Int. J. Mach. Learn. & Cyber.
DOI 10.1007/s13042-013-0203-3

ORIGINAL ARTICLE

Hierarchical classification with a topic taxonomy via LDA

Li He · Yan Jia · Zhaoyun Ding · Weihong Han

Received: 23 May 2013 / Accepted: 21 September 2013

Abstract Large scale hierarchical classification problem researches how to classify documents into a predefined taxonomy with thousands of categories. As the skewed category distribution over documents, that is, most categories have very few labeled documents, the data sparseness problem in the rare categories lead to a low classification performance. In this paper, we study the problem of web-page classification over the topic taxonomy of the DMOZ directory. For this hard task, we proposed a hierarchical classification model based on Latent Dirichlet allocation (LDA). We use LDA model as the feature extraction technique to extract latent topics to reduce the effects of data sparseness, and construct topic feature vectors associated with the corpus for training more robust classification models for rare categories. Experiments were conducted on the dataset of web pages from the Chinese Simplified branch of the DMOZ directory. The results show that our method achieves a performance improvement for rare categories over the hierarchical classification methods based on full-term and feature-word, and further improves the performance over the whole topic taxonomy.

This work was supported by the National High Technology Research and Development Program of China (No. 2010AA012505, 2011AA010702, 2012AA01A401 and 2012AA01A402), Chinese National Science Foundation (No. 60933005, 91124002, 61303265), National Technology Support Foundation (No. 2012BAH38B04) and National 242 Foundation (No. 2011A010).

L. He (✉) · Y. Jia · Z. Ding · W. Han
School of Computer Science, National University of Defense Technology, Changsha, China
e-mail: lihe@nudt.edu.cn

Keywords Text categorization · Hierarchical classification · Topic taxonomy · Latent dirichlet allocation (LDA) · Rare category

1 Introduction

With the development of information technology, Internet data and electronic data grow rapidly. In order to effectively organize and manage the massive Web information, a large scale class hierarchy of concepts or topics was used to label the web information to make information access easier, such as ODP[1] and the Yahoo! Directory[2]. The hierarchy is usually satisfied the partial order relation, typically a tree or a directed acyclic graph (DAG). Its scale is large that can reach thousands or even tens of thousands of categories. The large scale hierarchical classification problem researches how to classify the web documents into the categories among the class hierarchy using machine learning approach. Besides building a network resource directory, large scale hierarchical classification can also be applied to information retrieval, network resource management, green Internet, network reputation management, hazardous information filtering etc.

Lots of categories in the web taxonomy have very few samples, called rare categories. Rare categories are very common in the web taxonomy, such as ODP and the Yahoo! Directory. And about 70 % of the categories have no more than ten samples. Because rare category has too few instances, existing machine learning algorithm cannot learn effective models for it. For example, Liu et al. [11] found that there were 76 % of the categories in the Yahoo!

[1] http://www.dmoz.org/
[2] http://dir.yahoo.com/

Published online: 02 October 2013

Directory had <5 instances, and the performance of the SVM classifiers training for rare categories were very poor. For this regard, some methods [15, 17] try to increase the training samples of rare category using neighbors in the taxonomy. These methods take the instances of rare category's neighbors as the training samples of rare category. They use the class path or sub-tree where rare category located representing the rare category. Because its own instances are scare, this leads to rare category submerging to the class path or the sub-tree where it located. And this eventually leads to the prediction drift [9]. Even though rare categories are the majority in the taxonomy, there are still quite part of normal categories. Therefore, using the instance expansion strategy on the whole dataset is unreasonable, and this may be another reason causing errors. In addition to the instance expansion strategy, Marath [14] adopted the classifier of parent category to classify for rare category. It means that the prediction will be end when the given instance is classified to the parent node of the rare category in the hierarchy. Thus this is actually an incomplete rough classification. Obviously, classification of rare categories in the topic taxonomy should be solved further.

Latent dirichlet allocation (LDA) [2] is a unsupervised algorithm that models each document as a mixture of topics. The model generates automatic summaries of topics in terms of a discrete probability distribution over words for each topic, and further infers per-document discrete distributions over topics. LDA can be used to extract latent topics from document as a feature extraction method, convert documents from word-feature space to topic-feature space. If the topics learned by LDA better reflect the characteristic of a category and closely relate with the category-topic, for example, the latent topics are some subtopics of the category-topic, then we would expect that classification over topic-feature space should be easier for high-dimensional text. Latent topics cannot represent the distinction between various categories sometimes, Blei [1] proposed a supervised LDA for this. However, in the topic-taxonomy classification problem, the latent topics should be consistent with category-topics semantically. Therefore, in this paper, we use LDA to extract latent topics for each document, and construct a latent topics-document matrix associated with the corpus for training classifiers.

2 Related work

Instance expansion strategy is usually used for rare categories in large scale hierarchical classification. For the regard, Xue et al. [17] proposed a deep classification model, which consists of two stages: search and classification. In the first stage, a searching method is used to find the category candidates for the given document; the large hierarchy was pruned into a narrow one. In the second stage, classification is focused on a subset of categories, which are highly related to the given document. Based on this framework, Xue proposed an Ancestor-Assistant Strategy for rare category, which expands sample collection by the samples belonged to ancestors for each category. Oh et al. [15] further increase the sample extension scope, called Neighbor-Assistant Strategy, which expands the sample collection for each category by more neighbors, including ancestors, children and siblings. Both the two instance expansion strategies are increasing training instances of each category by neighbors in the taxonomy for training classifiers. Therefore, the essence of this approach is to discover the class path or sub-tree most similar to the given document. The samples of rare category are so few that rare category is very likely to be drowning in the class path or the sub-tree, which makes it difficult to improve the classification performance.

Beside the instance expansion method, Neighbor Based Approach is another common method. Adjacent categories are usually topical related in the taxonomy: vertical, categories on the same class path have derived relations; laterally, categories with the same parent are topical similar. In this regard, correlations between neighbors are used to help prediction. Neighbor based approach would alleviate over fitting problem of rare category to some extent by using adjacent categories. He et al. [9] make a systematic description for the method. They divide neighbors into three kinds: ancestors, descendants and siblings. For a given document, the similarity value between the document and a category is calculated by the similarity values between the document and the category's neighbors.

Using ancestors help classification decisions is a commonly used strategy. For example, Oh [15] proposed a naive Bayes combining both local and global information classifier, which utilizing global information from the top of the hierarchy help deciding which path would be more fruitful when there are more than two sub-trees to check. Gopal [7] proposed a set of Bayesian methods to model hierarchical dependencies among class labels, the parent–child relationships are modeled by placing a hierarchical prior over the children nodes centered around the parameters of their parents.

Another interesting line of related research aims to reduce the effects of scarcity of data by reducing the dimensions of the text content features of web documents. For example, Gomez et al. [6] proposed a feature extraction technique called Stratified Discriminant Analysis (sDA) that reduces the dimensions of the features of the web documents along the different levels of the hierarchy. The sDA model is intended to reduce the effects of scarcity of data by grouping and to identify the categories with few training examples.

3 LDA-based hierarchical classification model

Either expanding the instances collection for rare category, or utilizing neighbors to help predict, by its very nature is using the hierarchy information. However, these approaches may not enhance classification effectively according to previous related research so far [13]. Therefore, according to the characteristics of topic classification, we use LDA mining latent topics for each document, convert document-word matrix to document-topic matrix, and learn models over latent topics. Different to the existing methods, we attempt to reduce the data sparseness problem of rare categories by extracting the the topic features of the web documents. As a category usually contains a series of sub-topics in web taxonomy, thus the topics might better reflect the features of the web documents. And SVM is used to train classifiers to conquer the instance shortage problem of rare category.

3.1 Latent dirichlet allocation

The LDA model is a generative process where each document in the text corpus is modeled as a set of draws from a mixture distribution over a set of hidden topics. A topic is modeled as a probability distribution over words. The generative process for this vector is illustrated in Fig. 1.

Then the process of generating a corpus is as follows:

1. For each topic $k \in [1, K]$, choose a multinomial distribution ϕ_k from a Dirichlet distribution with parameter β, $\phi_k \sim \mathrm{Dir}(\beta)$;
2. For each document $d_m (m \in [1, M])$, choose a mutlinormial distribution θ_m from a Dirichlet distribution with parameter α, $\theta_m \sim \mathrm{Dir}(\alpha)$;
3. For each word token in document d_m, choose a latent topic $z_{m,n} \in [1, K]$ from the multinomial distribution θ_m, $z_{m,n} \sim \mathrm{Multi}(\theta_m)$;
4. Generate the word token $w_{m,n}$ from the V-dimensional multinormial distribution $\phi_{z_{m,n}}, w_{m,n} \sim Multi(\phi_{z_{m,n}})$, where V is the size of the vocabulary, $n \in [1, N_m]$.

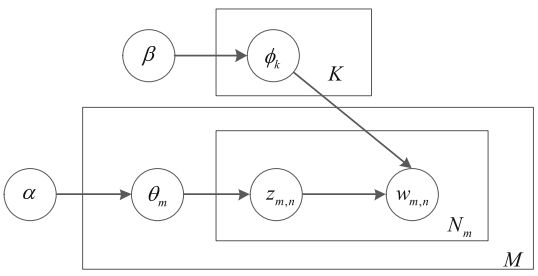

Fig. 1 Graphical model representation of LDA. K is the number of topics; M is the number of documents; N_m is the length of the m-th document; θ_m is the topic probability distribution of the m-th document; ϕ_k is the word probability distribution of the topic k; $w_{m,n}$ is the n-th word in the m-th document, and $z_{m,n}$ is the word's topic

Thus, the likelihood of generating corpus $\{d_m\}_{m=1}^{M}$ is $\prod_{m=1}^{M} \int p(\theta_m|\alpha)(\prod_{n=1}^{N_m} \sum_{z_{m,n}} p(z_{m,n}|\theta_m)p(w_{m,n}|z_{m,n}, \beta))d\theta_m$.

Given a set of documents, the principal task is to estimate the parameters $\{\phi_k\}_{k=1}^{K}$. This can be done by maximum likelihood, $\Phi^* = \arg\max_{\Phi} p(\{d_m\}; \Phi)$, where $\Phi \in \Re^{V \times K}$ is a matrix parameter whose columns $\{\phi_k\}_{k=1}^{K}$ are constrained to be members of a probability simplex. It is possible to treat the parameter Φ as well as the hyperparameters via Bayesian methods. In both the maximum likelihood and Bayesian framework it is necessary to integrate over θ_m to obtain the marginal likelihood, and this is accomplished either using variation inference or Gibbs sampling [2, 8].

3.2 Hierarchical classification via LDA

Because the topic of a category usually contains a series of sub-topics in web taxonomy, for example, the category of basketball includes the sub-topics of NBA, CBA, etc. Thus the latent topics might better reflect the topic features of a document. That's why we choose the LDA feature extraction technique. As the children nodes of a category can be seen as a series of sub-topics below the parent category, thus we extract latent topics in the parent category for all documents belong this parent category. And we train a binary classifier for each child category according to the latent topics. Support vector machines (SVM) provides a good out-of-sample generalization, and is less overfitting to noise. Thus SVM is suitable for dealing with the small samples classification problem of rare category. SVM has been broadly applied in machine learning and pattern recognition, such as semi-supervised classification [3], tremor canceling in microsurgery [12], separating hyper-planes in Banach space [10] and multi-category data classification [16]. We use LibLinear [5] which is very efficient for training large-scale problems as the binary classifier model.

In this paper, we use a top-down approach for training/testing in the classification. Top-down approach uses divide-and-conquer strategy to decompose a large scale global classification problem into a group of small scale local classification problems according to the hierarchy. It learns classification model respectively, finally classifying document from top to down. Therefore, top-down approach would alleviate the class imbalance problem in the training process of rare category to some extent.

The general HCL (Hierarchical Classification via LDA) training process consists of two stages: feature extraction and classifier training. In the first stage, with the LDA feature extraction technique we intend to convert document-word matrix to document-topic matrix at every non-leaf node which are marked as green in Fig. 2. Our aim is

compressing the documents content in less but very meaningful features. In the second stage, top-down approach trains a binary classifier over latent topics in every node of the hierarchy except root node which are marked by dashed squares in Fig. 2. For feature extraction, since we use a top-down approach for training/testing, thus we learn a topic model for each parent node and represent all documents of the node (its descendants) as the topic vectors. We choose the LDA for feature extraction. We illustrate how to control the number of topics in the following. We compare two strategies of controlling the number of topics. The first strategy is setting the number of topics to a constant. The second strategy is setting the number of topics adaptively to the number of children of the current node. And if the number of children is <10, then we set the number of topics to ten. According to the test result, we choose the second strategy. For training, we use the "sibling" policy [4] which is a natural method for top-down approach based on binary classifier. As shown in Fig. 3, we take the documents belonging to a given node (and its descendants) as the positive samples and the examples belonging to its siblings (and their descendants) as the negative samples. The topic representation vectors of documents used for training are generated by the topic model at the parent node of the given node.

When assigning categories to a new document during the testing phase, the HCL model first computes the latent topic vector of the given document in each fired node (predicts positive) of the hierarchy, later a top-down approach using SVM model is employed: first classifying the given document at the uppermost level, and then for each binary classifier that predicts positive, classifying the document at the next lower level, and changing the representation of the given document by using the LDA model of the fired node.

To elaborate the classification process of HCL clearly, we give an example of the predicting process. For each test document, we classify it from to down. For a given document, we first use the LDA model of ROOT to generate the topic vector of the given document, then we classify the document at every children node of ROOT. As shown in Fig. 3, if node1 predicts positive for the given document, first we use the LDA model of node1 to generate the topic vector of the given document, and then classifying the document at node1.1, node1.2 and node1.3. And the training set used by the classifiers of node1.1, node1.2 and node1.3 are all generated by the LDA model of node1, just with different sample labels of positive and negative. In our model we work with mandatory leaf-node predictions that a complete path from the root to a leaf node must be predicted. If there are labeled documents in a non-leaf node, we insert for such non-leaf node a new child node, and all

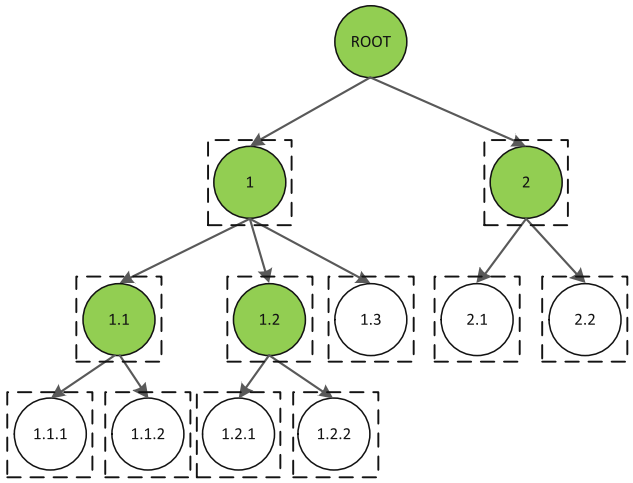

Fig. 2 Binary classifier based top-down approach, *circles* represent classes and *dashed squares* with *round corners* represent binary classifiers. The *green nodes* represent LDA models, and each LDA model is trained using the documents of that node and its descendants

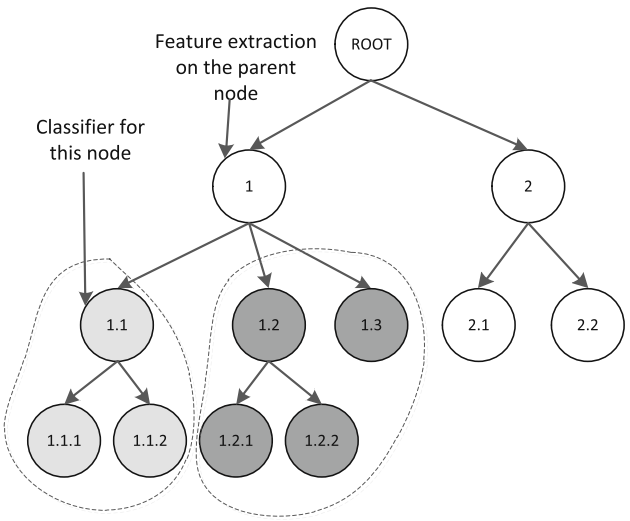

Fig. 3 Training a classifier for a given node of the hierarchy, the *light grey nodes* indicate the positive category and the *dark grey nodes* indicate the negative category. Topic model is trained on the parent node of the given node using all documents belong to the parent node (and its descendants) for generating the topic representation vectors of the documents

of these labeled documents on the non-leaf node will be transferred to the new child node. In this way, all documents are placed in leaf-node categories.

4 Experimental studies

Our experiments include three parts: dataset, model and results.

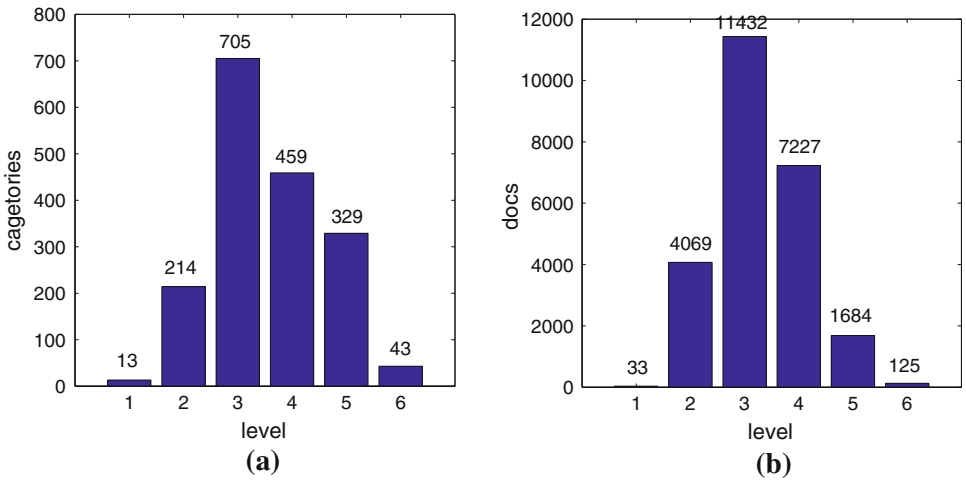

Fig. 4 Data distribution on different level: **a** Categories distribution, **b** Documents distribution

Table 1 Performance of the three models over the description dataset

Models	Macro-*P*	Macro-*R*	Macro-*F1*	Fe	Tr	Te
HFT	0.453	0.450	0.452	0	142	2
HFW	0.453	0.444	0.449	178	136	2
HCL	0.460	0.466	0.463	2,442	50	46

Performance of the models in terms of Macro precision (Macro-*P*), Macro recall (Macro-*R*), Macro-F_1, feature extracting (*Fe*) time, training (*Tr*) time and testing (*Te*) time. The times are expressed in seconds.

Table 2 Performance of the three models over the content dataset

Models	Macro-*P*	Macro-*R*	Macro-*F1*	Fe	Tr	Te
HFT	0.425	0.430	0.428	0	388	22
HFW	0.389	0.405	0.397	841	125	3
HCL	0.429	0.443	0.436	6,844	64	413

4.1 Dataset

There is no universally accepted experimental dataset for large scale hierarchical classification problem yet, so the previous work evaluated their algorithm on different datasets, such as ODP, the Yahoo! Directory, some other domain-specific datasets, etc. According to the practical application of large scale hierarchical classification, such as Internet navigation and Internet content-supervision, we used the Chinese web taxonomy of ODP hierarchy as the experiment subject. The Chinese web taxonomy is a hierarchy of height six, including 13 top categories: reference, commercial, leisure, sports, health, computer, news, family, social, games, art, shopping and science. The entire taxonomy contains 1,763 categories and 24,570 websites. The distribution of documents is shown in Fig. 4b, and the distribution of categories is shown in Fig. 4a.

There are 1,048 categories with instance number <10 in the dataset, about 60 % of the categories are rare categories. So the dataset is representable for the study of rare category classification in the web taxonomy. Because rare category has too few instances, it makes it harder for machine learning algorithm to train effective classifiers.

The documents we used in our experiment have obtained in two ways:

1. Content documents are documents obtained by directly crawling the web pages, using a standard indexing chain (crawling each website in the taxonomy, preprocessing, segmentation words, stop-word removal, representing each website as a document finally)
2. Description documents are documents obtained by indexing the ODP descriptions of the web pages. The ODP descriptions are manually created by ODP editors when placing new documents into the ODP hierarchy. They are thus available for each document in the ODP hierarchy, but not for new documents.

4.2 Model

We use two models as baselines for comparison with our HCL method. The two models are constructed with the word features: a hierarchical full-term classification model (HFT) and a hierarchical feature-word classification model (HFW). The HFT and HFW models are built in a similar way than the HCL, as described in sect. 3.2: it trains a binary classifier for each node except root, taking the examples belonging to a given node (and its descendants) as the positive samples and the examples belonging to its siblings (and their descendants) as the negative samples. During testing a top-down approach is employed: first classifying the given document at the uppermost level and then for each classifier that predicts positive, classifying the given document at the next lower level. The essential difference between the baseline models and the HCL model is in the use of features. The HCL model uses latent topic extracted by LDA. The HFT model uses the full terms. The

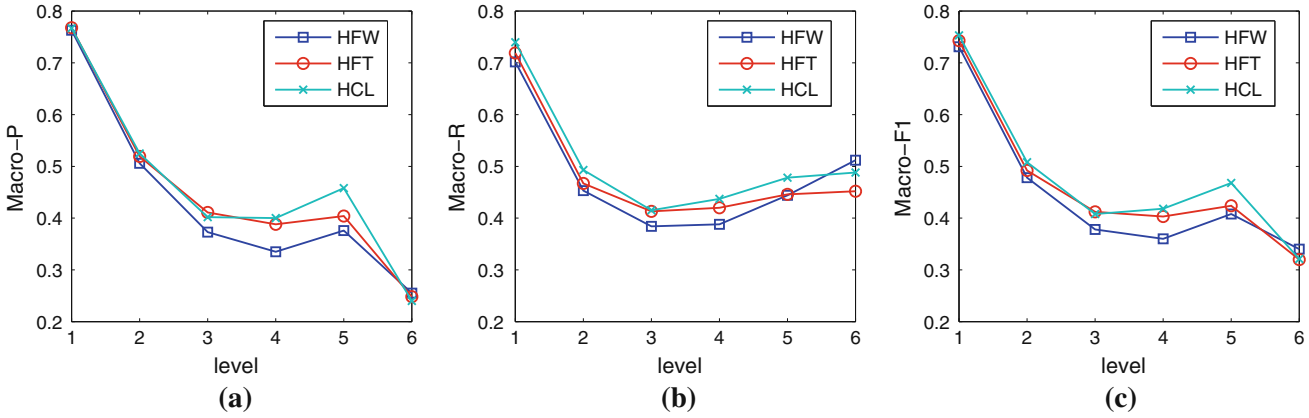

Fig. 5 Performance on different level over the content dataset: **a** Macro-*P* **b** Macro-*R* Macro-*F1*

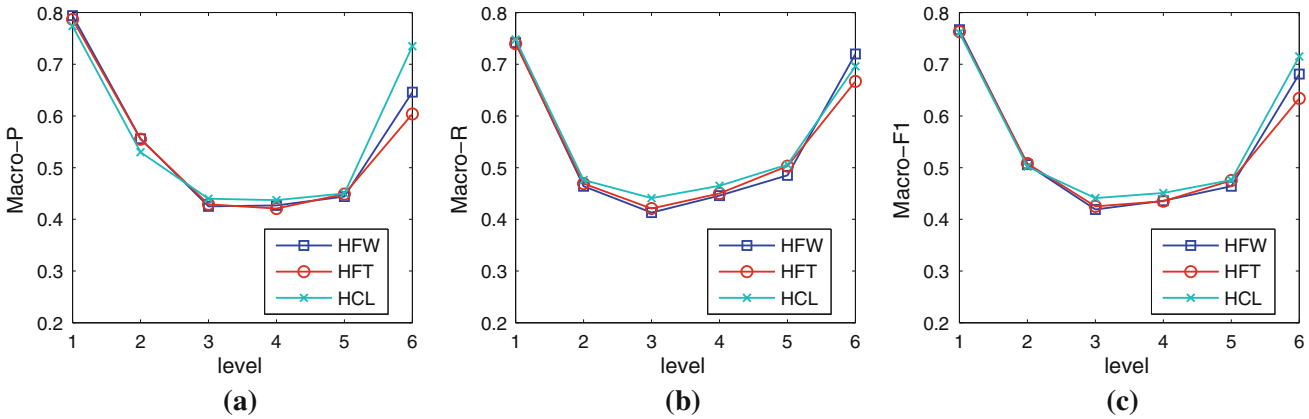

Fig. 6 Performance on different level over the description dataset: **a** Macro-*P* **b** Macro-*R* Macro-*F1*

HFW model uses the feature words selected by a "tfidf" feature selection method. The "tfidf" method first calculates the average $tf \times idf$ value of each word for each category, and takes the top-k words as the category feature set, then merges every category feature set into one set as the final feature set. The baseline models all use term frequency vectors for document representation.

In the three models we use a linear SVM as the binary classifier in each category node. Liblinear is a SVM classifier which is developed by Chih-Jen Lin [5]. Liblinear is very efficient for training large-scale problems, such as the large sparse data with a huge number of instances and features. Thus we use the Java version of the LibLinear [5] library for training linear SVMs in linear time HSVM. Hierarchical SVM (HSVM) is a top-down approach based on SVM. HSVM has been verified to be an efficient method for large scale classification problem. To improve the efficiency of the HSVM algorithm, we modified the corpus read interface of the LibLinear. We use the wrapper for LDA from the Mallet package[3].

[3] http://mallet.cs.umass.edu/

We conducted the experiments using a PC with a 2.53 Ghz Intel Core(TM)2 processor and with 8Gb in RAM. We split the corpus into ten parts randomly, one part as the test set, the rest as the training set, and then tested the performance. We do this for ten times, and use the average value as the final result.

4.3 Results

We use macro precision, macro recall and macro-F1 to compare the performance of the three models, which are defined as : $\text{Macro-}P = \frac{1}{C}\sum_{i=1}^{C}\frac{\text{TP}_i}{\text{TP}_i+\text{FP}_i}$, $\text{Macro-}R = \frac{1}{C}\sum_{i=1}^{C}\frac{\text{TP}_i}{\text{TP}_i}+\text{FN}_i$, $\text{Macro-}F_1 = \frac{2\times\text{Macro}-P\times\text{Macro}-R}{\text{Macro-}P+\text{Macro-}R}$. Macro averaging of the performance measures is used to estimate how well a model is performed along categories, disregarding their size by weighting the performance for each class equally. Since rare categories are very frequent in the web taxonomy, macro measure is used here which is able to avoid to bias the performance towards dense categories. The performance of the three models is shown in Tables 1, 2, including Macro-*P*, Macro-*R*, Macro-F_1, feature

extracting time, training time and testing time. The three time results are overall test results. Results show that the performance of HCL model over the DMOZ directory outperforms HFT and HFW, about 3 % improvement at macro averaging of the performance measures. The model HCL had the maximum Macro-F_1, a 2 % relative improvement compared to the model HFT and a 3 % relative improvement compared to the model HFW. Because HFT just used full terms as features, its feature extracting time is zero. HCL needs more feature extracting time because it needs training a LDA model at each non-leaf node. And because HCL first computes the latent topic vector of the given document in each fired node of the hierarchy when assigning categories to a new document, so its testing time is larger than the other two.

In order to evaluate the performance of the models on different levels in class hierarchy, Liu [11] proposed a level evaluation measure. When performance was calculated for the *i*-th level, it neglected the existence of the deeper levels in the hierarchy and put all documents in them into their parent categories at the *i*-th level. As the level evaluation measure is able to show the performance on different levels visually, here we use it to estimate the models on different levels of the hierarchy. The performance of the models on each level is shown in Figs. 5, 6. Obviously, the data points at the 6-th level correspond to the performance over the whole hierarchy that shown in the table. Compared to the other two methods, HCL performed better at the deep levels of the hierarchy. It means that the LDA model is suited to build topic representations for categories with only a few training examples, as most of rare categories are deep nodes.

5 Conclusion

In this paper, the data sparseness problem of rare categories in large scale hierarchical classification over the topic taxonomy is researched. We analyze the data sparseness problem of rare categories in the topic taxonomy firstly, and propose a hierarchical classification model based on LDA. We learn topic model for each non-leaf node in the hierarchy via LDA, and convert document-word matrix to document-topic matrix, then use a top-down approach for training/testing. Next, top-down approaches based on different feature extraction methods are compared. The experimental results show that compared to the full-term classification model and the feature-word classification model, the proposed model is suited to build topic representations for categories with only a few training examples, and improve the classification performance over the topic taxonomy.

References

1. Blei DM, McAuliffe JD (2010) Supervised topic models. arXiv:1003.0783
2. Blei D, Ng A, Jordan M (2003) Latent dirichlet allocation. J Mach Learn Res 3:993–1022
3. Chen WJ, Shao Y-H, Hong N (2013) Laplacian smooth twin support vector machine for semi-supervised classification. Intern J Mach Learn Cyber. doi:10.1007/s13042-013-0183-3
4. Fagni T, Sebastiani F (2007) On the selection of negative examples for hierarchical text categorization. In: Proceedings of the 3rd Language and Technology Conference (LTC07) pp 24–28
5. Fan R, Chang K, Hsieh C, Wang X, Lin C (2008) Liblinear: a library for large linear classification. J Mach Learn Res 9:1871–1874
6. Gomez JC, Moens M-F (2012) Hierarchical classification of web documents by stratified discriminant analysis. In: Multidisciplinary Information Retrieval, Springer, pp 94–108
7. Gopal S, Yang Y, Bai B, Niculescu-Mizil A (2012) Bayesian models for large-scale hierarchical classification. In: Advances in Neural Information Processing Systems 25: 2420–2428
8. Griffiths T, Steyvers M (2004) Finding scientific topics. Proc Natl Acad Sci USA 101(Suppl 1): 5228–5235
9. He L, Jia Y, Han W, Tan S, Chen Z (2012) Research and development of large scale hierarchical classification problem. In: Chinese Journal of Computers pp 2101–2115
10. He Q, Wu C (2011) Separating theorem of samples in banach space for support vector machine learning. Intern J Mach Learn Cybernet 2(1): 49–54
11. Liu T, Yang Y, Wan H, Zeng H, Chen Z, Ma W (2005) Support vector machines classification with a very large-scale taxonomy. ACM SIGKDD Explor Newslett 7(1):36–43
12. Liu Z, Wu Q, Zhang Y, Chen CP (2011) Adaptive least squares support vector machines filter for hand tremor canceling in microsurgery. Intern J Mach Learn Cybernet 2(1):37–47
13. Madani O, Huang J (2010) Large-scale many-class prediction via flat techniques. In: Large-Scale Hierarchical Classification Workshop of ECIR
14. Marath S (2010) Large-scale web page classification. Ph.D. thesis
15. Oh H, Choi Y, Myaeng S (2010) Combining global and local information for enhanced deep classification. In: Proceedings of the 2010 ACM Symposium on Applied Computing, ACM, pp 1760–1767
16. Wang X, Lu SX, Zhai JH (2008) Fast fuzzy multi-category svm based on support vector domain description. Intern J Patt Recogn Artif Intell 22(1):109–120
17. Xue G, Xing D, Yang Q, Yu Y (2008) Deep classification in large-scale text hierarchies. In: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, ACM, pp 619–626