Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 6947–6957
Marseille, 11–16 May 2020

ThaiLMCut: Unsupervised Pretraining for Thai Word Segmentation

Suteera Seeha[1,2], Ivan Bilan[2], Liliana Mamani Sanchez[2], Johannes Huber[2], Michael Matuschek[2], Hinrich Schütze[1]
[1]Center for Information and Language Processing (CIS), Ludwig Maximilian University of Munich, Germany
[2]TrustYou GmbH, Munich, Germany
suteera.seeha@gmail.com
{ivan.bilan, liliana.sanchez, johannes.huber, michael.matuschek}@trustyou.com

Abstract

We propose ThaiLMCut, a semi-supervised approach for Thai word segmentation which utilizes a bi-directional character language model (LM) as a way to leverage useful linguistic knowledge from unlabeled data. After the language model is trained on substantial unlabeled corpora, the weights of its embedding and recurrent layers are transferred to a supervised word segmentation model which continues fine-tuning them on a word segmentation task. Our experimental results demonstrate that applying the LM always leads to a performance gain, especially when the amount of labeled data is small. In such cases, the F1 Score increased by up to 2.02%. Even on a big labeled dataset, a small improvement gain can still be obtained. The approach has also shown to be very beneficial for out-of-domain settings with a gain in F1 Score of up to 3.13%. Finally, we show that ThaiLMCut can outperform other open source state-of-the-art models achieving an F1 Score of 98.78% on the standard benchmark, InterBEST2009.

Keywords: word segmentation, Thai word segmentation, Thai tokenizer, semi-supervised, character language model, pretrained language model

1. Introduction

Word segmentation or tokenization is the task of splitting texts into word units. It is an important building block for many Natural Language Processing (NLP) tasks such as Text Classification, Named Entity Recognition (NER), and Machine Translation. Incorrect tokenization leads to misinterpretation of the input text which could potentially affect the performance of the downstream tasks. Tokenizing Thai text is especially difficult because words are written continuously without word delimiters. Spaces can be used in most cases to identify word boundaries in e.g. English or German texts, but this is not the case for Thai and some other Asian languages like Chinese, Japanese, or Vietnamese. In Thai, spaces are used to separate sentences. However, they are used for other purposes as well, such as separating phrases, clauses, and listed items. In practice, the use of spaces in Thai is rather arbitrary due to the nature of the Thai language which allows for a lot of flexibility.

State-of-the-art supervised word segmentation systems for Thai report reaching a performance between 97% and 99% F1 Score (Nararatwong et al., 2018; Jousimo et al., 2017; Kittinaradorn et al., 2019; Phuriphatwatthana, 2017; Kongyoung et al., 2015). However, some studies suggest these models might not be able to handle non-standard texts efficiently. Ronran et al. (2016) found that the Thai Lexeme Analyser (TLex) (Haruechaiyasak and Kongyoung, 2009), a tokenizer based on Conditional Random Fields (CRFs) (Lafferty et al., 2001), was not able to properly segment Twitter[1] posts. Lertpiya et al. (2018) revealed that Sertis (Jousimo et al., 2017), a model based on a bi-directional Recurrent Neural Network (RNN) (Rumelhart et al., 1986), performed significantly worse when tested on user-generated web content from the finance domain: the F1 Score of 99.18% from the evaluation on the standard benchmark dropped to 88.2%.

This is not surprising, because these models are trained on a corpus which is very different from user-generated data. Unlike Thai standard corpora, user-generated web content commonly contains misspellings, slang words, keyword tags, and abbreviations. Due to the large number of unknown words during test time (out-of-vocabulary), the performance drops accordingly. InterBEST2009 (Kosawat et al., 2009) and ORCHID (Sornlertlamvanich et al., 2000) are two publicly available corpora for Thai word segmentation that are used for training most of supervised learning-based models. ORCHID is a small corpus for Part-of-Speech (POS) tagging created from a collection of technical papers. InterBEST2009 consists of about five million words from four domains: novel, article, news, and encyclopedia. These corpora are quite limited in domain variety.

Since the performance of neural models based on supervised learning relies a lot on labeled data, the lack of domain variety of annotated corpora could lead to poor segmentation performance in some scenarios. This limits the

[1] twitter.com

possibility to exploit and process textual data from sources such as online web content which has become very important nowadays, especially in the business sector.

To address this problem, ideally, the corpus needs to be extended to cover the target domain. However, this usually comes with high costs and requires time, so it is often not feasible. An alternative approach is to integrate unsupervised learning into a supervised system. Unsupervised learning allows making use of plenty of raw unlabeled data without the expensive cost of manual annotation.

In this paper, we propose using unsupervised pretrained character representations from a bi-directional Language Model (LM) in order to improve a supervised word segmentation system. We call our model ThaiLMCut which stands for Thai Language Model Cut. Our contributions are as follows:

- We show that our semi-supervised approach without any complex fine-tuning methods can boost word segmentation performance in general, and especially when the amount of labeled data is limited
- We show that our approach can enhance the segmentation performance in an out-of-domain scenario
- We provide an implementation of ThaiLMCut as a publicly available word segmentation library[2]

2. Related Work

2.1. Thai Word Segmentation

For over 30 years, researchers have been actively working on solving the word segmentation problem for Thai. Early works used dictionary based methods where a given text is segmented according to words that are defined in dictionaries. In the presence of multiple segmentation choices, a method for selecting the best one needs to be applied. Two classical algorithms to choose the best segmentation are Longest Matching (Poowarawan, 1986) and Maximal Matching (Sornlertlamvanich, 1993). Since dictionary based approaches segment ambiguities according to static predefined rules without considering the context of the word, they cannot handle unknown words and ambiguities efficiently.

Later, many statistical models using supervised machine learning were developed to overcome the drawbacks of dictionary based approaches. Such statistical approaches include: Decision Trees, Naive Bayes, Support Vector Machines (Haruechaiyasak et al., 2008), Trigram Markov Models (Kawtrakul and Thumkanon, 1997), and feature based models using feature extraction algorithms (Meknavin et al., 1997). Kruengkrai et al. (2009) also proposed a model based on word and character clustering. Regarding methods using machine learning, models based on Conditional Random Fields (CRFs) have proven to be among the most popular and suitable models for this task (Kruengkrai et al., 2006; Haruechaiyasak and Kongyoung, 2009; Kongyoung et al., 2015; Nararatwong et al., 2018).

Haruechaiyasak and Kongyoung (2009) have shown that the lexical property of Thai characters provides effective information for identifying the word boundaries. They introduced the Thai character type feature set for CRF-based models. The feature set categorizes characters in ten groups based on their lexical functions. For example, some characters can only be present at the beginning of a word, some cannot be at the word ending, and some cannot appear alone. This information has shown to increase the model performance and thus is often exploited in later works as well (Kongyoung et al., 2015; Nararatwong et al., 2018). A combination of a CRF based model and dictionaries proposed by Kongyoung et al. (2015) has shown to achieve a relatively high F1 Score of 97.50%. While the most commonly used evaluation corpus InterBEST2009 is defined so that compound words are split into smallest word units, the work from Nararatwong et al. (2018) aimed to keep compound words as one unit. They developed a compound word merging algorithm that operates on top of a CRF-based tokenizer. The model without the compound word merging extension reported a very high segmentation performance with about 99% F1 Score on InterBEST2009.

In recent years, models based on neural networks also have achieved remarkably accurate segmentation. A number of open source libraries for word segmentation have been developed. Deepcut (Kittinaradorn et al., 2019) is a popular word segmentation tool, which is based on Convolutional Neural Networks (CNNs) (Krizhevsky et al., 2012) and Thai character type features. It reported an F1 Score of 98.18% on InterBEST2009. The Attacut model (Chormai et al., 2019), motivated by Deepcut, was designed to speed up the tokenization process while still maintaining a reasonable performance. Sertis (Jousimo et al., 2017), a neural network model based on bi-directional Gated Recurrent Units (GRUs) (Cho et al., 2014), claimed to yield an F1 Score of 99.18%. SynThai (Phuriphatwatthana, 2017) is a word segmentation and POS tagging model based on a multi-layer bi-directional Long Short-Term Memory (Bi-LSTM) (Schuster and Paliwal, 1997). Boonkwan and Supnithi (2017) suggested that word segmentation should be trained in combination with POS tagging. They proposed a model based on a bi-directional LSTM which adopted character embeddings to deal with unknown words. Lapjaturapit

[2] https://github.com/meanna/ThaiLMCUT

et al. (2018) introduced multi-candidate word segmentation using bi-directional LSTM together with character and character cluster embeddings which should help identify prefixes and suffixes of words. Their multi-candidate model can yield a very high recall, however, precision drops with increasing number of segmentation candidates.

A few works also focused on improving word segmentation for content from social networks. Ronran et al. (2016) developed a method to optimize segmentation results for Twitter data by exploiting local context from Twitter and global context from Thai Wikipedia[3]. They reached an F1 Score of 64.90% on a small manually annotated corpus. Beside misspellings and slang, texts from social networks can often contain words with intentionally repeated characters like "มากกกกกกก" (equivalent to "a lotttttt" in English), which is difficult to segment properly using a general tokenizer. To handle such cases, Haruechaiyasak and Kongthon (2013) proposed a dictionary based system with a rule based extension to merge and remove repeated characters. This method, however, still does not solve the out-of-vocabulary and misspelling problems.

Concerning the semi-supervised approaches, Fujii et al. (2017) have proposed a hybrid model which is a combination of CRFs and a non-parametric Bayesian unsupervised model for word segmentation which utilizes the nested Pitman-Yor language modeling (Mochihashi et al., 2009). The model reached 95.4% F1 Score on the novel domain.

2.2. Transfer Learning

Transfer Learning (Pan and Yang, 2010) is a technique of exploiting knowledge learned from a task to use in another similar task. It allows the target task to save training time and resource costs. The technique is widely used in computer vision (Deng et al., 2009; Tausczik and Pennebaker, 2010; Antol et al., 2015) and has gained a lot of interest in NLP as well.

Word embeddings (Mikolov et al., 2013b; Mikolov et al., 2013a; Pennington et al., 2014) are an example of a successful application of transfer learning in NLP. Word embeddings are word representations that encode semantic information about words. They are typically applied as a lookup table in the first layer of a neural network that maps a given word to its corresponding representation. Utilizing word embeddings has shown to improve the performance of various NLP tasks including Question Answering (Zhou et al., 2016), Sentiment Analysis (Yu et al., 2018), Dependency Parsing (Chen et al., 2015) and Machine Translation (Zhou et al., 2016; Zhang et al., 2017; Chen et al., 2018). Word embeddings can be trained from large unlabeled corpora using methods like Continuous Bag of Words (CBOW), Skip-Gram (Mikolov et al., 2013a), co-occurrence counts (Pennington et al., 2014), and by training a neural language model (Bengio et al., 2003). A drawback of traditional word embeddings is that each word in the vocabulary is typically assigned one explicit representation, while in fact, many words are ambiguous and can have more than one meaning depending on the context.

Recently, many studies have focused on developing word representations which are more context sensitive, for instance embeddings from BERT (Devlin et al., 2019), ULMfit (Howard and Ruder, 2018), and ELMo (Peters et al., 2018). These representations are even richer than the traditional word embeddings, since the models also consider the context in which the word appears before assigning the representation. Instead of applying the learned knowledge to only the first layer of the model like in the traditional word embeddings approach, ULMfit transfers both weights of the embedding layer and the recurrent layer of the pretrained LM to the downstream model. This method has shown to be a great performance boost for text classification. The authors also suggested a few fine-tuning methods to adapt the pretrained LM to downstream tasks including discriminative fine-tuning, slanted triangular learning rates, and gradual unfreezing.

Our approach is similar to ULMfit in the sense that we transfer weights from both sources, that is, from embeddings and from all recurrent layers of the pretrained LM to the word segmentation model. Since developing a fine-tuning method is not the main focus of this study, we prefer to leave this aspect to future work.

2.3. Language Models

A language model computes the probability distribution over a sequence of tokens. Given a sequence $T = t_1, t_2, t_3, ..., t_n$, a language model estimates the probability

$$P(T) = P(t_1, t_2, t_3, ..., t_n)$$

where the token t could be a word or a character.

The joint probability can be formulated as products of the conditional probability of each word given its previous context using the chain rule:

$$P(T) = \prod_{i=1}^{n} P(t_i | t_1, t_2 ..., t_{i-1})$$

LMs are important components in many NLP applications such as Speech Recognition, Machine Translation, Text Generation, and Spelling Correction. In recent years, pretrained word representations from recurrent neural LMs have gained increased interest from the research community

[3] th.wikipedia.org

due to their ability to improve the performance of various downstream tasks (Peters et al., 2018; Howard and Ruder, 2018; Devlin et al., 2019). A recurrent neural LM estimates the sequence's probability distribution by predicting the next word for each word in a sequence. While word-level LMs can capture syntactic and semantic features of words, character-level LMs are used for extracting sub-word information and improving word level representations (Kim et al., 2016; Bojanowski et al., 2015; Gerz et al., 2018; Verwimp et al., 2017; Peters et al., 2018).

The work from Hahn and Baroni (2019) revealed that the hidden states of a recurrent neural character LM that has been trained on unsegmented English corpora encode information that can help identify word boundaries. Our approach is motivated by the idea that integrating such information into a word segmentation system could increase its performance.

2.4. Bi-directional LSTM

Due to the ability to capture information in long sequences from both forward and backward directions, Bi-LSTMs have been applied and achieved great success in various sequence labeling tasks including POS tagging, chunking, NER (Huang et al., 2015; Alzboun et al., 2018), and also word segmentation (Yao and Huang, 2016; Ma et al., 2018; Jousimo et al., 2017; Phuriphatwatthana, 2017). Recently, Ma et al. (2018) showed that their Bi-LSTM model for Chinese word segmentation outperformed other more complex models on various benchmarks. The model applies pretrained character and bigram embeddings to the first layer of the network. For Thai word segmentation, models based on Bi-LSTM have also reported highly accurate results (Jousimo et al., 2017; Phuriphatwatthana, 2017).

Bi-directional information is also an important component of modern contextual pretrained word representation models including BERT, ELMo, and ULMfit. Forward and backward information helps the model learn context-sensitive representation by taking the whole sequence into consideration. Peters et al. (2017) proposed a pretrained bi-directional LM for sequence tagging. The model uses the concatenation of separate forward and backward unidirectional LSTMs. Both LSTMs are trained separately with no shared parameters unlike in the traditional architecture proposed by Schuster and Paliwal (1997).

ELMo learns deep contextualized word representations from a bi-directional LM and uses all its layers in prediction. It uses a similar structure of Bi-LSTM as the one outlined by Peters et al. (2017), but shares some weights between directions instead of using completely independent parameters. Howard and Ruder (2018) showed that using a regular Bi-LSTM for pretrained LM in ULMfit model can also yield a performance boost for text classification.

Sachan et al. (2017) demonstrated that the pretrained LM based on a regular Bi-LSTM can outperform forward or backward only models in biomedical NER. Their language model also leads to faster convergence and requires fewer labeled examples during fine-tuning. They pretrained a Bi-LSTM LM on unlabeled data then transferred its weights to a NER model which has the same architecture. Our approach is based on a similar idea.

Motivated by the success of pretrained word-level bi-directional language models and the findings in the work of Hahn and Baroni (2019) regarding the presence of useful information about word boundaries in a recurrent character LM, our work investigates the potential of using a pretrained bi-directional character LM in order to improve word segmentation performance. We demonstrate that a pretrained character LM based on a Bi-LSTM architecture without any sophisticated fine-tuning methods can yield an improvement on the task of Thai word segmentation.

3. Datasets and Experimental Setup

3.1. Datasets

To train the language model we mainly used unlabeled data from hotel reviews and also some data from InterBEST2009 depending on the experiments. For training and evaluating the word segmentation model, we use InterBEST2009.

TrustYou hotel reviews[4] dataset consists of 1,715,630 user reviews (approximately 218,196,000 Thai characters) from hotel review websites such as Agoda[5], Booking[6], TripAdvisor[7], etc. Foreign words, informal expressions, misspellings, transliterations, and informal Internet abbreviations are commonly found in the reviews. We preprocess the corpus by removing non-Thai characters, digits, special characters, and spaces. The resulting corpus contains only Thai characters without spaces.

InterBEST2009 is a tagged corpus for word segmentation, created by the National Electronics and Computer Technology Center (NECTEC)[8] for the purpose of the Thai Word Segmentation Software Contest competition in 2009 (Kosawat et al., 2009). The corpus consists of 4,678,998 words (58,113,858 Thai characters) from four domains including news, novels, encyclopedia, and academic articles. We will refer to them as *news*, *novel*, *encyclopedia*, and

[4] www.trustyou.com

[5] www.agoda.com

[6] www.booking.com

[7] www.tripadvisor.com

[8] www.nectec.or.th/en/

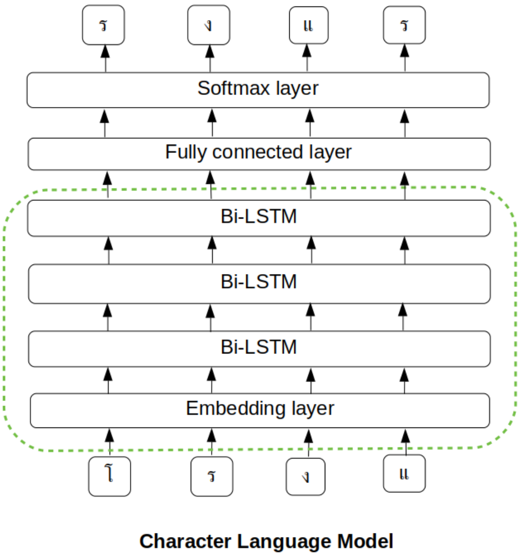


Figure 1: The architecture of the character language model

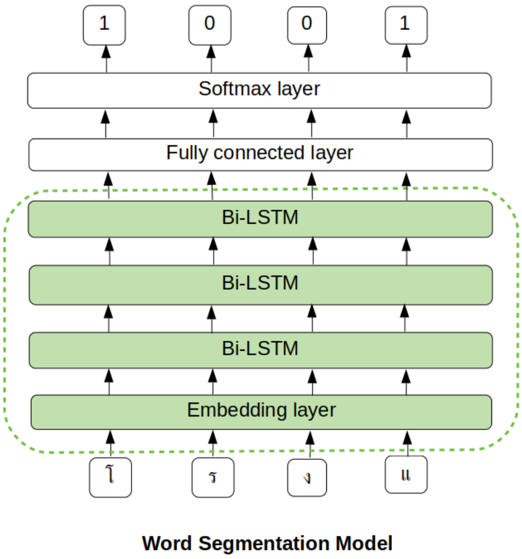


Figure 2: The architecture of the word segmentation model. The highlighted layers are the transferred parameters.

article, respectively. InterBEST2009 is the single benchmark for Thai word segmentation that is publicly available. The corpus is annotated with word boundary markers. Abbreviations, named entities, and poems are annotated using special tags. These tags are first removed, as are full stops that appear in the abbreviations. Named entities containing multiples words could cause inconsistency in the corpus. For example, “แม่น้ำ เจ้าพระยา” (River Chaopraya) is grouped as one named entity, while the word “แม่น้ำ” (river) is treated in general as an individual word. However, since named entities are important and appear often, we keep them in our corpus. On the other hand, a poem tag can cover multiple lines of a poem which were not annotated with boundary markers. Since poems do not contribute to the learning of the model, we remove all of them. After this step, we process the corpus the same way as the TrustYou corpus. The resulting corpus is then composed of only Thai characters without spaces.

3.2. Experimental Model Setup

In this section we describe the structure of our character language model and the word segmentation model together with their training and parameter settings.

3.2.1. Character Language Model

The first layer of the model is an embedding layer, followed by three Bi-LSTM layers, a linear fully connected output layer and a softmax activation function. Negative log-likelihood loss on the development set and Adam optimizer (Kingma and Ba, 2015) are used to optimize the model parameters.

We also applied dropout (Srivastava et al., 2014) at the embedding layer to prevent over-fitting and gradient clipping (Pascanu et al., 2013) to prevent the problem of vanishing gradients in the Bi-LSTM components.

Model parameters. The dataset for training the language model varies in each experiment. However, all LMs use the same hyperparameters. The embedding layer dimension is set to 200. The dimension of the hidden layer is 500. The learning rate of the Adam optimizer is 0.0001. Batch size and the sequence length are 60 and 100 characters respectively. This sequence length covers an average sentence length in Thai. Dropout is set to 0.01 and gradient clipping to 0.5.

3.2.2. Word Segmentation Model

Similar to other neural network-based word segmentation models, we formulate the task as a sequence labeling problem. For each character in a given input sequence, our word segmentation model learns to predict whether the character is a word boundary or not. The character is tagged with digit 1 for being a word boundary and digit 0 otherwise. The lower layers of the word segmentation model, including the embedding and the bi-LSTM layers, have the same structure as the language model. This allows to transfer weights from the pretrained LM to the model and later to fine-tune them for the word segmentation task. After a fully connected output layer, a softmax function classifies each character input into two classes (1 or 0). The model is trained to minimize the cross-entropy loss on the development set. Same as the LM, the word segmentation model also applies Adam optimizer, dropout, and gradient clipping. We apply the same hyperparameters as in the language model throughout all experiments, including the learning rate of 0.0001 which has shown to work well for the task.

3.3. Evaluation Metrics

To evaluate the word segmentation model, we report precision, recall, and F1 Score at the boundary-level, which is defined as follows:

$$\text{Precision} = \frac{\text{\# correctly predicted word boundaries}}{\text{\# characters predicted as word boundaries}}$$

$$\text{Recall} = \frac{\text{\# correctly predicted word boundaries}}{\text{\# real word boundaries}}$$

$$\text{F1} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision + recall}}$$

3.4. Impact on Training Data Sizes

Using pretrained representations has shown to be most beneficial when the amount of training data for the target task is small (Gururangan et al., 2019; Peters et al., 2017). In this experiment, we aim to answer how much impact the pretrained language model has on a word segmentation model which is trained on different dataset sizes.

To train and evaluate the tokenizer, we randomly split the InterBEST2009 corpus into 80% as the training set, 10% as the development set, and 10% as the test set. For the LM training, we combine the TrustYou corpus with all the data from InterBEST2009 except the 10% test portion. After merging and shuffling these two corpora, 90% of the resulting dataset are used as the training set and the remaining 10% as the development set to optimize the language model parameters.

After training the LM for 20 epochs, we transfer its weights and parameters to the word segmentation model, and train it until the error rate on the development set starts increasing (early stopping). In order to see the impact of the pretrained LM, we trained another word segmentation model whose weights are randomly initialized. We do the same experiment on the word segmentation models whose training data is reduced to 40%, 20%, 10% and 5% of the full dataset. To allow a fair comparison, the development and test portion, as well as hyperparameters and stopping strategy, are the same in all models.

3.5. Impact on Out-of-Domain Setting

In a real-world application, there will not always be annotated data available for the domain of interest and it might be difficult to create a new corpus for this specific domain. In this experiment, we try to find out whether our approach could improve the segmentation performance when the word segmentation model is trained on a different domain than the target domain.

To investigate this, we train a word segmentation model on each domain of InterBEST2009 and test the model on the other 3 domains. For example, we would use *news* domain for the training and evaluate the model on *novel*, *article*, and *encyclopedia*. Similar to the previous experiment, for each combination, we train a new language model using TrustYou corpus and InterBEST2009 without the test domain. If the representations learned from the language model lead to an improvement in word segmentation, it suggests that we might not need to retrain the LM on the target domain every time we deal with a new specific domain.

3.6. Final Model

This experiment compares the performance of our best model with other existing models. As a baseline, we use the Maximum Matching (Newmm) algorithm from the PyThaiNLP[9] library. It first generates all possible segmentation candidates using dictionaries, then selects the one that contains the fewest words. We also compare our model with three neural network-based models that reported high performance, Deepcut, Sertis, and Attacut. They are all trained on InterBEST2009 using the same (but not identical) partitioning of the dataset as us. We evaluate all models on the same 10% test portion from InterBEST2009 (the test set also used in section 3.4.).

4. Result and Analysis

4.1. Result: Impact on Training Data Sizes

Table 1 shows the comparison of word segmentation models with and without the use of the pretrained language model on different sizes of training data. In all cases, F1 Score has shown to increase to different extents when using the LM. We observe a trend that the performance gain becomes smaller with the increase in the training data size.

On 5% training data, adding the LM has shown the largest gain with the increase in F1 Score of 2.02%. On 10% training data, F1 Score increases by 1.25%, and on 20% training data by 0.65%. The performance gain becomes much smaller on 40% training set with 0.45% improvement in F1 Score and even smaller on the full training set (80% training data) with 0.15% improvement gain. Here, we observe that the performance of the model without the LM is already quite high with an F1 Score of 98.63% which was further boosted to 98.78% when integrating the pretrained LM.

Adding the language model in a word segmentation model has shown to be the most impactful when the training data is of limited amount. In the case of 5%, 10%, 20% setting, the models that utilize the language model can get even better performance than those trained on twice as much training data. A very high F1 Score in the original model without the pretrained LM also confirms that Bi-LSTM is a

[9] pypi.org/project/pythainlp/

Labeled data	Model WS			Model WS+LM			**Gain in F1**
	P	**R**	**F1**	**P**	**R**	**F1**	
5%	94.68	95.89	95.26	96.93	97.63	97.28	2.02
10%	96.69	95.95	96.32	97.70	97.43	97.57	1.25
20%	96.69	98.30	97.49	97.85	98.43	98.14	0.65
40%	98.00	98.03	98.01	98.31	98.60	98.46	0.45
80%	98.24	99.03	98.63	98.73	98.85	98.78	0.15

Table 1: Comparison of the WS model with weight transferring from the pre-trained LM, and the one without the LM on different sizes of training data

Model	**P**	**R**	**F1**
Newmm	93.13	81.77	87.08
Sertis	95.34	97.91	96.61
Attacut	98.21	98.56	98.39
Deepcut	98.28	98.52	98.40
ThaiLMCut	**98.73**	**98.85**	**98.78**

Table 2: Results of our best model (ThaiLMCut) compared to other word segmentation models

Train	Model WS				Model WS+LM				**Gain in F1**
	Test	**P**	**R**	**F1**	**Test**	**P**	**R**	**F1**	
news	novel	96.53	90.71	93.53	novel	96.93	96.38	96.66	3.13
	article	97.71	93.23	95.42	article	97.77	98.23	98.00	2.58
	ency	97.37	90.3	93.70	ency	97.44	96.17	96.80	3.10
novel	news	90.95	95.88	93.35	news	90.86	97.79	94.20	0.85
	article	95.06	97.28	96.15	article	95.39	98.51	96.92	0.77
	ency	94.73	96.14	95.43	ency	95.02	97.87	96.42	0.99
article	news	94.99	95.09	95.90	news	95.48	96.77	96.12	0.22
	ency	96.34	96.12	96.23	ency	97.33	96.23	96.78	0.55
	novel	96.19	92.58	94.35	novel	96.73	95.86	96.30	1.95
ency	news	91.57	96.22	93.83	news	93.83	96.42	95.11	1.28
	article	95.26	98.21	96.71	article	97.13	98.11	97.62	0.91
	novel	94.03	96.25	95.12	novel	96.39	96.18	96.29	1.17

Table 3: Comparison of the word segmentation model (WS) with weight transferring from the pre-trained language model (LM) and the one without the LM when test domain and train domain are different. "ency" refers to encyclopedia domain

suitable choice for the word segmentation task while adding pretrained representations can further enhance the performance even on a big dataset.

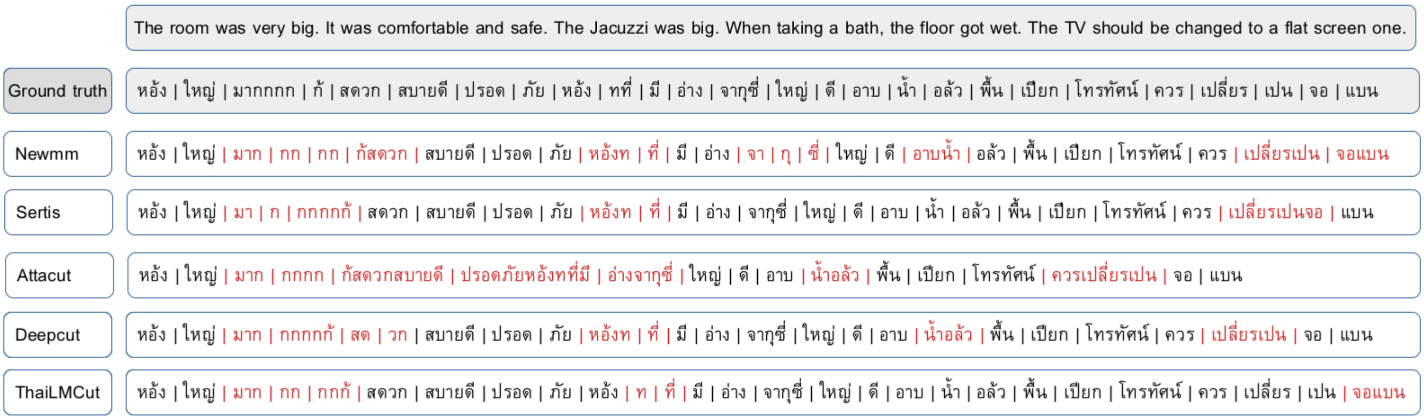
We also observe that all models that utilize the language model converge faster than the ones trained from scratch. For example, on the full training set the original model requires 11 epochs for the training and with the LM it requires only 7 epochs. Similar observations are found in other settings as well.

4.2. Result: Impact on Out-of-Domain Setting

Table 3 summarizes the results of the out-of-domain experiments. In all combinations, adding the pretrained LM has shown to improve word segmentation performance by up to 3.13% in terms of F1 Score. We observe that the pretrained LM constantly leads to a notable improvement when the tokenizer is tested on *novel* and *encyclopedia* while for *news* the improvement is rather modest.

In *news-novel* and *novel-encyclopedia* setting, the original models reach F1 Score of around 93% and when adding the language model F1 Score increases by more than 3% for both. On the other hand, in *novel-news* combination with similar initial F1 Score, the pretrained LM can only bring 0.85% improvement in F1 Score. The least improvement gain is observed when the model is trained on *article* and tested on *news* with the increase in F1 Score of 0.22%. The largest gain of 3.13% F1 Score is obtained on *news-encyclopedia* settings.

In most settings, the *news* domain has shown to benefit the least from the language model. When trained on *encyclopedia*, the *news* domain gets a bit more gain than *novel*. However, the improved F1 Score of *encyclopedia-news* is still below the one from *encyclopedia-novel*. Similar to the results from the first experiment on data size, the language model seems to bring more improvement when the initial performance of the word segmentation model is quite low than when the model already reaches highly accurate performance. One assumption about the different impact of the language model on each target domain would be that the unlabeled data that the LM is trained on might resemble *novel*



The room was very big. It was comfortable and safe. The Jacuzzi was big. When taking a bath, the floor got wet. The TV should be changed to a flat screen one.

Ground truth
หอ้ง | ใหญ่ | มากกกกก | ก้ | สดวก | สบายดี | ปรอด | ภัย | หอ้ง | ทที่ | มี | อ่าง | จากุซี่ | ใหญ่ | ดี | อาบ | น้ำ | อล้ว | พื้น | เปียก | โทรทัศน์ | ควร | เปลี่ยร | เปน | จอ | แบน

Newmm
หอ้ง | ใหญ่ | มาก | กก | กก | ก้สดวก | สบายดี | ปรอด | ภัย | หอ้งท | ที่ | มี | อ่าง | จา | กุ | ซี่ | ใหญ่ | ดี | อาบน้ำ | อล้ว | พื้น | เปียก | โทรทัศน์ | ควร | เปลี่ยรเปน | จอแบน

Sertis
หอ้ง | ใหญ่ | มา | ก | กกกกก้ | สดวก | สบายดี | ปรอด | ภัย | หอ้งท | ที่ | มี | อ่าง | จากุซี่ | ใหญ่ | ดี | อาบ | น้ำ | อล้ว | พื้น | เปียก | โทรทัศน์ | ควร | เปลี่ยรเปนจอ | แบน

Attacut
หอ้ง | ใหญ่ | มาก | กกกก | ก้สดวกสบายดี | ปรอดภัยหอ้งทที่มี | อ่างจากุซี่ | ใหญ่ | ดี | อาบ | น้ำอล้ว | พื้น | เปียก | โทรทัศน์ | ควรเปลี่ยรเปน | จอ | แบน

Deepcut
หอ้ง | ใหญ่ | มาก | กกกกก้ | สด | วก | สบายดี | ปรอด | ภัย | หอ้งท | ที่ | มี | อ่าง | จากุซี่ | ใหญ่ | ดี | อาบ | น้ำอล้ว | พื้น | เปียก | โทรทัศน์ | ควร | เปลี่ยรเปน | จอ | แบน

ThaiLMCut
หอ้ง | ใหญ่ | มาก | กก | กกก้ | สดวก | สบายดี | ปรอด | ภัย | หอ้ง | ท | ที่ | มี | อ่าง | จากุซี่ | ใหญ่ | ดี | อาบ | น้ำ | อล้ว | พื้น | เปียก | โทรทัศน์ | ควร | เปลี่ยร | เปน | จอแบน

Figure 3: Tokenization output of a hotel review with multiple misspellings. The parts marked with red are those that are wrongly predicted by the model.

and *encyclopedia* domain more than *news* domain. Accordingly, the language model yields high performance boost for both *novel* and *encyclopedia* in most settings, while the improvement for *news* is often modest.

We assume that the language model might be able to capture and learn the character type features (mentioned in Section 2.1.) from unlabeled data by itself and generate representations in a way that helps detect word boundaries. As a result, the representations from the LM have shown to have a positive impact on the word segmentation performance.

4.3. Result: Final Model

Table 2 demonstrates the performance of our model compared to other four word segmentation models on the same evaluation set. In our previous experiments, the model that yields the best performance is the one trained on the full dataset and utilizing the pretrained LM. The result shows that our proposed model performs better compared to other models reaching an F1 Score of 98.78%. All neural network-based models outperform the baseline Newmm as expected. The second best model is Deepcut with 98.40% F1 Score. It outperforms Attacut by a small margin, however, the segmentation speed of Attacut is substantially faster, also when compared to other models. Sertis achieves the lowest F1 Score among other neural network-based models, outperforming only the Newmm baseline. We notice that the result of Sertis is surprisingly low when considering the reported performance. We found that Sertis used a different evaluation method by counting all the predicted characters instead of only characters that mark word boundaries which is the standard way of evaluating a word segmentation system. This might explain their reported high F1 Score.

We suppose that the language model used in this study, which is trained mainly on hotel reviews, could be the most beneficial for segmenting user-generated data in the hotel domain. However, there is no annotated corpus for the hotel domain publicly available at the current time. Figure 3 demonstrates the performance of each model on a hotel review which contains multiple misspellings. Attacut seems to prefer long tokenizations and has the most difficulty dealing with misspellings, while other tokenizers produce outputs with minor mistakes. THAILMCUT has proven to be the most accurate in this example. For a more exhaustive evaluation of this domain, further investigation is needed.

5. Conclusion

We proposed a semi-supervised approach for Thai word segmentation using a pretrained character language model fine-tuning. After a Bi-LSTM language model is trained on substantial unlabeled corpora, its weights are transferred to a word segmentation model which has the same structure besides its output layer. The model then continues the training using labeled data to fine-tune the pretrained weights for the word segmentation task.

Our results showed that the approach consistently leads to a performance gain in various settings. The language model has proven to be the most beneficial when only a small amount of labeled data is available. In such cases, our results showed that F1 Score could be increased by up to 2.02%. The approach has also shown to boost the segmentation performance in all of the out-of-domain datasets in our experiment with the gain from 0.22% to 3.13% F1 Score. Our final model, THAILMCUT, outperforms other state-of-the-art neural network-based models achieving an F1 Score of 98.78%. In the future, we would like to investigate the performance of our model on the hotel review domain. Additionally, we would want to explore better fine-tuning methods and other options for training the LM which could be more efficient than the Bi-LSTM, for instance, CNNs or Attention-based approaches.

6. Bibliographical References

Alzboun, S. D. A., Tawalbeh, S. K., Al-Smadi, M., and Jararweh, Y. (2018). Using Bidirectional Long Short-Term Memory and Conditional Random Fields for Labeling Arabic Named Entities: A Comparative Study. In Proceedings of the 5th International Conference on Social Networks Analysis, Management and Security, pages 135–140.

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C. L., and Parikh, D. (2015). VQA: Visual question answering. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), pages 2425–2433.

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A Neural Probabilistic Language Model. *The Journal of Machine Learning Research*, 3:1137–1155.

Bojanowski, P., Joulin, A., and Mikolov, T. (2015). Alternative Structures for Character-Level RNNs. *CoRR*, abs/1511.06303.

Boonkwan, P. and Supnithi, T. (2017). Bidirectional Deep Learning of Context Representation for Joint Word Segmentation and POS Tagging. In Proceedings of the International Conference on Computer Science, Applied Mathematics and Applications, pages 184–196.

Chen, W., Zhang, M., and Zhang, Y. (2015). Distributed feature representations for dependency parsing. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):451–460.

Chen, K., Zhao, T., Yang, M., Liu, L., Tamura, A., Wang, R., Utiyama, M., and Sumita, E. (2018). A neural approach to source dependence based context model for statistical machine translation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(2):266–280.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pages 1724–1734.

Chormai, P., Prasertsom, P., and Rutherford, A. (2019). AttaCut: A Fast and Accurate Neural Thai Word Segmenter. *ArXiv*, abs/1911.07056.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Li, F.-F. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, pages 248–255.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*, abs/1810.04805.

Fujii, R., Domoto, R., and Mochihashi, D. (2017). Non-parametric Bayesian Semi-supervised Word Segmentation. *Transactions of the Association for Computational Linguistics*, 5:179–189.

Gerz, D., Vulić, I., Ponti, E., Naradowsky, J., Reichart, R., and Korhonen, A. (2018). Language Modeling for Morphologically Rich Languages: Character-Aware Modeling for Word-Level Prediction. *Transactions of the Association for Computational Linguistics*, 6:451–465.

Gururangan, S., Dang, T., Card, D., and Smith, N. A. (2019). Variational Pretraining for Semi-supervised Text Classification. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5880–5894.

Hahn, M. and Baroni, M. (2019). Tabula Nearly Rasa: Probing the Linguistic Knowledge of Character-level Neural Language Models Trained on Unsegmented Text. *Transactions of the Association for Computational Linguistics*, 7:467–484.

Haruechaiyasak, C. and Kongthon, A. (2013). LexToPlus: A Thai lexeme tokenization and normalization tool. In Proceedings of the 4th Workshop on South and Southeast Asian Natural Language Processing, pages 9–16. Asian Federation of Natural Language Processing.

Haruechaiyasak, C. and Kongyoung, S. (2009). TLex: Thai Lexeme Analyser Based on the Conditional Random Fields. In Proceedings of the International Symposium on Natural Language Processing.

Haruechaiyasak, C., Kongyoung, S., and Dailey, M. (2008). A comparative study on thai word segmentation approaches. In Proceedings of the 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, volume 1, pages 125–128.

Howard, J. and Ruder, S. (2018). Universal Language Model Fine-tuning for Text Classification. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 328–339. Association for Computational Linguistics.

Huang, Z., Xu, W., and Yu, K. (2015). Bidirectional LSTM-CRF Models for Sequence Tagging. In Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation.

Jousimo, J., Laokulrat, N., Carr, B., Thongthanomkul, E., and Satayamas, V. (2017). Thai word segmentation with bi-directional RNN. [online]. Available: `https://github.com/sertiscorp/`

thai-word-segmentation, accessed on 11-11-2019.

Kawtrakul, A. and Thumkanon, C. (1997). A Statistical Approach to Thai Morphological Analyzer. In Proceedings of the 5th Workshop on Very Large Corpora.

Kim, Y., Jernite, Y., Sontag, D., and Rush, A. M. (2016). Character-aware Neural Language Models. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, pages 2741–2749. AAAI Press.

Kingma, D. and Ba, J. (2015). Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980.

Kittinaradorn, R., Titipat, A., Chaovavanich, K., Srithaworn, K., Chormai, P., Kaewkasi, C., Ruangrong, T., and Oparad, K. (2019). DeepCut: A Thai word tokenization library using Deep Neural Network [online] . Available: http://doi.org/10.5281/zenodo.345770, accessed on 11-11-2019.

Kongyoung, S., Rugchatjaroen, A., and Kosawat, K. (2015). TLex+: a Hybrid Method using Conditional Random Fields and Dictionaries for Thai Word Segmentation. In Proceedings of the 10th Int. Conf. Knowl., Inform. and Creativity Support Syst. (KICSS), pages 112–125.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou and K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., pp. 1097–1105.

Kruengkrai, C., Sornlertlamvanich, V., and Isahara, H. (2006). A Conditional Random Field Framework for Thai Morphological Analysis. In Proceedings of the Fifth International Conference on Language Resources and Evaluation. European Language Resources Association.

Kruengkrai, C., Uchimoto, K., Kazama, J., Torisawa, K., Isahara, H., and Jaruskulchai, C. (2009). A Word and Character-Cluster Hybrid Model for Thai Word Segmentation. In Proceedings of the 8th International Symposium on Natural Language Processing, pages 1544 – 1549.

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In Proceedings of the 18th International Conference on Machine Learning, pages 282–289. Morgan Kaufmann Publishers Inc.

Lapjaturapit, T., Viriyayudhakorn, K., and Theeramunkong, T. (2018). Multi-Candidate Word Segmentation using Bi-directional LSTM Neural Networks. In Proceedings of the 2018 International Conference on Embedded Systems and Intelligent Technology International Conference on Information and Communication Technology for Embedded Systems.

Lertpiya, A., Chaiwachirasak, T., Maharattanamalai, N., Lapjaturapit, T., Chalothorn, T., Tirasaroj, N., and Chuangsuwanich, E. (2018). A Preliminary Study on Fundamental Thai NLP Tasks for User-generated Web Content. In Proceedings of the International Joint Symposium on Artificial Intelligence and Natural Language Processing.

Ma, J., Ganchev, K., and Weiss, D. (2018). State-of-The-Art Chinese Word Segmentation with Bi-LSTMs. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, page 4902–4908.

Meknavin, S., Charoenpornsawat, P., and Kijsirikul, B. (1997). Feature-based Thai Word Segmentation. In Proceedings of the Natural Language Processing Pacific Rim Symposium 1997, pages 41–46.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient Estimation of Word Representations in Vector Space. In Proceedings of the 1st International Conference on Learning Representations.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, pages 3111–3119. Curran Associates Inc.

Mochihashi, D., Yamada, T., and Ueda, N. (2009). Bayesian Unsupervised Word Segmentation with Nested Pitman-Yor Language Modeling. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, page 100–108.

Nararatwong, R., Kertkeidkachorn, N., Cooharojananone, N., and Okada, H. (2018). Improving Thai Word and Sentence Segmentation Using Linguistic Knowledge. *IEICE Transactions on Information and Systems*, pages 3218–3225.

Pan, S. J. and Yang, Q. (2010). A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28, pages III–1310–III–1318. JMLR.org.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543. Association for Computational Linguistics.

Peters, M., Ammar, W., Bhagavatula, C., and Power, R. (2017). Semi-supervised sequence tagging with bidirectional language models. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1756–1765, Vancouver, Canada. Association for Computational Linguistics.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep Contextualized Word Representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 2227–2237. Association for Computational Linguistics.

Phuriphatwatthana, W. (2017). Synthai: Thai Word Segmentation and Part-of-Speech Tagging with Deep Learning. [online]. Available: `https://github.com/KrakenAI/SynThai`, accessed on 11-11-2019.

Poowarawan, Y. (1986). Dictionary-based Thai Syllable Separation. In Proceedings of the 9th Electronics Engineering Conference.

Ronran, C., Unankard, S., Nadee, W., Khomwichai, N., and Sirirangsi, R. (2016). Thai Word Segmentation on Social Networks with Time Sensitivity. In Proceedings of the Knowledge Management International Conference 2016.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088):533–536.

Sachan, D. S., Xie, P., Sachan, M., and Xing, E. P. (2017). Effective Use of Bidirectional Language Modeling for Transfer Learning in Biomedical Named Entity Recognition. In Proceedings of the 3rd Machine Learning for Healthcare Conference, page 248.

Schuster, M. and Paliwal, K. K. (1997). Bidirectional Recurrent Neural Networks. *IEEE Transactions on Signal Processing*, pages 2673 – 2681.

Sornlertlamvanich, V. (1993). Word Segmentation for Thai in Machine Translation System. *Machine Translation, NECTEC*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Tausczik, Y. R. and Pennebaker, J. W. (2010). The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods. *Journal of Language and Social Psychology*, page 24–54.

Verwimp, L., Pelemans, J., Van Hamme, H., and Wambacq, P. (2017). Character-Word LSTM Language Models. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 417–427. Association for Computational Linguistics.

Yao, Y. and Huang, Z. (2016). Bi-directional LSTM Recurrent Neural Network for Chinese Word Segmentation. In Proceedings of ICONIP 2016, pages 345–353.

Yu, L.-C., Wang, J., Lai, K. R., and Zhang, X. (2018). Refining Word Embeddings Using Intensity Scores for Sentiment Analysis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pages 671 –681.

Zhang, B., Xiong, D., Su, J., and Duan, H. (2017). A Context-Aware Recurrent Encoder for Neural Machine Translation. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, pages 2424 – 2432.

Zhou, G., Xie, Z., He, T., Zhao, J., and Hu, X. T. (2016). Learning the Multilingual Translation Representations for Question Retrieval in Community Question Answering via Non-Negative Matrix Factorization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pages 1305 – 1314.

7. Language Resource References

Krit Kosawat and Monthika Boriboon and Patcharika Chootrakool and Ananlada Chotimongkol and Supon Klaithin and Sarawoot Kongyoung and Kanyanut Kriengket and Sitthaa Phaholphinyo and Sumonmas Purodakananda and Tipraporn Thanakulwarapas and Chai Wutiwiwatchai. (2009). BEST 2009 : Thai Word Segmentation Software Contest. Proceedings of the 8th International Symposium on Natural Language Processing, Available : `https://www.nectec.or.th/corpus/index.php?league=pm`.

Sornlertlamvanich, Virach and Takahashi, Naoto and Isahara, Hitoshi. (2000). Thai Part-of-Speech Tagged Corpus: ORCHID. Journal of the Acoustical Society of Japan, Available : `https://www.nectec.or.th/corpus/index.php?league=pm`.