# Comparison Thai Word Sense Disambiguation Method

Teerapong Modhiran*, Boontee Kruatrachue*, Thepchai Supnithi **

*Computer Engineering Department, Faculty of Engineering
King Mongkut's Institute of Technology Ladkrabang
Chalongkrung Road, Bangkok 10520, Thailand
(Tel : +66-6-320-4101; E-mail: s4061618@kmitl.ac.th, boontee@diamond.ce.kmitl.ac.th )
**Information Research and Development Division
National Electronic and Computer Technology Center, Thailand
(Tel : +66-2-564-6900; E-mail: thepchai@nectec.or.th)

**Abstract**: Word sense disambiguation is one of the most important problems in natural language processing research topics such as information retrieval and machine translation. Many approaches can be employed to resolve word ambiguity with a reasonable degree of accuracy. These strategies are: knowledge-based, corpus-based, and hybrid-based. This paper pays attention to the corpus-based strategy. The purpose of this paper is to compare three famous machine learning techniques, Snow, SVM and Naive Bayes in Word-Sense Disambiguation on Thai language. 10 ambiguous words are selected to test with word and POS features. The results show that SVM algorithm gives the best results in solving of Thai WSD and the accuracy rate is approximately 83-96%.

**Keywords:** Word Sense Disambiguation, Machine Learning, Natural language processing, SVM algorithm, Corpus-based

## 1. INTRODUCTION

Word Sense Disambiguation (WSD) is the mean to solve problem of assigning a sense to an ambiguous word. Resolving the word ambiguity is considered as the major bottleneck for large scale language understanding applications and their associate tasks such as machine translation (MT), information retrieval (IR), natural language understanding (NLU) and others. In MT, Word Sense Disambiguation is used to produce an appropriate translated sentence output. Many approaches have been proposed for eliminating the ambiguous. Most of the word sense researching is to assess in English sentence and to assist in English language translation some Thai words have several meaning or senses, and depend on context words which ambiguous between these senses. For example the word "เกาะ" ("Kor") can be represented as a noun with the meaning "an island" or as a verb with the meaning "to hold something". If we are able to solve this problem, translation Thai into other languages will be more accurate.

Previous paper for solving WSD in Thai sentence is applied to 2 words and applies the dicision list algorithm is used to solve the problem [1]. The accuracy rate of the result is approximately 80%

## 2. ALGORITHM

### 2.1 SNOW (Sparse Network of Winnow [8])

Snow is a multi-class learner, where each class label is represented as a linear function over the feature space. Each class is implementing using a Winnow [5] node, which learns to separate that class from all the rest.

Let $A_t = \{i_1,\ldots,i_n\}$ be the set of active features in a given example that are linked to each class (target) node *t*. Let $s_i$ be the real valued strength associated with feature *i* (defalut: 1) in the example. The feature $i_1$represent the occurrence of a word 1 or part of speech of a word in training sentence, which has binary value active or non-active.

Then we say that *t* predicted *positive* if and only if

$$\Omega^t(e) = \sum_{i \in A_t} w_i^t s_i \geq \theta^t \quad (1)$$

Where $w_i^t$ is the weight on the edge connecting the feature $i^{th}$ to target node *t*, and $\theta^t$ is *t*'s threshold.

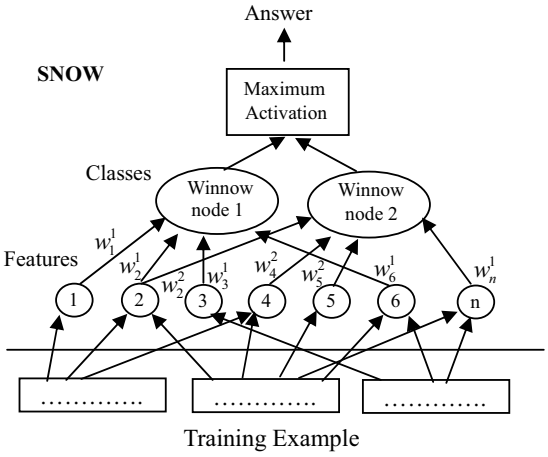


Fig. 1 Show the Snow architecture.

The winnow update rule has two update parameters are a promotion parameter $\alpha_t > 1$ and a demotion parameter $0 < \beta_t < 1$. These are used to update the current hypothesis in t (the set of $w_i^t$ weights) only when a mistake in prediction is made.

A winnow update proceeds as follows:

- Condition 1: If algorithm predicted negative ( $\sum_{i \in A_t} w_i^t s_i < \theta^t$ ) and the specified label is positive, the weights of features active in the current example are *promoted:* $\forall i \in A_t, w_i^t \leftarrow w_i^t \alpha_t^{s_i}$
- Condition 2: If algorithm predicted positive ( $\sum_{i \in A_t} w_i^t s_i \geq \theta^t$ ) and the specified label is negative weights of features active in the current example are *demoted:* $\forall i \in A_t, w_i^t \leftarrow w_i^t \beta_t^{s_i}$
- If the Winnow node predict correct, weights are unchanged.

All other parameters threshold, alpha, and beta are fixed in this research (for method to modify these value see [8]).

From Snow's equation if we have train data as follows.
1,1001,1002,1003
1,1001,1002,1003
2,1001,1004,1005
2,1001,1003,1006

Use $\beta$ = 0.8, $\alpha$ = 1.35, $\theta$ = 4 and default weight = 3.

We train this features to Snow. The first field is the class number and the following numbers is the features. In figure 2a-2d represent the Snow network processing. The features will be input consequently. Once it matches to the update-condition rules, the weight of feature will be updated.

Step1: 1,1001,1002,1003

Initialize feature for Class 1.

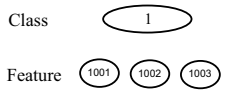


Fig 2a

Step2: 1,1001,1002,1003

Assign default weight.

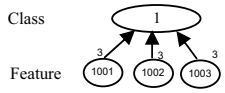


Fig 2b

Step3: 2,1001,1004,1005

Initialize feature for Class 2.

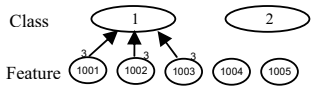


Fig 2c

Step4: 2,1001,1003,1006

Class 1 matched condition 2.
Class 2 matched condition 1.
The active feature's weight will be updated.

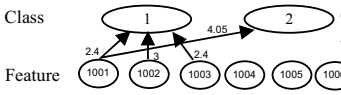


Fig 2d

Fig. 2 Example of Snow network generate.

The real-valued result of the summation in equation (1) is referred to as the target node's *activation.* Target node activations drive weight vector updates during training as well as predictions during testing. The default testing policy for multiple target nodes networks is a winner-take-all rule. Let $T$ be the set of all targets defined in the current architecture instantiation. The predicted target $t^*$ for the example $e$ with a set of active features $\Upsilon_{t \in T} A_t$ is:

$$t^*(e) = \text{argmax}_{t \in T} \sigma(\theta^t, \Omega^t(e)) \quad (2)$$

Where $\Omega^t(e)$ is the activation calculated by the summation in equation (1) for target node $t$ given $e$, and $\sigma(\theta, \Omega(e))$ is a learning algorithm specific sigmoid function. In Snow, winnow's sigmoid activation is calculated with the following formula.

$$\sigma(\theta, \Omega) = \frac{1}{1 + \frac{\theta}{\Omega}} \quad (3)$$

For example if we train this feature to Snow:

| CLASS | Example |
|---|---|
| 1 | 1001, 1002, 1003,1004,1005,1006 |
| 1 | 1007, 1008, 1009,1010,1011,1012 |
| 1 | 1008, 1013, 1014,1015,1016,1017 |
| 2 | 1018, 1019, 1020,1021,1022,1023 |
| 2 | 1018, 1020, 1021,1024,1025,1026 |
| 2 | 1027, 1028, 1029,1030,1031,1032 |
| 3 | 1033, 1034, 1035,1036,1037,1038 |
| 3 | 1033, 1039, 1040,1041,1042,1043 |
| 3 | 1033, 1052, 1053,1054,1055,1056 |
| 4 | 1015, 1027, 1044,1045,1046,1047 |
| 4 | 1025, 1027, 1046,1048,1049,1050 |
| 4 | 1024, 1037,1045,1051,1052 |

We obtain snow network as shown in figure 3. If we has a test data set (1033,1035,1038,1057,1058,1059), The prediction result belong to equation(2) and we will get the result as class 3.

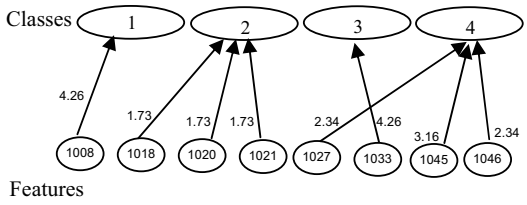


Fig. 3 Example of network from snow.

**2.2 NB (Naive Bayes Algorithm)**

Naive Bayes is based on Bayesian learning. Bayesian classifiers are statistical classifiers. They can predict the class membership probability that a given example belongs to. They are based on the Bayes Theorem [6].

Bayes Theorem:

- Given a hypothesis $H$ and evidence $E$ based on this hypothesis, then the probability of $H$ given $E$ is

$$P(H/E) = \frac{P(E|H)P(H)}{P(E)}$$

Naive Bayes Classifier:

Assume target function $f: X \rightarrow V$, where each instance $X$ described by attributes $(i_1, i_2, ..i_n)$ and V is set of word senses. A maximal hypothesis is $\upsilon_{SENSE}$ as follows.

$$\upsilon_{SENSE} = \arg\max_{v_j \in V} P(v_j \mid i_1, i_2 .....i_n)$$

$$\upsilon_{SENSE} = \arg\max_{v_j \in V} \frac{P(i_1, i_2 .....i_n \mid v_j)P(v_j)}{P(i_1, i_2 .....i_n)}$$

$$= \arg\max_{v_j \in V} P(i_1, i_2 .....i_n \mid v_j)P(v_j)$$

Naive Bayes assumption:

$$P(i_1, i_2 .....i_n \mid v_j) = \prod_i P(i_i \mid v_j)$$

Which gives Naive Bayes classifier:

$$\upsilon_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(i_i \mid v_j) \quad (4)$$

The input words are converted to string of integer feature ID as shown in Snow. We obtain probability for each feature $i_i$ given word sense, $P(i_i|v_j)$ where $i_i$ is feature i and $v_j$ is word sense j.

$$\frac{\text{Frequency of Feature } i_i + 1}{\text{Total Feature in word sense } v_j + \text{Total Feature of training data}} \quad (5)$$

If we use the same training input examples as in Snow,
Considering feature 1008 for Class 1
$P(i_{1008}|v_1) = 2+1/(18+1057) = 0.00279$

If the test data set is:
Test Data: 1033, 1035, 1038,1057,1058,1059

In order to identify the word sense of a test data, we calculate probability value $P(i_i|v_j)$ for each feature in the test data from the training data as shown above. The word sense that has maximal posteriori probability is the predicted word sense as in equation 4.

From this example, there are 3 features in the test data set that appears in the train set: 1033, 1035, 1038. The following is the example of calculation using Bayes classification.

Sense 1:
$P(v_1) = 0.25$
$P(i_{1033}| v_1) = 0.00093$,
$P(i_{1035}| v_1) = 0.00093$
$P(i_{1038}| v_1) = 0.00093$
$P(v_1)\prod_i P(i_i \mid v_1) = 0.25(0.00093)(0.00093)(0.00093) =$
0.00000000020108925

Sense 2:
$P(v_2) = 0.25$
$P(i_{1033}| v_2) = 0.00093$
$P(i_{1035}| v_2) = 0.00093$
$P(i_{1038}| v_2) = 0.00093$
$P(v_2)\prod_i P(i_i \mid v_2) = 0.25(0.00093)(0.00093)(0.00093) =$
0.00000000020108925

Sense 3:
$P(v_3) = 0.25$
$P(i_{1033}| v_3) = 0.0037$
$P(i_{1035}| v_3) = 0.0018$
$P(i_{1038}| v_3) = 0.0018$
$P(v_3)\prod_i P(i_i \mid v_3) = 0.25(0.0037)(0.0018)(0.0018) =$
**0.000000002997**

Sense 4:
$P(v_4) = 0.25$
$P(i_{1033}| v_4) = 0.00093$
$P(i_{1035}| v_4) = 0.00093$
$P(i_{1038}| v_4) = 0.00093$
$P(v_4)\prod_i P(i_i \mid v_4) = 0.25(0.00093)(0.00093)(0.00093) =$
0.00000000020108925

Since Class 3 gives the most probability results, the word in the test data has the sense number 3.

**2.3 SVM (Support Vector Machine [7])**

SVM is a supervised learning algorithm for binary classification problems [4]. In figure 4 shows how SVM builds a separating hyperplane for classification and vectors on margin lines (support vectors).

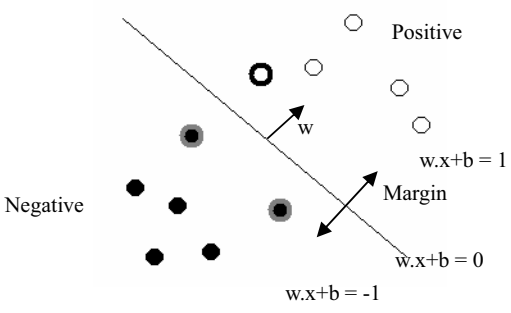


Fig.4 Show the concept of SVM

An exactly hyperplane is unspecified, then the optimal solution of this is using the maximal margin, which is a distance between negative set and positive set to classify all training set into their categories.

Training set is given in vector form.

$$S = (x_1, y_1)..., (x_n, y_n), x_i \in R^k, y_i \in \{+1,-1\}$$

When $x_i$ is a feature vector of the i-th training sample and $y_i$ is an additional label to specify whether $x_i$ is; positive (+1) or negative (-1). The separating hyperplane is given by

$$w \cdot x + b = 0, w \in R^n, b \in R$$

With maximal margins are given by

<w•x> + b = +1

<w•x> + b = -1

We can derive these relation to the following constrains

$$y_i(w \cdot x_i + b) - 1 \geq 0$$

From this, we implied that the size of margin is equal to 2/||w|| [7], we assume the following objective function J(w) for linear problem:

$$\underset{w,b}{Minimize} J(w) = \frac{1}{2} \| w \|^2$$

$$y_i(w.x_i + b) - 1 \geq 0$$

By solving a quadratic programming equation, we get the decision function f(x) = sgn(g(x)) is derived, where

$$g(x) = \sum_{i=1}^{u} \alpha_i y_i x_i . x + b$$

For handling a non-linear problem, simply substitution all occurrence of an inner product term in above equation with kernel function $K(x_i.x)$. Then we rewrite equation as:

$$g(x) = \sum_{i=1}^{u} \alpha_i y_i K(x_i, x) + b \quad (6)$$

In this paper, we apply a polynomial kernel function as in equation

$$K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0.$$

(We use $\gamma = 1$, r = 0, d = 1)

From equation 6 we get the decision function

$$f(x) = \text{sgn}(\sum_{i=1}^{l} \alpha_i y_i K(x_i.x) + b), \sum_{i=1}^{l} \alpha_i y_i = 0 \quad (7)$$

The function sgn is defined as sgn(x) = 1 when x ≥ 0,-1 otherwise. If we input same Snow's example to SVM then we will get file model as follows.

**File model:**

b = 0.06[1-2], 0.017[1-3], -0.036[1-4],
-0.044[2-3], -0.102[2-4], -0.055[3-4]

| Example | Pair-wise class ($.\alpha_i y_i$ value) | | | | | |
|---|---|---|---|---|---|---|
| | [1-2] | [1-3] | [1-4] | [2-3] | [2-4] | [3-4] |
| 1 | 0.026 | 0.027 | 0.028 | ------ | ------ | ------ |
| 2 | 0.025 | 0.026 | 0.028 | ------ | ------ | ------ |
| 3 | 0.025 | 0.026 | 0.028 | ------ | ------ | ------ |
| 4 | -0.023 | ------ | ------ | 0.023 | 0.024 | ------ |
| 5 | -0.023 | ------ | ------ | 0.023 | 0.026 | ------ |
| 6 | -0.023 | ------ | ------ | 0.029 | 0.031 | ------ |
| 7 | ------ | -0.026 | ------ | -0.025 | ------ | 0.028 |
| 8 | ------ | -0.026 | ------ | -0.025 | ------ | 0.027 |
| 9 | ------ | -0.026 | ------ | -0.025 | ------ | 0.028 |
| 10 | ------ | ------ | -0.023 | ------ | -0.022 | -0.022 |
| 11 | ------ | ------ | -0.024 | ------ | -0.024 | -0.027 |
| 12 | ------ | ------ | -0.037 | ------ | -0.036 | -0.039 |

Assume that the test data set is (1003, 1035, 1038, 1057, 1058, 1059). In order to find answer we use pair-wise method to compare the result for each class. Following equation 7 f(x) will be computed to each pair classes. Six combinations will be calculated and the results are as follows.

In case [1-2] Sgn(0.062) - Answer is Sense 1
In case [1-3] Sgn(-0.113) - Answer is Sense 3 *
In case [1-4] Sgn(-0.036) - Answer is Sense 4
In case [2-3] Sgn(-0.202) - Answer is Sense 3 *
In case [2-4] Sgn(-0.102) - Answer is Sense 4
In case [3-4] Sgn(0.084) - Answer is Sense 3 *

We conclude that the answer of test feature is Sense 3, since the number of results sense 3 is the most answer that is found in the comparison.

## 3. EXPERIMENTAL MOTHODOLOGY

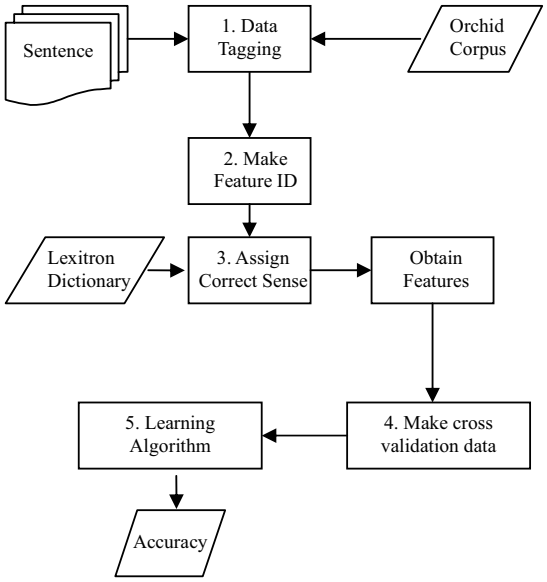


Fig.5 Show the overall experimental methodology.

From figure 5, Step 1-3 are processing data. Step 4-5 are the process for applying machine learning techniques to the data. The output is the accuracy of test data set.

Step 1: The 10,000 test sentences with ambiguous words are extracted from Thansethakij newspaper (1000 sentences for each ambiguous word). Each sentence is segmented and part of speech is tagged using Bigram algorithm [9] using the train data from Orchid corpus [2] which contain about 25,000 sentences. Sentences example and their POS tagging are shown in figure 6.

**เป็นเครือข่ายชนิดพิเศษที่มีระดับการยึด*เกาะ*กันสูงและโปร่งใสต่อผู้ใช้**
**สารที่มีคุณสมบัติยึด*เกาะ*ผิวพลาสติกได้ดี**
**เรืออื่น ๆ ที่ขึ้นล่องอยู่ระหว่างบริเวณอ่าวเล็ก ๆ และ*เกาะ*ต่าง ๆนอกฝั่งทะเล**
**เด็กวัยรุ่นที่มีบุคลิกลักษณะไม่เป็นตัวของตัวเองต้องค่อย*เกาะ*เกี่ยวผู้อื่นมันจะมีสาเหตุ**

(ระดับ NCMN)(การ FIXN)(ยึด VACT)***(TARGET เกาะ)***(กัน PDMN)(สูง VAT
(มี VSTA)(คุณสมบัติ NCMN)(ยึด VACT)***(TARGET เกาะ)***(ผิว NCMN)(พลาส
(อ่าว NCMN)(เล็ก VATT)(ๆ NPRP)(และ JCRG)***(TARGET เกาะ)***(ต่าง VSTA
(และ JCRG)(ต้อง XCMM)(คอย VACT)***(TARGET เกาะ)***(เกี่ยว VSTA)(พึ่งพา

Fig.6 Example data tagging.

In figure 6, the underline word is the target word that we are considering as an ambiguous word. We transform the raw data into two lists, word list and part of speech list (see table 4 for more details).

The senses of word are defined by using Lexitron dictionary. For example the distribution of sense of *เกาะ* /kor/ is presented in Table1.

Table 1 Definitions of sense เกาะ /kor/

| Senses | Definitions |
|---|---|
| Island(1) | A tract of land surrounded by water.<br>เขาไปที่***เกาะ***สีชังเมื่อวานนี้ |
| To gather(2) | To bring together, Collect.<br>มันอยู่รวม***เกาะ***กลุ่มกันดีทีเดียว |
| To hold(3) | To catch something.<br>ลิงตัวนั้น***เกาะ***ต้นไม้อยู่ |
| To be a parasite(4) | To seeks support from another without making an adequate return.<br>เขายังคง***เกาะ***พ่อแม่กินเหมือนเดิม |
| Name(5) | An identifying name or title<br>เขาอาศัยอยู่ที่อำเภอ***เกาะ***จังหวัดตรัง |
| Others(6) | Others case.<br>นั่งชมบอลแบบ***เกาะ***สนาม<br>ผู้ชายคนนั้น***เกาะ***แกะผู้หญิง |

Step 2: From word and POS lists assign integer id (feature ID) to each word and POS.

1001 W[กัน]
1002 W[การ]
1003 W[ยึด]
…..
1067 T[JCRG]
1068 T[VSTA]
1069 T[PDMN]

Step 3: After obtaining feature ID, we then get features information in order to build up the training set (as shown in Table 2). From the example, we test with feature of word and POS at the window size (span) = 2 to corpus word /kor/.

Table 2 Example Features

| Corpus Data: (ที่ JSBR) (มี VSTA) (ระดับ NCMN)(การ FIXN) (ยึด VACT) ***(TARGET เกาะ)*** (กัน PDMN) (สูง VATT) (และ JCRG) (โปร่งใส VSTA) | | | |
|---|---|---|---|
| Feature name | Feature obtained | Feature explanation | Sense number |
| Context word span 2 | {(การ), (ยึด), (กัน), (สูง)} | Tuple contain word left and right distance 2 | 2 |
| POS span 2 | {(FIXN),(VACT), (PDMN), (VATT)} | Tuple contain POS distance 2 | 2 |

In the test sentences, there are 10 ambiguous words (ที่ /Tee/, เกาะ /Kor/, กิน /Kin/, แก้ /Kae/, ขัน /Kan/, หัก /Hug/, เก็บ /Kept/, สาง /Sang/, ใส่ /Sai/, คน /Kon/). The number of neighbor words (span) or POS is changed.

Step 4: Sense of correct answer is added into each line to build up a train set with the correct answer. Each ambiguous word consists of 1,000 sentences. We train and test data ten times using 10% of random test set and 90% train set and then calculate the average accuracy of learning algorithms of the ten.

Step 5: The learning algorithm that we use are SVM, Snow, and Naive bayes.

## 4. RESULTS

We compute performance of 10 words. The recognition accuracy is shown in table 3, the 1st column shows an ambiguous word, 2nd column shows feature (WORD[-2,2] is context word span 2 previous words and 2 words after) and the 3rd, 4th, 5th column show the recognition accuracy from the 3 algorithms. It shows that SVM provide the most accurate result in comparison to other algorithms. The feature word is the appropriate for all selected ambiguous words. The span number depends on each ambiguous word, but for span greater than 4 has consistent lower accuracy.

## 5. CONCLUSION AND FUTRUE WORK

In this paper, we compare 3 machine learning algorithms SVM, Naive Bayes and Snow in solving WSD in Thai language. The comparison performs on 10 ambiguous words with different features word and POS. Word feature gives the most accurate result. SVM gives the most accuracy in solving WSD problem in Thai language.

For the further step, Firstly, more experiment on other ambiguous word is needed, the size of corpus data should be expanded. Secondly try to improve accuracy by modifying feature, such as the frequency of word features, collocation, prefix, suffix, window size.

## REFERENCES


[1] Wipharuk kanokrattananukul, "Word sense disambiguation in Thai using decisiton list," *Master Thesis*, Department of Linguistics Faculty of Arts Chulalogkorn University Academic, Year 2001.

[2] Sornlertlamvanich, Virach, Charoenporn, Thatsanee. And Isahara, Histoshi. "ORCHID: Thai Part-Of-Speech Tagged Corpus," *Technical Report Orchid Corpus*, National Electronics and Computer Technology Center: 1997.

[3] Charoenpornsawat, P. 1998. "Feature-Based Thai Word Segmentation." *Master Thesis*, Department of Engineering. Chulalongkorn University.

[4] Thorsten Joachims, "Text Categorization with support vector machine," *In proc.Of European Conference on Machine Learning (ECML),* 1998.

[5] Dan Roth, "Learning to Resolve Natural Language Ambiguities: A Unified Approach," Department of Computer Science University of Illinois at Urbana-Champaign Urbana, IL 61801.

[6] Yirong Shen and Jing Jiang. "Improving the Performance of Naïve Bayes for Text Classification," *CS224N Spring*, 2003

[7] Chin-Chung Chang and Chih-Jen Lin, "LIBSVM: a Library for Support Vector machine," 2002.

[8] Andrew J. Carlson, Chad M. Cumby, Jeff L.Rosen, Dan Roth. "Snow User Guide," Cognitive Computation Group, Computer Science Department University of Illinois, Urbana/Champaign.

[9] SWATH, "Smart Word Analysis for Thai," National Electronics and Computer Technology Center.

[10] Sornlertlamvanich, Virach, Potipiti, Tanapong and Charoenporn, Thatsanee. "Automatic Corpus-based Thai Word Extraction with C4.5 Learning algorithm," *Proceeding of the 18th International Conference on Computational Linguistics, Saarbrucken, Germany,* pp802-807,200.

Table 3 Experiment results

| Word | Feature | SVM (accuracy %) | | | NB (accuracy %) | | | SNOW (accuracy %) | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Min | Max | Average | Min | Max | Average | Min | Max | Average |
| ที่/Tee/ | WORD[-1,1] | 75 | 90 | **83.83** | 75 | 85 | 81.29 | 65 | 81 | 73.66 |
| เกาะ/Kor/ | WORD[-3,3] | 77 | 91 | **84.30** | 81 | 91 | 84.20 | 69 | 78 | 72.90 |
| กิน/Kin/ | WORD[-1,1] | 88 | 95 | **92.30** | 85 | 92 | 84.20 | 69 | 78 | 72.90 |
| แก้/Kae/ | WORD[-1,1] | 82 | 94 | **90.00** | 85 | 92 | 87.88 | 84 | 91 | 87.80 |
| ขัน/Kan/ | WORD[-1,1] | 91 | 100 | **95.00** | 64 | 88 | 73.80 | 69 | 82 | 70.50 |
| หัก/Hug/ | WORD[-2,2] | 92 | 99 | **96.60** | 91 | 96 | 93.60 | 91 | 97 | 93.50 |
| เก็บ/Kept/ | WORD[-2,2] | 81 | 92 | **85.10** | 76 | 87 | 79.50 | 72 | 84 | 78.70 |
| สาง/Sang/ | WORD[-4,4] | 91 | 98 | **94.70** | 82 | 90 | 86.30 | 87 | 97 | 93.70 |
| ใส่/Sai/ | WORD[-2,2] | 75 | 88 | **83.12** | 76 | 87 | 80.43 | 74 | 83 | 77.21 |
| คน/Kon/ | WORD[-1,1] | 79 | 89 | **85.30** | 77 | 88 | 81.20 | 71 | 84 | 75.50 |

Table 4 Example Thai Part-of-Speech from Orchid [2].

| POS | Description | Example |
|---|---|---|
| NPRP | Proper noun | โค้ก, พระอาทิตย์ (Coke, The sun) |
| NCNM | Cardinal number | หนึ่ง, สอง, สาม (1, 2, 3) |
| NCMN | Common noun | หนังสือ, อาหาร (book, food) |
| JCRG | Coordinating conjunction | และ, หรือ, แต่ (and, or, but) |
| VSTA | Stative verb | เห็น, รู้ (see, know) |
| VACT | Active verb | ทำงาน, ร้องเพลง (work, sing) |
| PDMN | Demonstrative pronoun | นี่, นั่น (this, that) |
| PNTR | Interrogative pronoun | ใคร, อะไร, อย่างไร (who, what, how) |