

Learning to Integrate Web Taxonomies with Fine-Grained Relations: A Case Study Using Maximum Entropy Model

Chia-Wei Wu^a, Tzong-Han Tsai^{ab}, Wen-Lian Hsu^{ac}

^aInstitute of Information Science, Academia Sinica, Nankang, Taipei, 115, Taiwan

^bDepartment of Computer Science and Information Engineering, National Taiwan University, Taipei, 640, Taiwan

^cDepartment of Computer Science, National Tsing Hua University, Hsingchu, 300, Taiwan
{cwwu, thtsai, hsu}@iis.sinica.edu.tw

Abstract. As web taxonomy integration is an emerging issue on the Internet, many research topics, such as personalization, web searches, and electronic markets, would benefit from further development of taxonomy integration techniques. The integration task is to transfer documents from a source web taxonomy to a target web taxonomy. In most current techniques, integration performance is enhanced by referring to the relations between corresponding categories in the source and target taxonomies. However, the techniques may not be effective, since the concepts of the corresponding categories may overlap partially. In this paper we present an effective approach for integrating taxonomies and alleviating the partial overlap problem by considering fine-grained relations using a Maximum Entropy Model. The experiment results show that the proposed approach improves the classification accuracy of taxonomies over previous approaches.

1 Introduction

A web taxonomy, or directory, is a hierarchical collection of categories and documents [2]. In the last decade, thousands of such taxonomies have been developed for various services, such as electronic auction markets, online book stores, electronic libraries, and search engines. Yahoo! and Google Directories are two good examples. The benefits of taxonomies include encouraging the serendipitous discovery of information, improving navigation among related topics, and enhancing full-text searching. In a web taxonomy, a category's concept is its parent's sub-concept [9].

Many of these taxonomies cover similar topics and knowledge. In recent years, integrating these taxonomies, which enables the reuse of information more efficiently and correctly, has become increasingly popular. For instance, Google News [1] collects news articles from various news web sites and categorizes them into its taxonomy, which is a typical example of assigning data from an existing taxonomy to another taxonomy. In B2B systems, millions of items need to be exchanged among thousands of taxonomies [8].

Given the enormous scale of the Web, manually integrating taxonomies is labor-intensive and time consuming. In recent years, various machine learning approaches, such as enhanced Naïve Bayes [3], Co-Bootstrapping [18], and SVM-based approaches [19] have been proposed. It is straightforward to formulate taxonomy integration as a classification task [3]. Suppose we want to integrate the BBC News web site with the Google News web site. The simplest way would be to assign news articles from BBC news to Google news based on the information contained in those articles. However, the relations between the categories in these two web sites could provide valuable information for assigning the articles. For example, if an article belongs to the *Sports* category of BBC news, it is likely that the article also belongs to the *Sports* category of Google news. Unfortunately, the relations between two categories in different taxonomies are inevitably fuzzy and noisy [19], since there are no standards for constructing taxonomies. In addition, taxonomies often overlap partially, as in *Software* and *Open source software*, which could undermine the accuracy of taxonomy integration.

Our taxonomy integration approach exploits the relations between a category in the source taxonomy and a category in the target taxonomy to improve the classification performance. We also consider the issue of partial concept overlap.

The remainder of this paper is organized as follows: In Section 2, we define the taxonomy integration task. In Section 3, state-of-the-art taxonomy technologies are briefly introduced. The features used in our taxonomy integration approach are presented in Section 4. In Section 5, we describe our experiments, including the dataset, settings, and results. Finally, we close the paper with some concluding remarks and also indicate possible future research directions in Section 6.

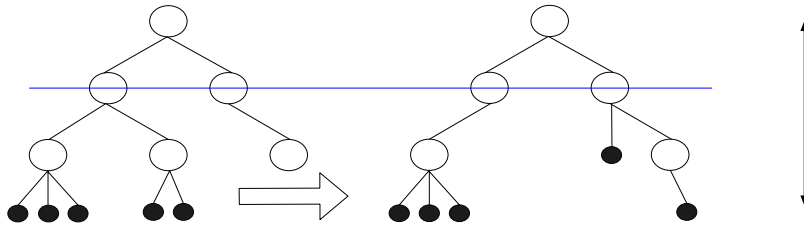


Fig. 1. The taxonomy integration task is to reclassify documents from a source taxonomy into a target taxonomy. The classification targets are the level-one categories in T .

2 Task Statement

The web taxonomy integration task in Fig.1 was originally defined by Agrawal and Srikant [3]. It can be formulated as the assignment of documents in a source taxonomy to a target taxonomy. The terms used in this task include:

- The source taxonomy, S , with a set of categories, $s_1, s_2, \dots, s_i, \dots, s_n$, each of which contains a set of documents.

- The target taxonomy, \mathbf{T} , with a set of categories, $t_1, t_2, \dots, t_l, \dots, t_m$, each of which contains a set of documents.

For each document x in \mathbf{S} , our task is to assign x to the target categories in \mathbf{T} . In this paper, we follow the settings of Agrawal and Srikant [3] and Zhang and Lee [19] [18], which simply consider level-one categories in \mathbf{T} as the target categories.

3 Related Works

Taxonomy integration is similar to text classification in that it also assigns a document to one or more target categories. However, in taxonomy integration, we have the additional information: the document’s source categories in \mathbf{S} . In addition, the relations between \mathbf{S} and \mathbf{T} can be used to enhance the accuracy of integration. For example, suppose most documents in category s_i are also in category t_j , we can then infer that these two categories are similar. Thus, any document x in s_i is likely to be categorized into t_j . From this example, we know that the relations between the source and target categories can be measured by estimating the degree of overlap between them. However, we have to resolve the following problems: (1) how to estimate the degree of overlap, and (2) how to use this information.

We now briefly introduce some state-of-the-art approaches that use the information about \mathbf{S} , or the relations between \mathbf{S} and \mathbf{T} . The source taxonomy provides information about the relations between corresponding categories, including the documents in them. For taxonomy integration, these relations can be used to augment inadequate information about the documents themselves. Zhang and Lee developed the cluster shrinkage algorithm (CS) [19], which combines information about documents in categories of the same category. The authors estimated that CS can achieve a 15% improvement over traditional SVM methods.

The Enhanced Naïve Bayes (ENB) algorithm [3] and Co-Bootstrapping (CB) algorithm [18] are the two main approaches that use the relations between the source and target taxonomies. The ENB algorithm, proposed by Agrawal and Srikant, initially used a Naïve Bayes (NB) classifier [13] to estimate the degree of overlap between the source and target categories. The estimated scores were then combined with the probabilities calculated by a second NB classifier. According to Agrawal and Srikant [3], ENB is 15% more accurate than NB.

Similarly, Co-Bootstrapping (CB) [18] exploits inter-taxonomy relationships by providing category indicator functions as additional features of documents. According to Zhang and Lee [18], CB achieves close to a 15% improvement over NB. We discuss the above approaches in more detail in Section 4.2.

4 Learning to Integrate Taxonomies

We also use the relations between corresponding categories in \mathbf{S} and \mathbf{T} to enhance the integration process; however, unlike previous approaches, we do not consider a flattened taxonomy only, i.e., a taxonomy reduced to a single level [3]. The relations be-

tween the level-one categories in \mathbf{S} and \mathbf{T} could be noisier than the relations between lower-levels, since the concept space of higher-level categories in taxonomies is more general. Therefore, we employ some features used in machine learning models to extract finer relations between lower-level categories in \mathbf{S} and \mathbf{T} .

In this section, we introduce five features used in our taxonomy integration approach. One feature is commonly used in text classification, two are derived from other taxonomy integration systems, and the remaining two are our own. We then introduce the Maximum Entropy (ME) model, a well-known classifier used in many applications. The section concludes with a discussion of ME's advantages and the process of our approach.

4.1 Features

Feature selection is critical to the success of machine learning approaches. In this section, we describe the features used in our system and discuss the effectiveness of each feature.

Word-TargetCat features (WT)

When classifying a document, the collection of words it contains is important. More specifically, a distinct feature is initiated for each word-category combination. In addition, if a word occurs often in one class, we would expect the weight for that word-category pair to be higher than if the word were paired with other categories. In text classification, features accounted for the number of times a word appears should improve classification. For example, Naïve Bayes implementations that use word counts outperform implementations that do not [14]. Since taxonomy integration is an extension of text classification, we adopt these features in our approach. For each word, w , and category, t' , in the target taxonomy, \mathbf{T} , we formally define the Word-TargetCat feature as:

$$f_{w,t'}(x,t) = \begin{cases} \frac{N(x,w)}{N(x)} & \text{if } t = t' \\ 0 & \text{Otherwise} \end{cases} \quad (1)$$

where $N(x,w)$ is the number of times a word, w , appears in a document x , and $N(x)$ is the number of words in x .

Normalized Word-TargetCat features (NWT)

In practice, the number of words contained in a document varies, and is relatively small compared to its vocabulary size in the document. According to the definition in Equation 1, most Word-TargetCat features will be zero. Therefore, it would be difficult to classify a web document by referring to a few words only. In text classification, solving this problem is difficult, since no more information can be used. In taxonomy classification, however, information about a document's category in both the source taxonomy and the target taxonomy is available. For each word, w , we can add the weight of w 's total count of the documents in the same category to w 's original count.

We regard this step as a kind of normalization, after which many zero Word-TargetCat features become non-zero values. Zhang and Lee [19] developed the cluster shrinkage (CS) algorithm to perform this normalization, which conceptually moves each document to the center of its level-one parent category. Zhang and Lee showed that this normalization significantly boosts the accuracy of taxonomy integration. Here, NWT is calculated by a modified version of CS. For each word w and category t' in the target taxonomy, \mathbf{T} , we define the NWT feature as:

$$f_{w,t'}(x, t, c_T) = \begin{cases} \eta \frac{N(x, w)}{N(x)} + (1 - \eta) \frac{N(c_T, w)}{N(c_T)} & \text{if } t = t' \\ 0 & \text{Otherwise} \end{cases} \quad (2)$$

where $N(x, w)$ is the number of times word w appears in a document x ; $N(x)$ is the number of words in x ; $N(c_T, w)$ is the number of times word w appears in x 's level-one category c_T ; $N(c_T)$ is the number of words in c_T ; and η is the weight to control the strength of normalization effect.

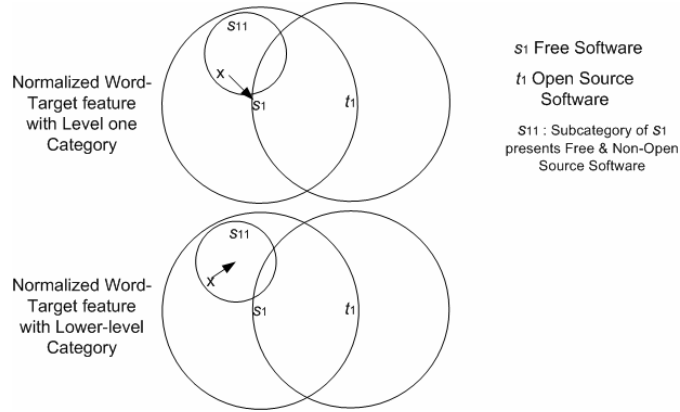


Fig. 2. With different level categories, the feature, NWTs, has different effects.

Normalized Fine-Grained Word-TargetCat features (NFWT)

As described in the last section, CS effectively helps Word-TargetCat features become non-zero values. That is, after applying CS, each document moves closer to its level-one parent category's center. In some cases, this can augment the correct information for classifying documents. However, the level-one parent category usually contains words that are too general or belong to cross-domains. Therefore, the accuracy improvement of taxonomy integration achieved by CS is reduced. In our approach, we consider the hierarchy's structure and regard the lowest-level parent category of the document as its category. Compared to the level-one parent category, the lowest-parent category contains more coherent information. As shown in Fig. 3, x is a document in the category s_{11} , and s_{11} is a subcategory of s_1 . If we use CS as our normalization algorithm and s_1 is x 's category, then x will be closer to s_1 after applying the normalization step, This could cause x to be misclassified into t_1 , but taking the

lowest-level as the document’s category would avoid this potential error of Word-TargetCat features.

For each word w and category t in the target taxonomy, \mathbf{T} , we define NFWT as:

$$f_{w,t'}(x, t, c_B) = \begin{cases} \eta \frac{N(x, w)}{N(x)} + (1 - \eta) \frac{N(c_B, w)}{N(c_B)} & \text{if } t = t' \\ 0 & \text{Otherwise} \end{cases} \quad (3)$$

where $N(x, w)$ is the number of times word w appears in a document x ; $N(x)$ is the number of words in x ; $N(c_B, w)$ is the number of times word w appears in x ’s lowest parent category c_B ; $N(c_B)$ is the number of words in c ; and η is the weight to control the strength of normalization effect..

Normalized TargetCat-SourceCat features (NTS)

For taxonomy integration, there is another type of information that can be used to decide a document’s target category, namely, the relations between corresponding source categories and the target categories. Zhang and Lee [18] initiated a distinct feature for each target-source category combination. In the training phase, documents in \mathbf{S} are used to train a multi-class classifier. Then, for each document y in \mathbf{T} , we use the classifier to decide y ’s category in \mathbf{S} , denoted as s' . The feature corresponding to the combination of t' and s' is enabled. In the test phase, when calculating t' ’s probability or score for each document x in \mathbf{S} , the feature corresponding to the combination of t' and x ’s level-one parent category is enabled. Zhang and Lee [18] showed that using this feature boosts the classification accuracy. We implement such features as Normalized TargetCat-SourceCat features (NTS). For each category t' in the target taxonomy \mathbf{T} , and each category s' in the source taxonomy \mathbf{S} , we define the NTS features as:

$$f_{t',s'}(t, s) = \begin{cases} 1 & \text{if } t = t' \text{ and } s = s' \\ 0 & \text{Otherwise} \end{cases} \quad (4)$$

Normalized Word-TargetCat-SourceCatSourceCat features (NWTS)

Although NTS improves the integration accuracy by relating each source and target category pair, the level-one category is so general that the relation between t' and s' is not precise enough to achieve a significant improvement. Since different words can have different impacts on a target-source category pair, we believe that these target-source combinations should be further divided by each distinct word. For each word w and category t' in the target taxonomy \mathbf{T} , and category s' in the source taxonomy \mathbf{S} , we define the normalized word count feature as:

$$f_{w,t',s'}(x, t, c, s) = \begin{cases} \eta \frac{N(x, w)}{N(x)} + (1 - \eta) \frac{N(c, w)}{N(c)} & \text{if } t = t' \text{ and } s = s' \\ 0 & \text{Otherwise} \end{cases} \quad (5)$$

where $N(x, w)$ is the number of times word w appears in a document x ; $N(x)$ is the number of words in x ; $N(c, w)$ is the number of times word w appears in the source

category c ; $N(c)$ is the number of words in c ; and η is the weight to control the strength of normalization effect.

We illustrate the use of the Word-TargetCat-SourceCat feature by the following example. As shown in Fig.2, to estimate the relation between t_1 and s_1 , we assign documents in t_1 to s_1 . From the classification result, we know that t_1 and s_1 overlap partially in the conceptual space. However, we still do not know which documents of s_1 should be classified into t_1 . In fact, the instances of s_{11} should not be classified into t_1 . We need other information to know the relations between the lower-level categories. Therefore, we add the word dimension to the original NTS, to combine the dimensions of the target and source categories.

4.2 Using Features in Maximum Entropy

We select the Maximum Entropy (ME) model [4] to implement our approach. ME is a statistical modeling technique used for estimating the conditional probability of a target label by the given information. ME computes the probability, $p(o|h)$, where o denotes all possible outcomes from the space, and h denoted all possible histories from the space. A history is all the conditioning data that enables one to assign probabilities to the space of outcomes. In the taxonomy integration task, the history can be viewed as all information derivable from the documents of the taxonomy relative to the current document, and the outcome can be viewed as the target category label. The computation of $p(o|h)$ in ME depends on a set of features, which is helpful for making predictions about the outcome.

Given a set of features and a training set, the ME estimation process produces a model where every feature f_i has a weight λ_i . From Berger [4], we can compute the conditional probability as:

$$p(o | h) = \frac{1}{z(h)} \exp \left(\sum_i \lambda_i f_i(h, o) \right) \quad (6)$$

$$Z(h) = \sum_o \exp \left(\sum_i \lambda_i f_i(h, o) \right), \quad (7)$$

The probability is derived by multiplying the weights of the active features (i.e., those $f_i(h, o) = 1$). The weight, λ_i is estimated by a procedure called *improved iterative scaling* (IIS) [7], which improves the estimation of weights iteratively. The ME estimation technique guarantees that for every f_i , the expected value of λ_i will equal the empirical expectation of λ_i in the training corpus. The feature sets we use were introduced in Section 4.1. The process of our approach is shown in Fig. 3.

Advantages of Maximum Entropy

We use ME to build the classifier for taxonomy integration because it has a proven competitive performance in various tasks, including part-of- speech tagging [15], named entity recognition [5], English parser [6], prepositional phrase attachment [16], and text classification [14].

As noted in [5], ME allows users to focus on finding features that characterize the problem, while leaving feature weight assignment to the ME estimation routine. When new features are discovered, users do not need to reformulate the model as in other machine-based approaches, because the ME estimation routine automatically calculates new weight assignments.

Although using the ME model is a good choice, other machine learning algorithms, such as Support Vector Machine [10], Conditional Random Field [11], or Boosting [17], could also be adopted in our approach to improve taxonomy integration.

4.3 The algorithm of our approach : NFWT+NWTS

Our proposed approach consists of NFWT and NWTS, which were introduced in Section 4.1. The procedure of NFWT+NWTS is shown in Fig.3.

```

T: target taxonomy S: source taxonomy
Taxonomy Integration Main (T, S)
1: Use labeled documents in S to induce a ME-based
classifier with NFWT features for transferring the
document from T to S and then use these results to
measure the similarity between those corresponding
categories in S and T.
2: Use labeled documents in T to induce a ME-based
classifier with NWTS and NFWT features for transfer-
ring the document in S to T.
3: Return classification result;

```

Fig. 3. Our approach includes NFWT and NWTS features.

There are two steps in NFWT+NWTS. The first step uses labeled documents in **S** as a training corpus to train a classifier of **S** with NFWT features. The classifier is used to generate feature values representing the source-target relations, which are necessary information of NWTS features. In the second step, we induce a classifier to transfer documents from **S** to **T** with both NWTS and NFWT features. Therefore, the final classifier will include NFWT, which uses lowest-level categories in the cluster shrinkage algorithm as well as NWTS, which considers word dimensions.

5 Experiments and Results

5.1 Datasets

We collected five datasets from the Google and Yahoo! Directories to evaluate our approach. Each dataset included a category in Google Directory, the corresponding category with a similar topic in Yahoo! Directory, and vice versa. Hyperlinks and web

pages within these two categories were also stored in the dataset. Table 1 shows the five dataset names and their paths in the directories.

In the Google and Yahoo! Directories, each link/document includes the web page’s title, URL, and description. For example:

Title: BBC News
URL : <http://news.bbc.co.uk/>
Description: offers U.K., world, business, science, and entertainment news.

In the experiment, we used the information from the title, description, and content of the web page as the information for training and testing. All documents were pre-processed by removing the stop words, and stemming.

Table 1. Datasets

	Google Directory	Yahoo! Directory
Disease	Top/Health/ Conditions_and_Diseases	Health/ Diseases_and_Conditions/
Book	Top/Shopping/Publications /Books/	Business_and_economy/ shopping_and_services/ books/
Movies	Top/Arts/Movies/Genres/	Entertainment/ movies_and_film/genres/
Garden	Top/Shopping/ Home_and_Garden/	Business_and_economy/ shopping_and_services/ home_and_garden/
Outdoor	Top/Recreation/Outdoors/	Recreation/Outdoors/

In Table 2, each row shows the dataset name, the number of links within each directory, and the number of shared links between the two directories. In each dataset, the shared links, identified by their URLs, are used as the testing data, while the rest of the links are used as the training data. Only a small proportion of links are shared by the two web taxonomies, which shows the benefit of integrating them.

The number of categories is shown in Table 3. As mentioned earlier, we use the level-one categories as the target classes in our classification task.

Table 2. Number of links in each dataset.

	Google	Yahoo!	Shared links
Disease	24,522	11,231	5,300
Book	58,46	8,168	567
Movies	30,438	19,223	2,560
Garden	13,058	4288	617
Outdoor	12,346	7,439	817
Total #	80364	50349	9861

Table 3. Number of categories

	Google	Yahoo!
	# of target categories	# of target categories
Disease	602	692
Book	44	43
Movies	23	23
Garden	37	18
Outdoor	37	65
Total #	743	841

5.2 Experimental Design

Our task is to classify documents from a source taxonomy into a target taxonomy. The experiment of each dataset consists of classifying a document from Google to Yahoo and vice versa. We use the documents of Yahoo (excluding the shared links) for training and classifying documents from Google into Yahoo. Similarly, we use documents from Google (excluding the shared documents) for training and classifying documents from Yahoo into Google. The shared documents are used as testing data.

To measure the correctness of all approaches, we defined the following classification accuracy:

$$\frac{\text{Number of instances in } S \text{ are classified correctly}}{\text{Number of instances in } S}, \text{ where } S \text{ is the source taxonomy.}$$

5.3 Settings

In the NB and ENB experiments, we implement the NB and ENB modules. The parameter w of ENB is selected from a series of numbers: {0, 1, 3, 10, 30, 100, 300, and 1,000} that have the best performance. The smoothing parameter [3] of the NB and ENB classifier is set to 0.1.

We use Maximum Entropy Toolkit [12] to implement the ME-based approaches. To compare our approach with normal text classification methods, we implement the ME-based text classification algorithm proposed by Kamal Nigam, John Lafferty, and Andrew McCallum [14]. We denote it as MEtext, which simply uses the Word-TargetCat features (WT).

We compare the features of our approach with the features used in previous approaches (NTS for [18] and NWT for [19] as discussed in Section 4). Although in previous works [18, 19], the features were implemented with other machine learning models, they can also be easily implemented with ME. The parameter η used in NWT, NFWT, and NWTS is set to 0.5.

5.4 Experimental Results

Table 4. Experimental results of ME-based text classification approach and our approach.

		MEtext	<i>NFWT+NWTS</i>	Improvement
G to Y	Disease	34.4%	46.3%	11.9%
	Book	48.8%	65.0%	16.2%
	Movies	56.8%	73.8%	17.0%
	Garden	75.2%	82.1%	6.9%
	Outdoor	59.6%	72.6%	13.0%
	Average	55.0%	68.0%	13.1%
Y to G	Disease	25.8%	47.0%	21.2%
	Book	39.7%	63.7%	24.0%
	Movies	44.7%	66.0%	21.3%
	Garden	59.4%	69.3%	9.9%
	Outdoor	62.7%	71.9%	9.2%
	Average	46.5%	63.6%	17.1%

In Table 4, we denote the ME-based text classification that only uses the WT feature as MEtext, and our proposed approach as *NFWT+NWTS*. To make a distinction between our approach and the others, we use boldface and italic style for *NFWT* and *NWTS*. One can see that *NFWT+NWTS* performs significantly better than normal text classification approaches [14] in all five topics. These results suggest that our approach can effectively exploit the relations between corresponding categories in the target and source taxonomies to enhance the classification accuracy.

Table 5. Experimental results of NB, ENB and our approach.

		NB	ENB	<i>NFWT+NWTS</i>	Improvement over NB	Improvement over ENB
G to Y	Disease	25.2%	25.6%	46.3%	21.1%	20.7%
	Book	44.7%	51.4%	65.0%	20.3%	13.6%
	Movies	54.5%	68.0%	73.8%	19.3%	5.8%
	Garden	75.1%	79.4%	82.1%	7.0%	2.7%
	Outdoor	54.0%	60.3%	72.6%	18.6%	12.3%
	Average	50.7%	56.9%	68.0%	17.3%	11.0%
Y to G	Disease	25.8%	29.1%	47.0%	21.2%	17.9%
	Book	38.3%	44.0%	63.7%	25.4%	19.7%
	Movies	47.7%	54.9%	66.0%	18.3%	11.1%
	Garden	65.4%	66.1%	69.3%	3.9%	3.2%
	Outdoor	61.4%	67.9%	71.9%	10.5%	4.0%
	Average	47.7%	52.4%	63.6%	15.7%	11.2%

Next, we compare our approach (*NFWT+NWTS*) with NB and ENB. In Table 5, one can see that, as previous works showed, ENB performs slightly better than NB. However, our proposed approach, *NFWT+NWTS*, outperforms NB and ENB by 17% and 11%, respectively. These results show that referring to the relationships between taxonomies and replacing NB with ME can improve the accuracy of taxonomy integration. The former is due to the high degree of relevance between the two taxonomies. The latter is because ME can catch more dependencies among different features that commonly exist in text categorization and taxonomy integration problems. Unlike other approaches, our approach retains the hierarchical structure of taxonomies, and estimates the relationship between lower-level categories. Since the number of words on a web page may be significantly fewer than in a normal news article, the results of web page classification are more likely to be affected by the sparseness of words. The information in the source and target taxonomies can provide a great deal of help in smoothing the word frequency vectors of web pages, or measuring the similarity between source and target categories.

One may further ask: How can information in the source and target taxonomies be used to achieve better performance? Next, we will compare our approach and previous approaches on taxonomy integration.

Table 6. Experimental results of NWT and *NFWT*.

		NWT	<i>NFWT</i>
G to Y	Disease	34.0%	38.0%
	Book	54.7%	58.6%
	Movies	61.5%	67.0%
	Garden	81.3%	82.3%
	Outdoor	65.8%	62.5%
	Average	59.5%	61.6%
Y to G	Disease	30.5%	38.2%
	Book	45.4%	48.1%
	Movies	49.1%	56.0%
	Garden	60.9%	69.4%
	Outdoor	67.6%	70.3%
	Average	50.7%	56.4%

In Table 6, the major difference between NWT and *NFWT* is that NWT uses level-one categories in the cluster shrinkage algorithm, while *NFWT* uses the lowest-level categories. The experimental results suggest that using lower-level categories yields a better performance than level-one categories. This supports our observation that level-one categories usually contain words that are too general or belong to cross-domains, which could undermine the performance. Even though the NWT is not as efficient as *NFWT*, its performance is still better than MEtext, as shown in Table 4.

Now, we compare the effects of two factors: (1) using level-one or lowest-level categories in the cluster shrinkage algorithm, and (2) using document or word dimensions to represent the source-target relations. Table 7 shows all combinations of these two factors: NWT+NTS, NWT+*NWTS*, *NFWT*+NTS, and *NFWT*+*NWTS*. The configuration name is composed of the features it uses. For example, NWT +NTS means

it uses level-one categories in cluster shrinkage algorithm as NWT and uses NTS to measure the source-target relations. In Table 8, we can see that *NFWT*+NTS outperforms NWT+NTS, and *NFWT*+*NWTS* outperforms NWT+*NWTS*. These results establish that lowest-level categories contain more precise information for categorization than level-one. Therefore, configurations using lowest-level categories in the cluster shrinkage algorithm (*NFWT*) outperform those using level-one categories (NWT). In addition, we can see that NWT+*NWTS* outperforms NWT+NTS, and *NFWT*+*NWTS* outperforms *NFWT*+NTS. These results demonstrate that using word-dimensions (*NWTS*) rather than document dimensions (NTS) to represent source-target relations could further alleviate the partial overlap problem.

Table 7. Four different configurations of feature combinations.

		Use level-one category in the cluster shrinkage algorithm	Use lowest-level category in the cluster shrinkage algorithm
Considering dimension	document	NWT +NTS	<i>NFWT</i> +NTS
Considering dimension	word	NWT + <i>NWTS</i>	<i>NFWT</i> + <i>NWTS</i>

Table 8. Experimental results of all combinations of features.

		NWT+NTS	NWT+ <i>NWTS</i>	<i>NFWT</i> +NTS	<i>NFWT</i> + <i>NWTS</i>
G to Y	Disease	38.7%	42.4%	44.0%	46.3%
	Book	59.0%	58.6%	64.2%	65.0%
	Movies	63.4%	68.4%	69.7%	73.8%
	Garden	78.2%	77.5%	81.8%	82.1%
	Outdoor	60.1%	68.2%	72.8%	72.6%
	Average	59.9%	63.0%	66.5%	68.0%
Y to G	Disease	39.8%	41.3%	46.9%	47.0%
	Book	52.2%	60.1%	59.5%	63.7%
	Movies	54.9%	59.4%	58.9%	66.0%
	Garden	60.5%	59.8%	71.1%	69.3%
	Outdoor	65.4%	68.2%	72.2%	71.9%
	Average	54.5%	57.8%	61.7%	63.6%

Generally speaking, using information of source category improves the categorization accuracy. We can see that NWT+NTS and NWT+*NWTS* outperform NWT, and *NFWT*+NTS and *NFWT*+*NWTS* outperform *NFWT*. Among these four configurations, the performance of NWT+NTS is the worst, such as in the Garden category. We believe this is because the classification criteria of Google’s Garden directory is much different with that of Yahoo!’s Garden directory. As a result, the partial overlap problem becomes very serious in Garden category. To further justify this argument, we compare the name of level-one categories of Garden in Yahoo! and Google. It is found that there is no common name between those categories in Google and Yahoo!’s Garden directory. From this observation, we conclude that, NWT+NTS, which

uses level-one categories in the cluster shrinkage algorithm and considers only document dimension in measuring the source-target relations, is influenced most deeply by the different classification criteria between the source and target taxonomy.

The experimental results of each taxonomy integration approach are shown in Fig. 4 and Fig. 5 respectively.

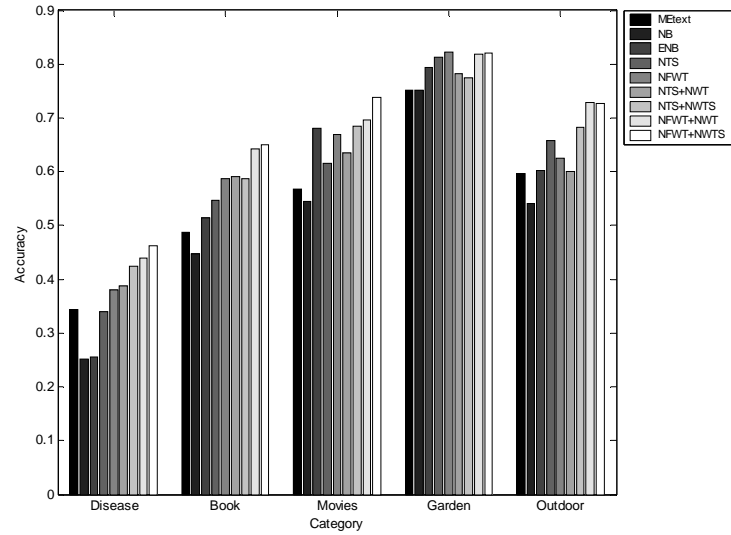


Fig. 4. Comparison of the experimental results of all taxonomy integration approaches and features (Google to Yahoo)

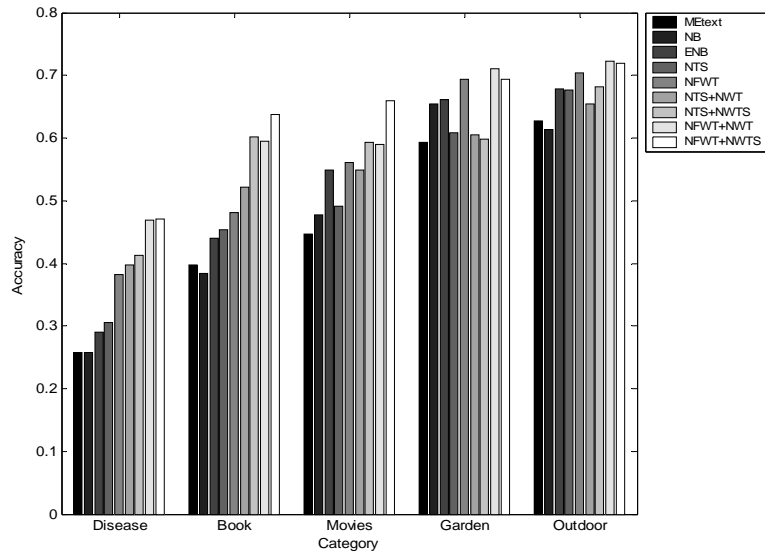


Fig. 5. Comparison of the experimental results of all taxonomy integration approaches and features (Yahoo to Google)

6 Conclusions

In this paper, we have proposed an approach that effectively uses the relations between corresponding categories in source and target taxonomies to improve taxonomy integration. Unlike previous works that use a flattened hierarchy as the information source, we utilize hierarchical information to extract fine-grained relations, which alleviates the partial concept overlap problem in taxonomy integration.

The proposed approach was tested using real Internet directories. Its performance was better than normal text classification approaches and other features in previous works. The experimental results also support the assumption that using a flattened hierarchy could cause the loss of valuable information about relations between corresponding categories.

In the future, more information, such as web resources, a third taxonomy, or existing knowledge ontology could be incorporated into our approach. It would also be interesting to see how our approach can be applied to other applications.

7 Acknowledgement

We are grateful for the support of National Science Council under grant NSC94-2752-E-001-001, and the support of the thematic program of Academia Sinica under grant AS91IIS1PP and 94B003.

References:

1. Google News: <http://news.google.com/>
2. Taxonomies of Knowledge: Uncovering Hidden Themes in Existing Corporate Data: <http://www.infotoday.com/it2001/presentations/pohs1.ppt>
3. Agrawal, R. and Srikant, R.: On Integrating Catalogs. *Proceedings of the Tenth International Conference on World Wide Web* (2001), 603 - 612.
4. Berger, A., Pietra, S. A. D., and Pietra, V. J. D.: A Maximum Entropy Approach to Natural Language Processing. *Computer Linguistics*, vol. 22, (1996), 39-71.
5. Borthwick, A.: A Maximum Entropy Approach to Named Entity Recognition. New York University, PhD Thesis, (1999).
6. Charniak, E.: A maximum-entropy-inspired parser. *Proceedings of the First Conference on North American chapter of the Association for Computational Linguistics* (2000), 132 - 139.
7. Darroch, J. N. and Ratcliff, D.: Generalized Iterative Scaling for Log-linear Models. *Annals of Mathematical Statistics*, vol. 43, (1972), 1470-1480.
8. Fensel, D., Ding, Y., Omelayenko, B., Schulten, E., Botquin, G., Brown, M., and Flett, A.: Product Data Integration in B2B E-Commerce. *IEEE Intelligent Systems*, vol. 16, no. 4, (2001).
9. Huang, C.-C., Chuang, S.-L., and Chien, L.-F.: Liveclassifier: Creating Hierarchical Text Classifiers through Web Corpora. *Proceedings of the Thirteenth International Conference on World Wide Web*, (2004), 184-192.

10. Joachims, T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features. *Tenth European Conference on Machine Learning* (1998), 137-142.
11. Lafferty, J., McCallum, A., and Pereira, F.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *Proceedings of the Eighteenth International Conference on Machine Learning*, (2001), 282-289.
12. Maximum Entropy Toolkit: <http://maxent.sourceforge.net>
13. Mitchell, T.: *Machine Learning*, Singapore, McGraw Hill, (1997).
14. Nigam, K., Lafferty, J., and McCallum, A.: Using Maximum Entropy for Text Classification. *In IJCAI-99 Workshop on Machine Learning for Information Filtering*, (1999), 61-67.
15. Ratnaparkhi, A.: A Maximum Entropy Model for Part-Of-Speech Tagging. *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (1996), 133-142.
16. Ratnaparkhi, A.: Statistical models for unsupervised prepositional phrase attachment. *Proceedings of the Thirty-Sixth conference on Association for Computational Linguistics*, vol. 2, (1998), 1079 - 1085.
17. Schapire, R. E. and Singer, Y.: BoosTexter: A Boosting-based System for Text Categorization. *Machine Learning*, vol. 39, (2000), 135-168.
18. Zhang, D. and Lee, W. S.: Web taxonomy integration through co-bootstrapping. *Proceedings of the Twenty-Seventh Annual International Conference on Research and Development in Information Retrieval* (2004), 410 - 417.
19. Zhang, D. and Lee, W. S.: Web taxonomy integration using support vector machines. *Proceedings of the Thirteenth International Conference on World Wide Web* (2004), 472 - 481.