

A Word and Character-Cluster Hybrid Model for Thai Word Segmentation

Canasai Kruengkrai, Kiyotaka Uchimoto, Jun'ichi Kazama,
Kentaro Torisawa, Hitoshi Isahara, and Chuleerat Jaruskulchai

Abstract—In this paper, we describe our system used in the InterBEST 2009 Thai Word Segmentation Shared Task. Our system is based on a word and character-cluster hybrid model which can effectively handle both known and unknown words. In addition, our model can be integrated with simple strategies for reducing annotation inconsistencies. Experimental results on in-domain and out-of-domain test data sets show the effectiveness of our system.

I. INTRODUCTION

SIMILAR to many Asian languages such as Chinese and Japanese, Thai is written without spaces between words. Word segmentation is an indispensable step in natural language processing (NLP) for Thai. NLP tasks, such as word sense disambiguation, summarization, and machine translation, need results of word segmentation as their inputs.

Research in automatic Thai word segmentation can be dated back to 1980's when word segmentation was initially considered as a simple process of syllable separation [1], [2], [3]. However, more meaningful units, words, are often required in many applications. Recent studies have applied machine learning techniques to Thai word segmentation, e.g., Markov models [4]; RIPPER and Winnow [5]; Decision Trees [6], [7]; naive Bayes, support vector machines, and conditional random fields [8]; and other unsupervised learning techniques [9], [10]. However, most previous methods still suffer from a serious problem caused by unknown words, which are defined as words that are not found in a training corpus nor in a system's word dictionary. The word boundaries of unknown words, which are very difficult to identify, cause numerous errors. Post-processing like unknown word identification [11] is a possible way to handle unknown words, but the performance of post-processing has been dismal.

In this paper, we propose a new approach to Thai word segmentation based on a word and character-cluster hybrid model, which represents the search space with a lattice consisting of word level and character-cluster level nodes. While word level nodes can handle *known* word ambiguities, character-cluster level nodes, which represent inseparable units of Thai

contiguous characters, can consistently handle *unknown* words. We describe an efficient method for training our model based on discriminative online learning. In addition, our model can be integrated with simple strategies for reducing annotation inconsistencies. We participated in the InterBEST 2009 Thai Word Segmentation Shared Task [12] and evaluated our system on test data sets consisting of 12 domains where 4 domains were not included in the training data. Experimental results on in-domain and out-of-domain test data sets show the effectiveness of our system.

The outline of this paper is as follows: Section II presents our word and character-cluster hybrid model, Section III describes learning and inference schemes, Section IV explains our development, Section V discusses inconsistencies in the InterBEST 2009 annotation, Section VI shows results on the test data sets, and Section VII concludes the paper.

II. WORD AND CHARACTER-CLUSTER HYBRID MODEL

Our word and character-cluster hybrid model is a novel variation of the word-character hybrid model [13], [14], which has shown excellent performance in Chinese and Japanese. Our modified model is motivated by a particular characteristic of the Thai writing system, which is an alphabetic system [15]. While a Chinese or Japanese character can be a word, a Thai character cannot be a word by itself. Several Thai characters have to be combined to form a Thai word. For example, the word “อิ” (if) contains three characters: the consonant “อิ”, the tone mark “ ”, and the vowel “ิ”. Particular sequences of Thai characters do not make sense if they are divided in the middle. Therefore, an unnecessarily large lattice can be produced if character level nodes are used for Thai.

In this paper, we broadly classify Thai characters into five types: consonants, final consonants, vowels, tone marks, and numerals. Table I shows all character types used in our model (including five non-Thai character types). In the Thai writing system, tone marks and vowels have to be associated with consonants to produce correct spellings. As shown in the previous example, the tone mark “ ” and the vowel “ิ” have to be combined with the consonant “อิ”. Based on this condition, we can generate unambiguous units called *character clusters* using Thai spelling rules. A character cluster can function as an inseparable unit which is larger than or equal to a character and smaller than or equal to a word. Once the character cluster is produced, it cannot be further divided into smaller units. The concept of character clusters was first proposed in [16] for document indexing, and we here apply it to our model. Using character clusters instead of characters can drastically reduce the lattice size in our hybrid model.

Canasai Kruengkrai, Kiyotaka Uchimoto, Jun'ichi Kazama, Kentaro Torisawa, and Hitoshi Isahara are with the National Institute of Information and Communications Technology (NICT), 3-5 Hikaridai, Seika-cho, Sorakugun, Kyoto 619-0289 Japan

{canasai,uchimoto,kazama,torisawa,isahara}@nict.go.jp
Canasai Kruengkrai and Hitoshi Isahara are also with the Graduate School of Engineering, Kobe University, 1-1 Rokkodai-cho, Nada-ku, Kobe 657-8501 Japan

Chuleerat Jaruskulchai is with the Department of Computer Science, Faculty of Science, Kasetsart University, 50 Phahon Yothin Rd, Chatuchak Bangkok 10900 Thailand fscichj@ku.ac.th

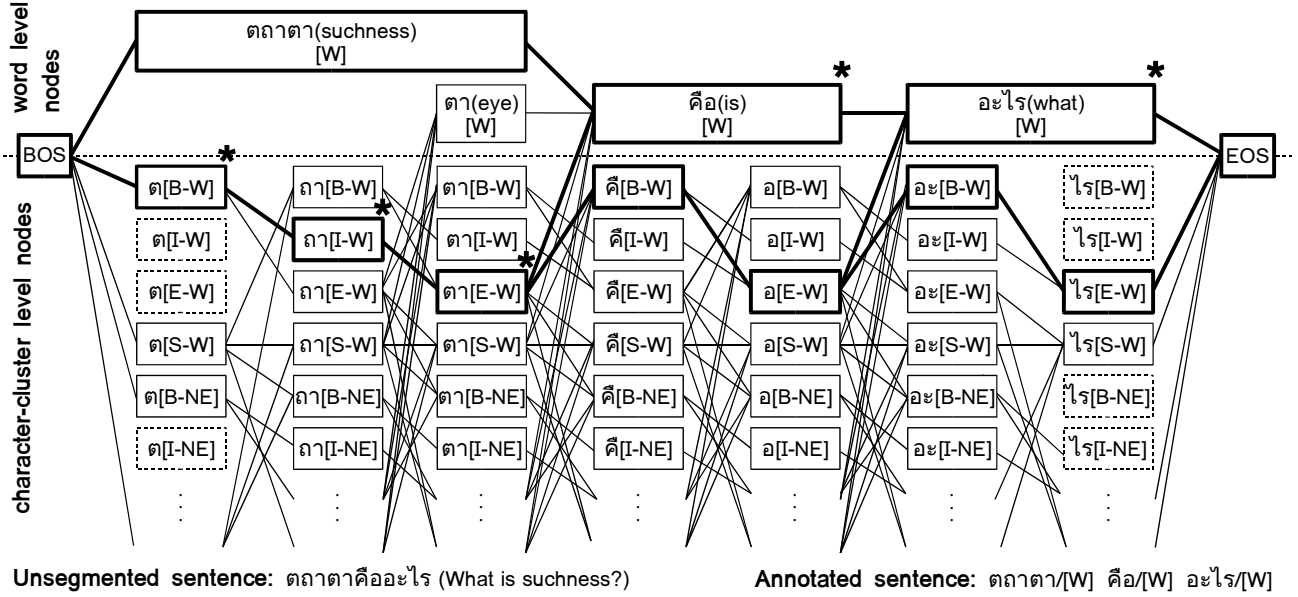


Fig. 1. Lattice used in our word and character-cluster hybrid model. Paths corresponding to the annotated sentence are marked in bold. The correct path used for training is marked with asterisks (*).

hence the characteristics of unknown words can be learned. For example, in Figure 1, if the word “ตถตา” is a rare word, we use the character-cluster level nodes instead of the word level node as the correct nodes. As a result, the correct path can contain both word level and character-cluster level nodes (marked with asterisks (*)).

Note that some nodes and state transitions are not allowed. For example, I and E nodes cannot occur at the beginning of the lattice (marked with dashed boxes), and the transitions from I to B nodes are also forbidden. These nodes and transitions are ignored during the lattice construction processing.

III. LEARNING AND INFERENCE

In this section, we describe our learning algorithm based on discriminative online learning. The goal is to learn a mapping from inputs (unsegmented sentences) $\mathbf{x} \in \mathcal{X}$ to outputs (segmented paths) $\mathbf{y} \in \mathcal{Y}$ based on training samples of input-output pairs $\mathcal{S} = \{(\mathbf{x}_t, \mathbf{y}_t)\}_{t=1}^T$. We apply a generalized version of the Margin Infused Relaxed Algorithm (MIRA) [19], [20] that can incorporate k -best decoding in the update procedure. To understand the concept of k -best MIRA, we begin with a linear score function:

$$s(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \langle \mathbf{w}, \mathbf{f}(\mathbf{x}, \mathbf{y}) \rangle, \quad (1)$$

where \mathbf{w} is a weight vector and \mathbf{f} is a feature representation of an input \mathbf{x} and an output \mathbf{y} .

Learning a mapping between an input-output pair corresponds to finding a weight vector \mathbf{w} such that the best scoring path of a given sentence is the same as (or close to) the correct path. Given a training example $(\mathbf{x}_t, \mathbf{y}_t)$, MIRA tries to establish a margin between the score of the correct path $s(\mathbf{x}_t, \mathbf{y}_t; \mathbf{w})$ and the score of the best candidate path $s(\mathbf{x}_t, \hat{\mathbf{y}}; \mathbf{w})$ based on the current weight vector \mathbf{w} that is proportional to a loss function $L(\mathbf{y}_t, \hat{\mathbf{y}})$.

In each iteration, MIRA updates the weight vector \mathbf{w} by keeping the norm of the change in the weight vector as small as possible. With this framework, we can formulate the optimization problem as follows [20]:

$$\begin{aligned} \mathbf{w}^{(i+1)} &= \operatorname{argmin}_{\mathbf{w}} \|\mathbf{w} - \mathbf{w}^{(i)}\| \\ \text{s.t. } \forall \hat{\mathbf{y}} \in \operatorname{best}_k(\mathbf{x}_t; \mathbf{w}^{(i)}) : \\ s(\mathbf{x}_t, \mathbf{y}_t; \mathbf{w}) - s(\mathbf{x}_t, \hat{\mathbf{y}}; \mathbf{w}) &\geq L(\mathbf{y}_t, \hat{\mathbf{y}}), \end{aligned} \quad (2)$$

where $\operatorname{best}_k(\mathbf{x}_t; \mathbf{w}^{(i)}) \in \mathcal{Y}_t$ represents a set of top k -best paths given the weight vector $\mathbf{w}^{(i)}$. The above quadratic programming (QP) problem can be solved using Hildreth’s algorithm [21], and $\operatorname{best}_k(\mathbf{x}_t; \mathbf{w}^{(i)})$ can be generated using a dynamic programming search [22].

We calculate the loss function through false positives (FP) and false negatives (FN). Here, FP means the number of output nodes that are not in the correct path, and FN means the number of nodes in the correct path that cannot be recognized by the system. We define the loss function by:

$$L(\mathbf{y}_t, \hat{\mathbf{y}}) = FP + FN. \quad (3)$$

This loss function can reflect how bad the predicted path $\hat{\mathbf{y}}$ is compared to the correct path \mathbf{y}_t .

Our basic features are similar to those of [13]. The main difference is in the design of character types. Here, we instead use character-cluster classes derived from spelling rules (Table III) as features.

In summary, we use k -best MIRA to iteratively update $\mathbf{w}^{(i)}$. The final weight vector \mathbf{w} is the average of the weight vectors after each iteration. As reported in [23], [24], parameter averaging can effectively avoid overfitting. For inference, we can use Viterbi-style decoding to search for the most likely path \mathbf{y}^* for a given sentence \mathbf{x} where:

$$\mathbf{y}^* = \operatorname{argmax}_{\mathbf{y} \in \mathcal{Y}} s(\mathbf{x}, \mathbf{y}; \mathbf{w}). \quad (4)$$

TABLE IV
TRAINING AND TEST DATA STATISTICS IN OUR DEVELOPMENT

Domain	Training Set		Test Set		OOV Rate
	# of sent.	# of words	# of sent.	# of words	
Article	14,553	997,467	2,568	196,395	0.0173 (3,388/196,395)
Buddhism	5,398	499,663	952	38,192	0.0283 (1,079/38,192)
Encyclopedia	43,013	1,003,609	7,591	157,093	0.0192 (3,021/15,7093)
Law	18,817	675,383	3,320	52,548	0.0118 (618/52,548)
News	26,533	1,346,847	4,682	312,606	0.0210 (6,557/312,606)
Novel	42,607	1,400,632	7,519	257,942	0.0245 (6,313/257,942)
Talk	3,479	345,863	614	62,983	0.0185 (1,166/62,983)
Wiki	16,952	661,827	2,992	129,711	0.0634 (8,222/129,711)

TABLE V
RESULTS OF OUR DEVELOPMENT

Domain	MM _{base}	MM _{top}	Our System									
			Trained on each domain					Trained on all domains				
			$r = 1$	$r = 2$	$r = 3$	$r = 4$	$r = 5$	$r = 1$	$r = 2$	$r = 3$	$r = 4$	$r = 5$
Article	91.25	98.13	97.73	97.74	97.71	97.63	97.70	97.75	97.78	97.78	97.75	97.80
Buddhism	93.79	99.11	98.05	98.45	98.85	98.77	98.78	98.18	98.19	98.17	98.36	98.36
Encyclopedia	90.25	98.42	96.87	96.82	96.87	96.87	96.85	97.14	97.17	97.18	97.17	97.15
Law	97.61	99.45	98.71	98.41	98.32	98.32	98.19	97.68	97.50	97.52	97.51	97.48
News	89.52	98.73	97.25	97.23	97.25	97.25	97.27	96.96	96.97	96.95	96.95	96.94
Novel	87.04	97.25	96.19	96.22	96.23	96.23	96.23	96.79	96.85	96.88	96.85	96.80
Talk	94.44	99.57	97.94	97.85	97.89	97.83	97.81	98.26	98.26	98.26	98.24	98.23
Wiki	88.13	99.02	95.70	95.69	95.55	95.53	95.48	95.66	95.70	95.73	95.74	95.68

IV. DEVELOPMENT

Our development consists of performing parameter tuning and checking the performance of models trained on each domain and all domains. Our model has three tunable parameters: the number of training iterations N ; the number of top k -best paths; and the rare word threshold r . Since we were interested in finding an optimal combination of word level and character-cluster level nodes for training, we focused on tuning r . We fixed $N = 10$ and $k = 5$ for all experiments, and varied r in the range of $[1, 5]$.

A. Data Sets

We used the training data sets from the InterBEST 2009 Thai Word Segmentation Shared Task [12] consisting of 8 domains: Article, Buddhism, Encyclopedia, Law, News, Novel, Talk, and Wiki. We made an approximate 85%/15% split for training/test set in each domain. Table IV shows training and test data statistics in our development.

B. Development Results

Table V shows F -measure results in our development. Following [25], we also conducted the baseline and topline experiments to estimate the lower and upper performance bounds. The well-known maximum matching algorithm that uses a word dictionary is used for these experiments. For the baseline experiment (MM_{base}), we construct the word dictionary from words in the training set. For the topline experiment (MM_{top}), we instead use words in the test set for making the word dictionary.

From the development results, we see that our system performs significantly better than the baseline and relatively

well compared to the topline. For our final in-domain model, we selected r which gave the best F -measure score in each domain. Thus, we set $r_{\text{article}} = 2$, $r_{\text{buddhism}} = 3$, $r_{\text{encyclopedia}} = 3$, $r_{\text{law}} = 1$, $r_{\text{news}} = 5$, $r_{\text{novel}} = 3$, $r_{\text{talk}} = 1$, and $r_{\text{wiki}} = 1$. For our final general-domain model, since the optimal r values varied across domains, we set $r_{\text{general}} = 3$ which provided good F -measure scores in all domains.

V. INCONSISTENCIES IN INTERBEST 2009 ANNOTATION

Error analysis in our development revealed inconsistencies in InterBEST 2009 annotation. Some inconsistencies distributed like random noise, but some did not. In this section, we discuss particular annotation inconsistencies and methods for automatically detecting and correcting them.

InterBEST 2009 annotation follows guidelines that provide general principles for segmenting words into their smallest units of meaning [12]. In our development, we found that segmentation inconsistencies exist in several word categories. For example, the compound word “หลักการ” (principle) is segmented 160 times as “หลัก|การ” (principle|-*suffix*) and 640 times as “หลัก|การ” on the training data from all domains. Table VI shows other examples of segmentation disagreements found in the training data.

Note that the different segmentations for a word are allowed, and contexts are needed to make decision. For example, the string “ตากลม” can be segmented into “ตา|กลม” (eye|round) or “ตา|ลม” (to dry|wind) depending on the contexts. This case is an example of genuine ambiguities in the language. However, we found that many segmentation disagreements in the InterBEST 2009 training data were caused by annotation inconsistencies. These annotation inconsistencies create problems for both training and evaluation [26]. While annotation

TABLE VII
RESULTS SHOWING THE PERFORMANCE OF OUR SYSTEM ON THE TEST DATA SETS USING DIFFERENT TRAINING STRATEGIES

Domain	Trained on each domain			Trained on all domains									InterBEST 2009 (submitted)
	<i>R</i>	<i>P</i>	<i>F</i>	Original			Revised f_{comp}			Revised $\kappa < 0.0$			
				<i>R</i>	<i>P</i>	<i>F</i>	<i>R</i>	<i>P</i>	<i>F</i>	<i>R</i>	<i>P</i>	<i>F</i>	
Article	98.206	96.903	97.550	98.491	96.986	97.732	98.571	97.110	97.835	98.584	97.104	97.839	97.839
Buddhism	98.295	97.374	97.832	98.763	98.029	98.394	98.854	98.197	98.524	98.856	98.136	98.494	98.524
Encyclopedia	97.589	96.480	97.031	98.136	97.239	97.685	98.137	97.263	97.698	98.077	97.205	97.639	97.698
Law	98.933	97.529	98.226	98.279	96.724	97.495	98.168	96.850	97.505	98.140	96.754	97.442	98.226
News	97.775	96.295	97.029	97.726	95.134	96.412	97.943	95.474	96.693	97.969	95.503	96.721	97.029
Novel	97.036	95.494	96.259	97.232	95.982	96.603	97.339	96.204	96.768	97.372	96.197	96.781	96.781
Talk	97.812	97.386	97.598	98.323	98.172	98.247	98.335	98.204	98.269	98.327	98.174	98.250	98.269
Wiki	97.025	95.253	96.131	96.928	95.089	96.000	97.260	95.384	96.313	97.168	95.248	96.198	96.313
NSC	–	–	–	97.548	97.679	97.613	97.676	97.887	97.781	97.796	97.938	97.867	97.867
Old doc	–	–	–	95.603	94.221	94.907	95.472	94.006	94.733	95.434	93.880	94.650	94.907
Royal news	–	–	–	95.753	88.594	92.034	96.075	88.809	92.299	95.949	88.506	92.077	92.299
TV news	–	–	–	97.569	95.499	96.523	97.598	95.538	96.557	97.562	95.480	96.510	96.557

TABLE VI
EXAMPLES OF SEGMENTATION DISAGREEMENTS FOUND IN THE INTERBEST 2009 TRAINING DATA [# OF OCCURRENCES]

จัด การ [575]	จัด การ [2,440]
ดำ เนิน การ [756]	ดำ เนิน การ [2,418]
ตัวอย่าง [101]	ตัวอย่าง [1,613]
ความ หมาย [134]	ความ หมาย [1,577]
ร้อย ล๑ [77]	ร้อย ล๑ [1,493]
ตกล ง [206]	ตกล ง [1,313]
ให้ การ [373]	ให้ การ [1,306]
วิธี การ [1,176]	วิธี การ [1,233]
จน ถึง [512]	จน ถึง [1,029]
แต่ ว่า [181]	แต่ ว่า [714]
แล้ว แต่ [271]	แล้ว แต่ [651]
ต้น ไม้ [199]	ต้น ไม้ [629]
สภา ผู้ แทน ราษฎร [130]	สภา ผู้ แทน ราษฎร [452]
ลูก หมู [103]	ลูก หมู [213]
ออก เสียง [111]	ออก เสียง [199]

inconsistencies introduce undesired patterns for learning, they also make evaluation less reliable.

We now describe our attempt to deal with annotation inconsistencies. Our motivation comes from the observation that part of annotation inconsistencies overlaps with test errors. We found that disagreements between the system output and the gold standard are sometimes caused by annotation inconsistencies. As a result, we begin by performing 10-fold cross-validation⁵ as follows:

- Divide the corpus into ten equal-sized sets.⁶
- For each trial, train the hybrid model using nine sets and

⁵Cross-validation is known to be an unbiased approach for assessing the generalization error.

⁶We first combined data from all domains into a single large corpus.

use this model to predict the remaining set.⁷

After 10-fold cross-validation, we can locate all segmentation disagreements between the system output and the gold standard in the corpus. Next, we need a quantitative measure that can distinguish between genuine ambiguities and annotation inconsistencies. We propose two simple methods: frequency comparison and agreement coefficient.

Suppose we have two segmentation disagreements, *sys* and *gold*, obtained from the system output and the gold standard, respectively. The idea of the frequency comparison method is to look back into the annotated corpus and count the numbers of *sys* and *gold* occurrences. If $freq(sys) > freq(gold)$, we change *gold* to *sys* and keep the correction rule⁸ for this disagreement; otherwise leave unchanged. As shown in Table VI, segmentation disagreements apparently occur in the annotated corpus. The second column shows alternative segmentations which were produced by our model. Note that the frequency comparison method may lead to overfitting since it tends to select many training errors for correcting.

The agreement coefficient method adds a condition to the frequency comparison method. We think of the system output as annotated results from another annotator and measure how much agreement between the system output and the gold standard using Cohen’s kappa coefficient κ . Note that other agreement coefficients can be used in this method. We chose κ since its calculation is very simple. For lack of space, we refer to [27] for a detailed description of κ and other agreement coefficients.

We revised the InterBEST 2009 training data to observe the results on the test data. Both methods made a very few changes in the training data: 0.84% (68,474/8,138,761) with 9,587 unique changes for the frequency comparison method and 0.67% (54,256/8,138,761) with 7,544 unique changes for

⁷We used nine sets for training, but constructed the system’s word dictionary from the whole corpus. Our aim is to avoid errors caused by unknown words and make the model more accurate. We also expect that the remaining test errors will consist of annotation inconsistencies rather than genuine ambiguities.

⁸An example rule is: change *gold* to *sys* if $gold = \text{“ตัวอย่าง”}$ and $sys = \text{“ตัวอย่าง”}$. Note that the correction rules of our simple methods are context-independent.

the agreement coefficient method.

VI. RESULTS

We conducted experiments on the InterBEST 2009 Thai Word Segmentation Shared Task test data sets consisting of 12 domains where 4 domains (NSC, Old doc, Royal news, and TV news) were not included in the training data, and evaluated performance through the InterBEST 2009 evaluation website.⁹ We compared different training strategies. For each domain, we trained the in-domain model with the optimal r derived from our development. We trained the general-domain model using all available training data with $r = 3$ (Original). We performed inconsistency detection and correction on the training data using the frequency comparison method (Revised $_{fcomp}$) and the agreement coefficient method (Revised $_{\kappa < 0.0}$ ¹⁰).

Results are given in Table VII. Standard evaluation metrics include recall (R), precision (P), and F -measure. The underlined scores (also shown in the last column) are our submitted results. The general-domain model trained on the original training data performs better than the in-domain model, except for Law, News, and Wiki domains. The in-domain models give the best performance on Law and News domains. Results of the general-domain model trained on the revised training data are superior to those trained on the original data, except for Old doc domain. These results indicate that our simple inconsistency detection and correction methods are useful for InterBEST 2009 annotation. Furthermore, detected inconsistencies can help to improve annotation guidelines to cover more realistic cases.

VII. CONCLUSION

We described our word and character-cluster hybrid model for Thai word segmentation. In addition, we discussed annotation inconsistencies, proposed simple methods to handle these inconsistencies, and showed very encouraging results on the benchmark data. We hope that our model can serve as a strong baseline for future developments in Thai NLP. Finally, we believe that word segmentation, POS tagging, and named entity recognition can be uniformly modeled using our framework.

ACKNOWLEDGMENTS

We would like to thank the InterBEST 2009 Thai Word Segmentation Shared Task annotators for their effort in creating the corpus and organizers for their kind support.

REFERENCES

- [1] Y. Thairatananond, *Towards the Design of a Thai Text Syllable Analyzer*. Asian Institute of Technology, Master Thesis, 1981.
- [2] S. Charnyapornpong, *A Thai syllable separation algorithm*. Asian Institute of Technology, Master Thesis, 1983.
- [3] Y. Poowarawan, "Dictionary-based thai syllable separation," in *Proceedings of the Ninth Electronics Engineering Conference*, 1986.
- [4] A. Kawtrakul and C. Thumkanon, "A statistical approach to thai morphological analyzer," in *Proceedings of of the 5th Workshop on Very Large Corpora*, 1997.
- [5] S. Meknavin, P. Charoenpornsawat, and B. Kijisirikul, "Feature-based thai word segmentation," in *Proceedings of of NLP/RS*, 1997.
- [6] V. Sornlertlamvanich, T. Potipiti, and T. Charoenporn, "Automatic corpus-based thai word extraction with the c4.5 learning algorithm," in *Proceedings of COLING*, 2000, pp. 802–807.
- [7] T. Theeramunkong and S. Usanavasin, "Non-dictionary-based thai word segmentation using decision trees," in *Proceedings of the First International Conference on Human Language Technology Research*, 2001, pp. 251–256.
- [8] C. Haruechaiyasak, S. Kongyoung, and M. N. Dailey, "A comparative study on thai word segmentation approaches," in *Proceedings of ECTI-CON*, 2008.
- [9] C. Jaruskulchai, "An automatic thai lexical acquisition from text," in *Proceedings of PRICAI*, 1998.
- [10] W. Aroonmanakun, "Collocation and thai word segmentation," in *Proceedings of the 5th SNLP & 5th Oriental COCOSDA Workshop*, 2002, pp. 68–75.
- [11] P. Charoenpornsawat, B. Kijisirikul, and S. Meknavin, "Feature-based thai unknown word boundary identification using winnow," in *Proceedings of the IEEE Asia-Pacific Conference on Circuits and Systems*, 1998.
- [12] Human Language Technology Laboratory, National Electronics and Computer Technology Center, "InterBEST 2009 Thai Word Segmentation: an International Episode," <http://thailang.nectec.or.th/interbest>.
- [13] C. Krueengkrai, K. Uchimoto, J. Kazama, Y. Wang, K. Torisawa, and H. Isahara, "An error-driven word-character hybrid model for joint chinese word segmentation and pos tagging," in *Proceedings of ACL-IJCNLP*, 2009.
- [14] T. Nakagawa and K. Uchimoto, "A hybrid approach to word segmentation and pos tagging," in *Proceedings of ACL Demo and Poster Sessions*, 2007.
- [15] N. Danvivathana, *The Thai writing system*. Forum Phonetikum 39, Helmut Buske Verlag Hamburg, 1987.
- [16] T. Theeramunkong, V. Sornlertlamvanich, T. Tanhermhong, and W. Chinnan, "Character cluster based thai information retrieval," in *Proceedings of the fifth international workshop on on Information retrieval with Asian languages*, 2000.
- [17] H. Baayen and R. Sproat, "Estimating lexical priors for low-frequency morphologically ambiguous forms," *Computational Linguistics*, vol. 22, no. 2, pp. 155–166, 1996.
- [18] A. Ratnaparkhi, "A maximum entropy model for part-of-speech tagging," in *Proceedings of EMNLP*, 1996, pp. 133–142.
- [19] K. Crammer, R. McDonald, and F. Pereira, "Scalable large-margin online learning for structured classification," in *NIPS Workshop on Learning With Structured Outputs*, 2005.
- [20] R. McDonald, *Discriminative Training and Spanning Tree Algorithms for Dependency Parsing*. University of Pennsylvania, PhD Thesis, 2006.
- [21] S. A. Z. Yair Censor, *Parallel Optimization: Theory, Algorithms, and Applications*. Oxford University Press, 1997.
- [22] M. Nagata, "A stochastic japanese morphological analyzer using a forward-DP backward-A* n-best search algorithm," in *Proceedings of the 15th International Conference on Computational Linguistics*, 1994, pp. 201–207.
- [23] M. Collins, "Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms," in *Proceedings of EMNLP*, 2002, pp. 1–8.
- [24] R. McDonald, F. Pereira, K. Ribarow, and J. Hajic, "Non-projective dependency parsing using spanning tree algorithms," in *Proceedings of HLT/EMNLP*, 2005, pp. 523–530.
- [25] R. Sproat and T. Emerson, "The first international chinese word segmentation bakeoff," in *Proceedings of the 2nd SIGHAN Workshop on Chinese Language Processing*, 2003, pp. 133–143.
- [26] M. Dickinson, *Error detection and correction in annotated corpora*. The Ohio State University, PhD Thesis, 2005.
- [27] R. Artstein and M. Poesio, "Inter-coder agreement for computational linguistics," *Computational Linguistics*, vol. 34, no. 4, pp. 555–596, 2008.

⁹The test data sets and the evaluation tool are publicly available at <http://www.hlt.nectec.or.th/Evaluation>.

¹⁰ $\kappa < 0.0$ can be interpreted as no agreement between the system output and the gold standard.