Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Multi-Instance Multi-Label Learning For Automatic Tag Recommendation

Chen Shen*†, Jun Jiao‡, Yahui Yang¶ and Bin Wang§
*Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100080, China
†School of Software and Electronics, Peking University, Beijing 100871, China
Email: scv119@gmail.com
‡Department of Computer Science and Technology, Nanjing University, Nanjing 210089, China
Email: failedjj@gmail.com
¶School of Software and Electronics, Peking University, Beijing 100871, China
Email:yhyang@ss.pku.edu.cn
§Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100080, China
Email: wangbin@ict.ac.cn

***Abstract*—Tag services have recently become one of the most popular Internet services on the World Wide Web. Due to the fact that a web page can be associated with multiple tags, previous research on tag recommendation mainly focuses on improving its accuracy or efficiency through multi-label learning algorithms. However, as a web page can also be split into multiple sections and be represented as a bag of instances, multi-instance multi-label learning framework should fit this problem better. In this paper, we improve the performance of tag suggestion by using multi-instance multi-label learning. Each web page is divided into a bag of instances. The experiments on real-word data from Del.icio.us suggest that our framework has better performance than traditional multi-label learning methods on the task of tag recommendation.**

***Index Terms*—multi-instance, multi-label, machine learning, tag recommendation**

I. Introduction

With the rapid development of the Web 2.0 applications such as Del.icio.us, the social bookmark services in which people can store, share and discover web tags have become a popular way to annotate and share web resources. The tagging service provides meaningful descriptors to the web pages which can be edited by not only the administrators but also users all around world. It has drawn much attention from both industry and academia.

Early research on tag recommendation includes exploring the tag space[1], investigating user patterns[3] and automatically generating personalized tags for Web pages[2]. As the task of automatic annotation can be modeled as a single-instance multi-label ranking problem, some researchers began to investigate the implementation of multi-label learning algorithms on tag suggestion. Latest works studied the extended frameworks of multi-label machine learning algorithms such as MMSG[4] and HOMER[5], both of which received significant improvement on tag recommendation. However, these work only regarded each web page as a single instance, which neglect that each web page can be divided into many sections and each section might be inherently associated with certain potential tags. We think that the web page can be represented as a bag of instances if we consider each section as an instance.

Zhou et al[6] proposed a multi-instance multi-label learning framework in which each training example is associated with not only multiple class labels but also multiple instances. Application to scene classification shows that solving some real-world problems in the MIML framework can achieve better performance than solving them in existing frameworks such as multi-instance learning and multi-label learning.

In this paper, we devise a tag recommendation framework based on multi-instance multi-label learning. Our framework is composed of two steps. First, we divide each web page into many sections by using TextTiling document segmentation algorithm. We consider each section as an instance and represent the web page as a bag of instances. The multi-instance multi-label Support Vector Machine(MIMLSVM) is then used to learn the relationship between labels and bags.

The rest of this paper is organized as follows. We first briefly review the related work in Section2. Section 3 introduce our framework to learn the labels for Web documents. In section 4, we demonstrate performance measures and experimental results on real-world tag suggestion problem. Finally, Section 5 concludes and indicates several issues for future work.

II. Related Work

This paper presents a novel approach to generate tags for Web pages by leveraging multi-instance multi-label learning frameworks. There exists a substantial amount of researches related to out work concerning the general goal of creating tags for Web pages from machine learning perspective, as well as document segmentation. The following subsections will discuss some of the important works in the research area of tag suggestion, Multi-Label Multi-Instance learning and document segmentation.

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

A. Automatic Tag Recommendation

Tag suggestion is a complicated problem which can be addressed in many aspects. Collaborative filtering[3] can be applied to suggest tags from users who share similar tagging behaviors. This method mines the usage patterns from current users and store a look-up table of users behavior similarity in advance. On the other hand, Chirita et all[2] proposed a method which automatically generates personalized tags by analyzing personal Desktop for personal interest and aligning keyword candidates for a given Web pages. As this method focused on personalized tags, every document stored on the users computer are regarded as personal interest corpus and exploited for personalized tag suggestion. Based on the assumption that users will annotate same tags to the similar Web pages, the other tag suggestion frameworks [1][5][4] try to find candidate tags from similar documents. Among them, Song et al[4] proposed a novel multi-label classification algorithm MMSG which transforms the tag suggestion problem into multi-label ranking problem. More over, Katakis et al[5] also models the tag suggestion as a multi-label text classification task.

B. Multi-Label Multi-Instance Learning

Multi-instance learning (or multi-instance single-label learning, MISL) coined by Dietterich et al [7]studies the problem where a training example represented by a number of instances is associated with one class label. Formally, the task of MISL is to learn a function $f_{MISL} : 2^{\mathcal{X}} \rightarrow \{-1, +1\}$ from a given data set $\{(X_1, y_1), (X_2, y_2), ..., (X_m, y_m)\}$, where $X_i \subseteq \mathcal{X}$ is a bag of instances $\{x_1^i, x_2^i, ..., x_n^i\}$ and $y_i \in \{+1, -1\}$ is the binary label of X_i.

On the contrary, multi-label learning (or single-instance multi-label learning, SIML) [10] studies the problem where a training example represented by one instance is associated with a number of class labels. The task of SIML is to learn a function $f_{SIML} : \mathcal{X} \rightarrow 2^{\mathcal{Y}}$ from a given data set $\{(x_1, Y_1), (x_2, Y_2), ..., (x_m, Y_m)\}$, where $x_i \in \mathcal{X}$ is a instance and $Y_i \subseteq \mathcal{Y}$ is a set of labels $\{y_1^i, y_2^i, ..., y_n^i\}$ associated with x_i.

This two learning frameworks address the ambiguity in representing real-world objects in machine learning perspective, and both have been successfully applied to numerous applications[7][8][10]. However, Zhou et al indicated that there exists many real-world objects associated with multiple instances and multiple labels simultaneously which do not fit these two multi- learning framework well[6]. In order to tackle such problem, Zhou et al[6] first proposed the multi-instance multi-label learning framework in which a training example is presented by multiple instances and associated with multiple class labels. In fact, we can treat the traditional supervised learning as a degenerated version of multi-instance learning as well as a degenerated version of multi-label learning. These three learning can be further recognized as degenerated versions of multi-label multi-instance learning[6] as Figure 1 illustrates.

Formally, the task of multi-instance multi-label learning is to learn a function $f_{MIML} : 2^{\mathcal{X}} \rightarrow 2^{\mathcal{Y}}$ from a given data set $\{(X_1, Y_1), (X_2, Y_2), ..., (X_m, Y_m)\}$, where $X_i \subseteq \mathcal{X}$ is a set of instances $\{x_1^i, x_2^i, ..., x_n^i\}$ and $Y_i \subseteq \mathcal{Y}$ is a set of labels $\{y_1^i, y_2^i, ..., y_l^i\}$ associated with X_i. Zhou et al [6]also proposed and applied two MIML algorithms–MIMLBOOST and MIMLSVM on the problem of scene classification. Their experiment on the the scene classification showed that the MIML framework achieved better performance than both multi-instance learning and multi-label learning[6].

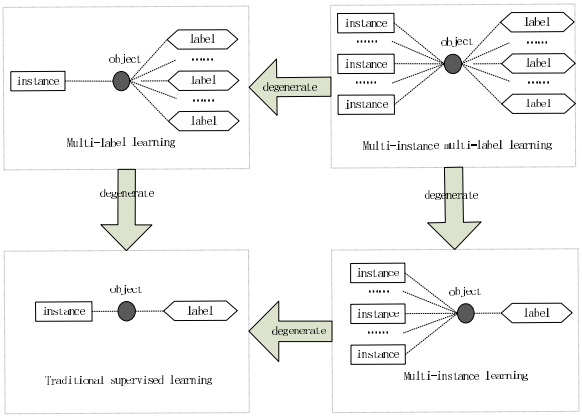

Fig. 1. Four different learning frameworks

C. Document Segmentation

Document segmentation is an important algorithm which has been applied in many natural language processing tasks such as information retrieval and summarization[12]. A basic document segmentation approach is to break down documents into blocks each of which contains same number of words. However, as the documents usually include various topics and each contains different number of sentences, it is more appropriate to divide documents by identifying and isolating topics. A lot of research has been done on text segmentation[11][12][13]. Beeferman et al[11] introduced a statistical approach to automatically partition text into related segments. In their method, a exponential model was built incrementally in order to extract features that are correlated with the presence of boundaries in labeled training text. Another classical algorithm, TextTiling [13], is a straight-forward algorithm that assigns a depthscore to each topic boundary candidate between two blocks based on a cosine similarity measure. If the depthscore of a boundary candidate exceeds a given threshold, the two blocks are regarded as belonging to different topics.

III. The Proposed Framework

In this section, we describe our framework which automatic suggest tags for a Web page in details. First, each web page is segmented into many parts. Then, MIMLSVM is engaged to learn the tags for web pages.

A. Web Page Segmentation

The TextTiling algorithm is an classical document segmentation algorithm which was firstly applied to topic segmentation in news[13]. We adapt this algorithm to our Web page segmentation task. as follows:

1) Similarity Score Calculation: For each Web page, we first remove all the HTML tags to obtain the text content. Then, the text content of Web pages are divided into sentences. We treat each inter-sentence gap along the text stream as a topic boundary candidate. At each inter-sentence gap, neighboring windows containing fixed number of sentences are used to compute the semantic similarity. The similarity score between adjacent windows at gap g is calculated by cosine metrics:

$$\begin{aligned} sim(g) &= cos(v1, v2) \\ &= \frac{\sum_{i=1}^{n} w_{i,v1} w_{i,v2}}{\sqrt{\sum_{i=1}^{n} w_{i,v1}^2 \sum_{i=1}^{n} w_{i,v2}^2}} \end{aligned} \tag{1}$$

where $v1$ and $v2$ are the term frequency vectors for the two adjacent windows to the inter window gap. i ranges over all the terms in the document and $w_{i,v1}$ and $w_{i,v2}$ are the tf-idf weight assigned to term i in $v1$ and $v2$ respectively.

2) Boundary Prediction: To identify whether a inter-sentence gap is a topic boundary, we follow [13] and only consider the gaps whose similarity value represent a "valley". The "valley" gaps are those gaps whose similarity value are less than those at the immediately preceding and following gaps. Specifically, the $depthscore$ is calculated for each gap g using the following equation:

$$\begin{aligned} depthscore(g) = &\max\{(sim(p) - sim(g)), 0\} \\ &+ \max\{(sim(f) - sim(g)), 0\} \end{aligned} \tag{2}$$

where p and f are the immediately preceding and following inter-sentence gaps of g. Note that the $depthscore$ only consider the drop in similarity and the non-valley gap is given a score of 0. From this formula we can see that a sharp drop in similarity value is more indicative of a boundary than a relatively gentler drop, in the case that the former's similarity may be higher then the latter's. With the depths scores computed above, we calculate a threshold value t as in [13] by subtracting the standard deviation σ from the mean $\mathcal{S}$ of all the depth scores:

$$t = \mathcal{S} - \sigma \tag{3}$$

With the threshold t, we consider the gaps whose depth scores are greater than t as the final topic boundaries. At last, the text content of each Web page is divided into sections by these predicted boundaries.

B. Learning with MIML

As each Web page is divided into sections and associated with several labels as well, we leverage the MIML learning framework to learn the relation between tags and Web pages. In this paper, we train a classifier based on the MIMLSVM algorithm proposed by Zhou et al[6] , which uses multi-label learning as the bridge.

In particular, as we mentioned in Section 2, the MIML learning task is to learn a function $f_{MIML} : 2^{\mathcal{X}} \to 2^{\mathcal{Y}}$ from a given data set $\{(X_1, Y_1), (X_2, Y_2), ..., (X_m, Y_m)\}$. Following [6], the MIMLSVM uses k-medoids clustering as a mapping ϕ to transform the multi-instance bags into traditional single-instances. Formally, with the $\phi : 2^{\mathcal{X}} \to \mathcal{Z}$, the former MIML learning task can be transformed into multi-label learning task which tries to learn a function $f_{SIML} : \mathcal{Z} \to 2^{\mathcal{Y}}$ where for any $z_i \in \mathcal{Z}$, $f_{SIML}(z_i) = f_{MIML}(X_i)$ if $z_i = \phi(X_i)$. Here the Multi-label SVM[6] is used to implement the function f_{SIML}.

Given the MIML training examples (X_i, Y_i) and the mapping $z_i = \phi(X_i)$ where $\phi : 2^{\mathcal{X}} \to \mathcal{Z}$, for any $y \in \mathcal{Y}$, let $\Phi(z_i, y) = +1$ if $y \in Y_i$ and -1 otherwise, where Φ is a function $\Phi : \mathcal{Z} \times \mathcal{Y} \to -1, +1$. The detailed pseudo-code of the MIMLSVM algorithm is shown in Algorithm 1.

Algorithm 1 MIMLSVM algorithm

Training Stage

For MIML examples $(X_u, Y_u)(u = 1, 2, ..., m)$,collect all X_u in a data set $\Gamma = \{X_u | u = 1, 2, ..., m\}$
Randomly select k elements from Γ to initialize the medoids $M_t(t = 1, 2, ..., k)$
repeat
  $\Gamma_t = M_t(t = 1, 2, .., k)$
  for each $X_u \in (\Gamma - M_t | t = 1, 2, ..., k)$ **do**
    $index = \arg\min_{t \in (1,...,k)} d_H(X_u, M_t)$
    $\Gamma_{index} = \Gamma_{index} \cup X_u$
  end for
  $M_t = \arg\min \sum_{A \in \Gamma_t} \sum_{B \in \Gamma_t} d_H(A, B)(t - 1, 2, ..., k)$
until all M_t do not change
Transform (X_u, Y_u) into a multi-label example $(z_u, Y_u)(u = 1, 2, ..., m)$ where $zu = (d_H(X_u, M_1), d_H(X_u, M_2), ..., d_H(X_u, M_k))$
for each $y \in \mathcal{Y}$ **do**
  Derive a data set $\mathcal{D}_y = \{(z_u, \Phi(z_u, y)) | u = 1, 2, ..., m\}$, and train an SVM $h_y = SVMTrain(\mathcal{D}_y)$
end for

Predicting Stage

Transform the given X^* into $z^* = (d_H(X^*, M_1), d_H(X^*, M_2), ..., d_H(X^*, M_k))$
Get a score set $S = \{s_y | s_y = h_y(z^*), y \in \mathcal{Y}\}$, and Return a ranked list by sort the all $s_y \in S$ in descending order.

As we can see from the Algorithm 1, the MIMLSVM first collects all the input of training examples, and then uses constructive clustering to cluster all the inputs into k medoids. While clustering, the Hausdorff distance[16] is used to measure the distance between two bags. With the help of these k medoids, the original multi-instance example X_u is transformed into a k dimensional numerical vector z_u, and the original MIML problem is transformed into a SIML problem.

In the predicting stage, the mulit-instance example is also transformed into single instance by calculating the Hausdorff

distance between the k medoids. Then, as we need to get a ranked lists of labels for the example, all the class labels with SVM scores are sorted in descending order.

IV. Experimental Results

We evaluate our proposal by conducting experiments in the real-world data collected from del.icio.us. Details of the experiments are omitted in the following subsections.

A. Experiment Set

1) Evaluation Metrics: To evaluate the performance of our tag recommendation framework, we measure the quality of the produced tags using *Tag-Precision*, *Tag-Recall*, *Top-k Accuracy* and *Exact-k Accuracy* all which were previously used in [14].

Tag-Precision: Tag-Precision is the percentage of correctly recommended tags among all tags recommended by the tag recommendation algorithm.

Tag-Recall: Tag-Recall is the percentage of correctly recommended tags among all tags annotated by the users.

Top-k Accuracy:Top-k Accuracy is the percentage of Web pages correctly annotated by at least one of the top k_{th} returned tags.

Exact-k Accuracy: Exact-k Accuracy is the percentage of Web pages correctly annotated by the k_{th} recommended tags.

2) Comparison Methods: In our experiment, we compare our framework with three multi-label learning algorithms, Binary Relevance, Label Powerset[17] and Multi-Label K-Nearest Neighbor(ML-KNN)[18], all of which have been successfully applied in multi-label classification tasks.

The Binary Relevance(BR Learning)[17] is the most widely-used problem transformation method which considers the multi-label problem as a set of independent binary classification task on each label. In our experiment, we uses Naive Bayes classifier as the binary learner for each class.

For a given unseen instance, the ML-KNN algorithm uses the statistical information gained from k nearest neighbors and utilizes MAP principle to determine the label [18]. For ML-kNN, the Parameter k are tuned between 10, 30 and 100 and we picked k as 10 which achieved the best performance.

The Label Powerset[17] algorithm treats the the multiple labels of each instance as a subset of the label's powerset. Comparing to BR, the advantage of the Label Powerset is that the learning process takes the correlations between labels into consideration.

B. Result and Analysis on Del.icio.us Data

In order to get real-world tagging data, we subscribed 20 popular tags on the del.icio.us. From these tags, we received 4914 URLs from Feb 1st, 2009 to Feb 8, 2009. We crawled the textual data of web pages and obtained their tags for each URL. Finally, we got 9847 distinct tags for all Web pages. To discharge the data set from rarely used tags, we removed tags used in less than 5 bookmarks. This led to a final set of 843 tags. Then, we randomly selected 1000 Web pages from the 4914 Web pages as the testing data set and the rest were used as training data. In the first-stage of

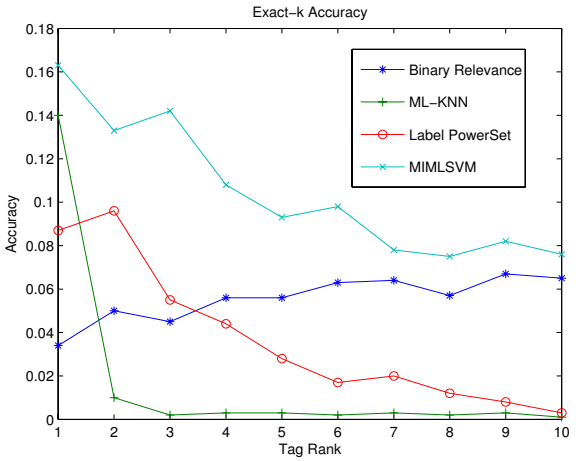

Fig. 2. Exact-k of Del.icio.us data set with 1000 test Web pages, 10 tags are recommended for each Web page

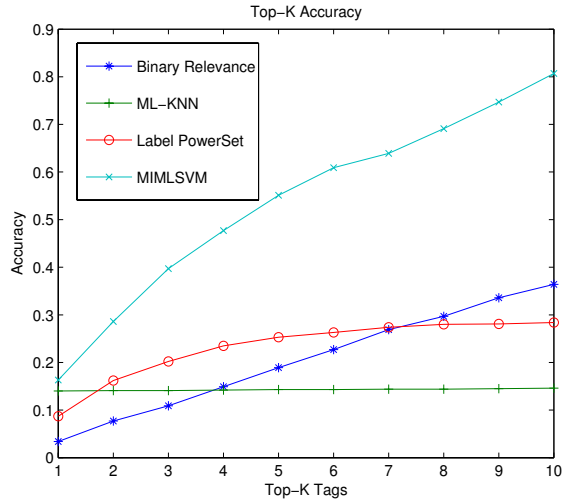

Fig. 3. Top-k of Del.icio.us data set with 1000 test Web pages, 10 tags are recommended for each Web page

our framework, the textual content of each Web pages was divided sections by the TextTiling algorithm. We used "Bag of Words" representation base on tf.idf to present each section. It is reported that dimensionality reduction by retaining the top 2% words with highest document frequency will not harm the effectiveness[15]. Thus we finally use 300 terms to get a 300-dimensional feature vector to present text context by removing stop words and performing dimensionality reduction.

Figure 2 shows the Exact-k precision of MIML, BR, LP and ML-KNN on the 1st to 10th recommended tags, from which we notice that MIML dominates the other algorithms in each top-k tag. As we can see in Figure 2, the correct

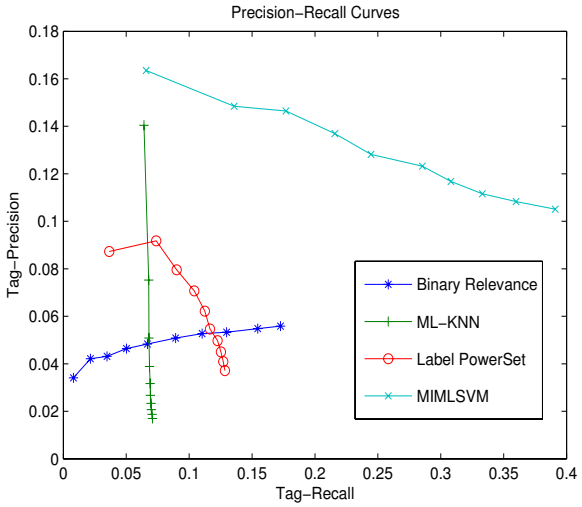

Fig. 4. P-R Curve of Del.icio.us data set with 1000 test Web pages, 10 tags are recommended for each Web page

recommendation on the top 1st tags is around 16.3% achieved by MIML, which is pretty slow, but the same metric on BR, LP and ML-KNN are around 14.0%,8.7% and 3.3%. We assume the reason for the poor accuracy is that the size of training data set is not sufficient to generalize the positive labels as most of the labels are negative. Although the exact-k precision is far from good, the top 10 recommended tags from MIML together are still able to overlap with more than 80% of the user-annotated tags whereas that of the other three algorithm is below 40%.

More over, the Precision and Recall of the MIML tag recommendation framework are significantly above these of other three algorithm, as is shown in Figure 4. With the number of tags increases from 1 to 10, the Precision drops from 16.3% to 10.5% while Recall arises from 6.5% to 39.1%.

V. Conclusion

In this paper, we proposed a learning framework which leverage multi-instance multi-label learning algorithm to tackle the task of automatic tag recommendation. In particular, we used TextTiling algorithm to divide the Web pages into bags of instances representation, and utilized MIMLSVM to learn the transformed MIML problem. We also generated realtime tag suggestions for Del.icio.us and conducted numerous experiments, which suggested that the our MIML framework outperformed the multi-label learning algorithm.

Future work would be to improve the accuracy of our algorithm for the top-most recommended tags. We assume that the segmentation algorithm we used to divide a Web page into a bag of instances neglects the connections between each section, which may lose some information. More over, the efficiency measure should be taken into consideration in the future work.

Acknowledgment

This work is supported by the the Natural Science Foundation (NSF) of China under grant No. 60603094, the NSF of Beijing under grant No. 4082030, the Major State asic Research Project of China under grant No. 2007CB311103 and the National High Technology Research and Development Program of China under grant No. 2006AA010105

References

[1] G. Begelman, P.Keller, and F.Smadja. Automated tag clustering: Improving search and exploration in the tag space. In *Collaborative Web Tagging Workshop at WWW2006*, Edinburgh, Scotland, 2006.

[2] P. A. Chirita, S. Costache, W. Nejdl, and S. Handschuh. P-tag: large scale automatic generation of personalized annotation tags for the web. In *WWW 07: Proceedings of the 16th international conference on World Wide Web*, pages 845C854, New York, NY, USA, 2007. ACM Press.

[3] S. Golder and B. Huberman. Usage patterns of collaborative tagging systems. *J. Inf. Sci.*, 2006.

[4] Song Yang and Zhang Lu and Giles C. Lee,. A sparse gaussian processes classification framework for fast tag suggestions. *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, New York, NY, USA, 2008.

[5] G. Tsoumakas, I. Katakis, and I. Vlahavas.Effective and efficientmultilabel classification in domains with large number of labels. In *Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data*

[6] Zhou, Z.-H., and Zhang, M.-L. 2007. Multi-instance multi-label learning with application to scene classification. *Proc. Advances in Neural Information Processing Systems*

[7] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Perez. Solving the multiple-instance problem with axisparallel rectangles. *Artificial Intelligence*, 89(1-2):31C71, 1997.

[8] O. Maron and A. L. Ratan. Multiple-instance learning for natural scene classification. In *Proceedings of the 15th International Conference on Machine Learning*, pages 341C349, Madison, MI, 1998.

[9] Z.-H. Zhou. Multi-instance learning: A survey. Technical report, AI Lab, Department of Computer Science & Technology, Nanjing University, Nanjing, China, 2004.

[10] G. Tsoumakas and I. Katakis. Multi-label classification: an overview. *International Journal of Data Warehousing and Mining*, 3(3):1C13, 2007.

[11] Doug Beeferman, Adam Berger, and John Lafferty. 1999. Statistical models for text segmentation. *Machine Learning*, 34(1-3):177C210.

[12] Masao Utiyama, Hitoshi Isahara. A Statistical Model for Domain-Independent Text Segmentation. *The 39th Annual Meeting of the Association for Computational Linguistics and 10th Conference of the European Chapter of the Association for Computational Linguistics*. 2001. 491-498.

[13] Marti A. Hearst. 1994. Multi-paragraph segmentation of expository text. *In Proc. of ACL94*.

[14] Y. Song, Z. Zhuang, H. Li, Q. Zhao, J. Li, W. C. Lee,and C. L. Giles. Real-time automatic tagrecommendation. In *SIGIR 08: Proceedings of the31st annual international ACM SIGIR conference onResearch and development in information retrieval*,pages 515C522, New York, NY, USA, 2008. ACM.

[15] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the 14th International Conference on Machine Learning*, pages 412C 420, Nashville, TN, 1997.

[16] G. A. Edgar. *Measure, Topology, and Fractal Geometry*. Springer, Berlin, 1990.

[17] G. Tsoumakas and I. Vlahavas. Random k-labelsets:An ensemble method for multilabel classification. In *Proceedings of the 18th European Conference on Ma-chine Learning (ECML 2007)*, pages 406C417, Warsaw,Poland, September 17-21 2007.

[18] Zhang, M.L., Zhou, Z.H.: A k-nearest neighbor based algorithm for multi-label classication. In *Proceedings of the 1st IEEE International Conference on Gran- ular Computing*. (2005) 718-721