A Preliminary Study on Fundamental Thai NLP Tasks for User-generated Web Content

Anuruth Lertpiya
Theerapat Lapjaturapit
Teerapat Chaiwachirasak
Tawunrat Chalothorn
Nattasit Maharattanamalai
Nutcha Tirasaroj
Kasikorn Labs
Nonthaburi, Thailand
{anuruth.l}{teerapat.ch}{nattasit.m}{theerapat.l}{tawunrat.c}{nutcha.t}@kbtg.tech

Ekapol Chuangsuwanich
Department of Computer Engineering
Chulalongkorn University
Bangkok, Thailand
ekapol.c@chula.ac.th

***Abstract*—Existing literature on Thai NLP often focuses on formally written texts with near-perfect spellings and boundaries between words or sentences. Such assumptions, however, do not hold in real-world NLP tasks, especially when dealing with User-generated web content (UGWC). So far, existing NLP research works on actual web data have been limited, making it unclear whether and how existing techniques can be applicable to UGWC. In this paper, several basic Thai NLP algorithms (word segmentation, sentence segmentation, word error detection, word variant detection, name entity recognition) are re-investigated and benchmarked against real-world, practical UGWC data set. The difference in performance between our data set and others are compared as a guidance for future research. Our baseline sentence segmentation on UGWC data set yields an average F-measure of 0.77. For name entity recognition and word variant / error detection tasks, our system yields the accuracy of 0.93 and 0.53, respectively.**

I. Introduction

High-level NLP tasks such as Social Listening and Chatbots typically require assistance from computer systems due to the sheer amount of data to be processed. Social listening (SL) is the process of monitoring social media for mentions of specific terms (such as companies, products, or services) and identifying the public perception of such terms. Companies use SL as a mean to gauge the success of their products or services and, at the same time, receive feedback for further improvement. The process of SL requires not only constant social media monitoring but also intensive historical data processing, both of which are either impractical or very costly to perform by human. Chatbots, on the other hand, are used to aid a company's customer service with chat-based support that imitates human, allowing trivial requests from customers to be processed quickly and reducing human operator workload. As chatbots become more sophisticated, more work can be off loaded to machine. Using computation methods to understand and process Thai language (Thai Natural Language Processing: Thai NLP) is a vital part of processing the massive and exponentially increasing user-generated web content (UGWC). While the access to UGWC is readily available to researchers, the complexity of Thai language in informal context (e.g., social media) makes developing NLP models a challenge when compared to formal text corpora (e.g., news articles, wikipedia articles). Our experiments have shown that processing a UGWC corpus is more challenging than other formal text corpora such as NECTEC's BEST [1] and NECTEC's ORCHID [2]. In this paper, we propose a common metric for five NLP tasks in the context of UGWC datasets: word segmentation, sentence segmentation, word error detection, word variant detection, and named entity recognition. All five tasks can be evaluated under framework of instance-detection and range-detection. Along with the formulation, we provide performance benchmarks using character-gram and word-gram models, establishing a common ground for future comparison especially when dealing with UGWC datasets. The paper is organized into four parts. We first describe the issues of UGWC and give an overview of related works. Then, we give an overview of the corpora used in this study along with the proposed

baseline models and evaluation metrics for each task. Section III presents the results and discusses the findings. Finally, we conclude and summarize the contribution of this paper and list some of possible future works.

II. Related Works

The section outlines previous works on each of the five tasks: word segmentation, sentence segmentation, word error detection, word variant detection, and named entity recognition.

A. Word & Sentence Segmentation

Word segmentation is an active research area in Thai NLP, with publicly available corpora annotated with word boundaries (e.g., NECTEC's ORCHID corpus [2], NECTEC's BEST corpus [1]) and open-source models based on artificial neural networks (e.g., Sertis [3], Deepcut [4]). Sertis [3] word segmentation algorithm, which is based on gated recurrent unit (GRU), achieved an F1 score of 0.992 on NECTEC's BEST corpus [1]. English sentence segmentation was first formalized by Riley [5] as a process of determining whether a punctuation marks the end of a sentence. Read et al. [6] later reformulated the task as an identification problem, i.e., to identify whether each word is the ending of a sentence. They then re-evaluated the task to find that the top performing model can achieve up to an F1 score of 0.992 using rule-based techniques. However, in Thai language, words are written without any word delimiter or sentence delimiter. Moreover, the definition of some Thai words and sentences are non-trivial and sometimes debatable [7]. Previous works on machine learning based Thai sentence segmentation algorithms have relied on segmented words along with part-of-speech information [8].

B. Word Error & Word Variant Detection

State-of-the-art English text classification have been achieved with the use of deep learning by Goodfellow et al. [9]. LeCun et al. [10] concluded that one advantage of deep learning algorithms over traditional NLP methods is the ability to learn and use distributed representation such as Word2Vec [11], which allows models to generalize on data outside the training set. Although the process of embedding lookup from a pre-train bank is trivial given that the word boundaries are already obtained, UGWC brings two issues that were previously neglected in Thai NLP: word errors and word variants. Word error, within the scope of this paper, is defined as misrepresentations of the original word due to many factors such as typos (human input errors as the writer types), misunderstandings of word usage, misspellings, language misuses, or incorrect fixes from spell-correction systems. In NLP, researchers commonly classified word errors into two groups: non-word-errors and real-word-errors [12]. Non-word-errors are errors that result in nonexistent words. Meanwhile, real-word-errors are errors that resulted in words whose meanings are valid but not coherent with the context. Any of these errors would result in an invalid embedding being input into a model, hampering the model's ability to understand and process sentences correctly. Another issue faced in UGWC is word variants, the usage of different words to convey the same idea. Ruder [13] outlines that although most existing word embedding can handle contextualization and disambiguation of words in contexts, many works try to explore the problem of polysemy, the ability of words to convey multiple ideas depending on the context. However, rule-based systems, most of which are still being used in many commercial applications such as keyword search, suffer heavily from this issue [14]. As UGWC are usually written without guidelines or quality check compared to other formal documents, the use of words are dependent on the writer's preferences.

C. Named Entity Recognition

Name entity recognition (NER) is a core task for information extraction and relevant tags detection. Several handcrafted, traditional methods were used to solve NER such as Support Vector Machine [15], Naive Bayes [16], Maximum Entropy Classifier [17], Hidden Markov Models [18], Conditional Random Field [19] and Decision Tree [20]. Research from [21] [22] have shown that neural networks based models can achieve better accuracy compared to traditional models. However, these techniques are quite difficult to adapt to Thai language because of the lack of explicit word boundaries in Thai language.

III. Methodology

A. Tasks

In this section, we outline the differences in our methodology (if any) along with the summary of the models, corpora, and evaluation metrics used in each task. The details of each model, corpora, and metrics are outlined later in the next subsections.

B. Word Segmentation

Previous works on Thai NLP have utilized corpora under the assumption that a given text is made up of only

words [23], [24]. Both NECTEC's BEST [1] and NECTEC's ORCHID [2] have been annotated under this assumption. However, the UGWC corpus used in this paper was labeled using ranges and thus requires a different evaluation metric (i.e., range detection metric) compared to previous works, which only measure the models' ability to identify the ending of position of a word. For word segmentation, the char-gram is tested on 3 datasets, NECTEC's BEST [1], NECTEC's ORCHID [2], and our UGWC dataset using the range detection metric. Section IV gives more details of the UGWC corpus used in this paper.

C. Sentence Segmentation

Due the lack of sentence boundaries annotation on NECTEC's BEST corpus [1], only NECTEC's ORCHID [2] and our UGWC dataset are evaluated in this task. Our UGWC corpus was labeled using ranges, allowing the corpus to contain text which cannot be identified as parts of a sentence (e.g., leftover markups from HTML, character encoding). Thus, the range detection metric is employed in this task. Since UGWC may contain text that may not be identified as words, the sentence segmentation task is performed as the first step of the NLP pipeline to extract relevant portions. Thus, we will only consider the raw text without any information from the word segmentation task. The char-gram is experimented on the sentence segmentation task on 2 datasets, NECTEC's ORCHID [2] and our UGWC corpus using range detection metric.

D. Word Error Detection & Word Variant Detection

In the scope of this corpus, the task of word error detection and word variant detection are viewed as a process of detecting corractable words in the data for the purpose of correcting them later on. However, word correction is outside the scope of any publicly available Thai corpora at the time of this research. In addition to experimenting on our UGWC dataset, we also create an erroneous corpus by inserting errors into clean, existing public corpora. The process of error injection is inspired by the Damerau–Levenshtein distance function from Setiadi [25] where edits are either insertion, deletion, substitution, or transpose of adjacent characters. For the task of word variants, only performance numbers on our corpus are provided as there is no realistic way to simulate word variance. The char-gram is experimented on 3 datasets (i.e., two simulated errorneous corpora and our UGWC corpus) using both instance detection metric and range detection metric.

E. Named Entity Recognition

Named entity recognition is not a new problem. However, there are only a few public Thai named entity corpora. For this task, we only use NECTEC's BEST corpus [1] and UGWC corpus. Named entities in NECTEC's BEST corpus [1] were fully labelled with word boundaries. On the other hand, named entities in our UGWC corpus only partly contain manually-tagged word boundaries. Since we want to evaluate word-gram method against char-gram method. We use an open-source word segmentation program by Sertis [3] as a pre-processing step to obtain word boundaries. The char-gram method is experimented on the two datasets using both instance detection metric and range detection metric. The word-gram method is experimented on the two datasets using only instance detection metric.

IV. Corpus

A. Public Corpora

Two publicly available corpora are evaluated in this paper. First is NECTEC's ORCHID [2] which was made in collaboration between Communications Research Laboratory (CRL) of Japan and National Electronics and Computer Technology Center (NECTEC) of Thailand. It is also one of the first publicly available corpora in Thai language. Second is NECTEC's BEST [1] from NECTEC, which was published as a corpus for the annual NECTEC BEST competition in 2009.

B. UGWC Corpus

The UGWC corpus used in this research consists of conversational text data related to financial domain. The data was collected from social media pages during a 3-month period (from January 2017 to March 2017). The corpus contains over 9 million raw characters along with manually-tagged annotations for word boundaries, sentence boundaries, word errors, word variants, and named entities. The annotation process on the UGWC corpus is described below. Please note that not the whole corpus was fully annotated for all of the five tasks (i.e., some parts of the corpus were annotated for some tasks only).

1) Word boundary: The guidelines proposed by Aroonmanakun et al. [7] was used for labeling word boundaries. Linguists were tasked to annotate words by identify ranges of characters in the collected raw text. This is similar to how NECTEC's ORCHID [2] corpus was labeled, as each word has its beginning and ending locations tagged. However, as our UGWC corpus may contain portions that

are not words (e.g., leftover markups), our annotation system does not require each word to be adjacent to the next word. A total of 3,163 lines out of 70,079 lines were annotated (4.5%), covering a total of 291,759 out of 9,948,378 characters (2.9%) making up of 31,619 words.

2) Sentence boundary: Our goal is not to develop a corpus that can perfectly separate boundaries in an ideal manner, but to identify meaningful segments from noisy text for higher-level NLP tasks and to create simpler and more defined tasks for the models to learn. For the sentence segmentation task, each annotation indicates if a character is either the beginning of a range that forms a sentence, the end of a range, or neither. Since Thai text does not have obvious sentence boundaries like full stop (.) in English, criterias from Aroonmanakun et al. [7] is used to indicate sentence boundaries in this research. First, the change of topics is used to indicate the beginning of new sentences. When the topic is changed, the sentence usually begins with a new subject or a conjunction such as อย่างไรก็ตาม (however), นอกจากนี้ (moreover). Second, a new subject and the conjunction joining sentences can be used as clues to indicate the new sentences. On the other hand, particles are used to indicate the end of sentences as they usually appear at sentence-ending positions. Other clues are punctuations and symbols. For example, the exclamation marks (!) and question marks (?) usually occur at the end of sentences so they can imply the sentence breaks. Moreover, bullets normally appear at the beginning of phrases or sentences; therefore, they can point to the beginning of new sentences. The sentences are also annotated as ranges as with the word boundaries. A total of 46,878 lines out of 70,079 lines were annotated (67%), covering a total of 6,080,134 out of 9,948,378 characters (80%) and making up of 143,964 sentences.

3) Word error and variant: For simplicity in labeling our data, word errors and word variant are labeled at the same time. Their labels are split into 7 categories: “misspelled words/errors”, “derivatives”, “slangs/new words”, “spoonerisms”, “transliterated words”, “abbreviations”, and “others”. "Misspelled words/errors" label are used for words that are spelled incorrectly (e.g. ‘ธนคาร’ instead of ‘ธนาคาร’). "Derivatives" are words with slight character changes for some purposes such as brevity. For example, ‘น่ามคาน’ is a derivative that comes from ‘น่ารำคาญ’. ‘น่าม’ is a blend word. It is formed by blending ‘น่า’ and ‘รำ’ to make the word sound like a spoken word. "Slangs/new words" are informal words that are used in a particular group or a particular period. For example, ‘อ้อย’ normally means “sugar cane” but some people use ‘อ้อย’ to mean “flirt” which is the new meaning. Then, ‘อ้อย’ in the context that means “flirt” should be marked as a slang. "Spoonerisms" are words in Thai that could be formed by transposing the sound. For example, ‘พับกบ’ /phap3 kop1/ comes from ‘พบกับ’ /phop3 kap1/. In this case, vowels of the first and last syllables are switched and hence these two words are considered to be spoonerisms. "Transliterated words" represent foreign words that are written in Thai language such as ‘ฟรี’ is used for ‘Free’. "Abbreviations" label is used for both formal and informal abbreviations. Formal abbreviations normally found in normal context such as news and articles. On the other hand, informal abbreviations are usually found only in social contexts such as ‘พน’ which used for represent ‘พรุ่งนี้’ (tomorrow). "Others" label is used for labelling words that cannot be found in Royal Thai Dictionary [26] including named entities, incorrectly segmented words and onomatopoeia such as ‘โรบินสัน’ (Robinson), which is the name of shopping mall, or ‘โอ๊ย’, which is an interjection. The linguists are tasked to review each sentence and check if any word is in any of the seven categories. Table I shows the result of the annotation done by the linguists. A total of 174,771 words out of 3,767,224 words (4.64%) were annotated as one of these 7 categories. In addition, Table II compare the amount of word error for each corpus. Note that errors in NECTEC’s BEST [1] and NECTEC’s ORCHID [2] are simulated using a simple algorithm that mimics human error.

4) Named entity recognition: Named entity recognition could be considered as a detection problem with

Table I
UGWC CORPUS DETAILS

Class	Token	Character
Total	3767224	14403448
Total error+variant	174771	955416
Transliterated	64648	355660
Misspelling	41804	167911
Derivatives	14450	61333
Abbreviations	22787	58116
Slangs/ New words	984	4566
Spoonerisms	62	369
Other	41119	307461

Table II
WORD ERROR

Type	ORCHID		BEST		UGWC	
	Word	*Character*	*Word*	*Character*	*Word*	*Character*
Total	414343	1569535	4485126	16344100	3767224	14403448
Word Error	57461	265999	575830	2444906	174771	955416
Error/Total	13%	16%	12%	14.96%	4.64%	6.60%

multiple classes. The NER corpus contains manually-tagged annotations for five categories: (1) The names of persons include full names, nicknames, and alias. (2) The names of locations include natural landscapes, man-made structures, buildings, and organization names that refer to locations such as 'ไปที่ธนาคารกสิกร' (go to Kasikorn bank). (3) The names of organizations including both government and non-government organizations, companies, and metonyms (i.e., common words used to refer to the organizations instead of their official names such as แบงก์เขียว refers to K-bank). (4) The names of products including brand names, trademarks, products' series. (5) The names of other entities which are not mentioned in the previous classes.

V. Model

Char-grams and word-grams were used in the research. The use of character-grams in Thai NLP was first explored by Watcharabutsarakham [27] in the domain of spell correction on OCR documents. The result concluded that the model could perform better but failed to generalize on larger grams due to limited training data. In our experiments, we found that character-gram models are suitable for performing range detection problems as the models simply calculate the probability of a gram being the a starting point (beginning index) or the an ending point (ending index) using the grams' statistical frequency. On the other hand, The use of word-grams, also known as n-grams are common in classical NLP research [28] and [29]. The word-grams model is only used in the NER task.

VI. Problem Formulation and Performance Metrics

As mentioned in the previous section, all of the fundamental tasks can be formalized as detection problems. In this section, we will go into detail about the differences of the two detection problems, instance detection and range detection.

A. Instance Detection Problem

Given a sequence of N units (characters or words), the goal of instance detection is to output another sequence of N class labels that specify whether the corresponding member of sequence is the class we are interested in (class label 1 or above) or not (class label 0). Word error, word variant and NER tasks can be formulated as instance detection problems.

B. Range Detection Problem

Given a sequence of N units (characters or words), the goal of range detection is to output another sequence of N class labels that specifies whether each member of the sequence is the first member of a subsequence of interest (class label 1), the last member of an subsequence of interest (class label 2) or neither (class label 0). All of the tasks stated above can be formulated as a range detection problem. The evaluation metric used is the standard F-measure, same as the instance detection problem.

VII. Results

A. Word Segmentation

Tables IV, V and VI show the results of word segmentation on the three corpora. The results are expected - for any gram models, as the gram size increases, the model's specificity increases, allowing the model to better differentiate positive and negative examples. Thus, precision increases. However, due to the limited size dataset, as the gram size increases, the model loses the ability to generalize due to overfitting, causing the recall to drop. The model with the best performance based on F-measure chooses a proper trade-off between precision and recall. For NECTEC's BEST corpus [1], note how the best performing configurations are symmetric, with the best char-gram being a 4-character-gram with 2 characters before the boundary and 2 characters after the boundary (the middle character included). This is likely due to fact that for the NECTEC's BEST corpus [1], the beginning of the next word is always right after the ending of the previous word. This is not true for our UGWC dataset. Table XII shows the result of the best char-gram model for detecting end range compared against the state-of-the-art method in word segmentation [3] on the UGWC corpus, both models are trained on the NECTEC's BEST corpus [1].

B. Sentence Segmentation

Tables VII, VIII and IX show the results of sentence segmentation. The results are not in-line with our expectation, as the model perform better on the UGWC corpus compared to NECTEC's ORCHID [2]. Both models are trained on the whole UGWC corpus and under-sampling to match the size of NECTEC's ORCHID [2]. After performing error analysis, we found that the sentence annotation in NECTEC's ORCHID [2] is inconsistent. This might be due the corpus creation method. Thus, we recommend discarding the performance number for NECTEC's ORCHID [2].

C. Word Error & Variant Detection

Table III shows the experimental results of N-gram on word error and variant detection of three corpus using

Table III
WORD ERROR AND VARIANT WITH CHAR-GRAM

	BEST			**ORCHID**			**UGWC**		
	Instance Detection	***Begin End Detection***		***Instance Detection***	***Begin End Detection***		***InstanceDetection***	***Begin End Detection***	
Gram Size		***Begin***	***End***		***Begin***	***End***		***Begin***	***End***
3	**0.428422**	0.299468	0.305331	**0.466428**	**0.409125**	**0.356181**	0.400026	0.212112	0.298713
5	0.374841	**0.303418**	**0.32387**	0.391542	0.317274	0.264011	**0.529302**	**0.457062**	**0.431757**
7	0.151633	0.130772	0.147203	0.229454	0.17214	0.148276	0.491228	0.429711	0.412221
9	0.053064	0.041963	0.046112	0.124761	0.091188	0.081511	0.435008	0.371202	0.359187
11	0.019584	0.015502	0.015516	0.07284	0.06095	0.053653	0.383118	0.326133	0.319502
Dict-based	0.5864035	0.6800373	0.591969	0.383937	0.307198	0.225843	0.329066	0.14597	0.185525

Table IV
WORD SEGMENTATION WITH CHAR-GRAM ON NECTEC'S BEST (16.4M CHARS) DATA USING RANGE (SEPARATE EVALUATION METHOD BETWEEN BEGIN, END) 5 FOLD AVERAGE

class	1 (Begin of Word)																							
left	0					1					2					3					4			
right	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3
precision	0.647	0.781	0.869	0.903	0.928	0.804	0.906	0.942	0.961	0.971	0.899	**0.955**	0.973	0.98	0.984	0.946	0.976	0.983	0.985	0.987	0.963	0.982	0.986	0.986
recall	0.618	0.819	0.924	0.938	0.916	0.871	0.936	0.947	0.9	0.807	0.937	**0.943**	0.903	0.802	0.667	0.938	0.889	0.819	0.692	0.553	0.883	0.776	0.692	0.549
f1	0.632	0.8	0.895	0.92	0.922	0.836	0.921	0.945	0.929	0.882	0.918	**0.949**	0.937	0.882	0.795	0.942	0.93	0.894	0.813	0.709	0.921	0.867	0.813	0.706
class	2 (End of Word)																							
left	0					1					2					3					4			
right	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3
precision	0.691	0.839	0.915	0.947	0.964	0.773	0.912	**0.958**	0.975	0.982	0.87	0.952	0.977	0.984	0.986	0.911	0.967	0.983	0.986	0.987	0.931	0.975	0.985	0.987
recall	0.472	0.877	0.942	0.942	0.885	0.85	0.945	**0.938**	0.881	0.773	0.93	0.939	0.875	0.785	0.659	0.933	0.882	0.758	0.657	0.521	0.906	0.78	0.622	0.524
f1	0.561	0.857	0.928	0.945	0.923	0.81	0.928	**0.948**	0.926	0.865	0.899	0.946	0.923	0.873	0.79	0.922	0.922	0.856	0.788	0.682	0.918	0.866	0.763	0.684

Table V
WORD SEGMENTATION WITH CHAR-GRAM ON ORCHID (1.5M CHARS) DATA USING RANGE EVALUATION METRIC 5 FOLD AVERAGE

class	1 (Begin of Word)																							
left	0					1					2					3					4			
right	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3
precision	0.616	0.753	0.837	0.869	0.898	0.771	0.895	0.927	0.942	0.954	0.893	**0.952**	0.962	0.968	0.971	0.938	0.967	0.971	0.973	0.974	0.952	0.972	0.974	0.975
recall	0.521	0.767	0.878	0.89	0.873	0.815	0.898	0.896	0.846	0.771	0.897	**0.878**	0.834	0.746	0.655	0.877	0.814	0.768	0.676	0.592	0.811	0.725	0.677	0.581
f1	0.564	0.76	0.857	0.88	0.885	0.792	0.896	0.911	0.891	0.853	0.895	**0.913**	0.894	0.843	0.782	0.906	0.884	0.858	0.798	0.736	0.876	0.83	0.799	0.728
class	2 (End of Word)																							
left	0					1					2					3					4			
right	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3
precision	0.645	0.807	0.908	0.936	0.949	0.767	0.908	0.958	0.966	0.971	0.851	**0.945**	0.971	0.974	0.976	0.889	0.957	0.975	0.976	0.977	0.909	0.964	0.977	0.978
recall	0.496	0.847	0.907	0.877	0.817	0.801	0.909	0.862	0.791	0.706	0.882	**0.88**	0.786	0.719	0.636	0.89	0.814	0.694	0.631	0.547	0.867	0.73	0.607	0.554
f1	0.561	0.826	0.907	0.906	0.878	0.783	0.908	0.907	0.87	0.818	0.866	**0.911**	0.869	0.827	0.77	0.89	0.88	0.811	0.767	0.702	0.887	0.831	0.749	0.707

Table VI
WORD SEGMENTATION WITH CHAR-GRAM ON UGWC (292K CHARS) DATA USING RANGE EVALUATION METRIC 5 FOLD AVERAGE

class	1 (Begin of Word)																							
left	0					1					2					3					4			
right	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3
precision	0.605	0.794	0.897	0.943	0.96	0.85	**0.949**	0.974	0.982	0.986	0.946	0.982	0.987	0.99	0.991	0.976	0.986	0.989	0.991	0.992	0.984	0.989	0.99	0.992
recall	0.69	0.84	0.896	0.837	0.752	0.853	**0.858**	0.785	0.667	0.557	0.839	0.708	0.641	0.541	0.458	0.742	0.623	0.576	0.487	0.418	0.62	0.523	0.489	0.414
f1	0.644	0.816	0.896	0.886	0.843	0.851	**0.901**	0.869	0.793	0.711	0.889	0.822	0.776	0.698	0.625	0.843	0.763	0.726	0.652	0.586	0.76	0.683	0.653	0.582
class	2 (End of Word)																							
left	0					1					2					3					4			
right	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3
precision	0.634	0.847	**0.953**	0.975	0.983	0.792	0.935	0.981	0.986	0.989	0.898	0.967	0.986	0.988	0.99	0.941	0.98	0.988	0.989	0.99	0.962	0.984	0.988	0.988
recall	0.316	0.847	**0.844**	0.744	0.631	0.775	0.858	0.707	0.621	0.534	0.876	0.787	0.627	0.561	0.485	0.83	0.659	0.511	0.464	0.404	0.731	0.548	0.427	0.394
f1	0.422	0.847	**0.895**	0.844	0.767	0.783	0.894	0.821	0.761	0.692	0.887	0.867	0.765	0.714	0.65	0.882	0.787	0.672	0.63	0.572	0.83	0.703	0.595	0.561

two performance metrics: Instance-Detection and Begin-End Detection. The table also shows the results of a dictionary-based approach where each word is checked with the correct dictionary obtained from the UGWC corpus. For UGWC, 5-grams outperforms other grams while in table NECTEC's BEST [1] and NECTEC's ORCHID [2]. One reason might be that our method of adding noise cannot completely mimic how users generate errors. In addition, the score in instance detection metrics is higher when compared to begin-end detection in every table as the scoring is more strict.

D. Named Entity Recognition

Table X shows the experimental results of character n-gram for NECTEC's BEST corpus [1] and UGWC corpus for named entity recognition. In both corpora, we performed 2 metrics, instance detection and begin end detection. From Table X and XI, the results show that using word 1-gram instance detection yields the best result. In case of no word segmentation, the 7 character-gram also provides a good result.

Table VII

SENTENCE SEGMENTATION WITH CHAR-GRAM ON ORCHID (1.5M CHARS) DATA USING RANGE EVALUATION METRIC 5 FOLD AVERAGE

class	1 (Begin of Word)																							
left	0					1					2					3					4			
right	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3
precision	1	0.919	0.757	**0.754**	0.748	0.911	0.815	0.739	0.763	0.778	0.752	0.739	0.757	0.78	0.803	0.631	0.716	0.763	0.793	0.829	0.56	0.72	0.79	0.823
recall	0.07	0.238	0.35	**0.388**	0.384	0.073	0.231	0.354	0.375	0.321	0.067	0.25	0.338	0.343	0.289	0.067	0.259	0.308	0.306	0.249	0.075	0.256	0.267	0.263
f1	0.131	0.378	0.479	**0.512**	0.507	0.135	0.36	0.479	0.502	0.454	0.123	0.374	0.467	0.476	0.424	0.122	0.381	0.438	0.441	0.383	0.133	0.378	0.398	0.399
class	2 (End of Word)																							
left	0					1					2					3					4			
right	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3
precision	NaN	0.658	0.618	0.592	0.625	0.924	0.759	**0.756**	0.75	0.77	0.815	0.739	0.763	0.778	0.816	0.739	0.757	0.78	0.803	0.843	0.716	0.763	0.793	0.829
recall	0	0.006	0.033	0.16	0.233	0.238	0.35	**0.388**	0.384	0.36	0.231	0.354	0.375	0.321	0.29	0.25	0.337	0.343	0.288	0.26	0.259	0.307	0.305	0.249
f1	NaN	0.012	0.063	0.252	0.34	0.378	0.479	**0.513**	0.508	0.49	0.36	0.479	0.502	0.454	0.428	0.374	0.467	0.476	0.424	0.397	0.381	0.438	0.441	0.383

Table VIII

SENTENCE SEGMENTATION WITH CHAR-GRAM ON UGWC (1.5M CHARS SIMULATED) DATA USING RANGE (SEPARATE EVALUATION METHOD BETWEEN BEGIN, END) 5 FOLD AVERAGE

class	1 (Begin of Word)																							
left	0					1					2					3					4			
right	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3
precision	0.534	0.579	0.607	0.61	0.607	0.836	0.782	0.762	0.775	0.785	0.804	**0.819**	0.846	0.873	0.897	0.828	0.874	0.897	0.923	0.942	0.862	0.907	0.921	0.942
recall	0.006	0.089	0.173	0.228	0.26	0.405	0.547	0.609	0.566	0.492	0.657	**0.652**	0.592	0.487	0.39	0.644	0.562	0.496	0.405	0.332	0.587	0.508	0.454	0.377
f1	0.011	0.154	0.269	0.332	0.364	0.546	0.643	0.677	0.654	0.604	0.723	**0.726**	0.696	0.625	0.544	0.725	0.684	0.638	0.563	0.491	0.698	0.652	0.608	0.539
class	2 (End of Word)																							
left	0					1					2					3					4			
right	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3
precision	0.526	0.754	0.812	**0.828**	0.859	0.832	0.773	0.827	0.875	0.902	0.757	0.761	0.855	0.905	0.921	0.783	0.774	0.879	0.922	0.934	0.754	0.803	0.899	0.937
recall	0.001	0.439	0.654	**0.648**	0.595	0.254	0.532	0.647	0.558	0.506	0.3	0.592	0.583	0.49	0.449	0.357	0.58	0.498	0.411	0.383	0.358	0.524	0.402	0.331
f1	0.001	0.555	0.725	**0.727**	0.703	0.39	0.631	0.726	0.681	0.648	0.43	0.666	0.693	0.636	0.604	0.49	0.663	0.636	0.569	0.544	0.485	0.634	0.556	0.489

Table IX

SENTENCE SEGMENTATION WITH CHAR-GRAM ON UGWC (7.6M CHARS) DATA USING RANGE (SEPARATE EVALUATION METHOD BETWEEN BEGIN, END) 5 FOLD AVERAGE

class	1 (Begin of Word)																							
left	0					1					2					3					4			
right	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3
precision	0.544	0.610	0.656	0.652	0.656	0.830	0.788	0.775	0.789	0.797	0.824	**0.825**	0.844	0.864	0.885	0.834	0.859	0.883	0.907	0.928	0.852	0.890	0.906	0.927
recall	0.005	0.083	0.165	0.252	0.293	0.414	0.566	0.657	0.648	0.612	0.662	**0.722**	0.703	0.630	0.547	0.704	0.678	0.627	0.543	0.469	0.677	0.622	0.575	0.502
f1	0.011	0.146	0.264	0.363	0.405	0.552	0.659	0.711	0.712	0.692	0.734	**0.770**	0.767	0.729	0.676	0.764	0.758	0.733	0.679	0.623	0.754	0.732	0.704	0.651
class	2 (End of Word)																							
left	0					1					2					3					4			
right	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3
precision	0.514	0.752	0.830	**0.830**	0.852	0.840	0.778	0.834	0.861	0.887	0.791	0.77	0.850	0.889	0.907	0.807	0.785	0.870	0.910	0.924	0.788	0.804	0.887	0.925
recall	0	0.446	0.662	**0.721**	0.697	0.255	0.542	0.707	0.677	0.627	0.299	0.625	0.680	0.615	0.569	0.373	0.630	0.619	0.539	0.503	0.386	0.607	0.544	0.461
f1	0.001	0.560	0.736	**0.771**	0.767	0.391	0.639	0.765	0.758	0.734	0.434	0.690	0.756	0.727	0.700	0.510	0.699	0.724	0.677	0.651	0.518	0.692	0.674	0.615

Table X

NAMED ENTITY RECOGNITION WITH CHAR-GRAM

	BEST			**UGWC**		
	Instance Detection	***Begin End Detection***		***Instance Detection***	***Begin End Detection***	
Gram Size		***Begin***	***End***		***Begin***	***End***
3	0.457	0.259	0.179	0.231	0.191	0.261
5	0.713	0.576	0.518	0.394	0.284	0.424
7	**0.729**	**0.609**	**0.532**	0.450	**0.333**	**0.460**
9	0.672	0.515	0.447	**0.453**	0.329	0.436
11	0.597	0.425	0.367	0.431	0.318	0.410

Table XI

NAMED ENTITY RECOGNITION WITH WORD-GRAM

	BEST	**UGWC**
Gram Size	***Instance Detection***	***Instance Detection***
1	**0.9271**	**0.6407**
3	0.7448	0.5005
5	0.5752	0.3584
7	0.5394	0.3225
9	0.528	0.2993
11	0.5214	0.2871

Table XII

WORD SEGMENTATION ON UGWC DATA USING RANGE EVALUATION METRIC (END OF RANGE DETECTION ONLY) WITH MODELS TRAIN ON NECTEC'S BEST CORPUS

model	Char-gram (Left 1, Right 2)	Sertis
precision	0.753	0.972
recall	0.856	0.807
f1	0.801	0.882

VIII. Conclusion

In this paper, we presented models and performance metrics for evaluating five fundamental Thai NLP tasks on a UGWC corpus. We hope that the results can serve as a benchmark to help measure future UGWC corpora's difficulties as well as to set a baseline comparison between minimally pre-processed UGWC and publicly available corpora that are based on more formal text. Those tasks were composed of word segmentation, sentence segmentation, word error detection,

word variant detection and named entity recognition. Three Thai text corpora were used in this research: NECTEC's BEST [1] and NECTEC's ORCHID [2] and UGWC corpus. After using those corpora with various algorithms, the results showed that, for word segmentation, Sertis, which achieved an F1 of 0.992 [3] on NECTEC's BEST corpus [1], achieved lower F1 value of 0.882 on UGWC corpus, indicating that the task is not yet fully solved on UGWC. In sentence segmentation task, the char-gram's result on UGWC (7.6m chars) data using range evaluation (separate evaluation method between beginning and ending) achieved better F1 than the others at 0.77. Moreover, for word error and variant detection, 5-gram model performs better than 3-gram model on UGWC corpus, achieving an F1 score at 0.529. For name entity recognition, we classified name entities into five classes which are person, location, organization, product and other. Within these classes, word-gram method outperforms char-gram method in instance detection metrics.

IX. Acknowledgement

We would like to thank Dr. Warasinee Chaisangmongkon and the linguist team: Ms. Sasiwimon Kalunsima and Dr. Tantong Champaiboon for their invaluable comments. This work was supported by Kasikorn Business-Technology Group (KBTG).

References

[1] K. Kosawat, M. Boriboon, P. Chootrakool, A. Chotimongkol, S. Klaithin, S. Kongyoung, K. Kriengket, S. Phaholphinyo, S. Purodakananda, T. Thanakulwarapas *et al.*, "BEST 2009: Thai word segmentation software contest," in *Natural Language Processing, 2009. SNLP'09. Eighth International Symposium on*. IEEE, 2009, pp. 83–88.

[2] V. Sornlertlamvanich, N. Takahashi, and H. Isahara, "Building a thai part-of-speech tagged corpus (orchid)," *Journal of the Acoustical Society of Japan (E)*, vol. 20, no. 3, pp. 189–198, 1999.

[3] J. Jousimo, "Thai word segmentation with bi-directional RNN." [Online]. Available: https://goo.gl/nWPWEq

[4] R. Kittinaradorn, "A Thai word tokenization library using deep neural network." [Online]. Available: https://goo.gl/MZEsu5

[5] M. D. Riley, "Some applications of tree-based modelling to speech and language," in *Proceedings of the workshop on Speech and Natural Language*. Association for Computational Linguistics, 1989, pp. 339–352.

[6] J. Read, R. Dridan, S. Oepen, and L. J. Solberg, "Sentence boundary detection: a long solved problem?" *Proceedings of COLING 2012: Posters*, pp. 985–994, 2012.

[7] W. Aroonmanakun *et al.*, "Thoughts on word and sentence segmentation in thai," in *Proceedings of the Seventh Symposium on Natural language Processing, Pattaya, Thailand, December 13–15*, 2007, pp. 85–90.

[8] P. Charoenpornsawat and V. Sornlertlamvanich, "Automatic sentence break disambiguation for thai," in *International Conference on Computer Processing of Oriental Languages (ICCPOL)*, 2001, pp. 231–235.

[9] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[11] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[12] K. Kukich, "Techniques for automatically correcting words in text," *Acm Computing Surveys (CSUR)*, vol. 24, no. 4, pp. 377–439, 1992.

[13] S. Ruder, "Multi-sense embeddings." [Online]. Available: https://goo.gl/9J9TZg

[14] J. D'Souza, "A sequence labeling approach to deriving word variants." in *AAAI*, 2015, pp. 4152–4153.

[15] S. B. Bam and T. B. Shahi, "Named entity recognition for nepali text using support vector machines," *Intelligent Information Management*, vol. 6, no. 02, p. 21, 2014.

[16] S. Amarappa and S. Sathyanarayana, "Named entity recognition and classification in kannada language," *International Journal of Electronics and Computer Science Engineering*, vol. 2, no. 1, pp. 281–289, 2013.

[17] M. P. Skënduli and M. Biba, "A named entity recognition approach for albanian," in *Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on*. IEEE, 2013, pp. 1532–1537.

[18] F. Ahmadi and H. Moradi, "A hybrid method for persian named entity recognition," in *Information and Knowledge Technology (IKT), 2015 7th Conference on*. IEEE, 2015, pp. 1–7.

[19] S. Manamini, A. Ahamed, R. Rajapakshe, G. Reemal, S. Jayasena, G. Dias, and S. Ranathunga, "A named entity recognition (ner) system for sinhala language," in *Moratuwa Engineering Research Conference (MERCon), 2016*. IEEE, 2016, pp. 30–35.

[20] A. Ekbal, S. Saha, and D. Singh, "Ensemble based active annotation for named entity recognition," in *Emerging Applications of Information Technology (EAIT), 2012 Third International Conference on*. IEEE, 2012, pp. 331–334.

[21] H.-K. Yi, J.-M. Huang, and S.-Q. Yang, "A chinese named entity recognition system with neural networks," in *ITM Web of Conferences*, vol. 12. EDP Sciences, 2017, p. 04002.

[22] L. Ouyang, Y. Tian, H. Tang, and B. Zhang, "Chinese named entity recognition based on bilstm neural network with additional features," in *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*. Springer, 2017, pp. 269–279.

[23] S. Tepdang, C. Haruechaiyasak, and R. Kongkachandra, "Improving thai word segmentation with named entity recognition," in *Communications and Information Technologies (ISCIT), 2010 International Symposium on*. IEEE, 2010, pp. 940–945.

[24] H. Chanlekha and A. Kawtrakul, "Thai named entity extraction by incorporating maximum entropy model with simple heuristic information," in *Proceedings of the IJCNLP*, 2004.

[25] I. Setiadi, "Damerau-levenshtein algorithm and bayes theorem for spell checker optimization, 6," 2013.

[26] "Royal thai dictionary." [Online]. Available: https://goo.gl/iZGxyB

[27] S. Watcharabutsarakham, "Spell checker for thai document," in *TENCON 2005 2005 IEEE Region 10*. IEEE, 2005, pp. 1–4.

[28] J. Pailai, R. Kongkachandra, T. Supnithi, and P. Boonkwan, "A comparative study on different techniques for thai part-of-speech tagging," in *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2013 10th International Conference on*. IEEE, 2013, pp. 1–5.

[29] A. Figueroa and J. Atkinson, "Contextual language models for ranking answers to natural language definition questions," *Computational Intelligence*, vol. 28, no. 4, pp. 528–548, 2012.