Abstract

Homograph ambiguity is an original issue in Text-to-Speech (TTS). To disambiguate homograph, several efficient approaches have been proposed such as part-of-speech (POS) n-gram, Bayesian classifier, decision tree, and Bayesian-hybrid approaches. These methods need words or/and POS tags surrounding the question homographs in disambiguation. Some languages such as Thai, Chinese, and Japanese have no word-boundary delimiter. Therefore before solving homograph ambiguity, we need to identify word boundaries. In this paper, we propose a unique framework that solves both word segmentation and homograph ambiguity problems altogether. Our model employs both local and long-distance contexts, which are automatically extracted by a machine learning technique called Winnow.

1 Introduction

In traditional Thai TTS, it consists of four main modules: word segmentation, grapheme-to-phoneme, prosody generation, and speech signal processing. The accuracy of pronunciation in Thai TTS mainly depends on accuracies of two modules: word segmentation, and grapheme-to-phoneme. In word segmentation process, if word boundaries cannot be identified correctly, it leads Thai TTS to the incorrect pronunciation such as a string “ตากลม” which can be separated into two different ways with different meanings and pronunciations. The first one is “ตา(eye) กลม(round)”, pronounced [ta:0 klom0] and the other one is “ตา(expose) ลม(wind)”, pronounced [ta:k1 lom0]. In grapheme-to-phoneme module, it may produce error pronunciations for a homograph which can be pronounced more than one way such as a word “พลชม” which can be pronounced [phlaw0] or [phe:0 la:0]. Therefore, to improve an accuracy of Thai TTS, we have to focus on solving the problems of word boundary ambiguity and homograph ambiguity which can be viewed as a disambiguation task.

A number of feature-based methods have been tried for several disambiguation tasks in NLP, including decision lists, Bayesian hybrids, and Winnow. These methods are superior to the previously proposed methods in that they can combine evidence from various sources in disambiguation. To apply the methods in our task, we treat problems of word boundary and homograph ambiguity as a task of word pronunciation disambiguation. This task is to decide using the context which was actually intended. Instead of using only one type of syntactic evidence as in N-gram approaches, we employ the synergy of several types of features. Following previous works [4, 6], we adopted two types of features: context words, and collections. Context-word feature is used to test for the presence of a particular word within +/- K words of the target word and collocation test for a pattern of up to L contiguous words and/or part-of-speech tags surrounding the target word. To automatically extract the discriminative features from feature space and to combine them in disambiguation, we have to investigate an efficient technique in our task.

The problem becomes how to select and combine various kinds of features. Yarowsky [11] proposed decision list as a way to pool several types of features, and to solve the target problem by applying a single strongest feature, whatever type it is. Golding [3] proposed a Bayesian hybrid method to take into account all available evidence, instead of only the strongest one. The method was applied to the task of context-sensitive spelling correction and was reported to be superior to decision lists. Later, Golding and Roth [4] applied Winnow algorithm in the same task and found that the algorithm performs comparably to the Bayesian hybrid method when using pruned feature sets, and is better when using unpruned sets or unfamiliar test set.

In this paper, we propose a unified framework in solving the problems of word boundary ambiguity and homograph ambiguity altogether. Our approach employs both local and long-distance contexts, which can be automatically extracted by a machine learning technique called Winnow. We then construct our system.
based on the algorithm and evaluate them by comparing
with other existing approaches to Thai homograph
problems.

2 Problem Description

In Thai TTS, there are two major types of text ambiguities
which lead to incorrect pronunciation, namely word
boundary ambiguity and homograph ambiguity.

2.1 Word Boundary Ambiguity (WBA)

Thai as well as some other Asian languages has no word
boundary delimiter. Identifying word boundary, especially
in Thai, is a fundamental task in Natural Lan-
guage Processing (NLP). However, it is not a simple
problem because many strings can be segmented into
words in different ways. Word boundary ambiguities for
Thai can be classified into two main categories defined
by [6]: Context Dependent Segmentation Ambiguity
(CDSA), and Context Independent Segmentation Ambi-
guity (CISA).

CISA can be almost resolved deterministically by
the text itself. There is no need to consult any context.
Though there are many possible segmentations, there
is only one plausible segmentation while other alternatives
are very unlikely to occur, for example, a string
"ตากลม" which can be segmented into two different
ways: "ตากลม (carry) ลม(deviate)" [paj0
ha:m4 he:4 si:4] and "ตากลม (see) ลม(deviate)" [paj0
ha:m4 ma:3 he:4 si:4]. Only the second choice is plausible.
One may say that it is not semantically ambiguous.
However, simple algorithms such as maximal matching
[6, 9] and longest matching [6] may not be able to dis-
criminate this kind of ambiguity. Probabilistic word
segmentation can handle this kind of ambiguity success-
fully.

CDSA needs surrounding context to decide which
segmentation is the most probable one. Though the
number of possible alternatives occurs less than the con-
text independent one, it is more difficult to disambugu-
ate and causes more errors. For example, a string
"หาหามเหสี" can be segmented into two different
ambiguity, we treat these problems as the problem of
disambiguation pronunciation. We construct a confusion
set of the string “หาหามเหสี” which is a boundary-
ambiguity string and C = { [ma:0 kwa:1] , [ma:k2 wa:2] } is the confu-
sion set of the homograph “หาหามเหสี”. We obtain the
features that can discriminate each pronunciation in the
set by Winnow based on our training set.

4.1 Winnow

Winnow algorithm used in our experiment is the algo-
risth described in [1]. Winnow is a neuron-like network
where several nodes are connected to a target node [4,
5]. Each node called specialist looks at a particular
value of an attribute of the target concept, and will vote
for a value of the target concept based on its specialty;
i.e. based on a value of the attribute it examines. The
global algorithm will then decide on weighted-majority
votes receiving from those specialists. The pair of (at-
tribute=value) that a specialist examines is a candidate of features we are trying to extract. The global algorithm updates the weight of any specialist based on the vote of that specialist. The weight of any specialist is initialized to 1. In case that the global algorithm predicts incorrectly, the weight of the specialist that predicts incorrectly is halved and the weight of the specialist that predicts correctly is multiplied by 3/2. The weight of a specialist is halved when it makes a mistake even if the global algorithm predicts correctly.

4.2 Features
To train the algorithm to resolve pronunciation ambiguity, the context around a homograph or a boundary-ambiguity string is used to form features. The features are the context words, and collocations. Context words are used to test for the presence of a particular word within +10 words and –10 words from the target word. Collocations are patterns of up to 2 contiguous words and part-of-speech tags around the target word. Therefore, the total number of features is 10; 2 features for context words, and 8 features for collocations.

5 Preliminary Experiment
To test the performance of the different approaches, we select sentences containing Thai homographs and boundary ambiguity strings from our 25K-words corpus to use in benchmark tests. Every sentence is manually separated into words. Their parts of speech and pronunciations are manually tagged by linguists. The resulting corpus is divided into two parts; the first part, about 80% of corpus, is utilized for training and the rest is used for testing.

In the experiment, we classify the data into three group depending on types of text ambiguity according to section 2: CDSA, CISA and Homograph, and compare the results from different approaches; Winnow, Bayesian hybrid [3] and POS trigram. The results are shown in Table 1.

<table>
<thead>
<tr>
<th></th>
<th>Trigram</th>
<th>Bayesian</th>
<th>Winnow</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDSA</td>
<td>73.02%</td>
<td>93.18%</td>
<td>95.67%</td>
</tr>
<tr>
<td>CISA</td>
<td>98.25%</td>
<td>99.67%</td>
<td>99.70%</td>
</tr>
<tr>
<td>Homograph</td>
<td>52.46%</td>
<td>94.25%</td>
<td>96.45%</td>
</tr>
</tbody>
</table>

Table 1: The result of comparing different approaches

6 Conclusion
In this paper, we have successfully applied Winnow to the task of Thai homograph disambiguation. Winnow shows its ability to construct networks that extract the features in data effectively. The learned features, which are context words and collocations, can capture useful information and make the task of Thai homograph disambiguity more accurate. The experimental results show that Winnow outperform trigram model and Bayesian hybrid. Our future works will investigate other machine learning techniques such as SNoW and SVM.

References