Design of a Text Markup System for Yorùbá Text-to-Speech Synthesis Applications

Ọdẹ́túnjí Àjàdí , ỌDẸ́JỌBÍ
Computer Science & Engineering Department
Faculty of Technology
Ọbáfẹ́mı Awólọ́wọ̀ University
Ilé-Ifẹ̀, Nigeria
oodejobi@oauife.edu.ng

ABSTRACT

The text intended as input into a text-to-speech (TTS) synthesis system can come from a number of digital sources, including: electronic emails, text-books, Internet webpages and newspapers. In order to generate the speech sound corresponding to the input text, a TTS system must extract the information relevant for computing speech signal and associated prosody. The ease with which this information can be extracted depends on a number of factors, including: the orthography of the target language, the domain of the text, and whether the content of the text is constrained by some writing rules or standards. Usually the information that accompanies such text are inadequate for computing the prosody of the corresponding text. A text markup system is then used to provide the missing information. This paper discusses how this task is achieved for the Standard Yorùbá language.

Keywords

text, spech synthesis, markup, text tagging

1. INTRODUCTION

The orthography of Standard Yorùbá (SY) has been described as *shallow*(Bird, 1998), implying that it is relatively easy to generate pronunciation from the text without the need for complicated syntactic analysis; such as part-of-speech analysis. For example, the pronunciations of the English words re*cord*(verb) and *re*cord (noun) differ because of their syntactic classes. This type of situation does not occur in SY. In the (SY) language orthography, diacritic marks are used to represent tones. This tonal information removes the ambiguity in the pronounication of a well written and properly accented SY text. The additional information provided by tonal marks is very important in the computation of intonation and prosody for a piece of text in Text to Speech Synthesis (TTS) system.

Despite the phonetic nature (i. e. closeness of the orthography to pronunciation) of the SY orthography, the information in the text is not enough for computing an accurate prosody in text to speech application (Ọdẹ́jọbí et al., 2008). The orthography of SY, and indeed any other language, is designed to provide enough information for a human reader. Paralinguistic aspects of expressions, such as emotion - which endows speech with its naturalness quality-cannot be reduced into writing but are normally guessed by human readers based on context. Other information, such as the structure and style for pronouncing the text are not explicitly represented in the orthography. There is, therefore, the need to augment the content of the text with additional information that will facilitate an unambiguous processing of prosodic data in an input text. In this paper we shall discuss the the design of a text markup system for adding additional prosody information into a SY text intended as input to a Text to Speech Synthesis (TTS) system.

1.1 The nature of written text

A written text can be described as an encoding of a speech utterance using symbols. The text encodes two important information: (i) the content and manner of speech, and (ii) how it should be spoken. The content of the speech, i.e. its written form, is defined by the letters, symbols and numerals in the text. The function of how an entity will be spoken is specified by punctuation marks such as comma (,), full-stop (.), semicolon (;), exclamation mark (!), question mark (?), etc. A space or a hyphen can also be used to indicates entities that are to be pronounced separately or as a discrete unit. For example, a space between two words indicates that each word is a single entity to be pronounced separately; whereas a hyphen used in the same context indicates that the words should be pronounced as if they are a single entity.

The domain of text considered in this work is derived from Yorùbá newspapers and standard textbooks. This class of text has two important attributes: (i) a *physical content*, and (ii) a *logical structure*. The physical content is described by tokens which form the component of the text. This includes the following:

- punctuation marks, including spaces;
- lexical items written using the SY orthography;
- numerals;
- symbols and acronyms;

- lexical items written using a foreign orthography (such as English words and proper names).

At a higher level is the grammar that guides the construction and organisation of the symbols and letters into syllables, words, phrases, and sentences. The logical structure of a text specifies how the text is organised into pronunciation units. The following elements constitute the logical structure of an SY text of reasonable length: (i) a title, (ii) one or more paragraphs, (iii) sentences, (iv) phrases, (v) words, (vi) syllables. The logical structure of an SY document can be described declaratively, and without reference to how a formatting program should realise the desired format.

1.2 Text models

There are three types of text model that can result from the above described SY text structure, namely: (i) *restricted*, (ii) *unrestricted*, and (iii) *markup*. In the restricted text model, specific rules that guide the acceptable contents and structure of the text are explicitly defined. For example, a rule may specify that all numerals or abbreviations in a text must be written in their lexical format. Another rule may specify the type and acceptable location of punctuation marks that must be used in the text. All input text must be written according to the defined rules and hence the input text is forced to conform to a format. The problem with restricted input is that it makes the TTS machine more difficult to use since the input format rules are normally not standardised and are difficult to standardise since there are divergent writing styles and genre of the text, e.g. drama and poem. In addition the text may be written to convey different concepts which will be compromised if the content or style of the text is restricted.

A softcopy of unconstrained text can be represented and stored in a plain text format (e.g. using UNICODE or ASCII). This type of representation is, however, not powerful enough for describing special features of the text that can provide a meaningful input for TTS machines. For example, in plain text, although there are several popular methods, there is no universally agreed convention for delimiting paragraph boundaries.

In the unrestricted text model, there is no restriction on the input text. The text may contain any symbol, letter or number and can represent text from diverse domains such as weather forecast, poems, incantation, etc. The text-preprocessing module of the TTS machine is then required to extract information necessary for synthesising the text. This approach imposes a more complicated text analysis task on the TTS machine. Predicting the prosodic attributes of the text by using automatic or rule based techniques is unlikely to produce an acceptable level of accuracy. (Quené and Kager, 1992) argued that this task can only be accurately done if the automatic process has access to the semantic and pragmatic information about the input text.

In a markup text model, the input text, usually unrestricted, is annotated with information that renders the prosody of the intended utterance transparent. Markup input can easily be built around a standard markup language, such as the eXtendible Markup Language (XML) making them easy to use by large group of users. Among the information that can be included in an annotated text include intonation phrases, phonetic attributes as well as descriptions of how foreign words and other text anomalies should be pronounced.

Many markup scheme for TTS systems, have been proposed particularly for non-tone languages (Taylor and Isard, 1997, Ogden et al., 2000). But such markup schemes are system specific and often use annotation schemes not specifically tailored for tone languages texts.

1.3 Issues in SY typesetting and markup

A very important feature of a fully tone-marked SY text is that the tones and under-dots are adequately indicated, hence tonal information of each syllable can be extracted from the text. This type of text can be typesetted using LaTeX. The standard LaTeX fonts have all of the components required to compose a Yorùbá text (Taylor, 2000). This is because the LaTeX provides markup for indicating diacritic and under-dots associated on letters. This features makes it possible to conveniently generate text with the correct SY orthography. The tones and phones are the two most important ingredients for generating accurate SY pronunciation.

Another feature of SY orthography, which the LaTeX system also represents accurately, is the use of unambiguous spaces between words. This information can be used to determine word boundaries. Therefore the typesetting of SY texts in LaTeX ensures that accented character format can be used to accurately represent the input to SY TTS. LaTeX provides annotation at the lower level of the physical content of text thereby facilitating the incorporation of more accurate utterance production information. In addition, LaTeX also facilitates the generation of portable documents, such as PDF and .DVI files, which can be read by humans and efficiently stored and transmitted.

On the word level, however, information about how a word must be pronounced in different contexts, e.g. rate of speech, speaking style, etc., cannot be adequately specified using LaTeX. Besides, the logical structure of text which controls the overall prosody of an utterance is better defined at levels high than word (Quené and Kager, 1992). Also, the same sequence of words can be read in different manners or styles depending on the context. Phrases, sentences, and paragraphs are not explicitly specified in LaTeX except through the use of punctuation marks, such as full-stop, comma, and semi-colon, and other escape sequences like the carriage return. For example, in the case of sentences, full-stop normally used for delimiting declarative statements may be ambiguous in some situations, e.g. in sentences also containing numbers with fractions, for example 12.45.

Predicting the phrase and sentence boundaries is a complicated problem if a LaTeX typesetted text is to be processed directly by a TTS system. Moreover, in some speech synthesis tasks, such as text containing dialogue, it may be required to control specific characteristics of the generated speech such as the loudness, rate of speech, age, and gender of the speaker, etc. A TTS system requires phrase and sentence level information to generate the appropriate prosody for a given text. For example, the information on whether a sentence is a question or an exclamation is best defined at sentence and phrase level since they affect utterance units longer than words.

Since LaTeX is so effective in describing the physical content of the text, a reasonable approach would be to design another markup system above LaTeX, which will describe the logical structure of the text. This will allow us to effectively describe the more abstract higher level prosody of the text.

2. TEXT MARK-UP IN SPEECH SYNTHESIS

When a piece of text is read aloud, how much information from the text is responsible for the sound-waveform generated? The text certainly conveys clues to the grouping of syllables in word sequences. It also contains cues about important syntactic boundaries that can be identified by way of punctuation marks. SY text, if written with the complete orthographic specification, contains cues about tones and syllables as well as their organisation into words, phrases and sentences. The identification and appropriate use of these cues can greatly improve the computation of an accurate intonation for the text.

In general, however, such syntactic cues are not enough to facilitate the generation of adequate prosody from text. For example, the same sequence of words can be spoken in different ways, depending on the context and mode. What really matters is the marking of structure and contents in such a way that pauses and emphases are placed correctly and the hierarchy of phrasing and prominence is equitably conveyed (Monaghan, 2001). (Taylor and Isard, 1997) have observed that plain text is the most desirable form of input to a TTS system from a human perspective due to its standard nature and universal understanding. Therefore, there is the need to render input text in such a way that it can be easily read by a human as well as easily processed by a TTS system.

The best way to achieve this goal is to explicitly annotate the input text with information that will aid further processing of the text. The idea of a text markup language was first introduced by (Goldfarb et al., 1970) with the design of the Generalised Markup Language (GML), which later evolved into the Standard Generalised Markup Language (SGML). Since then, a number of text markup languages have been developed and used.

Many TTS developers have designed text Markup Languages (ML) specifically for their TTS applications (e.g. (Taylor and Isard, 1997, Huckvale, 1999, Huckvale, 2001)). Some of these ML include: Spoken Text Markup Language (STML) and Speech Synthesis Markup Language (SSML) (Taylor and Isard, 1997, Burnett et al., 2002), VoiceXML (VoiceXML, 2000) as well as the Java Speech Markup Language (JSML) (JavaSpeechML, 1997). JSML was developed to facilitate a standard text markup and programming for TTS engine in the Java environment. Most of these mark-up languages provide text description tags that describe the structure of the document, and speaker directive tags that control the emphasis, pitch rate, and pronunciation of the text.

SABLE (Sproat et al., 1998) is a TTS markup language developed by combining STML and two other markup languages, i.e. Java Speech Markup Language (JSML) and Speech Synthesis Markup Language (SSML), to form a single standard. It is designed to be easily implemented in any TTS engine. It has been implemented in both the Edinburgh University's Festival speech synthesis system and the Bell Labs TTS engine (Taylor and Isard, 1997).

The following is a simple example of SABLE markup for SY two sentence paragraph: “Bàbá àgbẹ̀ ti ta cocoa 30 kg ní N500 kí ó tó mọ̀ pé àjọ NCB ti fi owólé cocoa. Nì ǹ'kan bíi dédé agogo 3:00 ọ̀sán ni bàbá àgbè délé” (meaning “*[Father framer has sold cocoa 30 kg before he knows that organisation NCB has add money to cocoa. At about time 3:00 afternoon father farmer got home.] The farmer has sold 30 kg of cocoa for N500 before realising that the NCB organisation has increased cocoa price. The farmer got home around 3:00 in the afternoon*”):

```
<DIV TYPE="paragraph">
    <DIV TYPE="sentence" >
       Baba agbe ti ta cocoa 30 kg ni
       N500 ki o to mo pe ajo
       NCB ti fi owole cocoa.
    </DIV>
    <DIV TYPE="sentence">
       Ni n'kan bii dede agogo
       3:00 osan ni baba agbe dele.
    </DIV>
</DIV>
```

In this example, the structure of the text to be pronounced is clearly marked. SABLE also includes tags to specify numeral and other text anomalies.

The markup systems discussed above are not suitable for SY text because they do not adequately describe data and structures found in typical SY text. Using them for marking SY text will lead to a complex representation system which is difficult to use. An alternative suggested by (Huckvale, 2001) is to provide an open, non-propriety textual representation of the data structures at every level and stage of processing. In this way, additional or alternative components may be easily added even if they are encoded in different format. This approach was used in the *ProSynth* project (Ogden et al., 2000).

In *ProSynth*, each composed utterance comprising a single intonation phrase is stored in a hierarchy. Syllables are cross-linked to the word nodes using linking attributes. This allows for phonetic interpretation rules to be sensitive to grammatical function of a word as well as to the position of the syllable in the word. Knowledge for phonetic interpretation is expressed in a declarative form that operates on prosodic structures. A special language called *ProXML* was used to represent knowledge which is expressed as unordered rules and it operates solely by manipulating the attribute on XML-encoded phonological structure.

The *ProXML* implementation suggests that the facility provided by XML matches the requirements to represent the phonological features of an utterance in a metrical prosodic structure, namely: nodes described by attribute-value pairs forming strict hierarchies (Huckvale, 1999). An important attribute of this structure for prosody modelling is that the phonetic descriptions and timings can be used to select speech unit and expresses their durations and pitch contour for output with a TTS system.

The apparent success of XML at representing phonological knowledge, as well as the additional advantage that the represented text can be published on the Internet, motivated our use of XML in developing the markup system for the SY language.

3. TEXT-TO-SPEECH MARKUP SYSTEM

The review presented in the previous section reveals two important facts. First, the design of a markup system is greatly influenced by the method selected for the implementation of the TTS high level synthesis module, the features of the target language (e.g. orthography) and the degree of freedom required in the control of prosody in the TTS system.

Second, the markup scheme used in most systems are based on the eXtensible Markup Language (XML). This is partly because XML allow users to add additional control commands in a flexible and easy to implement manner. XML provides a more abstract and powerful mean of describing speech prosody at utterance level higher that the syllable (i.e. word, phrase, sentence and paragraph) and facilitates possible publications and data sharing on the Internet.

3.1 Design of XML system

The logical structure of any SY text *document* to be synthesised can be viewed as a tree. The root of the tree is associated with the entire document. The *title* of the document is an optional attribute of the root element. The first sub-tree element is the *paragraph.* A document may contain one or more paragraph elements. Each paragraph is equivalent to a branch from the root document. The second sub-tree element is the *sentence.* A paragraph may contain one or more sentences. The third sub-tree element is the *phrase* and the fifth sub-tree element is the *word.* The phrase element is made up of one or more words and each word element is made up of one or more syllables. Each syllable can be adequately typesetted using LaTeX by indicating to diacritic marks and under-dot as required. For example, in syllable *bọ́*, the diacritic mark ´ indicates the tone (i.e. high) and *bọ* is the base. The vowel in the base is the letter *o* with an under-dot. Other LaTeX specifications for typesetting of SY text are shown in Table 1.

The structure of the XML tree defined above a LaTeX document is shown in Figure 1. The leaf node of the XML tree forms the root of the LaTeX part of the tree representation for a piece of text. Element at the same level of the tree are on same level in the linguistic structure of the utterance corresponding to the text. For example, two sentences in the same paragraph of a text share the same branch at the paragraph level. They are also on the same linguistic level in the utterance prosodic hierarchy. The text structure ends with syllable as the leaf elements.

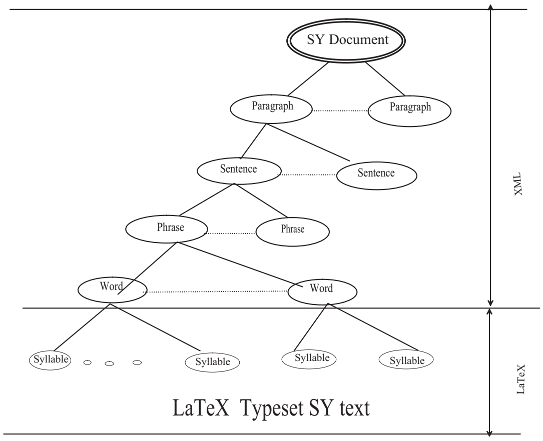

Figure 1: XML tree defined above LaTeX

Note that the tree is built from a hierarchical pattern of objects, each of which has a specific attribute which contributes to how the sentence is to be pronounced. We now discuss the design of higher level prosodic markup system using XML.

Table 1: LaTeX annotation for SY diacritic and under dots

Tag	Description	Result	Example
`\'`	High tone	é	Adé (Crown)
`\d`	Under dot	ẹ	Ọlẹ̀ (Fetus)
`` \` ``	Low tone	è	Ọ̀lẹ (Lazy)
\ (default)	Mid tone	ē or (e)	ewé (Leaf)

4. DOCUMENT TYPE DEFINITION (DTD)

A document type definition (DTD) is a set of declarations that conforms to a particular markup syntax and that describes a class, of "type" of XML documents, in terms of constrains on the logical structure of those document (Wikipedia, 2004). In a DTD for marking text for prosody modelling, the structure of a class of documents is described via:

element definition tags representing each element required to describe the prosody of SY text, and

attribute definition the combination of rules for elements and a list of the information which can be contained within the element. These rules are used to specify the contents of each element.

The beginning of an element is specified by the *start-tag* associated with the tag name and ends with a corresponding *end-tag.* The attribute list for an element contains information which will be included in its tag. An element may have a number of attributes, each of which will have values which can be set when the element is specified. Elements that do not appear in a particular tag may be given a default values

In the design of the tag names for our XML system, we observed that the some annotations LaTeX can be confused with standard XML tags. The annotation name specifications in LaTeX has the form `\tag_name` while XML tags is of the form `<tag_name>` or `</tag_name>`. But in situation where these names can be confused, we retain the name for LaTeX and use the first four upper-case letters of the name for defining the XML tag. Each element is defined by a `start-tag` (e.g. `<document>`) and an end tag (e.g. `</document>`).

There are two important criteria in the design of tag names in the XML our markup system. The first is that computer-literate non-speech-synthesis experts should be able to understand and use the markup. The second is that the markup should be portable across platforms so that a variety of speech synthesis systems should be able to use the additional information provided by the tags (Taylor and Isard, 1997). The general syntax of an XML document is dictated by a set of rules defined by the World Wide Web Consortium (W3C) (Burnett et al., 2002). It consists of a grammar-based set of production rules derived from the Extended Backus-Naur Form (EBNF). In the following, we discuss and illustrate the design of each tagging system using the following SY text.

```
 Title =  Ìdàmún àgbẹ̀
Bàbá àgbè ti ta cocoa 30 kg ní N500 kí ó tó mọ̀ pé àjọ
NCB ti fi fowó lé cocoa.  Ní ǹkan bìi dédé agogo 3:00
ọ̀sán ni bàbá àgbè délé.  Kíá tí àwon alágbèse tí ó bá
bàbá ṣiṣẹ́ ní oko cocoa rè gbọ́pé óti dé láti ọjà ni wọ́n
wátò sí enu ònà ilé bàbá àgbè làti gba owó iṣé tí bàbá jẹ
wọ́n.  Léyìn ìgbà tì bàbá ṣan owó àwọn alágbàse tán ló
tó ri wípé kò sí èrè rárá lórí oun tí òun tà.  Díè ni owó tí
ó kù fi lé ní N102.
```

4.1 The document tag

The `<document>` tag delimits the root element of an SY document. It has an optional *title* attribute which indicates the title of the document. The contents of *title* attribute include an optional text which specifies the title of the document. The `<Document>` tag encloses the text to be spoken. The syntax of the `<Document>` tag is as follows:

```
<document title= "text">
[
<!-- The content of the document will go here  -->
[
</document>
```

The plus sign (+) indicates that a document can containe one or more paragraphs. For example, the `<document>` tag is represented as follows:

```
<document title= ”Ìdàmún Àgbẹ̀” >
{
<!-- The content of the document will go here  -->
}
</document>
```

4.2 The paragraph tag

The paragraph tag, `<PARA>`, defines each paragraph in the document. We use the tag name `<PARA>` so that it will not be confused with the *paragraph* annotation in LaTeX. The paragraph contains an optional attribute which indicates the *style* that will be used in uttering the paragraph. The syntax of the `<PARA>` tag is as follows;

```
<PARA style = "Styletype">
{
sentence+ <!(i.e. The sentences in paragraph) >
}
</PARA>
```

The style attribute accepts five style types:

1. *Dialogue*- a paragraph spoken in the context of a dialogue conversation.

2. *Spontaneous*- a paragraph spoken spontaneously.

3. *Read* - a paragraph spoken as read speech (Default).

4. *Poem* - a paragraph spoken as read Yorùbá poem, e.g. Ewì (common poem), Oríkì (praise song), etc.

5. *Incantation*- a paragraph spoken as SY incantation, e.g. Ọfọ̀, Ògèdè, Àsẹ, etc..

The default value for style is *Read.* Using the paragraph tag, the first paragraph in the sample text will be tagged as follows:

```
<document  title = Ìdàmún Àgbẹ̀ >
  <PARA style= "Read">
        <!-- The content of each paragraph will go
here  -->
  </PARA>
</document>
```

4.3 The sentence tag

The sentence tag, `<sentence>`, delimits an SY sentence and contained in the paragraph element. All sentence elements within the same paragraph have same paragraph attributes. A sentence element contains at least one phrase element. The sentence element has many attributes which specify the prosodic information of a sentence. These attributes are useful in a multi-speaker text, such as a play or a dialogue. It is also required for annotating texts containing more than one reading styles, e.g. a poem. The sentence attributes include the following:

MODE: specifies the mode for speaking a sentence. The mode attribute can take one of the following values: *Question, Declaration, Exclamation, Statement.* The default value is Statement.

MOOD: specifies the mood for the speaking a sentence. The mood attribute can assume one of the following values: *Happy, Sad, Normal.* The default value is Normal.

RATE: specifies the rate at which a sentence is will be spoken. The attribute has three possible values: *Fast, Normal, Slow.* The default value is Normal.

PAUSE: signifies the duration of the pause that should be inserted between two words using a linguistic value. This attribute has 3 possible values: *Short, Long, Medium.*

LOUD: signifies the loudness or volume of the sentence as linguistic values. The values possible: *Low, Medium, High.* The default value is medium.

GENDER: specifies the type of voice, i.e. male or female, for synthesising the sentence. It has the following values: *Male, Female.* The default value is male.

AGE: specifies the approximate age of voice to be synthesises: It has the following values: *Child, Adult, and Old.* The default value is Adult.

The syntax for sentence tagging is therefore:

```
<sentence MODE= ”Statement” MOOD=”Normal”
STYLE=”Oro”
  RATE="Normal" PAUSE="Medium" LOUD="Medium" >
      <!-- The content of each sentence go here -->
</sentence>
```

All the parameters specified in this syntax are the default values of the attributes.

The annotation for the first sentence in our example text is as follows:

```
<sentence MODE=”Statement” MOOD=”Normal”
STYLE=”Oro”
  RATE="Normal" PAUSE="Medium" LOUD="Medium" >
Bàbá àgbè ti ta cocoa 30 kg ní N500 kí ó tó mọ̀ pé àjọ
NCB ti fi fowó lé cocoa.
</sentence>
```

The last two attributes specifies the kind of speaker's voice to be imitated by the speech synthesiser. If it is not specified an adult male native speaker of SY is assumed. This attribute is only useful for selecting the relevant database in a multi-speaker TTS environment, such as dialogue or story telling. The attribute will guide the TTS engine in selecting the appropriate database.

The sentence tags and attributes discussed above are designed in the manner stated above in order to facilitate the synthesis of text representing a dialogue between two or more people. This situation is very common in plays and newspaper interview texts. The scope of the speaker tag varies. In situation where only one language database is available all specified speaker attributes will be ignored.

4.4 The phrase tag

The phrase tag, <phrase>, can be inserted within the text of a sentence to indicate the presence of phrase boundaries. The <phrase> effectively breaks a sentence into intonation phrases even when punctuation marks are present. The syntax for the phrase element is as follows:

```
<phrase>
      <!-- The content of each phrase go here -->
</phrase>
```

The <phrase> tag for specifying the prosodic phrase for the previous example sentence is as follow:

```
<sentence MODE=”Statement” MOOD=”Normal”
STYLE=”Oro”
  RATE="Normal" PAUSE="Medium" LOUD="Medium" >
<phrase> Bàbá àgbè ti ta cocoa 30 kg ní N500 </phrase>
<phrase> kí ó tó mọ̀ pé àjọ NCB ti fi fowó lé cocoa
</phrase>.
</sentence>
```

5. TEXT CONTENTS DESCRIPTION TAGS

In a normal SY text, there are many textual anomalies which can occur as part of the content. Some examples of textual anomaly include numerals representing ordinal or cardinal data items, letters representing abbreviations, as well as a groups of letters in foreign -words and proper names, e.g. David. In order to remove the complexities involved in determining the exact pronunciation of these textual anomalies, we defined markup tags for them. These tags are built around the SAYAS tag in W3C (Burnett et al., 2002) and extended to incorporate features specific to SY text. The information provided by this tag will allow the High Level Synthesis module to determine the type of expansion to apply on each of the tagged items during text normalisation process.

5.1 The SAYAS tag

The text content elements are used to markup the content of a text in order to specify how they are to be spoken. For example, N500 must be spoken as currency; IADS is to be spoken as an acronym with specific pronunciation, whereas the abbreviation O.A.U. is to be spelt out as English alphabet using Yorùbá accent. The list of tags for content element markup is specified in Table 2.

Following the W3C format, we use the SAYAS tag with three specific attributes. The SUB attribute is used to specify a substitute text for an abbreviation or acronym. For example, the text *Kóànùn* can be substituted for COAN (COmputer Association of Nigeria). The CLASS attribute is used to specify the class of the pronunciation as stated in Table 2. When this parameter is not specified, the default is SY abbreviation pronounced with SY accent. Some examples of SAYAS tags is as follows:

```
<SAYAS SUB =”a.b.b.l.”> àti bèé bèé lọ </ SAYAS>
<SAYAS SUB =”i.n.p.”> ìyẹn ni pé </ SAYAS>
<SAYAS SUB =”f.w.” >fiwé </SAYAS
<SAYAS SUB =”b.a.” > bí àpẹẹrẹ </SAYAS>
<SAYAS SUB =”f.a.” > fún àpẹẹrẹ </SAYAS
<SAYAS SUB = ”w.o.r” > wò ó ore </SAYAS
<SAYAS SUB = ”e.n.p.” > èyí nipé</SAYAS >
```

5.1.1 *The SUB attribute*

The syntax for the SUB attribute is

```
< SAYAS SUB = ”text to be substituted” > text </SAYAS>
```

The SUB attribute is particularly useful in defining replacement text for abbreviations and other shorthand text. The substitution strings for some commonly used SY abbreviation are defined as bellow:

5.1.2 *The CLASS attribute*

The syntax for the CLASS attribute is as follows:

```
<SAYAS  CLASS = ”attribute” > text </SAYAS>
<SAYAS CLASS =”currency” > N500</SAYAS>
<SAYAS CLASS =”acronym” > AIDS </SAYAS\>
```

5.1.3 *The ABBRACENT attribute*

The third attribute is the abbreviation accent attribute, ABBRACENT. It determines whether an abbreviation will be spelt out as Yorùbá abbreviation using a Yorùbá accent or an English abbreviation (each letter is from the English alphabet) using Yorùbá accent. For example, the abbreviation “a.b.b.l.” (i.e. àti bèé bèé lọ) is a Yorùbá abbreviation and its component alphabet must be pronounced using SY phones. However, O.A.U. (Organisation of African Unit) is an English abbreviation which must be pronounced using SY accent. Below is an example of the usage of the above markup tags:

```
<SAYAS CLASS =”ABBREVIATION”
ABBRACENT=’English’ > O.A.U </SAYAS>
```

6. CONCLUSIONS

The text markup system for text intended as input to standard Yorùbá speech synthesis system is presented. Future work will be directed at the text normalisation system that will use the information provided in the text to expand textual anomalies.

Acknowledgments

Financial support from the *African Language Technologies Initiative* and encouragement from Dr. Túndé Adégbọlá and Professor Kọ́lá Owólábi to the development of the ideas presented in this paper is acknowledged and highly appreciated. The support from the Ọbáfẹ́mı Awólọ́wọ̀ University is here acknowledge.

7. REFERENCES

Bird, S. (1998). Strategies for representing tone in African writing systems: a critical review. URL:http://cogprints.org/2174/00/wll2.pdf. Visited: Jan 2003.

Burnett, D. C., Walker, M. R., and Hunt, A. (2002). Speech synthesis markup language version 1.0 w3c working draft 02. http://www.w3.org/TR/speech-synthesis/#S1.1. visited: Jun 2004.

Ọdẹ́jọbí, O. A., S., W. S. H., and Beaumont, A. J. (2008). A modular holistic approach to prosody modelling for standard yorùbá speech synthesis. *Computer Speech & Language*, 22:39–68.

Goldfarb, C. F., Mosher, E. J., and Peterson, T. I. (1970). An online system for integrated text processing. In *Proc. American Society for Information Science*, volume 7, pages 147–150.

Huckvale, M. (1999). Representation and processing of linguistic structures for an all-prosodic synthesis system using xml. In *Proc. of EuroSpeech ’99*, number 4, pages 1847–1850, Budapest.

Huckvale, M. (2001). The use and potential of extensible markup (XML) in speech generation. In Keder, E., Bailly, G., Monaghan, A., Terken, J., and Huckvale, M., editors, *Improvements in Speech Synthesis: Cost 258: The Naturalness of Synthetic Speech*, chapter 30, pages 298–305. Wiley Inter. Science.

JavaSpeechML (1997). Java speech markup language (jsml) specification , version 0.5. Visited: May 2005.

Monaghan, A. (2001). Markup for speech synthesis: a review and some suggestions. In Keder, E., Bailly, G., Monaghan, A., Terken, J., and Huckvale, M., editors, *Improvements in Speech Synthesis: Cost 258: The Naturalness of Synthetic speech*, chapter 31, pages 307–319. Wiley Inter. Science.

Ogden, R., Hawkins, S., House, J., Huckvale, M., Local, J., Carter, P., Dankovicova, J., and Heid, S. (2000). Prosynth: an integrated prosodic approach to device-independent natural-sounding speech synthesis. *Computer Speech & Language*, 14:177–210.

Quené, H. and Kager, R. (1992). The derivation of prosody for text-to-speech from prosodic sentence structure. *Computer Speech & Language*, 6:77–98.

Sproat, R., Hunt, A., Ostendorf, M., Taylor, P., Black, A., Lenzo, K., and Edgington, M. (1998). Sable: A standard for TTS markup. http://www.bell-labs.com/project/tts/sabpap/sabpap.html. Visited: Jun 2004.

Taylor, C. (2000). Typesetting african languages. http://www.ideography.co.uk/library/afrolin gua.html. Visited: Apr 2004.

Taylor, P. and Isard, A. (1997). SSML: A speech synthesis markup language. *Speech Communication*, 21:123–133.

VoiceXML (2000). Voice extensible markup language: VoiceXML. http://www.voicexml.org/specs/VoiceXML-100.pdf. Visited: May 2003.

Wikipedia (2004). Document type definition. http://en.wikipedia.org/wiki/Document _Type_Definition. visited: Jul 2005.

Appendix

Table 2: Tags for SY text

Token Classification	Description	Tag name
Date	String of numbers formatted as 99-99-99, or 99/99/99	DATE
Time	String of numbers formatted as 99:99, 99/99,	TIME
Currency	String of numbers prefixed by a currency symbol, e.g. N, $, £,	CURRENCY
Lexical	String of letters	LEXICAL
Ordinal digits	String of numbers prefixed by a noun	ORDINAL
Cardinal digit	String of numbers postfixed by a noun	CARDINAL
Loanword	Word with foreign language spelling, e.g. English, French, Arabic, e.t.c.	LOAN
Punctuation	Punctuation marks such as (;) , (:) , (.)	PUNCT
Acronym	Group of upper case letters such as FIFA, OAU, USA, e.t.c.	ACRONYM
Special Character	Characters such as *, +, etc.	SPEC
SI unit	SI unit to be expanded into SY accent pronunciation	SUNIT
Phone	Phone number (digit by digit pronunciation)	PHONE
Proper names	Proper names, usually of English origin	PRONAME