Enhanced Web Objects Classification using Social Tags

Moheb R. Girgis
Department of Computer Science,
Faculty of Science, Minia University,
El-Minia, Egypt
Email: moheb.girgis [AT] mu.edu.eg

Lamia M. Rady
Department of Computer Science,
Faculty of Science, Minia University,
El-Minia, Egypt

Abstract—**The automatic classification of Web objects into semantic categories is very important to facilitate indexing, browsing, searching, and mining these objects. But this is a very challenging task, because web objects often suffer from a lack of easy-extractable features with semantic information, interconnections between each other, and training examples with category labels. Social tags reflect the web objects semantics from users' points of view, which makes them an ideal web objects feature that overcomes the difficulties of web object classification. In this paper we study the impact of using social tagging on the performance of text classification techniques in web objects classification. An automated system for web objects classification has been developed that is based on social tags exploration. The system has three phases: data preprocessing, classification and evaluation phases. It accepts a training dataset that represents a set of web pages with its URLs, tags, titles and categories. Using this dataset, the system constructs a predictive model that is later used to assign labels to web objects based on their tags. In the classification step, the system employs three known text classification techniques namely, Support Vector Machine, Naïve Bayes, and Decision Tree, through the WEKA software. Experiments have been conducted to evaluate the effectiveness of using social tags with each one of the three text classification techniques in web objects classification. The experimental results indicate that using tags significantly improve the classification performance.**

Keywords-web objects classification; social tagging; text classification methods; WEKA software; cross validation.

I. INTRODUCTION

Web pages may contain, in addition to textual web objects, various non-textual web objects, such as photos, videos, and products. Automatically classifying these web objects into manageable semantic categories is very important to facilitate indexing, browsing, searching, and mining these objects. The explosive growth of heterogeneous web objects has made the problem of web classification increasingly challenging. Such objects often suffer from a lack of easy-extractable features with semantic information, interconnections between each other, as well as training examples with category labels [1].

Social network systems allow users to use descriptive tags to annotate the web objects that they are interested in. These tags are effectively utilized for information sharing and retrieval. A study of a large amount of user-generated tags in social network systems, such as del.icio.us, revealed that in general, user-generated tags are consistent with the web objects they are attached to, while more concise and closer to the understanding and judgments of users about the objects [2]. Hence, social tags reflect the semantics of the web objects from users' points of view, using a ubiquitous vocabulary for heterogeneous domains of objects. This property makes social tags an ideal web objects feature, which overcomes the difficulties of web object classification.

The aim of this research work is to study the impact of using social tagging on the performance of text classification techniques in web objects classification. To achieve this aim, we have developed an automated system for web objects classification that is based on social tags exploration. This system has three main phases: the data preprocessing phase, the classification phase and the evaluation phase. It accepts a training dataset that represents a set of web pages with its URLs, tags, titles and categories. Using the provided dataset, the system constructs a predictive model that is later used to assign labels to web objects based on their tags. In the classification step, the system employs the three known text classification techniques namely, Support Vector Machine (SVM), Naïve Bayes (NB), and Decision Tree (DT) through the WEKA software. Using the developed system, experiments have been conducted to evaluate the effectiveness of using social tags with each one of the three text classification techniques in web objects classification.

The paper is organized as follows: Section 2 presents a brief review of related work. Section 3 discusses the web objects classification challenges. Section 4 presents the classification methods used in the developed system. Section 5 describes the phases of the developed web objects classification system and their interfaces. Section 6 presents the results of the experiments that have been conducted to evaluate the effectiveness of using social tags in web objects classification. Section 7 presents the conclusion of this research work.

II. RELATED WORK

Web object classification has been investigated for a long time, especially web page classification [3, 4] and multimedia classification [5, 6]. To improve web page classification results, hyperlinks [7], html metadata [3], and query log [8], are explored. Chakrabarti et al. [7] proposed statistical models and a relaxation labeling technique for better classification by exploiting hyperlink information in a small neighborhood around web pages. An investigation by Ghani et al. [3] showed that metadata available for web pages can be extremely useful for improving classification accuracy. Shen et al. [8] observed that web pages that are clicked after the same queries are implicitly linked. Based on this observation, they provided an approach for automatically building the implicit links between web pages using web query logs, and used these links in web page classification, which improved classification results. For multimedia objects classification, both text feature and contextual information extracted from images or videos are combined to improve the performance. Kalva et al. [5] presented a method for the classification of images that combines information extracted from the images and contextual information. Lin and Hauptmann [6] combined text features from closed captions and visual features from images to classify broadcast news video. Xue et al. proposed an iterative reinforce categorization algorithm (IRC) [9], in which the category of one web object is propagated to reinforce the categorization of other interrelated web objects, and vice versa, as an iterative process.

Social tags can benefit web search [10, 11], information retrieval [12, 13], semantic web [14], web page clustering [15], and user interest mining [2]. Bao et al. [10] observed that the social tags can benefit web search in two aspects: 1) the tags are usually good summaries of corresponding web pages; 2) the count of tags indicates the popularity of web pages. Based on this observation, they proposed two algorithms, one for calculating the similarity between social tags and web queries; and another one for capturing the popularity of web pages. Heymann et al. [11] studied the potential impact the social tags may have on improving web search. Zhou et al. [12] proposed a unified framework to combine the modeling of social tags with the language-based methods for information retrieval. Schenkel et al. [13] developed a top-k algorithm for social search and ranking with two-dimensional expansions: social expansion considers the strength of relations among users, and semantic expansion considers the relatedness of different tags. Wu et al. [14] used a probabilistic generative model to model the user's annotation behavior and to automatically derive the emergent semantics of the tags, then they applied the derived emergent semantics to discover and search shared web bookmarks. Brooks and Montanez [15] studied the effectiveness of tags for clustering similar articles. They then showed that automated tagging produces more focused, topical clusters, whereas human-assigned tags produce broad categories. They also showed that clustering algorithms can be used to reconstruct a topical hierarchy among tags. Li et al. [2] proposed a social interest discovery approach based on user-generated tags.

However, there is little work on exploring social tagging for general web object classification. Yin et al. [1] explored social tags for web object classification. They formulated the web object classification problem as an optimization problem on a graph of objects and tags. Then, they proposed an algorithm, which utilizes social tags as enriched semantic features for the objects, and infers the categories of unlabeled objects from both homogeneous and heterogeneous labeled objects, through the implicit connection of social tags.

III. WEB OBJECTS CLASSIFICATION CHALLENGES

Web objects classification is the process of assigning a Web object to one or more predefined category labels [4]. Classification can be understood as a supervised learning problem in which a set of labeled data is used to train a classifier which can be applied to label future examples [16].

Classification of web objects is a very challenging task due to: (1) Lack of features. The limited text description is usually too sparse to provide enough semantic features. Content features of images or videos on the other hand, usually cannot be extracted in an accurate and efficient way. (2) Lack of interconnections. Web objects often exist in isolate settings, where interconnections between each other are limited. (3) Lack of labels. Web object classification usually suffers from a lack of training examples. Creating a large training set for certain types of web objects is laborious, sometimes even infeasible. [1]

Heterogeneous objects on the Web are tagged by users, with keywords freely chosen from their own vocabularies. These tags reflect the semantics of the web objects from users' points of view. This property makes social tags an ideal web objects feature, which overcomes the above difficulties of web object classification. First, a web object is associated with tagged keywords selected by many users, which provide enriched semantic features for web object classification. Second, through the intermediate connection of tags, a new link structure of web objects (and tags) is established, which makes it feasible to explore the latent relationships between web objects. Furthermore, since people are likely to tag different types of objects using the similar vocabularies, heterogeneous types of web objects are now connected through common tags. [1]

IV. CLASSIFICATION METHODS

Classification methods are used to construct a predictive model (a classifier) using a set of labeled data (training set). This model is later used to assign labels to future instances. Instances in the dataset that provides the input to the classifier are characterized by a set of features, or attributes. The outcome is the category to which the instance belongs. In this work, the instances are web objects, and their characterizing features are the social tags associated with them.

For our classification task we have used WEKA (Waikato Environment for Knowledge Analysis) [17], which is a collection of machine learning algorithms for data mining tasks, along with methods for data pre-processing,

classification, regression, clustering, association rules, visualization and for evaluating the result of learning schemes on any given dataset. From these algorithms, we have used the three known text classification techniques namely, Support Vector Machine (SVM), Naïve Bayes (NB), and Decision Tree (DT).

V. THE SYSTEM DESCRIPTION

The developed web objects classification system, which is written in Java, has three main phases: the data preprocessing phase, the classification phase and the evaluation phase.

A. The Data Preprocessing Phase

Our training data sample represents a set of web pages with its URLs, tags, titles and categories. This data set is taken from http://www.michael-noll.com/cabs120k08/. CABS120k08 is a large research data set about Web metadata in XML format. It contains a lot of data such as URLs of the web documents, number of Open Directory categories from (DOMZ), users who have bookmarked each URL, Google PageRank of the URL, number of del.icio.us most popular tags for this URL, … etc. [18]

In del.icio.us, when a user creates a bookmark for a URL that he/she wants to remember or share with other people, the user can add tags to this bookmark to describe it. The tags can later be used for searching, sharing, and categorizing the bookmarks. Users can add their own tags to the bookmarks pointing to the same URLs independently. This is called collaborative tagging. Different from traditional subject indexing for libraries and scientific literature, which are generated by experts, tags in del.icio.us are generated by creators and consumers of the content with freely chosen keywords rather than selected from a pre-defined term dictionary.

In our work, we need only the *URLs*, *tags*, *Titles* and *categories* organized in a format that WEKA accepts, which is the ARFF format.

The data preprocessing phase accepts the XML file and the desired category as input and picks the URLs that correspond to the specified categories with its tags and titles, and organizes them in ARFF format. Fig. 1 shows the interface of this phase. Fig. 2 shows an example XML file.

The input XML file is usually very large. So in order to accelerate its processing we used an XML splitter to split it to a number of small files and process those files automatically one by one.

After splitting the file into a number of small fragments, each small file is processed as follows:

1. Search for XML tags that have "top_tags" and "inLinks" attributes with value greater than "0".
2. Consider each tag that matches the above condition and search for the category that matches the specified category.

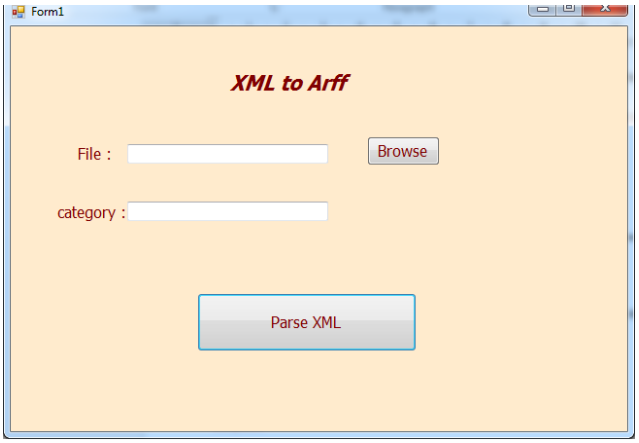

Figure 1. The interface of the data preprocessing phase that converts XML files to ARFF format

3. For each category that matches the specified category, extract from the XML file, the URL, and the tags and titles associated with it.
4. Finally, compose a line for that URL in the following format, which can be understood by WEKA, then append it to the output file:

['Bookmark URL'], ['tag1, tag2, tag3 ...'], ['title1, title2, title3 ...'], [category]

The output of this phase is a file that contains lines in the above format, as shown in the following examples:

'http://www.hittingacademy.com/', 'baseball, hitting, tips, training', 'Original Sports Fan Commentary Since 1998, Sports Central', sports
'http://www.southsidejohnny.com/', 'band,music,jazz', 'the Summer Institute, Classical Music Home, Courtesy of JazzStandards.com', arts
'http://www.nationalbreastcancer.org/', 'awareness, breast, breast cancer, cancer, medicine', 'National Medical Association, NMA, National Breast Cancer Foundation', health

The test set is an ARFF file similar to the training data file except that a question mark is placed in the place of the category name in the data section, as shown in Fig. 3.

B. The Classification Phase

In the Classification Phase, a predictive model is constructed, using one of the classification methods provided by WEKA, such as Decision Tree, Support Vector Machine or Naïve Bayes. The created model is later used to assign labels to web objects.

Instances (web objects) in the dataset that provides the input to the Classification Phase are characterized by a set of features, or attributes. In this case, there are three attributes: URL, tags, and titles. The outcome is the category to which the instance belongs.

```
<document url="http://www.cooldiamonds.com/" users="8" categories="2" searches="9" inlinks="8" top_tags="1" tags="9" pagerank="4">
    <category name="top/shopping/jewelry/diamonds" />
    <search query="diamond rings" aol500k_id="1502786" date="2006-03-08" time="22:52:09" rank="7" />
    <search query="www.conflictdiamonds.com" aol500k_id="11644951" date="2006-04-02" time="19:15:50" rank="8" />
    <search query="designer diamond rings" aol500k_id="3716825" date="2006-03-27" time="02:53:06" rank="1" />
    <search query="diamonds" aol500k_id="9894762" date="2006-03-06" time="20:34:59" rank="10" />
    <search query="size differences with diamonds" aol500k_id="12170639" date="2006-05-23" time="23:39:49" rank="3" />
    <search query="diamonds" aol500k_id="4046236" date="2006-03-22" time="12:23:12" rank="10" />
    <search query="engagement rings" aol500k_id="12079285" date="2006-04-19" time="14:04:52" rank="92" />
    <search query="london jewelers" aol500k_id="1013004" date="2006-04-25" time="05:34:40" rank="4" />
    <search query="designer engagement rings" aol500k_id="6054043" date="2006-05-20" time="11:02:23" rank="8" />
    <inlink anchor_text="Cool Diamond" />
    <inlink anchor_text="www.cooldiamonds.com" />
    <inlink anchor_text="Top Stories" />
    <top_tag name="diamonds" count="5" />
    <bookmark user="rlance" tags="system:unfiled" date="2006-01" />
    <bookmark user="hywel" tags="diamond" date="2006-03" />
    <bookmark user="DA2" tags="diamonds" date="2006-04" />
    <bookmark user="niall_f_oneill" tags="diamonds" date="2006-05" />
    <bookmark user="lou" tags="boolou" date="2007-01" />
    <bookmark user="ashleyiveson" tags="engagement, ring, diamonds, emerald" date="2007-02" />
    <bookmark user="donnadenston" tags="donna" date="2007-05" />
    <bookmark user="davidlethbridge" tags="shopping" date="2007-05" />
</document>
```

Figure 2. XML file example

The Classification Phase applies the following algorithm to create a model:

Create Model Algorithm:

1. Load the training data ARFF file that is created in the data preprocessing phase.
2. Select the feature or attribute that characterize each individual in the training set during the classification process.
3. Select the filter that will be applied (say, StringToWordVector [19])
4. Set the properties for this filter.
5. Select the classifier: Decision Tree, Support Vector Machine or Naïve Bayes.
6. Then, use the FilterdClassifier class, provided by WEKA metalearner algorithms, which wraps the learning algorithm (classifier) into the filtering mechanism to build the classification model (Classifier).
7. Evaluate the model using 10-fold cross validation.
8. The output will be
   a) A model that can be saved and used, and
   b) Summary of the classification process and output.

C. *The Evaluation Phase*

To predict the performance of a classifier on new data, we need to assess its error rate on a dataset that played no part in the formation of the classifier. This independent dataset is called the test set. We assume that both the training data and the test data are representative samples of the underlying problem.

In the Evaluation Phase, we either apply the current model, which is just created by the Classification Phase, on the test set, or load a previously created and saved model, then apply it on the test set.

The Evaluation Phase uses the following algorithm to apply a model to the test data:

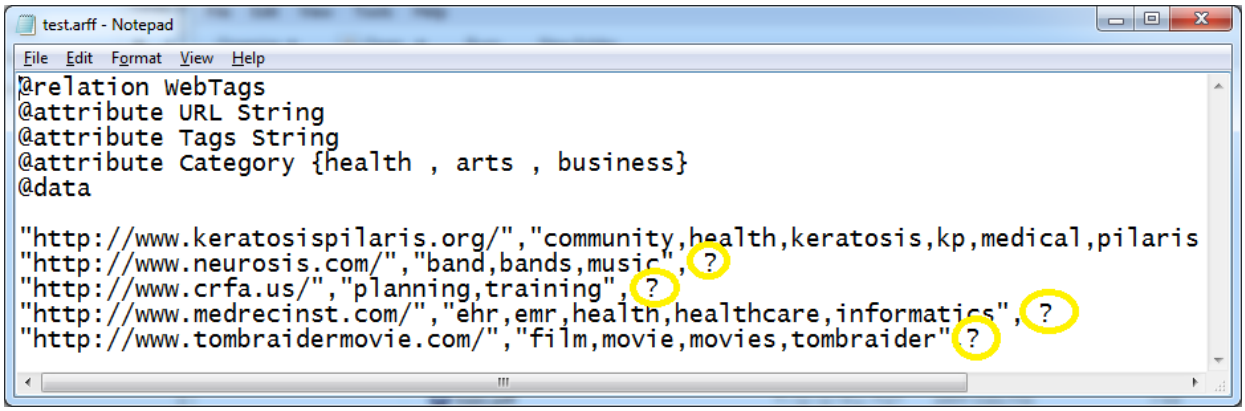

Figure 3. The test set ARFF file

Apply Model Algorithm:

1. Load the test data ARFF file that is created in the data preprocessing phase.
2. Load the model to be applied and serialize it, if it is not already loaded.
3. Evaluate the model by applying it to the test set.
4. The output will be
   a) ARFF file that represents the test file with the predicted categories.
   b) Summary of the classification process and output.

Fig. 4 shows the interface of the classification and evaluation phases. This interface includes two tabs: *Build Model* tab and *Apply Model* tab. As shown in the figure, the Build Model tab allows the user to select and load the training data file, select the attribute (tag or title) that will be used to characterize each individual in the training set, and select the classifier (DT, SVM or NB) that will be applied. When the Build Model button is pressed, the model is created and saved, and a summary of the classification process is displayed the textbox labeled Result.

Fig. 5 shows the Build Model tab that allows the user to select either to apply the current model, which is just created by the Classification Phase, or load a previously created and saved model. It also allows the user to load the test data file to which the model will be applied. In this case, when the Apply Model button is pressed, the result will be a copy of the test data file in which the question mark next to each instance is replaced by the category predicted by the model for this instance. As an alternative option, the Build Model tab allows the user to provide an individual URL of a web object and the associated tag(s) to determine its corresponding category, as shown in Fig. 6. In this case, when the Apply Model button is pressed, the predicted category for the given URL will be shown in a message box, as shown in Fig. 7.

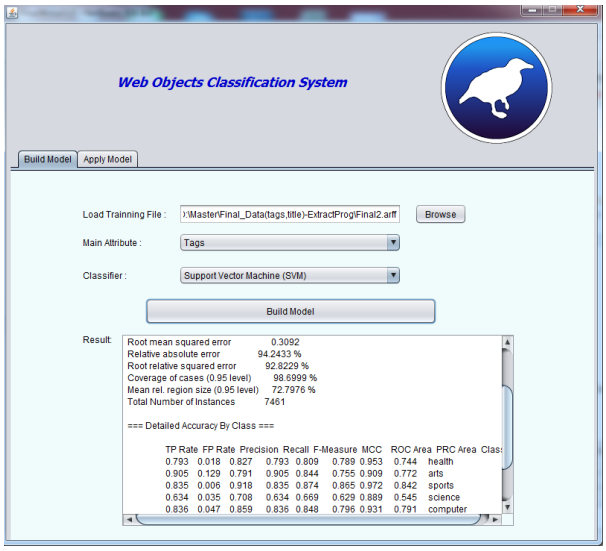

Figure 4. The interface of the classification and evaluation phases showing the contents of the Build Model tab

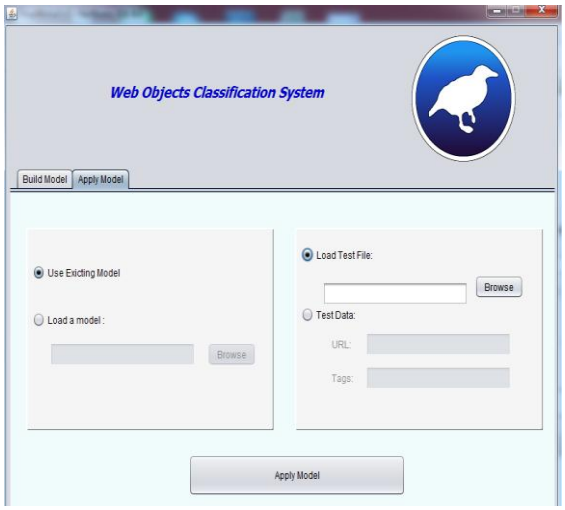

Figure 5. The interface of the classification and evaluation phases showing the contents of the Apply Model tab when the user selects to apply the current model to a test set

VI. EXPERIMENTS

As mentioned in Section 5.1, our data set is taken from http://www.michael-noll.com/cabs120k08/. The entries in the training data set correspond to 7461 web objects (web pages), with 7 categories (health, arts, sports, science, computer, games, kids). This data set is used to generate the classification model. The evaluation is carried out using 10-fold cross validation.

A. Evaluation Methods

We use the standard F1 measure, accuracy, recall, and precision for evaluating the effectiveness of classification results. These measures are defined as follows:

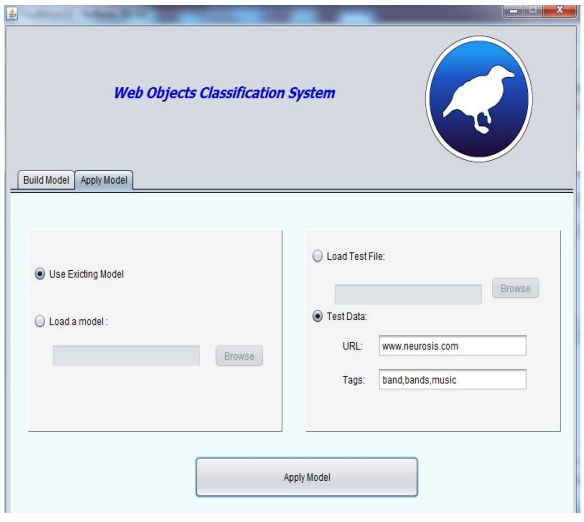

Figure 6. The contents of the Apply Model tab when the user selects to apply the current model to determine the category of an individual URL of a web object

Figure 7. The result of applying the current model to determine the category of a given web object

Accuracy is the percentage of correctly classified instances.

Precision is the proportion of the predicted positive cases that were correct, and is calculated using the equation:

$$\text{Precision} = TP / (TP + FP). \quad (1)$$

Recall or ***Sensitivity*** or ***TP Rate*** is the proportion of positive cases that were correctly identified, and is calculated using the equation:

$$\text{Recall} = TP / (TP + FN). \quad (2)$$

where ***TP*** (True Positives) is the number of items correctly labeled as belonging to the positive class; ***TN*** (True Negatives) is the number of items correctly labeled as not belonging to the positive class; ***FP*** (False Positives) is the number of items incorrectly labeled as belonging to the class; ***FN*** (False Negatives) is the number of items which were not labeled as belonging to the positive class but should have been.

The ***F-measure*** computes some average of the information retrieval precision and recall metrics, and is calculated using the equation:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (3)$$

Note that the computed F-measure values are between 0 and 1 and a larger F-Measure value indicates a higher classification quality. [20]

B. Classification Features and Methods

In order to evaluate the effectiveness of tags in web objects classification, we compared tags with other features. In our experiments, we used the title as another feature space, so web objects can also be represented as sets of words in the titles. We evaluated the effectiveness of tags compared to titles with three classification methods, namely: Decision Tree (DT), Support Vector Machine (SVM) and Naïve Bayes (NB), as follows:

- SVM+TITLE: SVM using titles as feature.
- SVM+TAG: SVM using tags as feature.
- DT+TITLE: Decision Tree using titles as feature
- DT+TAG: Decision Tree using tags as feature
- NB+TITLE: Naïve Bayes using titles as feature
- NB+TAG: Naïve Bayes using tags as feature

C. Experimental Results

Firstly, we have conducted experiments with the 10-fold Cross Validation test option using the developed system. Table I shows a comparison between the classification results of these experiments for different classification methods with tags and with titles. It can be seen from this table that the SVM+Tags gives the highest accuracy and F-measure values. It can be seen also that the accuracy and F-measure values of the three classification methods with tags are higher than with titles. Considering the time taken to build the model, it can be seen that NB took the smallest time, and the time taken by the three methods with tags was smaller than with titles.

Fig. 8 shows a comparison between the precision and recall values for different classification methods with tags and with titles. It can be seen from this figure that the SVM+Tags gives the highest precision and recall values, and NB+Tags gives higher precision than DT+Tags, but DT+Tags gives higher recall than NB+Tags. Fig. 8 shows also that the precision and recall values of the three classification methods with tags are higher than with titles.

TABLE I. A COMPARISON BETWEEN THE CLASSIFICATION RESULTS OF DIFFERENT CLASSIFICATION METHODS WITH TAGS AND WITH TITLES

Classifier	Accuracy	F-measure	Time to build model (sec)
DT+Tags	73.77%	0.725	151.67
DT+Titles	53%	0.513	715.2
SVM+Tags	80.7%	0.799	15.38
SVM+Titles	61.5%	0.597	39.87
NB+Tags	67.3%	0.685	7.8
NB+Titles	46.08%	0.495	8.6

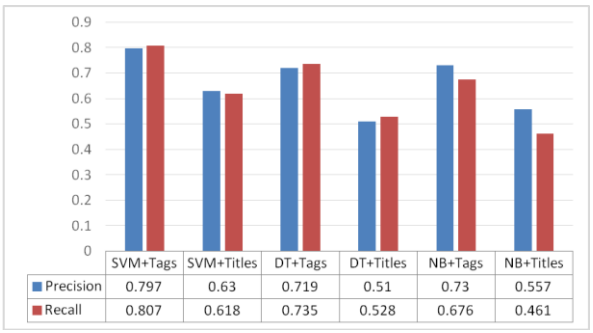

Figure 8. A comparison between precision and recall values for different classification methods with tags and with titles

TABLE II. A COMPARISON BETWEEN THE ACCURACY OF DIFFERENT CLASSIFICATION METHODS WITH DIFFERENT TRAINING DATA PERCENTAGE

Training data percentage (tdp)	SVM		Decision Tree		Naïve Bayes	
	Tags	*Titles*	*Tags*	*Titles*	*Tags*	*Titles*
10%	70.9%	47.9%	68.2%	41.4%	69.3%	42.7%
30%	76.2%	54.9%	70.6%	44.7%	68.4%	42.2%
50%	78.3%	58.2%	72.6%	47.8%	67.5%	44.9%
65%	79.2%	60.2%	71.4%	48.9%	66.3%	46.5%
80%	79%	61.6%	71.9%	53.3%	65.3%	47%
90%	79.4%	63.4%	71.9%	51.6%	62.5%	48%

In order to study the effect of the size of the training set on the effectiveness of the classification results, we have conducted experiments with the percentage split test option through WEKA software. Table II shows a comparison between the accuracy of different classification methods with different training data percentage (tdp). The tdp values used were 10, 30, 50, 65, 80, and 90. It can be seen from this table that the SVM+Tags and SVM+Titles give better accuracy when tdp = 90%, DT+Tags gives better accuracy when tdp = 50%, DT+Titles gives better accuracy when tdp = 80%, and NB+Tags gives better accuracy tdp = 10%, and NB+Titles gives better accuracy when tdp = 90%. All the results in this table indicate that when using tags the classification methods perform better than when using titles for all percentages of the training data used.

Fig. 9 shows the changes in the accuracy of different classification methods with different tdp. It can be seen from this figure that: the accuracy of SVM+Tags and SVM+Titles increases when the tdp increases; The accuracy of DT +Tags increases when the tdp increases up to 50%, and it decreases when the tdp increases up to 65%, then it almost stabilizes; The accuracy of DT+Titles increases when the tdp increases up to 80%, then it decreases; The accuracy of NB+Tags decreases when the tdp increases; and the accuracy of NB+Titles almost increases when the tdp increases. These results indicate that NB+Tags works well with small percentage of training data.

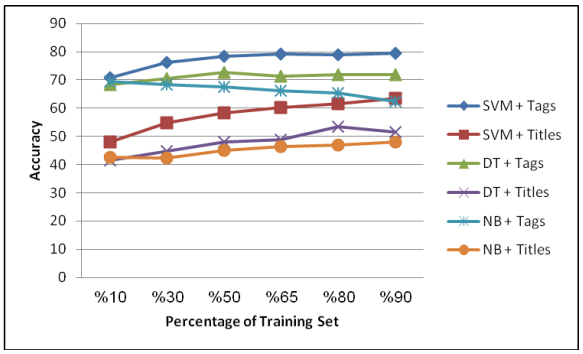

Figure 9. The changes in the accuracy of different classification methods with different training data percentage

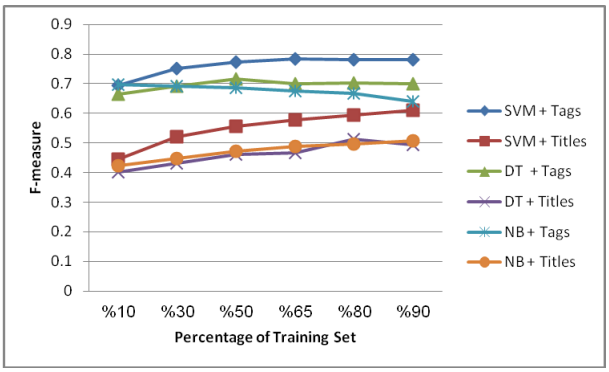

Figure 10. The changes in the F-measure values of different classification methods with different training data percentage

Figures 10, 11 and 12 show that the changes in the F-measure, precision and recall values with the increase in the tdp for all methods are similar to the changes in the accuracy values shown in Fig. 9, respectively.

All the above experimental results indicate that the classification methods perform better when using tags than when using titles. Thus, it can be concluded that tags are very useful features for web object classification task. it can be concluded also that SVM+Tags is best of all.

VII. CONCLUSION

This paper presented a study of the impact of using social tagging on the performance of text classification techniques in web objects classification. The study involved the development of an automated system for web objects classification that is based on social tags exploration. This system has three main phases: the data preprocessing phase, the classification phase and the evaluation phase. It accepts a training dataset that represents a set of web pages with its URLs, tags, titles and categories. Using the provided dataset, the system constructs a predictive model that is later used to assign labels to web objects based on their tags. In the classification task, the system employs the three known text classification techniques namely, Support Vector Machine, Naïve Bayes, and Decision Tree, through the WEKA software. In the evaluation phase, 10-fold cross validation is used.

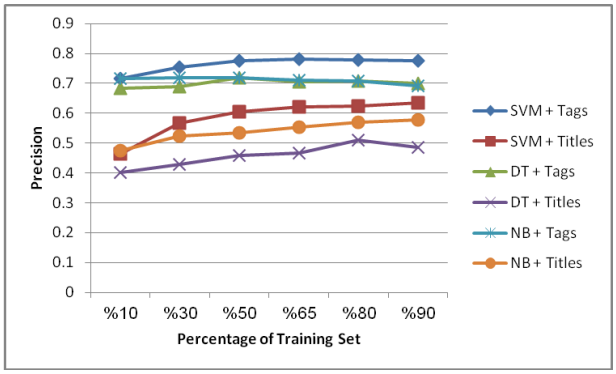

Figure 11. The changes in the precision values of different classification methods with different training data percentage

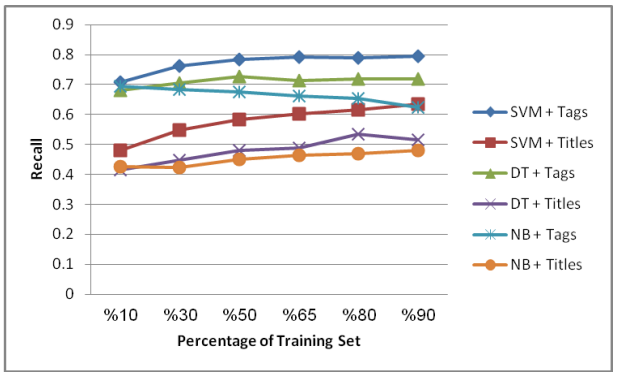

Figure 12. The changes in the recall values of different classification methods with different training data percentage

Experiments have been conducted to evaluate the effectiveness of using social tags with each one of the three text classification techniques in web objects classification. The experimental results indicate that the classification methods perform better when using tags than when using other web objects features, such as titles. The results also indicate that the Support Vector Machine with Tags is the best of all, and Naïve Bayes with Tags works well with small percentage of training data.

REFERENCES

[1] Zhijun Yin, Rui Li, Qiaozhu Mei, and Jiawei Han, "Exploring social tagging graph for web object classification", KDD'09, June 28–July 1, 2009, Paris, France.

[2] Xin Li, Lei Guo , and Yihong (Eric) Zhao, "Tag-based social interest discovery" , In Proceedings of the 17th international conference on World Wide Web (WWW '08), April 21–25, 2008, Beijing, China, pp. 675–684.

[3] R. Ghani, S. Slattery, and Y. Yang, "Hypertext categorization using hyperlink patterns and metadata", In Proceedings of ICML-01, 18th International Conference on Machine Learning, pp. 178–185, 2001.

[4] X. Qi and B. D. Davison, "Web page classification: features and algorithms. acm computing surveys", vol. 41, no. 2, Article 12, 2009.

[5] P. R. Kalva, F. Enembreck, and A. L. Koerich, "Web image classification based on the fusion of image and text classifiers", In Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), vol. 1, 23-26 Sept. 2007, pp. 561–568, Parana.

[6] W.-H. Lin and A. G. Hauptmann, "News video classification using svm-based multimodal classifiers and combination strategies", In Proceedings of the tenth ACM international conference on Multimedia (MULTIMEDIA '02), pp. 323–326, 2002.

[7] S. Chakrabarti, B. Dom, and P. Indyk, "Enhanced hypertext categorization using hyperlinks", In Proceedings of the 1998 ACM SIGMOD international conference on Management of data (SIGMOD '98), pp. 307–318, 1998.

[8] D. Shen, J.-T. Sun, Q. Yang, and Z. Chen, "A comparison of implicit and explicit links for web page classification", In Proceedings of the 15th international conference on World Wide Web (WWW '06), pp. 643–650, 2006.

[9] G.-R. Xue, D. Shen, Q. Yang, H.-J. Zeng, Z. Chen, Y. Yu, W. Xi, and W.-Y. Ma, "IRC: An iterative reinforcement categorization algorithm for interrelated web objects", In Fourth IEEE International Conference on Data Mining (ICDM '04), pp. 273–280, 2004.

[10] S. Bao, G.-R. Xue, X. Wu, Y. Yu, B. Fei, and Z. Su, "Optimizing web search using social annotations, In Proceedings of the 16th international conference on World Wide Web (WWW '07), pp. 501–510, 2007.

[11] P. Heymann, G. Koutrika, and H. Garcia-Molina, "Can social bookmarking improve web search?", In: First ACM International Conference on Web Search and Data Mining (WSDM'08), February 11-12, 2008, Stanford, CA., pp. 195–206.

[12] D. Zhou, J. Bian, S. Zheng, H. Zha, and C. L. Giles, "Exploring social annotations for information retrieval, In Proceedings of the 17th international conference on World Wide Web (WWW '08), pp. 715–724, 2008.

[13] R. Schenkel, T. Crecelius, M. Kacimi, S. Michel, T. Neumann, J. X. Parreira, and G. Weikum, "Efficient top-k querying over social-tagging networks", In Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '08), pp. 523–530, 2008.

[14] X. Wu, L. Zhang, and Y. Yu, "Exploring social annotations for the semantic web", In Proceedings of the 15th international conference on World Wide Web (WWW '06), pp. 417–426, 2006.

[15] C. H. Brooks and N. Montanez, "Improved annotation of the blogosphere via autotagging and hierarchical clustering", In Proceedings of the 15th international conference on World Wide Web (WWW '06), pp. 625–632, 2006.

[16] Gabriel Fiol-Roig, Margaret Miró-Julià, Eduardo Herraiz, "Data Mining Techniques for Web Page Classification", Advances in Intelligent and Soft Computing, vol. 89, 2011, pp 61-68.

[17] Ian H. Witten, Eibe Frank, and Mark A. Hall, Data Mining Practical Machine Learning Tools and Techniques, Third Edition, Morgan Kaufmann Publishers, Elsevier Inc., 2011

[18] Michael G. Noll, "CABS120k08", http://www.michael-noll.com/cabs120k08/ , Last accessed 2/5/2015.

[19] Jose Maria Gomez Hidalgo, "Text Mining in WEKA: Chaining Filters and Classifiers", http://jmgomezhidalgo.blogspot.com/2013/01/text-mining-in-weka-chaining-filters.html, Last accessed 2/5/2015.

[20] "Performance Evaluation", http://www.seas.gwu.edu/~bell/csci243/lectures/performance.pdf , Last accessed 2/5/2015.