Thai Sentence-Breaking for Large-Scale SMT

Glenn Slayden
thai-language.com
glenn@thai-language.com

Mei-Yuh Hwang
Microsoft Research
mehwang@microsoft.com

Lee Schwartz
Microsoft Research
leesc@microsoft.com

Abstract

Thai language text presents challenges for integration into large-scale multi-language statistical machine translation (SMT) systems, largely stemming from the nominal lack of punctuation and inter-word space. For Thai sentence breaking, we describe a monolingual maximum entropy classifier with features that may be applicable to other languages such as Arabic, Khmer and Lao. We apply this sentence breaker to our large-vocabulary, general-purpose, bidirectional Thai-English SMT system, and achieve BLEU scores of around 0.20, reaching our threshold of releasing it as a free online service.

1 Introduction

NLP research has consolidated around the notion of the *sentence* as the fundamental unit of translation, a consensus which has fostered the development powerful statistical and analytical approaches which incorporate an assumption of deterministic sentence delineation. As such systems become more sophisticated, languages for which this assumption is challenged receive increased attention. Thai is one such language, since it uses space neither to distinguish syllables from words or affixes, nor to unambiguously signal sentence boundaries.

Written Thai has no sentence-end punctuation, but a space character is always present between sentences. There is generally no space between words, but a space character may appear within a sentence according to linguistic or prescriptive orthographic motivation (Wathabunditkul 2003), and these characteristics disqualify sentence-breaking (SB) methods used for other languages, such as Palmer and Hearst (1997). Thai SB has therefore been regarded as the task of classifying each space that appears in a Thai source text as either sentence-breaking (**sb**) or non-sentence-breaking (**nsb**).

Several researchers have investigated Thai SB. Along with a discussion of Thai word breaking (WB), Aroonmanakun (2007) examines the issue. With a human study, he establishes that sentence breaks elicited from Thai informants exhibit varying degrees of consensus. Mittrapiyanuruk and Sornlertlamvanich (2000) define part-of-speech (POS) tags for **sb** and **nsb** and train a trigram model over a POS-annotated corpus. At runtime, they use the Viterbi algorithm to select the POS sequence with the highest probability, from which the corresponding space type is read back. Charoenpornsawat and Sornlertlamvanich (2001) apply Winnow, a multiplicative trigger threshold classifier, to the problem. Their model has ten features: the number of words to the left and right, and the left-two and right-two POS tags and words.

We present a monolingual Thai SB based on a maximum entropy (ME) classifier (Ratnaparkhi 1996; Reynar and Ratnaparkhi, 1997) which is suitable for sentence-breaking SMT training data and runtime inputs. Our model uses a four token window of Thai lemmas, plus categorical features, to describe the proximal environment of the space token under consideration, allowing runtime classification of space tokens with possibly unseen contexts.

As our SB model relies on Thai WB, we review our approach to this problem, plus related preprocessing, in the next section. Section 2 also discusses the complementary operation to WB, namely, the re-spacing of Thai text generated by SMT output. Section 3 details our SB model and evaluates its performance. We describe the integration of this work with our large-scale SMT system in Section 4. We draw conclusions in Section 5.

Proceedings of the 1st Workshop on South and Southeast Asian Natural Language Processing (WSSANLP), pages 8–16,
the 23rd International Conference on Computational Linguistics (COLING), Beijing, August 2010

2 Pre- and Post-processing

As will be shown in Section 3, our sentence breaker relies on Thai WB. In turn, with the aim of minimizing WB errors, we perform Unicode character sequence normalization prior to WB. As output byproducts, our WB analysis readily identifies certain types of named entities which we propagate into our THA-ENG SMT; in this section, we briefly summarize these preliminary processing steps, and we conclude the section with a discussion of Thai text re-spacing.

2.1 Character Sequence Normalization

Thai orthography uses an alphabet of 44 consonants and a number of vowel glyphs and tone marks. The four Thai tone marks and some Thai vowel characters are super- and/or sub-scripted with respect to a base character. For example, the อิ้ sequence consists of three code points: อ ◌ิ ◌้. When two or more of these combining marks are present on the same base character, the ordering of these code points in memory should be consistent so that orthographically identical entities are recognized as equivalent by computer systems. However, some computer word processors do not enforce the correct sequence or do not properly indicate incorrect sequences to the user visually. This often results in documents with invalid byte sequences.

Correcting these errors is desirable for SMT inputs. In order to normalize Thai input character sequences to a canonical Unicode form, we developed a finite state transducer (FST) which detects and repairs a number of sequencing errors which render Thai text either orthographically invalid, or not in a correct Unicode sequence.

For example, a superscripted Thai tone mark should follow a super- or sub-scripted Thai vowel when they both apply to the same consonant. When the input has the tone mark and the vowel glyph swapped, the input can be fully repaired:

อ า ◌่ น → อ ◌่ า น → อ่าน
อ ◌้ ◌ิ น → อ ◌ิ ◌้ น → อิ้น

Figure 1. Two unambiguous repairs

Other cases are ambiguous. The occurrence of multiple adjacent vowel glyphs is an error where the intention may not be clear. We retain the first-appearing glyph, unless it is a pre-posed vowel, in which case we retain the last-appearing instance. These two treatments are contrasted in Figure 2. Miscoding (Figure 3) is another variety of input error that is readily repaired.

จะะา → จะ
ใไป → ไป

Figure 2. Two ambiguous repairs

Within the Infoquest Thai newswire corpus, a low-noise corpus, about 0.05% of the lines exhibit at least one of the problems mentioned here. For some chunks of broad-range web scraped data, we observe rates as high as 4.1%. This measure is expected to under-represent the utility of the filter to WB, since Thai text streams, lacking intra-word spacing and permitting two unwritten vowels, have few re-alignment checkpoints, allowing tokenization state machines to linger in misaligned states.

อ ◌ำ า → อ ◌ำ → อำ
เ เ อ → แ อ → แอ

Figure 3. Two common mis-codings

2.2 Uniscribe Thai Tokenization

Thai text does not normally use the space character to separate words, except in certain specific contexts. Although Unicode offers the Zero-Width Space (ZWSP) as one solution for indicating word breaks in Thai, it is infrequently used. Programmatic tokenization has become a staple of Thai computational linguistics. The problem has been well studied, with precision and recall near 95% (Haruechaiyasak et al. 2008).

In our SMT application, both the sentence breaker and the SMT system itself require Thai WB, and we use the same word breaker for these tasks (although the system design currently prohibits directly passing tokens between these two components). Our method is to apply post-processing heuristics to the output of *Uniscribe* (Bishop et al. 2003), which is provided as part of the Microsoft® Windows™ operating system interface. Our heuristics fall into two categories: "re-gluing" words that *Uniscribe* broke too aggressively, and a smaller class of cases of further breaking of words that *Uniscribe* did not break.

Re-gluing is achieved by comparing *Uniscribe* output against a Thai lexicon in which desired breaks within a word are tagged. Underbreaking by *Uniscribe* is less common and is restricted to a number of common patterns which are repaired explicitly.

2.3 Person Name Entities

In written Thai, certain types of entities employ prescriptive whitespace patterns. By removing these recognized patterns from consideration, SB precision can be improved. Furthermore, because our re-gluing procedure requires a lookup of every syllable proposed by *Uniscribe*, it is efficient to consider, during WB, additional processing that can be informed by the same lookup. Accordingly, we briefly mention some of the entity types that our WB identifies, focusing on those that incorporate distinctive spacing patterns.

Person names in Thai adhere to a convention for the use of space characters. This helps Thai readers to identify the boundaries of multi-syllable surnames that they may not have seen before. The following grammar summarizes the prescriptive conventions for names appearing in Thai text:

```
<name-entity> ::= <honorific>  <full-name>
<full-name> ::= <first-name> [<last-name>]
<first-name> ::= <name-text> space
<last-name> ::= <name-text> space
<name-text> ::= <thai-alphabetic-char>+
<thai-alphabetic-char> ::= ก | ข | ฃ | ค | ...
```

Figure 4. Name entity recognition grammar

The re-glue lookup also determines if a syllable matches one of the following predefined special categories: name-introducing honorific (h), Thai or foreign given name (g), token which is likely to form part of a surname (s), or token which aborts the gathering of a name (i.e. is unlikely to form part of a name).

.../ว่า/นาย/จิ/ระ/นุช/ /วิ/นิจ/จกู/ล/ /ว่า/...											
ว่า	นาย	จิ	ระ	นุช		วิ	นิจ	จกู	ล		ว่า
	h	g0	g1	g2	sp0	s0	s1	s2	s3	sp1	
that	Mr.	<oov>	hit	beloved		<oov>	stable	<oov>	<oov>		said
...that	Mr.	Chiranut				Winichotkun					said...

Figure 5. Thai person-name entity recognition

Figure 5 shows a Thai name appearing within a text fragment, with *Uniscribe* detected token boundaries indicated by slashes. In the third row we have identified the special category, if any, for each token. The fourth line shows the English translation gloss, or <oov> if none. The bottom row is the desired translation output.

Our name identifier first notes the presence of an honorific {h} **นาย** followed by a pattern of tokens {g0-g*n*}, {s0-s*n*} and spaces {sp0, sp1} that is compatible with a person name and surname of sensible length.

Next, we determine which of those tokens in the ranges {g} and {s} following the honorific do not have a gloss translation (i.e., are not found in the lexicon). These tokens are indicated by <oov> in the gloss above. When the number of unknown tokens exceeds a threshold, we hypothesize that these tokens form a name. The lack of lexical morphology in Thai facilitates this method because token (or syllable) lookup generally equates with the lookup of a stemmed lemma.

2.4 Calendar Date Entities

Our WB also identifies Thai calendar dates, as these also exhibit a pattern which incorporates spaces. As a prerequisite to identifying dates, we map Thai orthographic digits {๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙} to Arabic digits 0 through 9, respectively. For example, our system would interpret the input text **๒๕๔๐** as equivalent to "2540."

.../ใน/วัน/ที่/ /14/ มีนาคม/ /๒๕๔๐/ /และ/...									
ใน	วัน	ที่	sp	14	มีนาคม	sp	๒๕๔๐	sp	และ
on	day	which		14	March		2540		and
...on	March 14th, 1997								and...

Figure 6. Date entity recognition

Figure 6 shows a fragment of Thai text which contains a calendar date for which our system will emit a single token. As shown in the example, our system detects and adjusts for the use of Thai Buddhist year dates when necessary. Gathering of disparate and optional parts of the Thai date is summarized by the grammar in Figure 7.

```
<date-entity> ::= [<cardinal-words>] [space] <date>
<cardinal-words> ::= วันที่ | ที่
<date> ::= month-date [space] year
<year> ::= <tha-digit> <tha-digit> <tha-digit> <tha-digit>
<year> ::= <ara-digit> <ara-digit> <ara-digit> <ara-digit>
<month-date> ::= <day> [space] <month>
<day> ::= <thai-digit>+
<day> ::= <ara-digit>+
<month> ::= <month-full> | <month-abbr>
<month-full> ::= มกราคม | กุมภาพันธ์ | มีนาคม | ...
<month-abbr> ::= ม.ค. | ก.พ. | มี.ค. | ...
<tha-digit> ::= ๐ | ๑ | ๒ | ๓ | ๔ | ๕ | ๖ | ๗ | ๘ | ๙
<ara-digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

Figure 7. Date recognition grammar

2.5 Thai Text Re-spacing

To conclude this section, we mention an operation complementary to Thai WB, whereby Thai words output by an SMT system must be re-spaced in accordance with Thai prescriptive convention. As will be mentioned in Section 4.2, for each input sentence, our English-Thai system has access to an English dependency parse tree, as well as links between this tree and a Thai transfer dependency tree. After using these links to transfer syntactic information to the Thai tree, we are able to apply prescriptive spacing rules (Wathabunditkul 2003) as closely as possible. Human evaluation showed satisfactory results for this process.

3 Maximum Entropy Sentence-Breaking

We now turn to a description of our statistical sentence-breaking model. We train an ME classifier on features which describe the proximal environment of the space token under consideration and use this model at runtime to classify space tokens with possibly unseen contexts.

3.1 Modeling

Under the ME framework, let B={**sb**, **nsb**} represent the set of possible classes we are interested in predicting for each space token in the input stream. Let C={linguistic contexts} represent the set of possible contexts that we can observe, which must be encoded by binary features, $f_j(b,c), 1 \le j \le k$, such as:

$$f_1(b,c) = \begin{cases} 1 \text{ if the previous word is English } \textit{and } b = \textbf{nsb}. \\ 0 \text{ otherwise.} \end{cases}$$

This feature helps us learn that the space after an English word is usually not a sentence boundary.

$$f_2(b,c) = \begin{cases} 1 \text{ if the distance to the previous honorific} \\ \text{is less than 15 tokens } \textit{and } b = \textbf{nsb} \\ 0 \text{ otherwise.} \end{cases}$$

This feature enables us to learn that spaces which follow an honorific are less likely to mark sentence boundaries. Assume the joint probability *p(b,c)* is modeled by

$$p(b,c) = Z \prod_{j=1}^{k} \alpha_j^{f_j(b,c)}$$

where we have k free parameters $\{\alpha_j\}$ to estimate and Z is a normalization factor to make $\sum_{b,c} p(b,c) = 1$. The ME learning algorithm finds a solution $\{\alpha_j\}$ representing the most uncertain commitment

$$\max \; H(p) = -\sum p(b,c) \log p(b,c)$$

that satisfies the observed distribution $\hat{p}(b,c)$ of the training data

$$\sum p(b,c) f_j(b,c) = \sum \hat{p}(b,c) f_j(b,c), \; 1 \le j \le k \, .$$

This is solved via the *Generalized Iterative Scaling* algorithm (Darroch and Ratcliff 1972). At run-time, a space token is considered an **sb**, if and only if p(**sb**|c) > 0.5, where

$$p(\textbf{sb}|c) = \frac{p(\textbf{sb},c)}{p(\textbf{sb},c) + p(\textbf{nsb},c)} \, .$$

3.2 Feature Selection

The core context of our model, {*w*, *x*, *y*, *z*}, is a window spanning two tokens to the left (positions *w* and *x*) and two tokens to the right (positions *y* and *z*) of a classification candidate space token.

c	**token characteristic**
yk	Yamok (syllable reduplication) symbol ๆ
sp	space
๐๙	Thai numeric digits
num	Arabic numeric digits
ABC	Sequence of all capital ASCII characters
c*nn*	single character (derived from hex)
c*kkmmnn*	single character (derived from UTF8 hex)
ascii	any amount of non-Thai text
(Thai text)	Thai word (derived from lemma)

Table 1. Categorical and derived feature names

The possible values of each of the window positions {*w*, *x*, *y*, *z*} are shown in Table 1, where the first match to the token at the designated position is assigned as the feature value for that position. Foreign-text tokens plus any intervening space are merged, so a single "ascii" feature may represent an arbitrary amount of non-Thai script with interior space.

Figure 8 shows an example sentence that has been tokenized. Token boundaries are indicated by slashes. Although there are three space tokens in the original input, we extract four contexts. The shaded boxes in the source text—and the shaded line in the figure—indicate the single **sb** context that is synthesized by wrapping, to be described in Section 3.4.

For each context, in addition to the {*w, x, y, z*} features, we extract two more features indicated by {*l* ,*r*} in Figure 8. They are the number of

tokens between the previous space token (wrapping as necessary) and the current one, and the number of tokens between the current space token and the next space token (wrapping as necessary). These features do not distinguish whether the bounding space token is **sb** or **nsb**. This is because, processing left-to-right, it is permissible to use a feature such as "number of tokens since last **sb**," but not "number of tokens until next **sb**," which would be available during training but not at runtime.

ลักษณะการอ้างอิงแบบ R1C1 ถูกแปลงไปเป็นลักษณะการอ้างอิงแบบ A1

"R1C1 reference style was converted to A1 reference style."

/ลักษณะ/การ/อ้าง/อิง/แบบ/ /R1C1/ /ถูก/แปลง/ไป/เป็น/ลักษณะ/การ/อ้าง/อิง/แบบ/ /A1/

b	c=*w*	*c=x*	*c=y*	*c=z*	*c=l*	*c=r*
nsb	อิง	แบบ	ABC	sp	5	1
nsb	sp	ABC	ถูก	แปลง	1	9
nsb	อิง	แบบ	ABC	sp	9	1
sb	sp	ABC	ลักษณะ	การ	1	5

Figure 8. A Thai sentence and the training contexts extracted. Highlighting shows the context for **sb**.

In addition to the above core features, our model emits certain extra features only if they appear:

- An individual feature for each English punctuation mark, since these are sometimes used in Thai. For example, there is one feature for the sentence end period (i.e. full-stop);
- The current nest depth for paired glyphs with directional variation, such as brackets, braces, and parentheses;
- The current parity value for paired glyphs without directional distinction such as "straight" quotation marks.

The following example illustrates paired directional glyphs (in this case, parentheses):

.../ยูนิลิเวอร์/ /(/ประเทศ/ /ไทย/)/ /จำกัด/ /เปิดเผย/ว่า/...
...Unilever (Thailand) Ltd. disclosed that...

b	c=*w*	*c=x*	*c=y*	*c=z*	*c=pn*
nsb	(	ประเทศ	ไทย	)	1

Figure 9. Text fragment illustrating paired directional glyphs and the context for the highlighted space

In Figure 9, the space between **ประเทศ** "country" and **ไทย** "Thai," generates an **nsb** context which includes the features shown, where "pn" is an extra feature which indicates the parenthesis nesting level. This feature helps the model learn that spaces which occur within parentheses are likely to be **nsb**.

Parity features for the non-directional paired glyphs, which do nest, are true binary features. Since these features have only two possible values (*inside* or *outside*), they are only emitted when their value is "inside," that is, when the space under consideration occurs between such a pair.

3.3 Sentence Breaker Training Corpus

Thai corpora which are marked with sentence breaks are required for training. We assembled a corpus of 361,802 probable sentences. This corpus includes purchased, publicly available, and web-crawled content. In total it contains 911,075 spaces, a figure which includes one inter-sentence space per sentence, generated as described below.

3.4 Out-of-context Sentences

For SB training, paragraphs are first tokenized into words as described in Section 2.2. This process does not introduce new spaces between tokens; only original spaces in the text are classified as **sb**/**nsb** and used for the context features described below. To keep this distinction clear, token boundaries are indicated by a slash rather than space in the examples shown in this paper.

For 91% of our training sentences, the paragraphs from which they originate are inaccessible. In feature extraction for each of these sentences, we wrap the sentence's head around to its tail to obtain its **sb** context. In other words, for a sentence of tokens t_0-t_{n-1}, the context of **sb** (the last space) is given by

$$\{ w=t_{n-2}, x=t_{n-1}, y=t_0, z=t_1 \}.$$

This process was illustrated in Figure 8. Although not an ideal substitute for sentences in context, this ensures that we extract at least one **sb** context per sentence. The number of **nsb** contexts extracted per sentence is equal to the number of interior space tokens in the original sentence. Sentence wrapping is not needed when training with sentence-delimited paragraph sources. Contexts **sb** and **nsb** are extracted from the token stream of the entire paragraph and wrapping is used only to generate one additional **sb** for the entire paragraph.

3.5 Sentence Breaker Evaluation

Although evaluation against a single-domain corpus does not measure important design requirements of our system, namely resilience to broad-domain input texts, we evaluated against the ORCHID corpus (Charoenporn et al. 1997) for the purpose of comparison with the existing literature. Following the methodology of the studies cited below, we use 10-fold ×10% averaged testing against the ORCHID corpus.

Our results are consistent with recent work using the Winnow algorithm, which itself compares favorably with the probabilistic POS trigram approach. Both of these studies use evaluation metrics, attributed to Black and Taylor (1997), which aim to more usefully measure sentence-breaker utility. Accordingly, the following definitions are used in Table 2:

$$\text{space-correct} = \frac{(\#\text{correct sb}+\#\text{correct nsb})}{\text{total \# of space tokens}}$$

$$\text{false break} = \frac{\#\text{sb false positives}}{\text{total \# of space tokens}}$$

It was generally possible to reconstruct precision and recall figures from these published results[1] and we present a comprehensive table of results. Reconstructed values are marked with a dagger and the optimal result in each category is marked in boldface.

	Mittrapiyanuruk et al.	Charoenpornsawat et al.	Our result
method	**POS Trigram**	**Winnow**	**MaxEnt**
#sb in reference	10528	1086†	2133
#space tokens	33141	3801	7227
nsb-precision	90.27†	91.48†	**93.18**
nsb-recall	87.18†	**97.56**†	94.41
sb-precision	74.35†	**92.69**†	86.21
sb-recall	79.82	77.27	**83.50**
"space-correct"	85.26	89.13	**91.19**
"false-break"	8.75	**1.74**	3.94

Table 2. Evaluation of Thai Sentence Breakers against ORCHID

Finally, we would be remiss in not acknowledging the general hazard of assigning sentence breaks in a language such as Thai, where source text authors may intentionally include or omit spaces in order to create syntactic or semantic ambiguity. We defer to Mittrapiyanuruk and Sornlertlamvanich (2000) and Aroonmanakun (2007) for informed commentary on this topic.

[1] Full results for Charoenpornsawat et al. are reconstructed based on remarks in their text, including that "the ratio of the number of [**nsb** to **sb**] is about 5:2."

4 SMT System and Integration

The primary application for which we developed the Thai sentence breaker described in this work is the Microsoft® BING™ general-domain machine translation service. In this section, we provide a brief overview of this large-scale SMT system, focusing on Thai-specific integration issues.

4.1 Overview

Like many multilingual SMT systems, our system is based on hybrid generative/discriminative models. Given a sequence of foreign words, $\boldsymbol{f}$, its best translation is the sequence of target words, $\boldsymbol{e}$, that maximizes

$$\boldsymbol{e}^* = \text{argmax}_{\boldsymbol{e}}\, p(\boldsymbol{e}|\boldsymbol{f}) = \text{argmax}_{\boldsymbol{e}}\, p(\boldsymbol{f}|\boldsymbol{e})p(\boldsymbol{e})$$

$$= \text{argmax}_{\boldsymbol{e}}\, \{\log p(\boldsymbol{f}|\boldsymbol{e}) + \log p(\boldsymbol{e})\}$$

where the translation model $p(\boldsymbol{f}|\boldsymbol{e})$ is computed on dozens to hundreds of features. The target language model (LM), $p(\boldsymbol{e})$, is represented by a smoothed n-grams (Chen 1996) and sometimes more than one LM is adopted in practice. To achieve the best performance, the log likelihoods evaluated by these features/models are linearly combined. After $p(\boldsymbol{f}|\boldsymbol{e})$ and $p(\boldsymbol{e})$ are trained, the combination weights λ_i are tuned on a held-out dataset to optimize an objective function, which we set to be the BLEU score (Papineni et al. 2002):

$$\{\lambda_i^*\} = \max_{\{\lambda i\}} \text{BLEU}(\{e^*\},\{r\})$$

$$\boldsymbol{e}^* = \text{argmax}_{\boldsymbol{e}}\, \{\sum_i \lambda_i \log\, p_i(\boldsymbol{f}|\boldsymbol{e}) + \sum_j \lambda_j \log p_j(\boldsymbol{e})\}$$

where $\{r\}$ is the set of gold translations for the given input source sentences. To learn λ_i we use the algorithm described by Och (2003), where the decoder output at any point is approximated using n-best lists, allowing an optimal line search to be employed.

4.2 Phrasal and Treelet Translation

Since we have a high-quality real-time rule-based English parser available, we base our Eng-

lish-to-Thai translation (ENG-THA) on the "treelet" concept suggested in Menezes and Quirk (2008). This approach parses the source language into a dependency tree which includes part-of-speech labels.

Lacking a Thai parser, we use a purely statistical phrasal translator after Pharaoh (Koehn 2004) for THA-ENG translation, where we adopt the name and date translation described in Sections 2.3 and 2.4.

We also experimented with phrasal ENG-THA translation. Though we actually achieved a slightly better BLEU score than treelet for this translation direction, qualitative human evaluation by native speaker informants was mixed. We adopted the treelet ENG-THA in the final system, for its better re-spacing (Section 2.5).

4.3 Training, Development and Test Data

Naturally, our system relies on parallel text corpora to learn the mapping between two languages. The parallel corpus contains sentence pairs, corresponding to translations of each other. For Thai, quality corpora are generally not available in sufficient quality for training a general-domain SMT system. For the ENG-THA pair, we resort to Internet crawls as a source of text. We first identify paired documents, break each document into sentences, and align sentences in one document against those in its parallel document. Bad alignments are discarded. Only sentence pairs with high alignment confidence are kept in our parallel corpus. Our sentence alignment algorithm is based on Moore (2002).

For our ENG-THA translation system, we assembled three resources: a parallel training corpus, a development bitext (also called the lambda set) for training the feature combination weights $\{\lambda_i\}$, and a test corpus for BLEU and human evaluation. Both the lambda and the test sets have single reference translations per sentence.

Data Set	#Sentences
(ENG||THA) training	725K
(ENG,THA) lambda	2K
(ENG,THA) test	5K
THA LM text	10.3M
ENG LM text	45.6M

Table 3. Corpus size of parallel and monolingual data

Although it is well known that language translation pairs are not symmetric, we use these same resources to build our THA-ENG translation system due to the lack of additional corpora.

Our parallel MT corpus consists of approximately 725,000 English-Thai sentence pairs from various sources. Additionally we have 9.6 million Thai sentences, which are used to train a Thai 4-gram LM for ENG-THA translation, together with the Thai sentences in the parallel corpus. Trigrams and 4-grams that occur only once are pruned, and n-gram backoff weights are re-normalized after pruning, with the surviving KN smoothed probabilities intact (Kneser and Ney 1995). Similarly, a 4-gram ENG LM is trained for THA-ENG translation, on a total of 45.6M English sentences.

For both the lambda and test sets, THA LM incurs higher out-of-vocabulary (OOV) rates (1.6%) than ENG LM (0.7%), due to its smaller training set and thus smaller lexicon. Both translation directions define the maximum phrase/treelet length to be 4 and the maximum re-ordering jump to be 4 as well.

4.4 BLEU Scores

To evaluate our end-to-end performance, we compute case insensitive 4-gram BLEU scores. Translation outputs are WB first according to the Thai/English tokenizer, before BLEU scores are computed. The BLEU scores on the test sets are shown in Table 4. We are not aware of any previously published BLEU results for either direction of this language pair.

	BLEU
THA-ENG	0.233
ENG-THA	0.194

Table 4. Four-gram case-insensitive BLEU scores.

Figures 10 and 11 illustrate sample outputs for the each translation direction, with reference translations.

INPUT: ในประเทศไทยมีกล้วยไม้ประมาณ ๑๗๕ ชนิด ถ้าสูญพันธุ์ไปจากประเทศไทย ก็หมายถึงสูญพันธุ์ไปจากโลก
OUTPUT: In Thailand a Orchid approximately 175 type if extinct from Thailand. It means extinct from the world.
REF: In Thailand, there are about 175 species of Orchid. If they disappear from Thailand, they will be gone from the world.

Figure 10. THA-ENG Sample Translation Output

INPUT: In our nation the problems and barriers we face are just problems and barriers of law not selection or development.
OUTPUT: ในประเทศชาติของเรา ปัญหาและอุปสรรคที่เราเผชิญอยู่เพียงปัญหาและอุปสรรคของกฎหมายไม่เลือกหรือพัฒนา
REF: ในประเทศของเราปัญหาและอุปสรรค ก็เป็นปัญหาอุปสรรคทางด้านกฎหมาย แต่ไม่เป็นปัญหาอุปสรรคในการคัดเลือกและพัฒนาพันธุ์

Figure 11. ENG-THA Sample Translation Output

Although the translation quality is far from being perfect, SMT is making good process on building useful applications.

5 Conclusion and Future Work

Our maximum entropy model for Thai sentence-breaking achieves results which are consistent with contemporary work in this task, allowing us to overcome this obstacle to Thai SMT integration. This general approach can be applied to other South-East Asian languages in which space does not deterministically delimit sentence boundaries.

In Arabic writing, commas are often used to separate sentences until the end of a paragraph when a period is finally used. In this case, the comma character is similar to the space token in Thai where its usage is ambiguous. We can use the same approach (perhaps with different linguistic features) to identify which commas are sentence-breaking and which are not.

Our overall system incorporates a range of independent solutions to problems in Thai text processing, including character sequence normalization, tokenization, name and date identification, sentence-breaking, and Thai text respacing. We successfully integrated each solution into an existing large-scale SMT framework, obtaining sufficient quality to release the Thai-English language pair in a high-volume, general-domain, free public online service.

There remains much room for improvement. We need to find or create true Thai-English directional corpora to train the lambdas and to test our models. The size of our parallel corpus for Thai should increase by at least an order of magnitude, without loss of bitext quality. With a larger corpus, we can consider longer phrase length, higher-order n-grams, and longer reordering distance.

References

W. Aroonmanakun. 2007. Thoughts on Word and Sentence Segmentation in Thai. In *Proceedings of the Seventh International Symposium on Natural Language Processing, Pattaya, Thailand, 85-90*.

F. Avery Bishop, David C. Brown and David M. Meltzer. 2003. Supporting Multilanguage Text Layout and Complex Scripts with Windows 2000. *http://www.microsoft.com/typography/developers/uniscribe/intro.htm*

A. W. Black and P. Taylor. 1997. Assigning Phrase Breaks from Part-of-Speech Sequences. *Computer Speech and Language*, 12:99-117.

Thatsanee Charoenporn, Virach Sornlertlamvanich, and Hitoshi Isahara. 1997. *Building A Thai Part-Of-Speech Tagged Corpus (ORCHID)*.

Paisarn Charoenpornsawat and Virach Sornlertlamvanich. 2001. Automatic sentence break disambiguation for Thai. In *International Conference on Computer Processing of Oriental Languages (ICCPOL)*, 231-235.

S. F. Chen and J. Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, 310-318. Morristown, NJ: ACL.

J. N. Darroch and D. Ratcliff. 1972. Generalized Iterative Scaling for Log-Linear Models. *The Annals of Mathematical Statistics*, 43(5): 1470-1480.

Choochart Haruechaiyasak, Sarawoot Kongyoung, and Matthew N. Dailey. 2008. A Comparative Study on Thai Word Segmentation Approaches. In *Proceedings of ECTICON 2008*. Pathumthani, Thailand: ECTI.

Reinhard Kneser and Hermann Ney. 1995. Improved Backing-Off for M-Gram Language Modeling. In *Proceedings of International Conference on Acoustics, Speech and Signal Procesing (ICASSP)*, 1:181-184.

Philipp Koehn. 2004. Pharaoh: a Beam Search Decoder for Phrase-Based Statistical Machine Translation Models. In *Proceedings of the As-*

sociation of Machine Translation in the Americas (AMTA-2004).

Arul Menezes, and Chris Quirk. 2008. Syntactic Models for Structural Word Insertion and Deletion during Translation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*.

P. Mittrapiyanuruk and V. Sornlertlamvanich. 2000. The Automatic Thai Sentence Extraction. In *Proceedings of the Fourth Symposium on Natural Language Processing*, 23-28.

Robert C. Moore. 2002. Fast and Accurate Sentence Alignment of Bilingual Corpora. In *Machine Translation: From Research to Real Users (Proceedings, 5th Conference of the Association for Machine Translation in the Americas, Tiburon, California)*, Springer-Verlag, Heidelberg, Germany, 135-244

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41th Annual Meeting of the Association for Computational Linguistics*. Stroudsburg, PA: ACL.

David D. Palmer and Marti A. Hearst. 1997. Adaptive Multilingual Sentence Boundary Disambiguation. *Computational Linguistics*, 23:241-267.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual meeting of the Association for Computational Linguistics*, 311–318. Stroudsburg, PA: ACL.

Adwait Ratnaparkhi, 1996. A Maximum Entropy Model for Part-of-Speech Tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 133-142.

Jeffrey C. Reynar and Adwait Ratnaparkhi. 1997. A Maximum Entropy Approach to Identifying Sentence Boundaries, In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, 16-19.

Suphawut Wathabunditkul. 2003. Spacing in the Thai Language. *http://www.thailanguage.com/ref/spacing*