THESIS APPROVAL

GRADUATE SCHOOL, KASETSART UNIVERSITY

Master of Engineering (Computer Engineering)

DEGREE

Computer Engineering — **FIELD**

Computer Engineering — **DEPARTMENT**

TITLE: Thai-related Foreign-language Specific Web Crawler

NAME: Mister Tanaphol Suebchua

THIS THESIS HAS BEEN ACCEPTED BY

_____ **THESIS ADVISOR**

(Associate Professor Arnon Rungsawang, Ph.D.)

_____ **THESIS CO-ADVISOR**

(Mister Bundit Manaskasemsak, D.Eng.)

_____ **DEPARTMENT HEAD**

(Associate Professor Anan Phonphoem, Ph.D.)

APPROVED BY THE GRADUATE SCHOOL ON _____

_____ **DEAN**

(Associate Professor Gunjana Theeragool, D.Agr.)

THESIS

THAI-RELATED FOREIGN-LANGUAGE SPECIFIC WEB CRAWLER

TANAPHOL SUEBCHUA

A Thesis Submitted in Partial Fulfillment of
the Requirements for the Degree of
Master of Engineering (Computer Engineering)
Graduate School, Kasetsart University
2014

Tanaphol Suebchua 2014: Thai-related Foreign-language Specific Web Crawler. Master of Engineering (Computer Engineering), Major Field: Computer Engineering, Department of Computer Engineering. Thesis Advisor: Associate Professor Arnon Rungsawang, Ph.D. 41 pages.

National web archive that preserves national knowledge for generations to come has been successfully made available through a domain-specific web crawler for years. However, that kind of crawler still misses many foreign language web pages which are also related to the nation. This thesis proposes new machine learning based focused crawling approach to collect national related web pages written in a foreign language, especially the English web pages that related to Thailand. We first propose a notion of website segment which groups the related web pages from their same longest directory paths. Thus, rather than looking for a target web page as proposed in many traditional focused crawling approaches, an ensemble classifier is trained with several features to predict the relevancy of the website segments. The most relevant website segments in the crawling frontier are then enqueued to download. The preliminary experiments on the real web space show that our approach can provide better promising harvest results than the Breadth-First and Best-First baselines for the Thai-tourism, Thai-estate and Thai-diving topics which are written in English language.

_____ _____ __ / __ / __

Student's signature Thesis Advisor's signature

ACKNOWLEDGEMENTS

First of all, I would like to thank to thanks to my parents for their unconditional love and support. I would like to thank Associate Professor Dr. Arnon Rungsawang and Dr.Bundit Manaskasemsak for their teaching and advice to complete this thesis. I gratefully acknowledge to the Junior Science Talent Project (JSTP), NSTDA, Thailand, for funding this research.

Tanaphol Suebchua

June 2014

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

THAI-RELATED FOREIGN-LANGUAGE SPECIFIC WEB CRAWLER

INTRODUCTION

To build a national web archive (British Library, 2011; National Diet Library, 2011; National Library Singapore, 2013; National Library of Australia, 2013; Gomes *et al.*, 2008), we have to gather web pages which are related to the specific country as much as possible. Naively, one may think of using a web crawler to collect every web page from the Internet and filter out the irrelevant ones from the collection. However, this method is impractical because of limited computational resource and time. Many researchers and search engine administrators then attempt to use another type of web crawler call "domain specific web crawler" to collect the web pages in a specific country domain name (Baeza-Yates *et al.*, 2004, 2005; Gomes *et al.*, 2008). For example, to construct a Thai web archive, we could limit our web crawler to download only web pages in ".th" domain name.

Although, the domain-specific web crawling strategy may have been used in many web archive researches, it cannot download a lot of related web pages which are resided in other domain names, such as .com, .net and etc. To solve that problem, a group of researchers assume that the web pages which are related to a specific country could be written in its native language, and propose another crawling strategy called "language-specic web crawler" which is able to find relevant web pages by not restricting itself only in a national domain name (Somboonviwat *et al.*, 2006; Tamura *et al.*, 2007; Srisukha *et al.*, 2008; Alabbad *et al.*, 2009; Tadapak *et al.*, 2010). For example, to build a Thai national web archive, web pages which belong to .th domain name and those which are written in Thai language have to be accumulated as much as possible.

However, the aforementioned two types of crawlers cannot easily manage to find web pages whose contents are written in other foreign languages, but still related to the nation. The missing web pages may also contain some important information

such as aspect, thought and some useful information to foreigners. For example, English web pages which contain information about traveling points and places in Thailand could be beneficial to foreigners who want to visit Thailand. In order to gather those web pages, in this thesis, we propose a new focused web crawling approach for finding Thai related web pages written in a foreign language.

Instead of considering that the largest unit of crawling is a web page or a website as proposed by other researchers, we rather propose to find the set of related web pages which share the same longest logical directory paths, called the website segment. We hypothesize that there are some relations between features extracted from the source website segment and the characteristic of the destination website segment. Thus, in this thesis, we design our crawling approach to consist of two main parts: Segment Predictor and Segment Crawler. The Segment Predictor is a machine learning based component which is trained with several features from the downloaded website segments. It is used to predict whether the unvisited destination website segments could hosts relevant web pages, and prioritize those segments in the crawling frontier. The website segment with the highest priority will be dequeued and sent to the Segment Crawler for collecting all web pages within that website segment later.

The Internet evaluation results on the Thai-tourism, Thai-estate, and Thai-diving topics in English language show that our proposed approach can provide better harvest rates comparing to the Breadth-First and Best-First baselines. In the other words, our crawler can find more relevant web pages more than the others within the same downloaded times.

OBJECTIVE

To develop an effective crawling framework for collecting Thai-related web page which are written in foreign language.

LITERATURE REVIEW

1. Web crawler

Web crawler, or also known as web spider, is an automated program that is used to collect web content from the Internet. As shown in figure 1, the simple web crawler is composed of three main component, i.e., (1) Downloader, (2) Parser and (3) Crawling Frontier. The crawling process begins after the user assigns the starting URLs to the Crawling Frontier. These URLs is then dequeued and downloaded by the Downloader. After finished downloading, the HTML Parser component is then used to extract all URLs from the downloaded web pages and save those newly unvisited URLs into the Crawling Frontier later. The web crawler will dequeue the unvisited URLs from the Crawling Frontier and repeat the crawling process gain. This looping process will be stopped only if there is no URL left in the Crawling Frontier, or it meets the conditions that are set by user. For instance, user can set the crawler to stop crawling after retrieving a specified number of downloaded web pages.

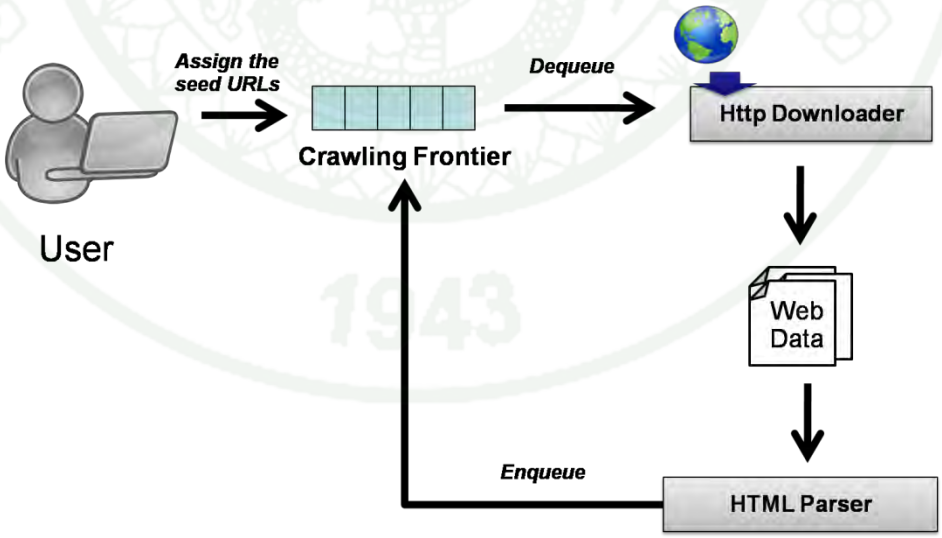

Figure 1 Web crawling process

2. Focused web crawler

Focused web crawler (Chakrabarti *et al.*, 1999) is a type of web crawler that is designed for finding web pages with some specific characteristic, for instance, finding the web pages written in Thai language. The focused web crawler normally uses heuristic rules or machine learning technique to predict the relevancy of unvisited URLs. The prediction result is later used to prioritize the Crawling Frontier. The URLs which are the most probable relevant web pages will be downloaded first. Therefore, if the rules or the machine learning classifiers work accurately, this type of crawler can find more relevant web pages more than the simple web crawler within the same downloaded times.

To our knowledge, Somboonviwat *et al.* (2006) and Tamura *et al.* (2007) were the first research team who proposed an adaption of focused crawler, called the "language-specific web crawler", to collect web pages written in a specific language. They used an N-gram based text categorization, called TextCat (1994) , to first detect the language of web pages, and then developed a set of heuristic rules concluded from the observation of the link characteristics on a small sample set of Thai web graph to direct their crawler to the unvisited target web pages. Srisukha *et al.* (2008) also first proposed a machine learning based language-specific web crawler to predict the relevancy of the unvisited web pages. Features extracted from link characteristics had been used to train a Naive Bayes classifier to further predict the relevancy of web pages. Following the Srisukha *et al.* work, Tadapak *et al.* (2010) proposed a framework to predict the relevancy of a website rather than an individual web page. Features extracted from already downloaded websites had been used to train an SVM classifier. The entire web pages of the highest relevant website were then downloaded.

Unlike those language-specific web crawlers, we here propose to find the relevant website segment, instead of relevant web pages or the websites that has high probability to host relevant web pages. We extract several features from the already downloaded website segments, and use them to train an ensemble classifier to predict

the relevancy of unvisited website segments in the crawling frontier. The most relevant ones will be chosen to download first.

3. Classifier Ensemble

The classifier ensemble or classifier fusion (Ranawana *et al.*, 2006) is one of the machine learning techniques that is used to enhance the classification accuracy in many classification tasks. It is based on the hypothesis that classifiers which are trained with different training dataset, and/or classifier models, could predict the samples differently. Therefore, rather than using only single classifier for a classification task, this technique uses multiple classifiers for prediction instead. Parallel Classifier ensemble that we apply in our crawler here is also one of the variations of this technique. During the prediction, an input feature vector will be fed into each classifier in the system. After all classifiers finished predicting, a combiner function will gather all classification results and produce the final prediction later.

4. Imbalance dataset problem

An imbalance dataset (He *et al.*, 2009) is one of the main problems in many machine learning applications, including the machine learning based focused crawler. It is the dataset which the numbers of positive and negative examples are much different. Thus, this kind of dataset is not suitable for training any classifier.

4.1. Under-sampling technique

The under-sampling technique (He *et al.*, 2009; Garcia *et al.*, 2009) is one of the basic techniques that is used for solving the imbalance dataset problem. This technique will randomly filter out the examples from majority class until the number of examples in each class is equal to each other. For example, if the under-sampling is applied to the dataset in table 1, around eight hundred of positive samples will be randomly removed from the training dataset.

Table 1 Sample of an imbalance dataset

Class	Number of samples
Positive	1000
Negative	200

4.2. Performance metric for imbalance learning

When evaluating the classifier on the imbalance dataset, the simple accuracy may not be suitable to use as the performance metric since it is affected by the class distribution of the imbalance dataset. For instance, suppose that an imbalance dataset contains many positive examples and a few negative examples. If we evaluate a classifier, it will always predict the input sample as positive, with this dataset. The accuracy will be high although the classifier cannot predict the negative examples correctly. Thus, various metrics have also been proposed to measure the performance of the crawler in an imbalance dataset. The Geometric-mean (He *et al*., 2006) is one of them. For the binary classification task, it can be calculate by using the following equation (1).

$$G-mean=\sqrt{\frac{TP}{TP+FN}\times\frac{TN}{TN+FP}} \quad (1)$$

Here we suppose to classify the sample into two classes named positive and negative, and the term TP, FN, TN and FP can be calculated by using equation (2-5).

$$TP\ (True\,Positive)=\frac{Number\,of\ \ positive\ samples\ that\ \ are\ classified\ \ as\ \ positive}{Number\,of\ all\ \ positive\ samples\ in\ \ dataset} \quad (2)$$

$$FP\ (False\,Positive)=\frac{Number\,of\ \ nagative\ samples\ that\ \ are\ classified\ \ as\ \ positive}{Number\,of\ all\ \ \ positive\ samples\ in\ \ dataset} \quad (3)$$

$$FN\ (False\ Negative) = \frac{Number\ of\ positive\ samples\ that\ are\ classified\ as\ negative}{Number\ of\ all\ negative\ samples\ in\ dataset} \quad (4)$$

$$TN\ (True\ Negative) = \frac{Number\ of\ nagative\ samples\ that\ are\ classified\ as\ negative}{Number\ of\ all\ negative\ samples\ in\ dataset} \quad (5)$$

5. k-fold cross validation

The k-fold cross validation is a technique to estimate the accuracy of the classifier when it performs in real application. This technique initially divides the dataset into k fold equally. After finished division, it will repeat the evaluation process k times. In each time, a sample fold is selected as the validation fold for validating the classifier model that is built by the remaining k-1 folds. Finally, the average accuracy from all validation folds is calculated. After finished evaluation process k times, which all the sample fold are used as the validation fold, the average accuracy is finally calculated.

In the case that the dataset is imbalance, it may not suitable to use k-fold cross validation to estimate the accuracy of the model since all sample folds are also imbalance. Therefore, we apply the under-sampling technique to the training folds before we use them to build the classifier model. In the validation phase, we rather observe the Geometric-mean instead of the accuracy too.

6. Harvest rate and Coverage rate

In focused crawling research, the objective is to use the limited resource to find many relevant web pages as much as possible. Therefore, the harvest rate and the coverage rate are used as the performance metrics. As shown in the following equation (6-7), the harvest rate is the ratio of the number of relevant web pages and the number of all web pages that has been downloaded at that time. The crawler that archives higher harvest rate means that it can find more relevant web pages than the others within the same downloaded times.

$$Harvest\,rate = \frac{Number\ of\ relevant\ web\ pages\ \ retreived}{Number\ of\ retreived\ web\ pages} \qquad (6)$$

For the coverage rate, it is the ratio of the number of relevant web pages and the number of all relevant web pages resided in the dataset.

$$Coverage\ rate = \frac{Number\ of\ relevant\ web\ pages\ \ retreived}{Number\ of\ all\ \ relevant\ \ web\ pages\ in\ dataset} \qquad (7)$$

MATERIALS AND METHODS

Materials

1. Hardware Equipments
   1.1. Intel ® Xeon ™ 2.80 GHz
   1.2. RAM 24 Gigabyte
   1.3. Hard disk 6 Terabyte
   1.4. Gigabit Ethernet network

2. Software Equipments
   2.1. Linux CentOS 5.1
   2.2. Java Software Development Kit (JDK) 7.0
   2.3. Eclipse IDE JUNO
   2.4. InetAddressLocator 2.23
   2.5. JerhichoHTMLParser 4.1
   2.6. Berkeley DB 5.0
   2.7. WEKA Library

Methods

1. Relevant web page classification

In this research, a web page is relevant only if that web page is written in a specified language and also contains content related to a specified target topic. As mentioned in the scope section, our target web pages in this work are English web pages which are related either to Thai-tourism, Thai-estate, or Thai-diving topic only. Since we will evaluate our crawler one topic at a time, we thus designed our relevant web page classification process for Thai-tourism and Thai-estate as shown in figure 2. We first use the language classifier from LangDetect (2011) library to detect the language of the input web page fist. If the web page is not written in English, we then

consider it as the irrelevant web pages. Otherwise, the web page will be sent to the topic classifier for classifying whether the web page is related to the target topic later.

To create the topic classifier model for Thai-tourism and Thai-estate topics, we first build the training dataset by selecting some websites URLs from the following categories in Open Directory Project (2013) dataset:

- English language websites whose contents are related to target topic in Thailand
- English language websites whose contents are unrelated to the target topic

We launch the Breadth-First search crawler to collect web pages at most 300 pages for each website. The web pages which are not written in target language are filtered out by the language classifier. The remaining web pages from the first category websites are labeled as positive examples. The leftover are labeled as negative ones. During the pre-processing, terms from TITLE, BODY and ANCHOR tags have been extracted, and parsed through stopwords and stemming routines. Each web page is then represented by its term-frequency feature vector (Manning *et al.*, 2008).

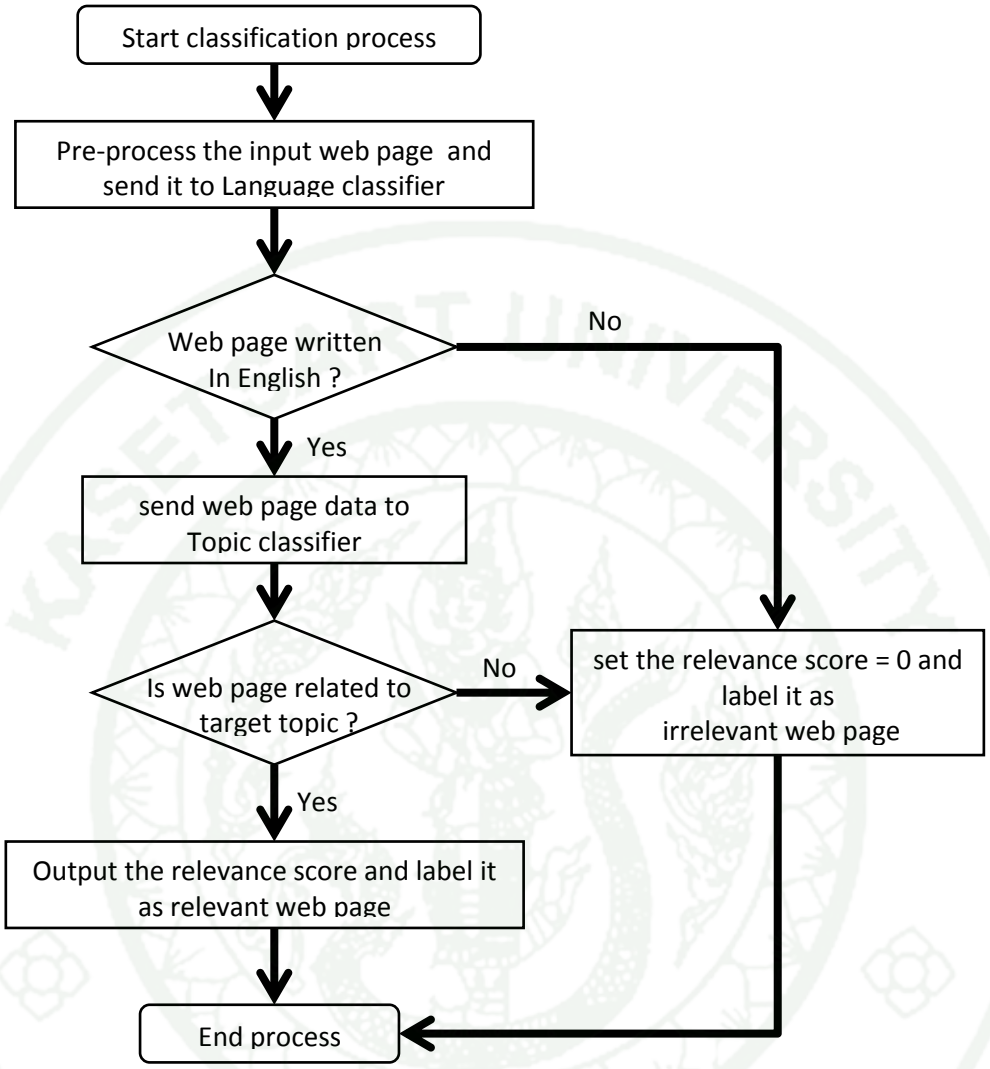

Figure 2 The web page classification process for Thai-tourism and Thai-estate topic

Table 2 Statistics of the training dataset for the topic classifier in Thai-tourism and Thai-estate topic

	Class	Number of samples
Thai-tourism	Positive examples	6,112
	Negative examples	80.648
Thai-estate	Positive examples	11,528
	Negative examples	39,301

Table 2 concludes the number of positive and negative samples for Thai-tourism and Thai-estate topic. It can be seen that these dataset is imbalance. Thus, we build the classifier for each topic by using 10-fold cross validation with the under-sampling technique. After repeating the experiments 10 times, the optimal classifier model for Thai-tourism and Thai-estate give us 95.78%, and 94.81% of Geometric mean value, respectively. These results show that our relevant web page classification process is efficient enough to classify the relevant web pages.

To classify the English web page in Thai-diving topic, we designed the process as shown in figure 3. Since Thai-diving can be considered as a subset of Thai-tourism topic, we thus use the Thai-tourism topic classifier to check whether the input web page is related to Thai-tourism first. If the classifier classify that the input web page is related to Thai-tourism, and it contains some diving related words, such as, "scuba", "snorkeling", "diving", and etc., we will consider that web page is our relevant web page.

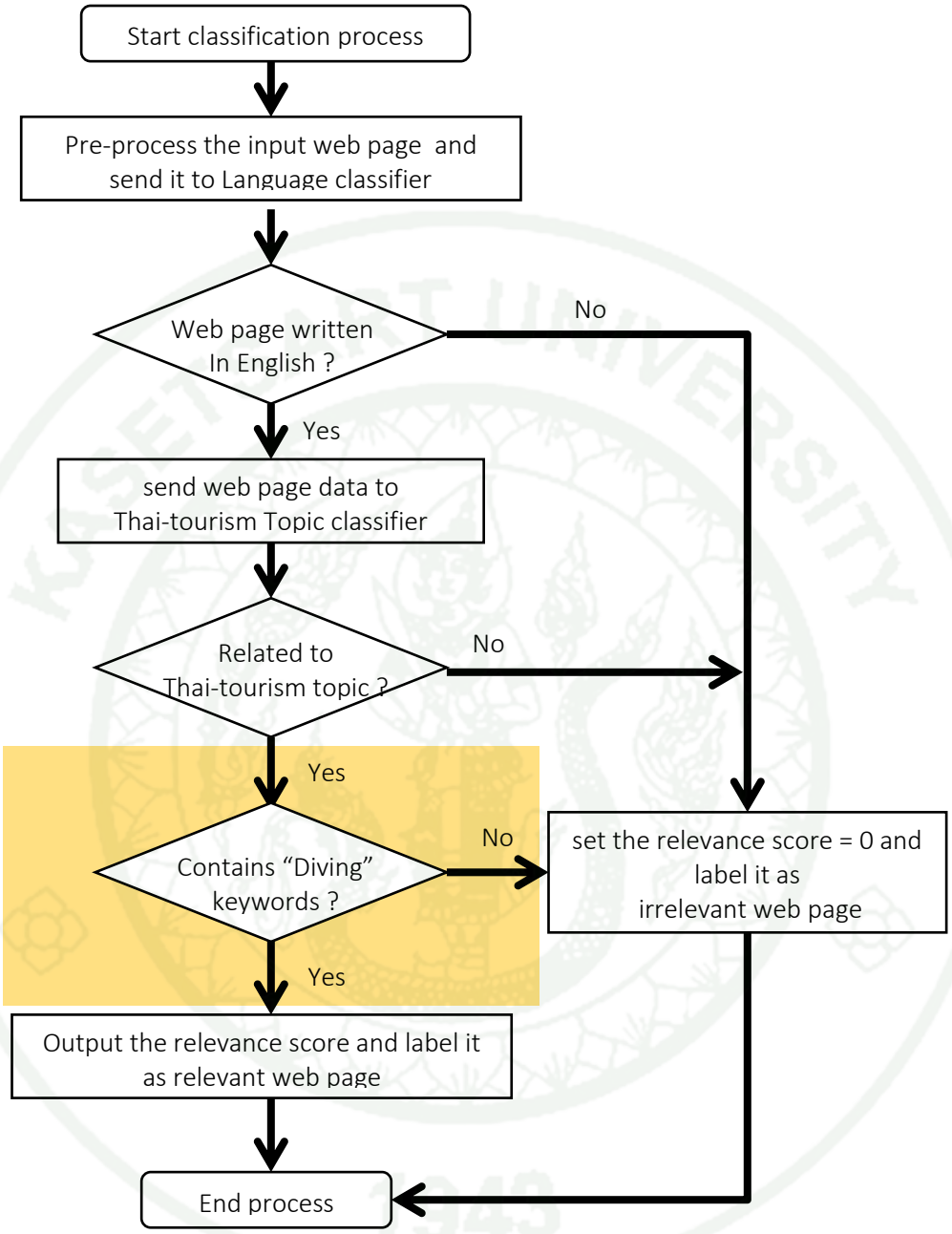

Figure 3 The web page classification process for Thai-diving topic

2. Finding Thai-related keyword in an English text

In this research, we use the Thai-related dictionary as reference to find out how many Thailand-related keywords in an input text. The result is used to extract some of the feature, which is used for predicting the group of unvisited web pages in the crawling frontier later. In this section, we will provide more detail on how we build the Thailand-related dictionary first. Then, we will present how we find the Thailand-related keywords from the English text later.

Figure 4 Thailand-related lists category in English Wikipedia

To build the dictionary of Thai related words, in this thesis, we choose to consider terms extracted from the well hand-craft knowledge source, such as the Wikipedia. As shown in figure 4, there is a category named "Thailand related lists" in Wikipedia. In this category, it contains many links to subcategories. Each

subcategory contains many links to the hub articles, named "List of" articles", which cite to many Thailand related articles later. Therefore, we build the Thai-related dictionary by first collecting the following individual "List of" articles and "List of" articles under the following subcategories from English Wikipedia

- The name of cities, provinces and all districts in Thailand (http://en.wikipedia.org/wiki/Category:Thailand geography-related lists)
- The name of buildings and structures in Thailand (http://en.wikipedia.org/wiki/Category:Lists of buildings and structures in Thailand)
- The name of organizations and companies in Thailand (http://en.wikipedia.org/wiki/Category:Lists of organizations based in Thailand)
- The name of schools, universities and museums in Thailand (http://en.wikipedia.org/wiki/Category:Thailand education-related lists)
- The name of Thai foods (http://en.wikipedia.org/wiki/List of Thai dishes)
- The name of Buddhist temples in Thailand (http://en.wikipedia.org/wiki/List of Buddhist temples in Thailand)

After finished collecting, the anchor texts from those "List of" articles are extracted, and examined whether they are relevant or related to Thailand by three annotators. Terms with majority voted as relevant will be added into our Thai related words list. At this stage, we also let the annotators add the shortest form of some keywords into the list manually. For example, “Samui” can be referred as “Koh Samui” which is the one of famous touristic islands in Thailand. Finally, our Thai related dictionary contains around 8,000 keywords.

To find the Thailand-related words in an input text, naively, we can use this dictionary to check each term we found in an input text. If we found the word in dictionary, that word is the Thailand-related words. However, there is some problem with this method since there are various ways to write native Thai words in English. For example, in our dictionary, “Nakhon Pratom”, the name of a province in Thailand, can also be written in many forms, such as “Nakon patom” or “NakornPrathom”. Thus, in this thesis, we solve this problem by utilizing the soundex technique. We use the soundex algorithm implemented in Apache Common

Codec (2013) to construct the additional soundexed dictionary of all Thailand-related words. When an input string feed into our process, we first tokenize it with white space and some special character (e.g., !, #, :, =, etc.) delimiters. Then we try to combine consecutive tokens to form the longest term found in the Thai related dictionary, or its soundex found in the soundexed one. All longest terms will then be used to compute the relevant keyword feature later. The figure 5 illustrates the mentioned text-tokenizer algorithm.

```
1: procedure TEXTTOKENIZER(inputString)
2:     tokens ← SimpleTokenizer(inputString)
3:     ThaiWords ← ∅
4:     NormalWords ← ∅
5:     usedTokens ← ∅
6:     for i ← 0; i < size(tokens); i++ do
7:         if usedTokens contains i
8:             continue
9:         end if
10:        longestWord = null
11:        word ← ""
12:        for j ← i; j < size(tokens); j++ do
13:            word ← word + " " + tokens[j]
14:            soundWord = Soundex(word)
15:            if word is in Thai related dictionary
16:                longestWord ← word
17:                usedTokens ← usedTokens ∪ {i, . . . , j}
18:            else if soundWord is in soundex dictionary
19:                longestWord ← word
20:                usedTokens ← usedTokens ∪ {i, . . . , j}
21:            end if
22:        end for
23:        if longestWord ≠ null
24:            ThaiWords ← ThaiWords ∪ longestWord
25:        else
26:            NormalWords ← NormalWords ∪ tokens[i]
27:            usedTokens ← usedTokens ∪ {i}
28:        end if
29:    end for
30:    returns ThaiWords and NormalWords
31: end procedure
```

Figure 5 Thai-related dictionary based text tokenizer algorithm

3. Definitions

3.1. Website segment

Traditional focused crawlers were mostly designed to find the relevant web pages, while a recent one has rather been designed to find the relevant websites (Tadapak *et al.*, 2010). When a website is judged relevant, the whole website, or most part of it, will be downloaded. However, from our observation on the target English web pages whose contents are related to one of our interest topics, i.e., the Thai tourism, we found that many foreign target websites serve relevant web pages only in some sections. For example, http://www.lonelyplanet.com hosts Thai tourism related web pages only in the http://www.lonelyplanet.com/thailand section. If we directly follow Tadapak et al. approach to download the whole website, we will finally end up wasting large amount of network resource by downloading many irrelevant web pages from other non-related sections. To solve this problem, in this thesis, we propose to download only portion of a website that hosts relevant target web pages. We define the notion of a website segment as those web pages which share the same longest logical directory path to be our largest crawling unit. Thus, if we find a probable website segment which could host relevant web pages, we will allow the crawler to download the web pages only within that website segment.

Figure 6 illustrates a sample set of URLs in a crawling frontier which we can later group them based on their logical directory paths into following four segments.

- http://www.kpnews.org/
- http://www.kpnews.org/high-alert/
- http://www.kpnews.org/genearl/
- http://www.kpnews.org/travel/

Thus, if the http://www.kpnews.org/travel/ is determined as a relevant website segment, all three web pages in that segment will consecutively be downloaded.

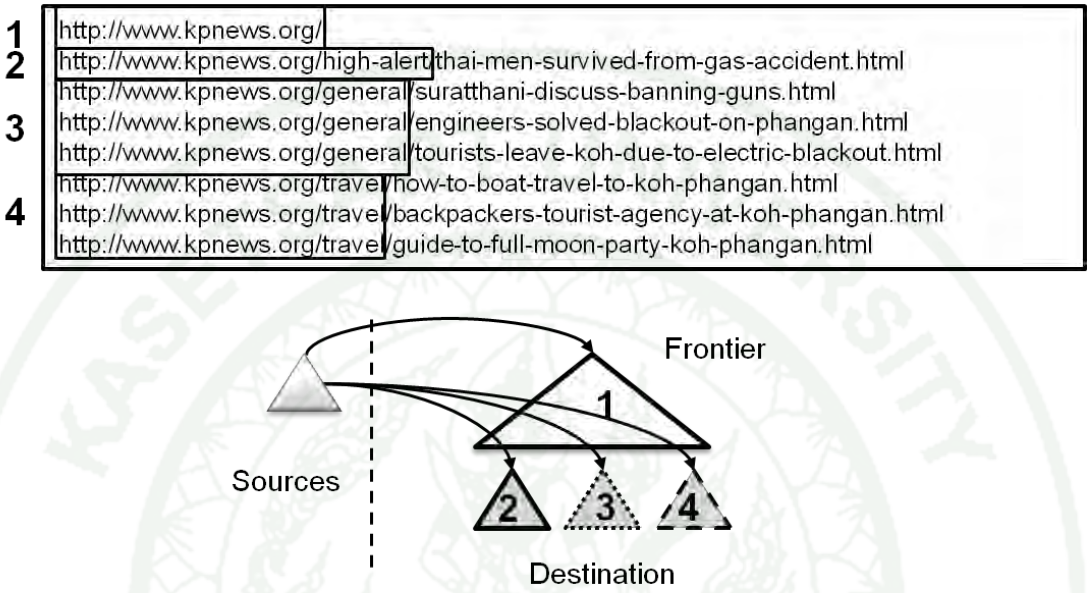

Figure 6 Sample of website segments from http://www.kpnews.org

3.2. Relevance degree

The relevance degree is defined as the ratio between relevant web pages and total web pages found within a website segment. For example, suppose that there are 7 web pages from a website segment are classified, by the web pages classifier, as relevant. Thus, if this website segment contains 10 web pages, the relevance degree of this website segment is 70%.

3.3. Relevant website segment

The relevant segment is the website segment that we would direct our crawler to visit. In this thesis, we define the relevant segment as a website segment which hosts relevant web pages more than a predefined relevance degree threshold. For example, if we set the relevance degree threshold to 50%, a sample segment which contains 7 relevant and 3 irrelevant web pages can be considered as the

relevant website segment. The suitable relevance degree threshold for each topic is experimentally determined by several crawling attempts later in the experimental section.

4. Thai-related foreign language specific web crawling approach

In this thesis, we aim to collect web pages within a relevant website segment. Thus, as shown in Figure , we design our crawler to be composed of three main components, i.e., Segment Crawler, Segment Identifier and Segment Predictor. The Segment Crawler is responsible for collecting web pages within an assign segment. After finished downloading, the Segment Identifier will identify all destination website segments from the already downloaded source website segments. Several features of each destination website segment will be extracted by the Segment Predictor. After extracting, the Segment Predictor will then use the machine learning classifiers to predict the relevancy of the destination website segment. All probable relevant website segments will be inserted into the crawling frontier, and will be prioritized by the probability scores obtained from the Segment Predictor. The most relevant website segments are then first downloaded by the Segment Crawler later. The further details of these three components are given as follows.

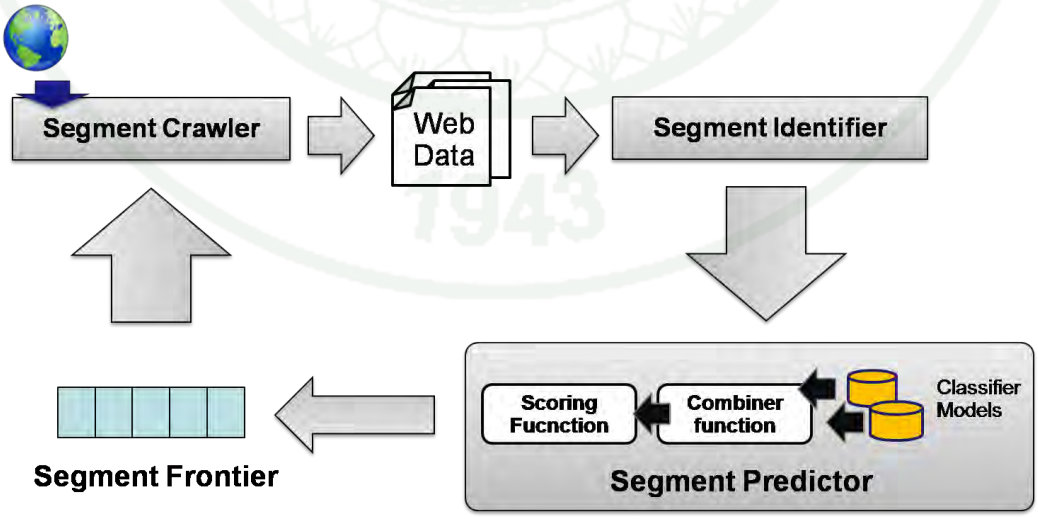

Figure 7 The architecture of the Thai-related foreign-language web crawler

4.1. Segment Identifier

As detailed in previous definition section, the Segment Identifier is used to extracts all destination URLs from the downloaded source website segments. After extracting, all those URLs will be grouped into website segment later.

4.2. Segment Predictor

At present, we observe the Thai related website segments' characteristics, and extract five features from the source segment in which there is a link pointing to the destination segment under consideration, as the following.

4.2.1.1. Source Relevance degree feature

Following the definition of relevance degree, and the topical locality idea of Davidson *et al.* (2006), we hypothesize that a relevant website segment should be recommended by another highly relevant degree source website segment, i.e., relevant website segment. Thus, we use the relevant degree of the source website segment as the feature. For example, if there are 5 relevant and 5 irrelevant web pages in the source website segment, thus the relevant degree feature can be written as 0|5.

4.2.1.2. Relevant keyword feature

For this feature, we hypothesize that if there are some Thai related words in (1) anchor texts and their surrounding words which cite to some web pages in the destination segment or (2) URLs of web pages within the destination segment, that destination website segment should be related to Thailand. Therefore, the relevant keyword feature vector can be constructed as follows.

- The ratio of Thai related words found in anchor texts and their surrounding words within 100 characters (excluding stopwords) which cite to the destination segment.

- The ratio of Thai related words found in URLs of web pages within the destination segment (excluding stopwords, special characters and word tokens from its hostname).

For instance, if “Pattaya”, and “Thailand" are those words included in a Thai related word list, and “is”, “a”, “in”, “http”, and “html” are included in the stopwords list, the relevant keyword feature vector extracted from the sample anchor text and its surrounding words, as well as the destination segment URL, shown in figure 8, can be simply written as 0.6|1.0.

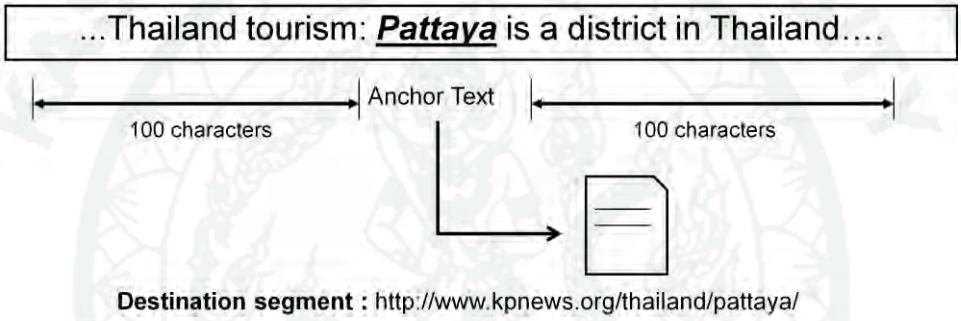

Figure 8 Example of an anchor text feature extraction

4.2.1.3. Geo-location feature

From observation, we found that many relevant destination segments are normally cited by the source segments whose web servers locate in some specific locations. We then hypothesize that (1) hosts, located in some country locations, whose the source segments are often found to link to the relevant destination segments, have a high probability to host other relevant segments, and (2) the destination segment whose host is located in the country which is often found relevant segments should be the relevant segment too. Thus, we use the InetAddressLocator (2007) to find the geo-location country code of source and the destination website segments and use them to construct the geo-location feature vector. For instance, if the source and destination segment’s hosts are located in

Thailand and United State, respectively, the geo-location feature vector can then be written as TH|US.

4.2.1.4. Domain feature

Following the same idea of the above geo-location feature, we can construct the domain name feature vector by extracting the domain name of the source and destination website segments. For example, if source and destination segments are located in ".com" and ".th" domains, respectively, the domain name feature vector can then be represented as COM|TH.

4.2.1.5. Word occurrence feature

Following the same idea of the relevant keyword feature, we include the occurrence frequency of words, excluding the stopwords, found in the anchor texts, and their surrounding words, as well as those from the destination segment URL, to form the word occurrence feature.

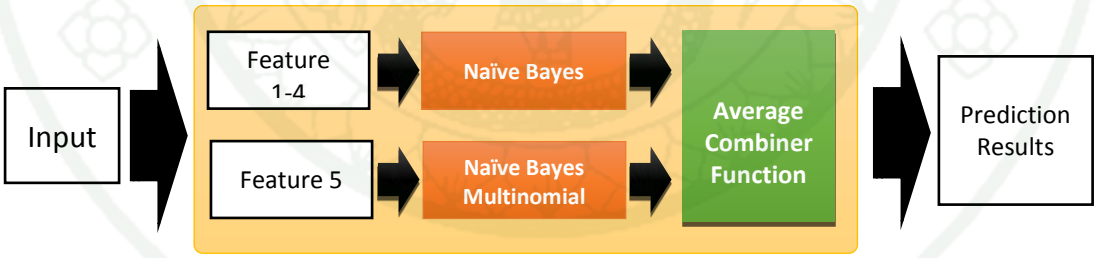

Figure 9 Structure of the classifier ensemble in Segment Predictor

To improve the classification performance, we choose to apply the ensemble technique to our Segment Predictor. Thus, we design our segment predictor as illustrated in figure 9. From the features that we proposed here, we can categorize them into two sets of features: link-based features (i.e., features 4.2.1.1-4.2.1.4) and word occurrence feature (feature 4.2.1.5). For the first set, we first discretize the real-values of both relevant keyword and relevance degree features into 10 discrete values. We then train all link-based features with the Simple Naive Bayes classifier, called

later the "link-based classifier". For the second one, we train it with the Naive Bayes Multinomial classifier, called later the "word-occurrence based classifier".

Suppose that during a crawling process we obtain $\boldsymbol{n}$ downloaded (i.e., source) website segments $\boldsymbol{x_1, x_2, \cdots, x_n}$ with each segment $\boldsymbol{x_i}$ containing $\boldsymbol{n_i}$ web pages $\boldsymbol{p_{i1}, p_{i2}, \cdots, p_{in_i}}$. If an individual segment $\boldsymbol{x_i}$ has a link to the same destination website segment $\boldsymbol{y}$, then the process of prioritizing $\boldsymbol{y}$ for the next crawling is considered based on an average prediction score calculated by using equation (8).

$$\boldsymbol{score(y) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{n_i}\sum_{j=1}^{n_i} P_w[rel|p_{ij}] \times P_E[rel|y_i]\right)} \quad (8)$$

$\boldsymbol{P_w[rel|p_{ij}]}$ is the relevance score obtained from the web page classifier, for the web page $\boldsymbol{p_{ij}}$ belonging to the source website segment $\boldsymbol{x_i}$. $\boldsymbol{P_E[rel|y_i]}$ is the ensemble probability of the segment $\boldsymbol{y}$ obtained from considering the link that is pointed by the source segment $\boldsymbol{x_i}$. This value is calculated from equation (9) which is an average of the relevant probabilities obtained from link-based and word-occurrence based classifier, respectively.

$$\boldsymbol{P_E[rel|y_i] = \frac{1}{2}(P_L[rel|y_i] + P_W[rel|y_i])} \quad (9)$$

4.3. Segment Crawler

The Segment Crawler is responsible for downloading web pages within a specified website segment. To avoid downloading too many irrelevant pages from a low relevant segment, we implement two heuristic rules into our Segment Crawler as follows.

4.3.1. Segment threshold

We define the segment threshold *S* to limit the crawler to download the irrelevant web pages in a website segment. For example, suppose that we assign *S = 2*, if the crawler has downloaded two irrelevant web pages consecutively, that website segment will be discarded, all un-downloaded web pages will be ignored.

4.3.2. Distance threshold

We define the distance threshold D to limit the crawler to download irrelevant website segments. For example, as illustrate in figure 10, if we assign D = 2 and the current destination segment has been recommended by two consecutive irrelevant source segments, then that segment will be discard from the crawling frontier.

Figure 10 Segment Crawler setting D=2

5. Experimental setup

In this section, we will provide you the details on how we construct the training dataset, and use them to build the Segment Predictor model for each topic.

5.1. Building the training dataset for Segment Predictor

To construct the training dataset for each target topic, we manually select URLs whose contents are related to our target topic from the Open Directory Project as the initial seeds. We then use the Google to find their back-links, and then launch the BFS crawler to collect web pages within 2 hops from those back links. All downloaded web pages in each hop are grouped into website, and the web graph of all

website segments is created. Finally, the relevance degree of each website segment is also calculated. Figure 11 provides an illustration of our training dataset preparation.

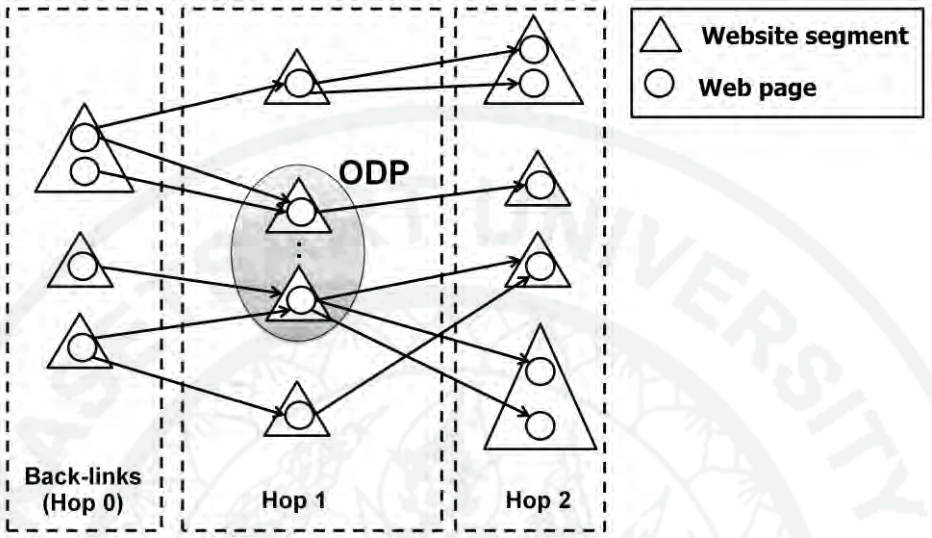

Figure 11 Dataset preparation for the Segment Predictor

5.2. Finding the optimal degree threshold

To train the Segment Predictor for each topic, we have to divide the collected dataset into two sets. The first one composes of feature vectors that have been extracted from relevant destination website segments. On the other hand, the second one composes of the feature vectors that have been extracted from the irrelevant website segments. As mentioned in definition section, in this thesis, the relevance degree is used to classify whether a website segment is relevant or not. Therefore, we have to find the suitable relevance degree which gives us the optimal crawling results first.

To find the optimal relevance degree threshold, we first set the relevance degree threshold as 0.25. Therefore, the website segment which has relevance degree greater than 0.25 are labeled as relevant website segment. We use all back-link website segments (Hop 0 in figure 11) in our dataset as the starting seeds. By using these seeds, we then simulate the crawling processes by letting the segment crawler to

follow the path that lead to the relevant website segment only. After finished crawling, we vary the threshold to 0.5, 0.75 and 1.0 and start the crawling process again. At the end of each simulated crawling attempt, we record the crawling performance in term of harvest rate and coverage rate. Since both harvest rate and coverage rate can be considered as precision and recall values, we can then calculate the F-measure, which is the average performance, for each simulated crawling attempt as shown in equation (10).

$$F - Measure = \frac{2 \times Harvest Rate \times Coverage\ rate}{Harvest\ rate + Coverage\ rate} \quad (10)$$

Table 3 concludes results from this simulation study on Thai-tourism, Thai-estate and Thai-diving topics. We can see that the relevant degree threshold setting to 0.5 provides an optimal performance for all target topics. Therefore, we then choose this relevance threshold value for classifying the relevance website segment.

Table 3 Harvest rate, Coverage rate and F-Measure obtained from varying relevance degree threshold

Dataset	Relevance degree Threshold	Harvest rate	Coverage rate	F-Measure
Thai-tourism	0.25	94.8%	91.6%	93.2%
	0.5	96.6%	90.6%	**93.5%**
	0.75	98.4%	86.4%	92.0%
	1	99.5%	76.0%	86.2%
Thai-estate	0.25	91.1%	85.5%	88.2%
	0.5	91.7%	85.1%	**88.3%**
	0.75	98.0%	75.4%	85.2%
	1	99.1%	70.4%	82.3%
Thai-diving	0.25	93.8%	93.7%	93.8%
	0.5	97.8%	91.2%	**94.4%**
	0.75	98.5%	89.8%	93.9%
	1	99.5%	66.0%	79.4%

5.3. Building the Segment Predictor models

Observation from the web graph built from all training website segments reveals that segments extracted from the same web site are mostly found to link to other segments within the same website. Therefore, the feature vectors extracted from those segments may be similar and redundant to each other. To avoid the over-fitting problem, we remove all those redundant feature vectors from the training dataset only if the source and destination website segment are located in the same website. Table 4 concludes the number of the remaining relevant and irrelevant samples in each training dataset.

It can be clearly seen that this dataset is imbalance. Therefore, we build the Segment Predictor model using 10-fold cross validation with undersampling technique. After repeating the experiments 10 times, we finally obtain an optimal

Segment Predictor model for Thai-tourism, Thai-estate, and Thai-diving topic with 87.3%, 88.6%, and 85.6% G-mean, respectively.

Table 4 Training dataset for the Segment Predictor

Dataset	Class	Number of samples
Thai-tourism	Relevant website segments	9,550
	Irrelevant website segment	33,669
Thai-estate	Relevant website segments	1,900
	Irrelevant website segment	10,710
Thai-diving	Relevant website segments	1,120
	Irrelevant website segment	4,467

RESULTS AND DISCUSSION

Results

In this thesis, we evaluate the crawling performance of the following web crawler on the real internet environment.

1. Our proposed crawling approach, i.e., Thai related Foreign Language-Specific Web Crawler (TFLSWC).
2. Best website segment crawler (Best-segment) - It is our proposed crawler which uses only the average relevance score to prioritize the Segment Frontier.
3. Best-First crawler (Best-page) - It is a simple focused crawler which first follows the destination URL whose parent web page is the most related to the target language and topic (Menczer *et al*., 2004) . In other words, we use the relevancy score of the download source web page, which is returned from web page classification process, to prioritize the destination URLs which are cited by the source.
4. Breadth-First search crawler (Breadth-First).

To build the seed URLs for each target topic, we first use the topic name, e.g., Thailand tourism, as the query and send it to Google. From the top two search result pages, we manually select the unseen relevant URLs to be our seeds. We finally use these seeds to launch all web crawlers at the same period and observe their harvest rate within 10,000 web pages. All crawlers are limit to collect only unseen web pages at most 300 pages per website. The segment threshold and the distance threshold parameters for our proposed crawler and Best website segment crawler are set to S = 2 and D=2, respectively.

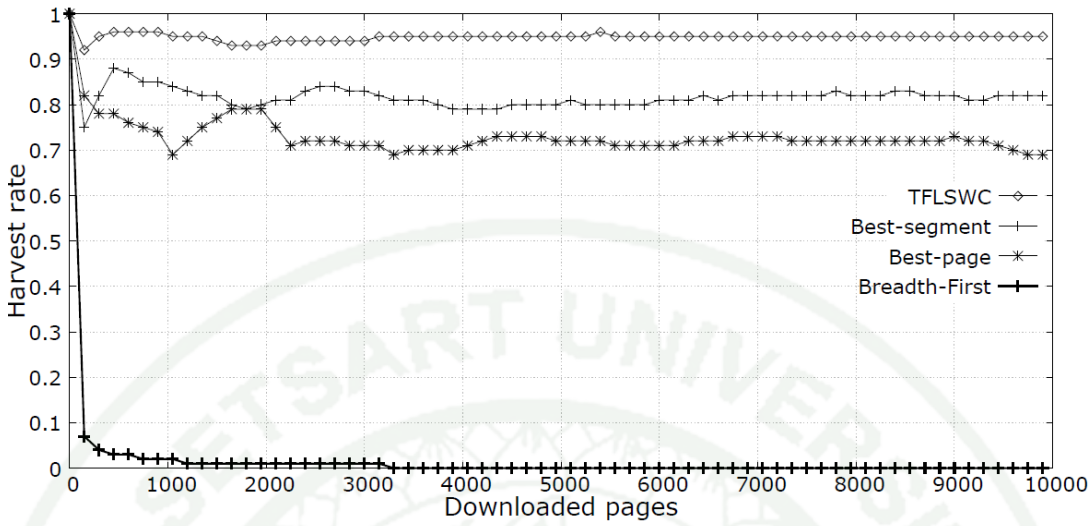

Figure 12 Harvest rate results on Thai-tourism topic

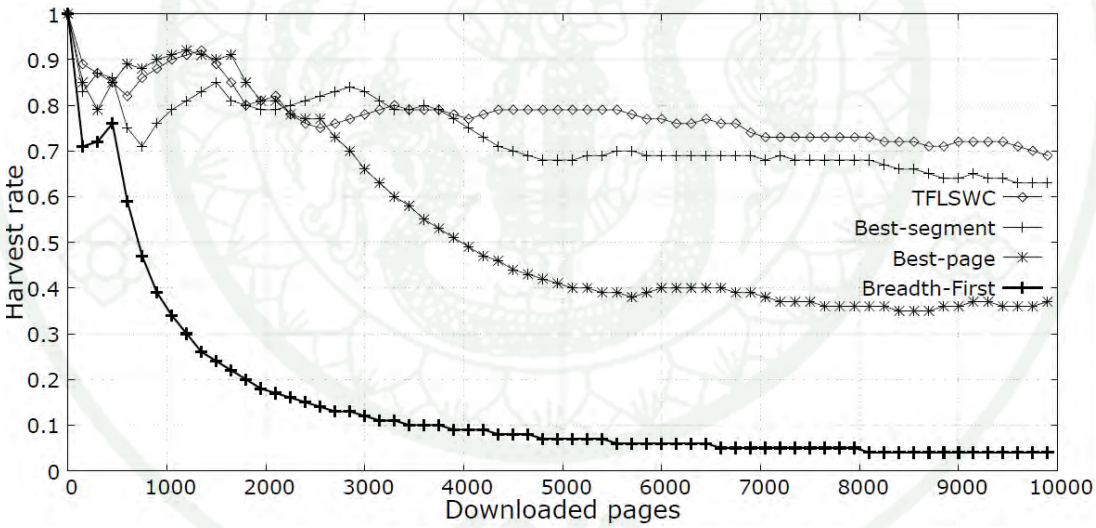

Figure 13 Harvest rate results on Thai-estate topic

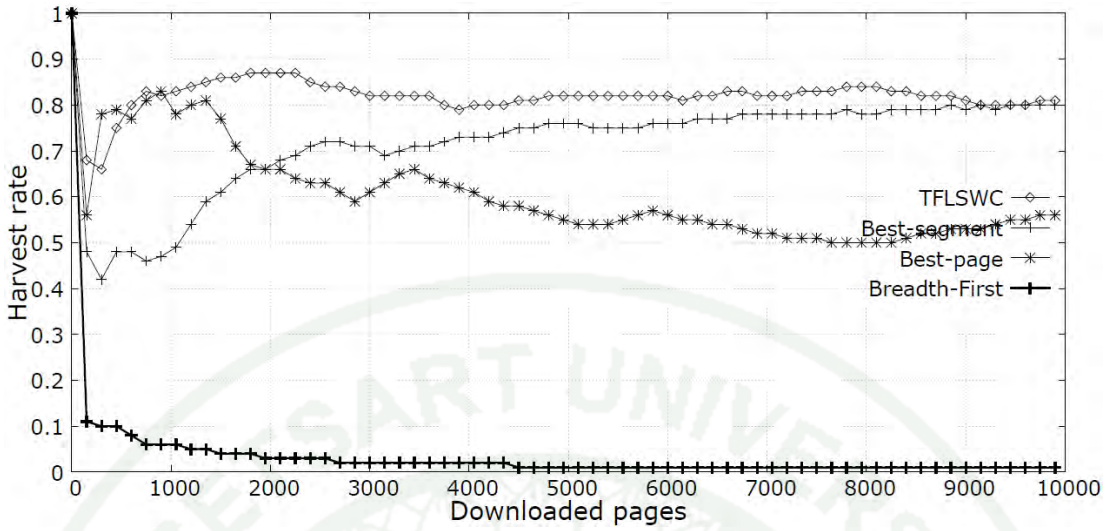

Figure 14 Harvest rate results on Thai-diving topic

Figure 12-14 graphically conclude the harvest rates of all above crawling strategies on Thai-tourism, Thai-estate and Thai-diving topics, respectively. It can be clearly seen that our proposed crawler archive better harvest rate than other baselines.

To understand why our crawler provide the better harvest rate results, we observed all the website segment links, which TFLSWC, Best-segment, and Breadth-First crawler found during the crawling. Table 5 concludes the ratio of the links to destination relevant website segments which are cited from the source relevant and irrelevant ones. The results show that the relevant website segment is likely to recommend to another relevant website segment with the probability around 60% at least. On the contrary, if the source is irrelevant, the destination should be irrelevant ones too.

Table 5 Ratio of links to destination relevant website segment whose source are relevant and irrelevant website segment

			Ratio of the link which destination is relevant website segment
Source website segment	Thai-tourism	Relevant website segment	82.35%
		Irrelevant Website segment	0.46%
	Thai-estate	Relevant website segment	59.39%
		Irrelevant Website segment	22.01%
	Thai-diving	Relevant website segment	62.47%
		Irrelevant Website segment	6.73%

Table 6 shows the ratio of the segment links to destination relevant website segment with and without Thai-related keywords found in anchor text and its surrounding words. Table 7 shows the ratio of the website segment links to destination relevant website segment with and without Thai-related keywords found in URLs of web pages within the destination segment. From these results, it can be seen that if we found any Thailand-related keywords in anchor text and surrounding words, or in the URLs path compositions, that destination website segment should be relevant.

Table 6 Ratio of links to destination relevant website segment with and without Thai-related keywords found in anchor text and surrouding words

			Ratio of the link which destination is relevant website segment
Source website segment	Thai-tourism	Found Thai-related keywords	83.73%
		Not found Thai-related keywords	37.55%
	Thai-estate	Found Thai-related keywords	63.42%
		Not found Thai-related keywords	48.14%
	Thai-diving	Found Thai-related keywords	69.10%
		Not found Thai-related keywords	39.29%

Table 7 Ratio of links to destination relevant website segment with and without Thai-related keywords found in URLs of some web pages in destination relevant website segment

			Ratio of the link which destination is relevant website segment
Source website segment	Thai-tourism	Found Thai-related keywords	91.98%
		Not found Thai-related keywords	36.81%
	Thai-estate	Found Thai-related keywords	80.41%
		Not found Thai-related keywords	45.81%
	Thai-diving	Found Thai-related keywords	81.45%
		Not found Thai-related keywords	40.70%

Discussion

According to the Harvest rate results, it can be seen that our crawler can find more relevant web pages than the others within the same downloaded time. In other words, our proposed approach can direct the crawler to the target topics and find more relevant web pages than other baselines. Furthermore, our analysis results also show that (1) if the source website segment is relevant, the destination should be relevant, and (2) if we found any Thai-related keywords in the website segment link, the destination should also be the relevant website segment too.

CONCLUSION AND RECOMMENDATION

Conclusion

In this thesis, we propose a novel crawling approach for collecting Thai related web pages which are written in English language. Instead of finding relevant web pages or relevant websites as proposed by other researchers, we rather explore the relevant website segments. The link-based and word occurrence features are extracted from the source segments which have at least one link to the destination segment under consideration. Those extracted features are used to train the Segment Predictor to predict the relevancy of the destination segments in the crawling frontier. According to the experimental results on the Thai-tourism, Thai-estate and Thai-diving topics, our proposed crawling approach give more promising efficiency in term of the harvest rate result than both the Breadth-First and Best-First baselines.

Recommendation

Although we obtain quite satisfying crawling performance with our proposed approach, there are still many interesting things to further explore. For example, we may also have to analyze the subset of Thailand tourism web pages in more detail to find the optimal S and D parameters for the Segment Crawler. Instead of combining all Thai related keywords from Wikipedia to build only one dictionary for all topics, we can build one specific dictionary for each Thai related topical crawler and examine its performance. Moreover, we also anticipate to exploring the other topics that are related to Thailand.

LITERATURE CITED

Alabbad, S.H. and S. Alanazi. 2009. Language Based Crawling: Crawling the Arabic Content of the Web, pp. 83-88. *In* **Proceeding of International Conference on Internet Computing 2009**. Las Vegas, USA.

Apache Foundation. 2013. **Apache Commons Codec**. Available Source: http://commons.apache.org/codec/, December 19, 2013.

Baeza-Yates, R., C. Castillio and V. Lopez. 2005. Characteristic of the Web of Spain. **Cybermetrics** 9 (1): 1-41.

_____ and F. Lalanne. 2004. Characteristics of the Korean Web. **Korea-Chile IT Cooperation Center ITCC**, Korea.

British Library. 2013. **UK web archive**. Available Source: http://www.webarchive.org.uk/ukwa/**,** December 19, 2013.

Cavnar, W.B. and J.M. Trenkle. 1994. N-Gram-Based Text Categorization, pp. 161-175. *In* **Proceedings of 3rd Annual Symposium on Document Analysis and Information Retrieval.** Last Vegas, USA.

Chakrabarti, S., M. Van den Berg and B. Dom. 1999. Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery. **Computer Networks: The International Journal of Computer and Telecommunications Networking** 31 (11): 1623-1640.

Davision, B.D. 2000. Topical locality in the web, pp.272-279. *In* **Proceeding of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval**. ACM, USA**.**

Garcia, S. and F. Herrera. 2009. Evolutionary undersampling for classification with imbalanced datasets: proposals and taxonomy. **Evolutionary Computation**. 17 (3): 275-306

Gomes, D., A. Nogueira, J. Miranda and M. Costa. 2008. Introducing the Portuguese Web Archive Initiative. *In* **Proceeding of 8th International Web Archiving Workshop**. Denmark.

He, H. and E.A. Garcia. 2009. Learning from Imbalanced Data. **Transactions on Knowledge and Data Engineering** 21 (9): 1263-1284.

InetAddressLocator. 2007. **JAVA IP(InetAddress) Locator**. Available Source: http://javainetlocator.sourceforge.net/, December 19, 2013.

Manning, C.D., P. Raghavan and H. Schutze. 2008. **Introduction to Information Retrieval**. Cambridge University Press, England.

Menczer, F., G. Pant and P. Srinivasan. 2004. Topical Web Crawlers: Evaluating Adaptive Algorithms. **ACM Transactions on Internet Technology** 4 (4): 278-419.

Nakatani, S. (2011). **LangDetect**. Available Source: http://code.google.com/p/language-detection/, December 19, 2013**.**

National Diet Library. 2011. **Web Archiving Project**. Available Source: http://warp.ndl.go.jp, December 19, 2013.

National Library Board Singapore. 2013. **Web Archiving Singapore**. Available Source: http://was.nl.sg/, December 19, 2013.

National Library of Australia. 2013. **PANDORA Australia's Web Archive**. Available Source: http://pandora.nla.gov.au/, December 19, 2013.

Netscape Communication Corporation. 2013. **Open Directory Project**. Available Source: http://www.dmoz.org/, December 19, 2013.

Ranawana, R. and V. Palade. 2006. Multi-Classifier System - Review and a Roadmap for Developers. **Hybrid Intelligent Systems**. 3 (1): 35-61

Somboonviwat, K., T. Tamura and M. Kitsuregawa. 2006. Finding Thai Web Pages in Foreign Web Spaces, pp.. *In* **Proceeding of 22nd International Conference on Data Engineering Workshops**. IEEE, USA.

Srisukha, E., S. Jinarat, C. Haruechaiyasak and A. Rungsawang. 2008. Naïve Bayes Based Language-Specific Web Crawling, pp. 113-116. *In* **Proceedings of the 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology.** IEEE, USA.

Tadapak, P., T. Suebchua and A. Rungsawang. 2010. A Machine Learning Based Language Specific Web Site Crawler, pp.155-161. *In* **13th International Conference on Network-Based Information Systems**. IEEE, USA.

Tamura, T., K. Somboonviwat and M. Kitsuregawa. 2007. A Method for Language-Specific Web Crawling and Its Evaluation. **Systems and Computers in Japan** 38 (2): 10-20.

CURRICULUM VITAE

NAME : Mr. Tanaphol Suebchua

BIRTH DATE : December 19, 1988

BIRTH PLACE : Bangkok, Thailand

EDUCATION :

YEAR	**INSTITUTE**	**DEGREE/DIPLOMA**
2010	Kasetsart University	B.Eng.(Computer)

POSITION/TITLE : -

WORK PLACE : -

SCHOLARSHIP/AWARDS : JSTP Scholarship, NSTDA