Enhance Web Page Classification by Using a Topic Model and Integrating Neighboring Pages

Wongkot Sriurai*, Phayung Meesad2 and Choochart Haruechaiyasak3

Abstract
This paper applies a topic model to represent the feature space for learning the web page classification model. Latent Dirichlet Allocation (LDA) algorithm is applied to generate a probabilistic topic model consisting of term features clustered into a set of latent topics. Words assigned into the same topic are semantically related. In addition, we propose a method to integrate the additional term features obtained from the neighboring pages (i.e., parent, child and sibling pages) to further improve the performance of the classification model. In the experiments, we evaluated among three different feature representations: (1) applying the simple bag of words (BOW) model, (2) applying the topic model on current page, and (3) integrating the neighboring pages via the topic model. From the experimental

Received 21 December 2009; Accepted 15 March 2010

1 Student, Department of Information Technology, Faculty of Information Technology, King Mongkut's University of Technology North Bangkok.

2 Assistant Professor, Department of Teacher Training in Electrical Engineering, Faculty of Technical Education, King Mongkut’s University of Technology North Bangkok.

3 Researcher, Human Language Technology Laboratory (HLT), National Electronics and Computer Technology Center (NECTEC).

* Corresponding Author, Tel. 08-9717-6189, E-mail: swongkot@gmail.com
results, the approach of integrating current page with the neighboring pages via the topic model yielded the best performance with the F1 measure of 85.01%; an improvement of 23.81% over the BOW model.

**Keywords:** Web Page Classification, Bag of Words, Topic Model

1. **Introduction**

Today, the amount of web documents (e.g., web pages, blogs, emails) is increasing with an explosive rate. Text categorization is a widely applied solution for managing and organizing those documents and also assists the information retrieval process in filtering the documents for a specific topic. Text categorization process usually adopts the supervised machine learning algorithms for learning the classification model [1], [2]. To prepare the term feature set, the bag of words (BOW) is usually applied to represent the feature space. Under the BOW model, each document is represented by a vector of weight values calculated from, for example, the term frequency-inverse document frequency (TF-IDF), of a term occurring in the document. The BOW is very simple to create, however, this approach discards the semantic information of the terms, (i.e., synonym). Therefore, different terms whose meanings are similar or the same would be represented as different features. As a result, the performance of a classification model learned by using the BOW model could become deteriorated.

In this paper, we apply a topic model to represent the feature space for learning the web page classification model. Under the topic model concept, words (or terms), which are statistically dependent, are clustered into the same topics. Given a set of documents $D$ consisting of a set of terms (or words) $W$, a topic model generates a set of latent topics $T$ based on a statistical inference on the term set $W$. In this paper, we applied the Latent Dirichlet Allocation (LDA) [3] algorithm to generate a probabilistic topic model from a web page collection. A topic model can capture the hypernyms, hyponyms and synonyms of a given word. For example, the words “vehicle” (hypernym) and “automobile” (hyponym) would be clustered into the same topic. In addition, the words “automobile” (synonym) and “car” (synonym) would also be clustered into the same topic. Therefore, the topic model improve the performance of a classification model by (1) reducing the number of features or dimensions and (2) mapping the semantically related terms into the same feature dimension.

In addition to the concept of topic model, our proposed method also takes an advantage of hyperlink structure of the web. Given a web page (i.e., current page), we use nearby pages with three kinds of neighboring pages: parent pages, child pages and sibling pages. Using the additional terms from the neighboring pages could increase more evidence for learning the classification model. However, the terms from current page should be weighted higher than terms from neighboring pages. Therefore, the proposed method for integrating neighboring information provides a function for varying the weight values of terms from the parent pages, the child pages and the ones from the sibling pages. Using the information gain [1] as the feature selection method and the Support Vector Machines (SVM) [4], [5] as the classification algorithm, the experimental results showed that by integrating the
additional neighboring information via a topic model, the classification performance under the F1 measure was significantly improved over the simple BOW model.

The rest of this paper is organized as follows. In the next section we provide a brief review of related works. Section 3 presents the proposed framework of feature representation via the topic model for learning the web page classification models. Section 4 presents the experiments with the discussion on the results. In Section 5, we conclude the paper and put forward the directions of our future works.

2. Background and Related Works

In this section, we review text categorization and feature selection and description of the topic model based on the Latent Dirichlet Allocation algorithm.

2.1 Text Categorization and Feature Selection

Text categorization (also known as document classification) is a supervised learning task, concerning the assigning of category labels to new documents based on the information learned from a labeled training data [1], [2]. Text categorization is a well-studied research area related to information retrieval, machine learning and text mining. A number of machine learning algorithms have been introduced and applied for the task of text classification including the Support Vector Machines (SVM).

Support Vector Machines algorithm (SVM) [4], [5] was developed from the statistical learning theory since the 60s. SVM has been applied for text classification in many previous works [6], [7]. SVM is based on the structural risk minimization principle from computational theory. The algorithm addresses the general problem of learning to discriminate between positive and negative members of a given class of n-dimensional vectors. The SVM has been shown to yield the best performance compared to other classification algorithm in many previous works. In this paper, we adopt the SVM in our experiments.

Most known text categorization algorithms represent a document collection as BOW. Using the BOW usually leads to an explosion in the number of features, so that even tens of thousands of features. The major problem of this representation is the high dimensions of feature space and information loss of the original texts. Feature selection (FS) [1] is one technique to deal with such problems. The main idea of feature selection is to select a subset of terms occurring in the training set and using only this subset as features in text categorization. There are several previous research works, which proposed and compared among many feature selection methods. For example, Dash and Liu [8] gave a survey of feature selection methods for classification. In a comparative study of feature selection methods in statistical learning of text categorization, Yang and Pedersen [1] evaluated document frequency (DF), information gain (IG), mutual information (MI), chi-square (CHI) and term strength (TS); and found IG and CHI to be the most effective. In this paper we use information gain (IG) method.

In the domain of the web, text categorization has been applied for classifying web pages. Recent works in web page classification proposed some methods to include the information from
neighboring web pages to learn the model. The information of neighboring pages are, for example, title and surrounding text of anchor text [6], [9]. Fürnkranz [10] proposed a classification method using anchor text, surrounding text of anchor text that precedes the hyperlink. Shen et al. [11] proposed an approach to compare implicit and explicit links for web page classification. The experimental results showed that the use of the implicit links is better than using explicit links in classification.

Qi and Davison [12] proposed a method to improve web page classification by utilizing the class information from neighboring pages in the link graph. The categories represented by four kinds of neighbors (parents, children, siblings and spouses) are combined to help with the page in question. Experiments showed that sibling pages are the most important type of neighbor to use. Qi and Davison [13] proposed a method by utilization a weighted combination of the contents of neighbors to generate a better virtual document for classification. Their experimental results showed that using a weighted of neighboring pages could improve the performance of web page classification. Chen and Choi [14] presented an automatic genre-based web page classification system, which can work either independently or in conjunction with other topic-based web page classification system.

In this paper, we also apply the neighboring information for improving the classification model. However, we adopt the topic model to represent the feature space. There have been many studies on discovering latent topics from text collections [3].

### 2.2 Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) has been introduced as a generative probabilistic model for a set of documents [3], [15]. The basic idea behind this approach is that documents are represented as random mixtures over latent topics. Each topic is represented by a probability distribution over the terms. Each article is represented by a probability distribution over the topics. LDA has also been applied for identification of topics in a number of different areas such as classification, collaborative filtering [3] and content-based filtering [16]. In the generation process of LDA, a word is generated from combination of topics $z$ in a document $d$ and sampling a word from topic-word distribution. The process is generated using Equation (1)

$$P(w_i) = \sum_{j=1}^{T} P(w_i | z_i = j) P(z_i = j)$$

where $P(z_i = j)$ represents the probability of topic $j$ was sampled for word token $w_i$ in document; $P(w_i | z_i = j)$ represents the probability of under topic $j$, and $T$ is the number of topics. To simplify notation, let $\phi^{(j)} = P(w_i | z = j)$ refer to multinomial distribution over words for topic $j$, and $\theta^d = P(z)$ refer to multinomial distribution over topics for document $d$. The estimated parameters $\phi$ and $\theta$ are the basis for latent-semantic representation of words and documents.

Generally, an LDA model can be represented as a probabilistic graphical model as shown in Figure 1 [3]. There are three levels to the LDA representation. The variables $\alpha$ and $\beta$ are the corpus-level parameters, which are assumed to be sampled during the process of generating a corpus.

The $\alpha$ and $\beta$ are hyperparameters for the document-topic and topic-word Dirichlet distributions, respectively. The variables $\theta$ is a document-level
variable, sampled once per document. Finally, the variables \( z \) and \( w \) are word-level variables and are sampled once for each word in each document. The variable \( N_d \) is the number of word tokens in a document \( d \) and variable \( D \) is the number of documents.

In the LDA model, there are many algorithms for estimating parameters, such as variation Bayes [3], expectation propagation [17] and Gibbs sampling [15], [18]. This paper, we choose Gibbs sampling algorithm. Gibbs sampling algorithm is easy to implement and requires little memory. The procedure for LDA model uses Gibbs sampling to estimate topics from the document collection as well as estimate the word-topic and topic-document probability distributions. We can compute the conditional probability for the Gibbs sampler by Equation (2).

\[
P(z_i = j | z_i \neq j, w_i, d_i, \alpha, \beta, \phi, \theta) \propto \frac{c_{wi}^{WT} + \beta}{\sum_{w=1}^{W} c_{wi}^{WT} + W \cdot \beta} \cdot \frac{c_{dj}^{DT} + \alpha}{\sum_{r=1}^{T} c_{dj}^{DT} + T \cdot \alpha}
\]

where \( z_i = j \) represents the topic assignment of word token \( w_i \) to the topic \( j \), \( z_i \neq j \) refers to the topic assignments of all other word tokens, and “.” refers to all other known or observed information. \( W \) is the number of word tokens, \( D \) is the number of documents, and \( T \) is the number of the topics. The \( C^{WT} \) and \( C^{DT} \) are matrices of counts with dimensions a word-topic matrix and a topic-document matrix, respectively. \( C_{wi}^{WT} \) is composed of the number of times word \( w_i \) is assigned to topic \( j \), not including the current word token \( w_i \) and \( C_{dj}^{DT} \) is composed of the number of times topic \( j \) is assigned to some word token in document \( d \), not including the current word token \( w_i \). The suitable values for \( \alpha \) and \( \beta \) are discussed by Steyvers and Griffiths [15], [18]. Next the sampling process, we can estimate parameters \( \phi \) and \( \theta \) with equations (3) and (4).

\[
\hat{\phi}_{i}^{(j)} = \frac{c_{wi}^{WT} + \beta}{\sum_{w=1}^{W} c_{wi}^{WT} + W \cdot \beta}
\]

\[
\hat{\theta}_{j}^{(d)} = \frac{C_{dj}^{DT} + \alpha}{\sum_{r=1}^{T} C_{dj}^{DT} + T \cdot \alpha}
\]

3. The Proposed Framework

Figure 2 illustrates the proposed framework of feature representations for learning the web page classification models. In our proposed framework, we evaluated among three different feature representations: (1) applying the simple BOW model on current page, (2) applying the topic model on current page, and (3) integrating the neighboring pages via the topic model. Each approach is described in details as follows.

**Approach 1 (BOW):** Given a web page collection, the process of text processing is applied to extract terms. The set of terms is then filtered by using the feature selection technique of...
information gain (IG). Once the term features are obtained, we apply the Support Vector Machines (SVM) to learn the classification model. The model is then used to evaluate the performance of category prediction.

**Approach 2 (TOPIC_CUR):** Given a web page collection, the process of text processing is applied to extract terms. The set of terms is then generated by using the topic model based on the LDA algorithm. The output from this step is the topic probability for each article. The Support Vector Machines (SVM) is also used to learn the classification model.

**Approach 3 (TOPIC_INTEGRATED):** The main difference of this approach from Approach 2 is the integration of some additional term features obtained from the neighboring pages to improve the performance of web page classification. The process of integrating the neighboring pages is explained as follows.

Figure 3 shows three types of neighboring pages, parent, child and sibling pages. Given a web page (i.e., current page), there are typically incoming links from parent pages, outgoing links to child pages and links from sibling pages. A parent child and sibling pages are collectively referred to as the neighboring pages. Using the additional terms from the neighboring pages could help increase more evidence for learning the classification model.

In this paper, we vary a weight value of neighboring pages from zero to one. A weight value equals to zero means the neighboring pages are not included for the feature representation. Under this
approach, terms from different page types (i.e., current, parent, child and sibling) are first transformed into a set of $n$ topics (denoted by $T_0, \ldots, T_{n-1}$) by using the LDA algorithm. The weight values from 0 to 1 are then multiplied to the topic dimension $T_i$ of parent, child and sibling pages. The combined topic feature vector by integrating the neighboring topic vectors with adjusted weight values can be computed by using the algorithm listed in Table 1.

The Integrating Neighboring Pages algorithm that we present in this paper incorporates term features obtained from the neighboring pages (i.e., parent, child and sibling pages) into the classification model. Using additional terms from the neighboring pages could help increase more evidence for learning the classification model. In this algorithm, we propose a function for varying the weight values of terms from parent pages (PDT), child pages (CDT) and sibling pages (SDT). The probability values from all neighboring pages are integrated with the current page (CurDT) to form a new integrated matrix (IDT).

Table 1 The Integrating Neighboring Pages Algorithm

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Integrating Neighboring Pages via Topic Model</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong></td>
<td>CurDT, PDT, CDT, SDT, $W_p$, $W_c$, $W_s$</td>
</tr>
<tr>
<td><strong>for all</strong></td>
<td>$d_i$ in CurDT do</td>
</tr>
<tr>
<td><strong>for all</strong></td>
<td>$t_j$ in CurDT do</td>
</tr>
<tr>
<td><strong>CurDT</strong></td>
<td>$\leftarrow$ getPValue(CurDT, $i$, $j$)</td>
</tr>
<tr>
<td><strong>PP</strong></td>
<td>$\leftarrow$ getPValue(PDT, $i$, $j$) * $W_p$</td>
</tr>
<tr>
<td><strong>PC</strong></td>
<td>$\leftarrow$ getPValue(CDT, $i$, $j$) * $W_c$</td>
</tr>
<tr>
<td><strong>PS</strong></td>
<td>$\leftarrow$ getPValue(SDT, $i$, $j$) * $W_s$</td>
</tr>
<tr>
<td><strong>setPValue</strong></td>
<td>IDT, CurDT + PP + PC + PS, $i$, $j$</td>
</tr>
<tr>
<td><strong>end for</strong></td>
<td></td>
</tr>
<tr>
<td><strong>return</strong></td>
<td>IDT</td>
</tr>
</tbody>
</table>

Parameters and variables:
- **CurDT**: document-topic matrix from current page
- **PDT**: document-topic matrix from parent pages
- **CDT**: document-topic matrix from child pages
- **SDT**: document-topic matrix from sibling pages
- **IDT**: integrated document-topic matrix
- **PP**: $P$-Value from PDT at specific index
- **PC**: $P$-Value from CDT at specific index
- **PS**: $P$-Value from SDT at specific index
- **$W_p$$, W_c$$, W_s$$**: weight value for parent pages, child pages and sibling pages, respectively, $0.0 \leq W_j \leq 1.0$
- **$P$-Value**: probability value
- **getPValue**($M$, $r$, $c$): function for getting $P$-Value from row $r$ and column $c$ of matrix $M$
- **setPValue**($M$, $p$, $r$, $c$): function for setting $P$-Value on row $r$, column $c$ of matrix $M$ with value $p$

The process of algorithm begins with the results from the LDA model; that is document-topic matrices from all page types. The algorithm starts by
getting data from document-topic matrices (CurDT, PDT, CDT, SDT) using getPValue function. All P-Values of the document-topic matrices are then multiplied by the weight values of each document-topic matrix except for the current page matrix. Finally all P-Values from four matrices are summed up and then sent to IDT using setPValue function. After the integrating process, we use the IDT matrix for learning the classification model.

4. Experiments and Discussion

4.1 Web Page Collection

In our experiments, we use a collection of articles obtained the Wikipedia Selection for Schools, which is available from the SOS Children’s Villages web site. There are 15 categories: art, business studies, citizenship, countries, design and technology, everyday life, geography, history, IT, language and literature, mathematics, music, people, religion and science. The total number of articles is 4,625.

<table>
<thead>
<tr>
<th>Category</th>
<th>No. of Articles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Art</td>
<td>74</td>
</tr>
<tr>
<td>Citizenship</td>
<td>224</td>
</tr>
<tr>
<td>Design and Technology</td>
<td>250</td>
</tr>
<tr>
<td>Geography</td>
<td>650</td>
</tr>
<tr>
<td>IT</td>
<td>64</td>
</tr>
<tr>
<td>Mathematics</td>
<td>45</td>
</tr>
<tr>
<td>People</td>
<td>680</td>
</tr>
<tr>
<td>Science</td>
<td>1068</td>
</tr>
<tr>
<td>Business Studies</td>
<td>88</td>
</tr>
<tr>
<td>Countries</td>
<td>220</td>
</tr>
<tr>
<td>Everyday life</td>
<td>380</td>
</tr>
<tr>
<td>History</td>
<td>400</td>
</tr>
<tr>
<td>Language and literature</td>
<td>196</td>
</tr>
<tr>
<td>Music</td>
<td>140</td>
</tr>
<tr>
<td>Religion</td>
<td>146</td>
</tr>
</tbody>
</table>

Table 2 lists the first-level subject categories available from the collection. Organizing articles into the subject category set provides users a convenient way to access the articles on the same subject. Each article contains many hypertext links to other articles which are related to the current article.

4.2 Experiments

We used the LDA algorithm provided by the linguistic analysis tool called LingPipe to run our experiments. LingPipe is a suite of Java tools designed to perform linguistic analysis on natural language data. In this experiment, we apply the LDA algorithm provided under the LingPipe API and set the number of topics equal to 200 and the number of epochs to 2,000.

For text classification process, we used WEKA, an open-source machine learning tool, to perform the experiments.

4.3 Evaluation Metrics

The standard performance metrics for evaluating the text classification used in the experiments are precision, recall and F1 measure [2]. We tested all algorithms by using the 10-fold cross validation.

Precision (P) is the percentage of the predicted documents for a given category that are classified correctly, defined as:

\[
\text{precision} = \frac{\text{categories found and correct}}{\text{total categories found}}
\]
Recall (R) is the percentage of the documents for a given category that are classified correctly, defined as:

\[
\text{recall} = \frac{\text{categories found and correct}}{\text{total categories correct}}
\]  \hspace{1cm} (6)

F1 measure is a single measure that tries to combine precision and recall. F1 measure ranges from 0 to 1 and the higher the better. F1 measure can be defined as follows:

\[
F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}
\]  \hspace{1cm} (7)

4.4 Experimental Results

We start by evaluating the weight values of neighboring pages under TOPIC_INTEGRATED (Approach 3). Table 3 shows the results of weight value adjustment on parent pages, child pages and sibling pages based on our algorithm. The best weight value of parent pages is equal to 0.4 with the F1 measure of 0.8463. For the child pages, the maximum value of F1 measure is 0.8463 with the weight value of 0.2 and the sibling pages, the maximum value of F1 measure is 0.8461 with the weight value of 0.3. The results showed that using information from parent pages is more effective than child pages and sibling pages, while weight value of sibling pages higher than child pages.

From the results in Table 3, we combine the weight value of neighboring pages on our algorithm. The results show the combination of neighboring pages in Table 4. The best combination of neighboring pages with the F1 measure of 0.8501 by weight of parent pages, child pages and sibling pages equal to 0.4, 0.0 and 0.3, respectively. The results showed that using information from parent pages and sibling pages are more effective than child pages for improving the performance of a classification model.

### Table 3

Weight value adjustment of parent pages, child pages and sibling pages under TOPIC_INTEGRATED (Approach 3)

| Neighbors | \( w_p \) | \( \text{P} \) | \( \text{R} \) | \( \text{F1} \) | Neighbors | \( w_c \) | \( \text{P} \) | \( \text{R} \) | \( \text{F1} \) | Neighbors | \( w_s \) | \( \text{P} \) | \( \text{R} \) | \( \text{F1} \) |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| Parent pages | 0.1 | 0.8587 | 0.8319 | 0.8440 | 0.1 | 0.8590 | 0.8343 | 0.8455 | 0.1 | 0.8577 | 0.8331 | 0.8444 |
| 0.2 | 0.8575 | 0.8321 | 0.8435 | 0.2 | 0.8577 | 0.8335 | 0.8463 | 0.2 | 0.8595 | 0.8338 | 0.8454 |
| 0.3 | 0.8575 | 0.8323 | 0.8439 | 0.3 | 0.8576 | 0.8332 | 0.8442 | 0.3 | 0.8603 | 0.8344 | 0.8461 |
| 0.4 | 0.8569 | 0.8313 | 0.8463 | 0.4 | 0.8548 | 0.8305 | 0.8416 | 0.4 | 0.8597 | 0.8345 | 0.8459 |
| 0.5 | 0.8566 | 0.8318 | 0.8428 | 0.5 | 0.8560 | 0.8333 | 0.8437 | 0.5 | 0.8589 | 0.8333 | 0.8449 |
| 0.6 | 0.8533 | 0.8279 | 0.8391 | 0.6 | 0.8576 | 0.8331 | 0.8444 | 0.6 | 0.8571 | 0.8302 | 0.8425 |
| 0.7 | 0.8541 | 0.8274 | 0.8391 | 0.7 | 0.8549 | 0.8285 | 0.8404 | 0.7 | 0.8577 | 0.8314 | 0.8433 |
| 0.8 | 0.8543 | 0.8251 | 0.8378 | 0.8 | 0.8539 | 0.8286 | 0.8400 | 0.8 | 0.8571 | 0.8311 | 0.8428 |
| 0.9 | 0.8527 | 0.8225 | 0.8358 | 0.9 | 0.8548 | 0.8293 | 0.8407 | 0.9 | 0.8578 | 0.8294 | 0.8421 |
| 1.0 | 0.8524 | 0.8225 | 0.8355 | 1.0 | 0.8547 | 0.8289 | 0.8403 | 1.0 | 0.8577 | 0.8287 | 0.8417 |

### Table 4

Combination of neighboring pages with highest F1 measure

<table>
<thead>
<tr>
<th>Approaches</th>
<th>( w_p )</th>
<th>( w_c )</th>
<th>( w_s )</th>
<th>( \text{P} )</th>
<th>( \text{R} )</th>
<th>( \text{F1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. BOW</td>
<td>0.4</td>
<td>0.0</td>
<td>0.3</td>
<td>0.8583</td>
<td>0.8337</td>
<td>0.8501</td>
</tr>
<tr>
<td>2. TOPIC_CUR</td>
<td>0.4</td>
<td>0.2</td>
<td>0.0</td>
<td>0.8571</td>
<td>0.8299</td>
<td>0.8451</td>
</tr>
<tr>
<td>3. TOPIC_INTEGRATED ((w_p=0.4,w_c=0.0, w_s=0.3))</td>
<td>0.4</td>
<td>0.2</td>
<td>0.3</td>
<td>0.8569</td>
<td>0.8351</td>
<td>0.8447</td>
</tr>
<tr>
<td>4.</td>
<td>0.0</td>
<td>0.2</td>
<td>0.3</td>
<td>0.8565</td>
<td>0.8348</td>
<td>0.8445</td>
</tr>
</tbody>
</table>

### Table 5

Evaluation of different feature representation approaches

<table>
<thead>
<tr>
<th>Approaches</th>
<th>( \text{P} )</th>
<th>( \text{R} )</th>
<th>( \text{F1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. BOW</td>
<td>0.6000</td>
<td>0.6610</td>
<td>0.6120</td>
</tr>
<tr>
<td>2. TOPIC_CUR</td>
<td>0.7960</td>
<td>0.7710</td>
<td>0.7840</td>
</tr>
<tr>
<td>3. TOPIC_INTEGRATED ((w_p=0.4, w_c=0.0, w_s=0.3))</td>
<td>0.8583</td>
<td>0.8337</td>
<td>0.8501</td>
</tr>
</tbody>
</table>
The experimental results of three feature representation approaches are summarized in Table 5. From the table, applying the topic model on current page (TOPIC_CUR) improved the performance over the BOW by 17.2% based on the F1 measure. The approach of integrating current page with the neighboring pages via the topic model (TOPIC_INTEGRATED), however, yielded the best performance with the F1 measure of 85.01%; improvement of 23.81% over the BOW model. Thus, integrating the additional neighboring information, especially from the parent pages and sibling pages, via a topic model could significantly improve the performance of a classification model. The reason is the parent pages often provide terms, such as in the anchor texts, which provide additional descriptive information of the current page.

5. Conclusions and Future Works

To enhance the performance of web page classification based on the bag of words feature representation, we propose a method based on a topic model to integrate the additional term features obtained from the neighboring pages to improve the performance of web page classification. We apply the topic-model approach based on the Latent Dirichlet Allocation algorithm to cluster the word (or term) features into a set of latent topics. Words assigned into the same topic are semantically related.

From the experimental results, the approach of integrating current page with the neighboring pages via the topic model yielded the best performance with the F1 measure of 85.01%; an improvement of 23.81% over the BOW model.

In our future work, we plan to evaluate our proposed method on more interesting web 2.0 contents such as blogs. Link analysis may be included into classification approach for further improvement.

References


[8] M. Dash and H. Liu, “Feature Selection for Classification,” Intelligent Data Analysis,


