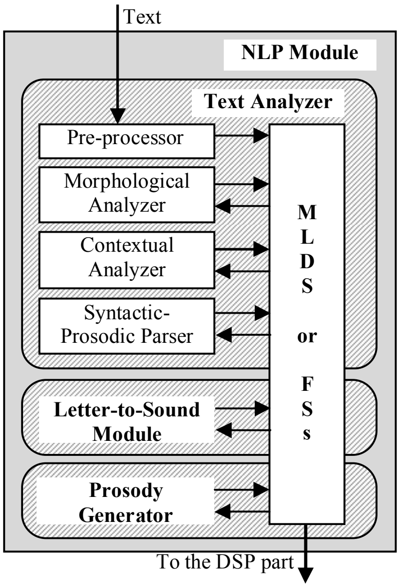

Fig. 2. The NLP Module of a Typical Text-to-Speech Conversion System [5].

There are two main strategies in transforming phonetic code into speech in the DSP part: synthesis by rule and synthesis by concatenation. Rule-based synthesizer creates rules that formally describe the influence of phonemes on one another. This synthesizer always appears in the form of Formant synthesizer [5]. On the other hand, concatenation-based synthesizer stores examples of phonetic transitions and co-articulations in speech units that will be concatenated.

Two criteria in evaluating TTS systems are intelligibility and naturalness. Klatt [6] in Lemmety [7] proposed additional criterion: suitability with application that will be used. For example, in a machine reader for visually impaired people, intelligibility is more important than naturalness.

Intelligibility evaluation can be made in several levels, such as phoneme, word, or sentence level. One of the tests on phoneme level intelligibility is cluster identification test (consonant-vowel-consonant (CVC) test or vowel-consonant-vowel (VCV) test). This test asks listener to insert non-sensible words (with CVC or VCV structure) in short carrier phrases, and only provides listeners with the overall list of words to be played [5]. On sentence level, semantically unpredictable sentences (SUS) test involves the use of syntactically acceptable but semantically anomalous sentences [4].

Problem area in speech synthesis is very wide [7]. In the pre-processing stage, one of the problems is how to handle symbols of punctuation that are ambiguous. As an example, period marks sentence end but it often appears in abbreviations as well, such as: "dr." for "dokter" (Eng: "doctor"), in Indonesian language. Other problems are how to pronounce numbers, abbreviations ("dsb." stands for "dan sebagainya" (Eng: "etc."), "kg" for "kilogram", "km" for "kilometer", etc.), and acronyms ("ABRI", "SMA", etc.). For Indonesian language, another problem is pronunciation of the ambiguous "e" letter. It can be pronounced *taling* /E/[1] as in the word "medan" (in English as in "bed") or *pepet* (schwa) /@/ as in the word "beras" (in English as in "collide"). Another problem is how to provide correct prosody and pronunciation analysis from a written text.

B. *IndoTTS*

IndoTTS is a TTS synthesizer for Indonesian language developed by Arry Akhmad Arman from Bandung Institute of Technology (ITB), Indonesia. In the NLP part, this system converts text to phoneme and adds prosody information using an intonation model for Indonesian language. Phonetic code and the produced prosody information then become input for MBROLA software as the DSP part. MBROLA uses concatenation-based synthesizer with diphone as concatenation unit. MBORLA diphone database for Indonesian language is also created by Arry Akhmad Arman. IndoTTS system configuration is shown in Fig. 3.

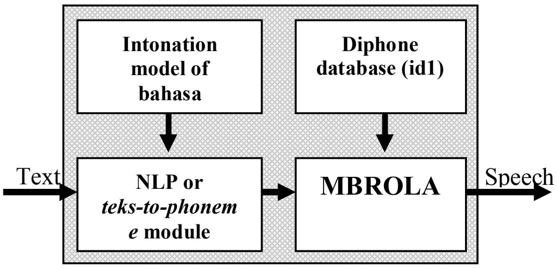

Fig. 3. IndoTTS System Configuration [1].

Prosody features have been implemented in this program through an intonation model for Indonesian language. This model uses Fujisaki's model approach and contour pitch theory by first analyzing the utterance to obtain prosodic curve of Indonesian

[1] The phoneme mnemonics are based on Kirshenbaum (http://www.kirshenbaum.net/IPA/ascii-ipa.pdf) that represent International Phonetic Alphabet in ASCII character.

language [8].

The pronunciation produced by IndoTTS sounds considerably natural with the prosodic features. Punctuation that is located inside sentences (such as comma ",") have been considered in generating appropriate prosody. However, this program does not provide the prosody differences caused by sentence-end punctuation (period ".", exclamation mark "!", or question mark "?").

The pronunciation is also intelligible. In most cases, this program pronounces "e" letter (whether "e" *pepet* or "e" *taling*) correctly. However in some cases, errors are still discovered. Some symbols have been correctly pronounced, such as dollar sign ("$"), ampersand ("&"), plus sign ("+"), and percent ("%"), while others have not been handled correctly like dashes ("-") that located inside a string. For example, "Tanggal 17-19 Juli" is pronounced "tanggal juli", where the string that contains the dash is omitted.

In handling abbreviations, abbreviation such as "Rp" (stands for "rupiah" (Eng: "rupiahs")) is pronounced correctly. For example, "Rp 7000" is pronounced "tujuh ribu rupiah". Other common abbreviations such as "dll", which stands for "dan lain-lain" (Eng: "etc.") is pronounced as their sequence of letters ("/dE El El/"). "kg", abbreviation for "kilogram", is pronounced as a word similar to "/k@g/". Abbreviations with period inside the string, e.g. "a.n" ("atas nama"), are not pronounced. Acronyms are spoken letter by letter although it can be pronounced as word (for example: "NATO" is pronounced /En A tE O/ instead of "/nAtO/"). There is an uneven case in pronouncing acronyms with complex letter like "MBROLA", the program read the first four letter only "/Em bE Er O/", remaining letters are omitted.

Also, the program is not yet equipped with morphological analyzer and contextual analyzer modules, hence ambiguous words in sentence have not been handled properly. There are also cases of different pronunciation between words in its root form with the same word after it gets affixes, while it should be the same (for example: "resep" is pronounced /r @ s E p/, but "resepnya" is pronounced /r @ s @ p n^ A/ instead of /r @ s E p n^ A/).

In pronouncing numbers, IndoTTS is capable of translating the numbers up to hundreds of thousands units. However the program does not always succeed in reading numbers with period separator (in Indonesian language, period is used to separate numbers). For example, "1000" is read correctly as "seribu" (Eng: "one thousand"), but "1.000" is only pronounced "dot". In pronouncing fractions, slash sign which is normally read as "per" is pronounced "garis miring" (e.g. "1/2" is pronounced "satu **garis miring** dua" (Eng: "one **slash** two") instead of "satu per dua". Decimal numbers are correctly spoken, e.g. 43,13 is pronounced "empat puluh tiga koma satu tiga" (Indonesian language uses comma to separate decimal numbers). Negative numbers are not pronounced, unless they are followed by a decimal (e.g. "-9,2" but only pronounced "koma dua"). Also, the program has not been equipped with roman numbers translation (e.g. roman number 4, "IV" is read "/I fE/").

C. eSpeak

eSpeak is an open source software text-to-speech synthesizer that is available in many languages, including Indonesian. In synthesizing speech, eSpeak uses Formant synthesis method. This allows many languages to be provided in a small size. The produced speech is clear and can be used at high speeds, but it is not as natural or smooth as larger synthesizers which are based on human speech recordings [2].

eSpeak for Indonesian language is still an experimental attempt, and when this paper is written, there has not been any adequate feedback from Indonesian native speaker [2]. The produced speech is not like speech produced by Indonesian native speaker, though it is clear.

This program requires a lot of improvements in pronouncing "e". Except the words that are excluded, "e" pronunciation always follows the given rules. For instance, the letter 'e', which is located on the first syllable of the word of two syllables, is always pronounced *taling* /E/, while the letter 'e', which is located on the first syllable of the word of three syllables, is pronounced *pepet* /@/. Like IndoTTS, in eSpeak is also found cases of the pronunciations of 'e' that are different when they are located in a root form word with the same word after gets affixes. For example, the word "bersih" (Eng: "clean") is pronounced "/bErsIh/", while "bersihkan" is pronounced "/b@rsIhkAn/".

Symbols that appear often are correctly pronounced, but several symbols are pronounced with additional word of "tanda" (Eng: "sign") in front of the symbol (e.g. "Romeo & Juliet" is read "/r O m E O/ **tanda** dan /dZ U l I t/"), which gives an odd impression. There are also pronunciations of symbol that are transferred to English, e.g. the symbol "½" is pronounced "half" with subtle English accent. Meanwhile, if the symbol is written as three-character fraction "1/2", it will be read as "satu **garis miring** dua" (Eng: "one **slash** two") just like indoTTS. Dash symbol ("-") in most cases are not pronounced.

eSpeak is able to pronounced numbers up to billion units. There is a difference when pronouncing numbers that are written using numerical characters with numbers written in text. For example, "4" is pronounced "/@mpAt/" with the *pepet* 'e', while "empat" is pronounced "/Empat/" with the *taling* 'e', where the correct pronunciation is with *pepet* 'e'. The program pronounces negative number, with the minus sign pronounced "tanda kurang" (e.g. "-2" is pronounced "**tanda kurang** dua"). Decimal number is pronounced correctly (e.g. "43,13" is pronounced "empat puluh tiga koma satu tiga"). In addition, the program is also able to utter roman numbers that consist of two or more character (e.g. roman number "IV" for "4" is correctly read as "/@mpAt/"). One character roman numbers (e.g. "X", "V") is read as letter ("/Eks/", "/fE/").

In dealing with abbreviations, eSpeak simply spells the letters or pronounces it as a word ("Rp" is read "/Er pE/", "Prof." is read "/prOf/"). This is also the case for acronyms.

eSpeak has a fine level of naturalness. This program considers punctuation in sentence to produce appropriate prosody, such as pause at the comma, or a rising intonation on interrogative sentence. According to the developers, in terms of naturalness, this program still requires improvement in the stress placement on words.

D. Google Translate TTS

Google Translate also provides text-to-speech feature. Google Translate uses eSpeak to convert text to speech [3]. However there are slight differences between pronunciations of eSpeak and Google Translate.

Intelligibility of the speech is basically the same as eSpeak. There is a difference in pronouncing 'e' letter that is located at the end of a word. Google Translate TTS pronounces it with *taling* 'e' "/E/", while eSpeak produces *pepet* 'e' "/@/". For example the word "sore" (Eng: "afternoon") read by Google Translate as "/sOrE/", and "/sOr@/" by eSpeak, where the correct pronunciation is "/sOrE/". Also, for the pronunciation of words with two syllables where the first syllable is "tel" such as "telur" (Eng: "egg"), eSpeak pronounces it "/tElUr/", while Google Translate produces "/tElUr/" (the correct pronunciation is "/t@lUr/"). The remaining pronunciation of numbers, symbols, acronyms, and abbreviations in Google Translate and eSpeak are identical.

In converting from text to speech, Google Translate limits the length of the sentence up to 100 characters. When the input exceeded the limit, the TTS feature is not available.

III. TTS Evaluation Using Semantically Unpredictable Sentences (SUS) Test

Following the qualitative evaluation of TTS as mentioned in the previous section, we further conducted a quantitative evaluation to observe intelligibility of the systems using semantically unpredictable sentences (SUS) test. With this method listeners are asked to write sentences that are syntactically correct, but semantically anomalous (e.g. "Angin menghindari kasus yang berdiri", Eng: "The wind avoids the case that stands."). Anomalous meaning of sentences is intended to minimize contextual cues provided by meaningful sentences that could affect intelligibility test score.

The SUS test is conducted following the scheme discussed in [4]. It involved five syntactic structures with some adaptations for Indonesian language. Syntactic structures in this test are as follows:

Structure 1

a) Subject + intransitive verb + adverbial: **intransitive** structure

b) Noun + verb (intr.) + preposition + noun

e.g. : "Bulan tiba di hati."

(Eng: "The moon arrives in the heart.")

Structure 2

a) Subject + verb + direct object: **transitive** structure

b) Noun + adjective + verb (trans.) + noun

e.g. : "Rumah tajam mengatur kapal."

(Eng: "The sharp house organizes the ship."

Structure 3

a) Verb + direct object: **imperative** structure

b) Verb (trans. no-affixed) + noun + conjuction + noun

e.g. : "Sebut orang dan lomba!"

(Eng: "Call the person and the contest!")

Structure 4

a) Question word + subject + verb + direct object: **interrogative** structure

b) Question word + noun + verb (trans.) + noun

e.g. : "Kapan sungai melihat contoh?"

(Eng: "When does the river see the sample?"

Structure 5

a) Subject + verb + complex object: **relative** structure

b) Noun + verb (trans.) + noun + relative pronoun + verb (intr.)

e.g. : "Angin menghindari kasus yang berdiri."

(Eng: "The wind avoids the case that stands.")

For each syntactic structure, there were seven sentences, resulting in 35 test sentences for every TTS system. The sentences were randomly generated using fixed vocabulary. Words in the vocabulary are the most frequent word taken from SIDoBI Indonesian language corpus [9]. Two additional sentences were prepared for each structure to be used in the training session prior to testing.

Sentences to be presented were recorded before, with 15 seconds pauses for every sentence. From 105 total sentences (7 sentences for each 5 syntactic structure for 3 synthesizer), each sentence was presented entirely in random order. Listeners were asked to write down what they heard.

The test was applied to ten native listeners. The correct answer from each person is summed, so there were a total of 350 sentences for each synthesizer. The word count was 147 for one testing set, so for ten listeners there were a total of 1470 words for each synthesizer.

The simplest and the fastest way to score results is to only take into account the sentences that are entirely correct [4]. All words (including conjunction,

TABLE I
PERCENTAGE OF CORRECT SENTENCES ON SUS TEST

	eSpeak	IndoTTS	Google Translate TTS
Percentage of Correct Sentence	20.29 %	24.00 %	8.00 %

preposition, and relative pronoun) must be correct and were in the right position. Evaluation results for the correct sentence are shown in Table I.

Among these three systems, IndoTTS produced the highest score of 24.00%, followed by eSpeak with 20.29%, and Google Translate TTS with 8.00%. It can be concluded that the correct sentence percentages for these systems were considerably low.

The low percentage shows that intelligibility of these TTS synthesizer for Indonesian language still needs much improvement. Several aspects that caused listeners' misperception are: pronunciation of the 'e' letter (*pepet* or *taling*) that is still incorrect in some cases, pronunciation of consonants in similar words (as in "pola" (Eng: "pattern") and "bola" (Eng: ball)), as well as error in distinguishing between diphthong and non-diphthong vowel cluster (e.g. vowel cluster "**ai**" on "perm**ai**" read as diphthong "/p @ r m **aI**/" while it is "/b @ r m **A I** n/" in "berm**ai**n").

IndoTTS outperformed the other synthesizers. One of the possible reasons is its capability of distinguishing 'e', whether *pepet* or *taling,* is better than the other two. Another reason is IndoTTS voice is Indonesian native speaker's voice, while eSpeak and Google Translate TTS have foreign-accented voices, where the pronunciation of 'r' that is unclear due to the accent may reduce the intelligibility of the systems.

Google Translate TTS appears to have very low intelligibility score. Although Google Translate also uses eSpeak as its TTS engine, the intelligibility score is far enough different. This may occur because the speech rate for Indonesian language in eSpeak was adjusted to be slightly slower than the default in order to clarify the pronunciation, while on Google Translate the speech rate is set to default and could not be changed.

Calculation for correct words were also performed (Table II). Again, IndoTTS delivered the highest score of 62.38% followed by eSpeak with 60.95% and Google Translate TTS with 44.49%. At this word level, Google Translate TTS also produced the lowest score, yet the difference with eSpeak was not as far as scores in the sentence level.

In addition to the aspects that have been described, the test results are also influenced by technical things on the implementation such as the recording process, presentation to the listeners, and randomization process. However, the test was done as fair as possible.

TABLE II
PERCENTAGE OF CORRECT WORDS ON SUS TEST

	eSpeak	IndoTTS	Google Translate TTS
Percentage of Correct Sentence	60.95 %	62.38 %	44.49 %

IV. CONCLUSION

Evaluation of TTS system for Indonesian language was conducted for eSpeak, IndoTTS, and Google Translate TTS. Sentence level intelligibility test using semantically unpredictable sentences (SUS) shows the three TTS systems for Indonesian language still require improvements in their intelligibility. Several problems that were found include the mispronounced of 'e' letter (whether /E/ or /@/), lack of clarity in pronouncing consonants of similar words, as well as errors in distinguishing vowel cluster (whether diphthong or not diphthong).

REFERENCES

[1] http://indotts.melsa.net.id/ (Accessed : August 28, 2011)

[2] http://espeak.sourceforge.net/ (Accessed : August 28, 2011)

[3] http://translate.google.com/support/ (Accessed: August 28, 2011)

[4] C. Benoit, M. Grice, V. Hazan, "The SUS test: A method for the assessment of text-to-speech synthesis intelligibility using Semantically Unpredictable Sentences". *Speech Communication*, vol.18 pp. 381-392.

[5] T. Dutoit, *An Introduction to Text-to-Speech Synthesis*. Dordrecht: Kluwer Academic Publishers, 1997, pp. 13, 14, 63, 72, 179, 196.

[6] D. Klatt, "Review of text-to-speech conversion for English". *Journal of the Acoustical Society of America*, JASA vol. 82 (3), pp. 737-793.

[7] S. Lemmety, "Review of speech synthesis technology", M.S. thesis, Dept. Elec. and Comm. Eng., Helsinki University of Technology, 1999.

[8] A. A. Arman (2008) Text to Speech Bahasa Indonesia dan Perkembangan Teknologi Bahasa.[online] available: http://kupalima.files.wordpress.com/2008/08/pelbba19.ppt (Accessed : August 27, 2011)

[9] B. Prasetyo, T. Uliniansyah, O. Riandi, "SIDoBI: Indonesian language document summarization system," in *Proc. of International Conference on Rural Information and Communication Technology*, pp. 378-382, Bandung, Indonesia, 2009.