Soft Comput (2015) 19:29–38
DOI 10.1007/s00500-014-1374-x

FOCUS

# Open-categorical text classification based on multi-LDA models

**Ruiji Fu · Bing Qin · Ting Liu**

Published online: 31 July 2014


**Abstract** We present a new and realistic problem, open-categorical text classification, which requires us to classify documents without the categorization system known beforehand. To solve this problem, we propose a novel approach to construct the categorization system and classify documents based on multi-latent Dirichlet allocation (LDA) models. We cluster topics and extract topical keywords to help category annotation. Subsequently, the LDA models are applied to predict the categories of documents comprehensively. Our result, a macro-averaged F1 measure of 84.02 %, outperforms the state-of-the-art supervised and semi-supervised text classification methods.

**Keywords** Topic model · Text classification · Categorization system construction

## 1 Introduction

Text classification (TC) is a task consisting in labeling automatically a document with a certain category, based on its content. Traditional TC classifies documents into predefined categories. Many machine learning-based approaches are exploited to deal with this task, such as Naive Bayesian (Kim et al. 2002), k Nearest Neighbors (Danesh et al. 2007), and Sport Vector Machine (Joachims 1998; Lin 2002; Donghui and Zhijing 2010; Qin and Wang 2009; Fu and Lee 2012). These methods need manually labeled data with predefined categories to tune parameters. However, in some practical applications, we do not know the categories beforehand or only know a part of the categories. For example, given a set of documents, we may not know which and how many categories are included in it. Because of this, it is difficult to label training data for supervised machine learning methods.

In this paper, we propose a novel approach based on latent Dirichlet allocation (LDA) model to deal with this problem. LDA is a generative probabilistic model for collections of discrete data such as text corpora. It is a three-level hierarchical Bayesian model, in which each item of a collection is modeled as a finite mixture over an underlying set of topics. Each topic is, in turn, modeled as an infinite mixture over an underlying set of topic probabilities. First, we apply LDA on the whole data set and get the latent topics which are represented as an allocation of words in the text. Subsequently, we select good topics based on the entropy of word distributional probabilities in the topics. We then cluster the good topics and extract topical keywords to help the categorization system construction. Our approach is easier than labeling documents one-by-one. Using the latter approach, we can only label limited training data because this approach is costly and time consuming. Moreover, some minority categories may be missed because no instance of these categories may be labeled. On the contrary, our approach can cover all categories because the topics are generated from the whole data. When the topics are clustered and labeled with category tags, we can estimate the topic distributions of a given document based on the LDA model and classify the document into the category of the top one topic.

A deficiency of LDA is that the number of topics need be set beforehand. If the number is too small, different cat-

Communicated by L. Xie.

R. Fu · B. Qin · T. Liu (✉)
Harbin Institute of Technology, 6th Floor, No.29, Jiaohua Street, Nangang District, Harbin 150001, People's Republic of China
e-mail: tliu@ir.hit.edu.cn

R. Fu
e-mail: rjfu@ir.hit.edu.cn

B. Qin
e-mail: bqin@ir.hit.edu.cn

egories may be mixed together in a topic, which makes it difficult to assign a single proper category to the topic. Therefore, the model cannot distinguish these categories. If it is too big, one category may be split up into several topics. The performance of LDA will be harmed. To avoid this deficiency, we employ multi-LDA models and induce and map the topics from different LDA models onto unified categories. Then, we use the multi-models to classify documents together. Our experiments show the proposed approach outperforms other supervised or semi-supervised approaches.

Our contributions are as follows:

- We propose a new and realistic problem, open-categorical text classification (OCTC). It is different from the traditional TC problem that OCTC requires us to classify documents without the categorization system known beforehand. Hence, it is difficult to label training data for the supervised machine learning method which are widely used for traditional TC.
- We propose an approach based on multi-LDA models to construct the categorization system and classify documents. The experiments show that our approach is effective.

## 2 Related work

### 2.1 Text Classification

In the last decades, TC is solved using supervised and semi-supervised classification algorithms. Multiple approaches to TC were presented using some well-known classifiers. The most widely used model for text categorization is the vector space model (VSM) (Gerard Salton et al. 1975). Under the model, a feature space is extracted based on a set of unique, uncommon and frequent words which are evaluated for each document. Each document is represented by a real-valued vectors with dimensions corresponding to unique words. When some of the documents' actual categories are known and used for training, many well-known classifiers in supervised machine learning, such as support vector machines (SVM) (Joachims 1998; Lin 2002; Donghui and Zhijing 2010; Qin and Wang 2009; Fu and Lee 2012), k Nearest Neighbors (kNN) (Danesh et al. 2007), Naive Bayes (NB) (Kim et al. 2002), Decision tree (DT) (Johnson et al. 2002; Vateekul and Kubat 2009) and Neural Network (Ng et al. 1997; Trappey et al. 2006; Li and Park 2009), can then be applied to categorize documents.

As we all know, supervised methods need labeled corpora for training. If there is not enough training data, their numerous parameters cannot be learned well. Therefore some researchers propose semi-supervised methods to obtain competitive models with limited labeled data with a large amount of unlabeled data, such as bootstrapping methods. Bootstrapping methods are used to train a model on a small number of labeled data and predict the labels of unlabeled data. They iteratively expand the set of labeled data using high-confidence results and improve the performance by training a new model on the larger labeled data. Such methods have shown promise in applications such as web page classification (Blum and Mitchell 1998), named entity classification (Collins and Singer 1999), parsing (McClosky et al. 2006), machine translation (Ueffing 2006), and information extraction (Carlson et al. 2010).

Cheng et al. (2013) propose a method to improve SVM by bootstrapping unlabeled data with self-training. The SVM classifier is iteratively refined through the augmentation of the training set.

Supervised or semi-supervised methods need some labeled data with predicted categories. However, in our task, we do not know categories beforehand. Therefore, it is difficult to apply the methods to the task directly.

### 2.2 Topic models

Topic models are a type of statistical models for discovering the abstract topics that occur in a collection of documents. Well-known topic models include probabilistic latent semantic analysis (pLSA) (Hofmann 1999), latent Dirichlet allocation (LDA) (Blei et al. 2003b) and their varieties (Blei et al. 2003a; Blei and McAuliffe 2007; Petinot et al. 2011; Mao et al. 2012).

LDA is widely used for identifying the topics in a set of documents, building on previous work about pLSA by Hofmann (1999). It is an unsupervised algorithm. In this model, the topic distribution is assumed to have a Dirichlet prior. Each document is represented as a mixture of a mixed number of topics, with topic $z$ receiving weight $\theta_z^d$ in document $d$. Each topic is a probability distribution over a finite vocabulary of words, with word $w$ having probability $\phi_w^z$ in topic $z$.

In this paper, we apply LDA to help category system construction and document categorization.

## 3 Our approach

In this section, we first define the task formally. Then, we elaborate on our proposed approach composed of three major steps, namely, topic discovery, category system construction, and document categorization (Fig. 1).

### 3.1 Task definition

The goal of OCTC task is to classify documents without the categories known beforehand. The input of the task is

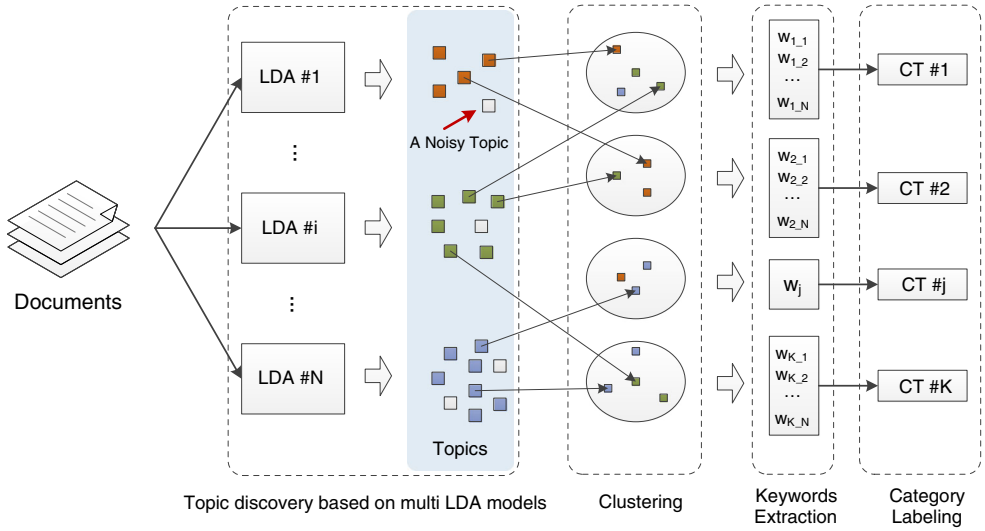


**Fig. 1** The framework of category system construction

**Table 1** Examples of LDA topics

| ID | # of topics | word distribution | category |
|---|---|---|---|
| 1 | 150 | 电影(movie):0.2235; 视频(video):0.0919; 影城(cineplex):0.0628; 数字(digital):0.0524; 影视(film and television):0.0458; 影院(cinema):0.0442 | 电影 movie |
| 2 | 250 | 医院(hospital):0.1812; 医疗(medical treatment):0.0539; 治疗(cure):0.0353; 妇科(gynecology):0.0280; 专家(expert):0.0264; 疾病(disease):0.0238 | 医疗 health care |
| 3 | 500 | 医院(hospital):0.1731; 医疗(medical treatment):0.0484; 治疗(cure):0.0327; 咨询(consult):0.0280; 专家(expert):0.0276; 妇科(gynecology):0.0272 | 医疗 health care |
| 4 | 500 | 销售(sell):0.1258; 汽车(automobile):0.1070; 服务(service):0.0385; 维修(repair):0.0230; 大众(Volkswagen):0.0172; 丰田(Toyota):0.0170 | 汽车 automobile |
| 5 | 500 | 知道(know):0.4916; 事儿(thing):0.1266; 告诉(told):0.1041; 关注(attention):0.0878; 事情(thing):0.0159; 秘密(secret):0.0156 | noisy |

only a set of unlabeled documents $D = \{d_1, d_2, \ldots, d_M\}$. There, $M$ denotes the total number of documents. We need first abstract and construct the categorization system $C = \{c_1, c_2, \ldots, c_K\}$ from $D$. $K$ is the number of categories in all. When a new document $d_i$ is given, the final goal is to assign a proper category $c_j$ to it.

### 3.2 Topic discovery

To discover topics from messages, we choose to directly apply LDA. The basic idea of LDA is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. Our experiments show that we can obtain meaningful topics from our data set using standard LDA. We set multi-numbers of topics and ran 1,000 iterations of Gibbs sampling using the GibbsLDA++ toolkit.[1]

### 3.3 Category system construction

After we obtain topics, we induce and annotate the categories using the distributions of words in topics (Fig. 1). In LDA, each topic is represented as a distribution over words as Table 1 shows. We apply multi-LDA models with different numbers of topics. However, manually annotating so many topics one-by-one is still time consuming. To solve the problem, we first cluster the topics and then extract keywords of each cluster to help category annotation. Specifically, to

[1] http://gibbslda.sourceforge.net/.

measure the similarity between two topics $t$ and $t'$, we use JS-divergence between the word distributions of the two topics, denoted as $\mathbf{P}_t$ and $\mathbf{P}_{t'}$. Each distribution vector is composed of distributional probabilities of words in the corresponding topic.

$$D_{\mathrm{JS}}(\mathbf{P}_t \parallel \mathbf{P}_{t'}) = \frac{1}{2}(D_{\mathrm{KL}}(\mathbf{P}_t \parallel \mathbf{P}_{\mathrm{a}}) + D_{\mathrm{KL}}(\mathbf{P}_{t'} \parallel \mathbf{P}_{\mathrm{a}})) \quad (1)$$

where word distributions are represented as vectors:

$$\mathbf{P}_t = (p(w_1|t), p(w_2|t), \ldots, p(w_L|t)) \quad (2)$$

Each dimension corresponds to the distributional probability of a separate word on topic $t$. $L$ is the number of words in the vocabulary (the number of distinct words occurring in the corpus). $\mathbf{P}_{\mathrm{a}}$ is the average word distribution of topic $t$ and $t'$. That is $p(w_i|a) = \frac{1}{2}p(w_i|t) + \frac{1}{2}p(w_i|t')$ for $i \in \{1, 2, \ldots, L\}$. $D_{\mathrm{KL}}$ is the KL-divergence which is a non-symmetric measure of the difference between two probability distributions.

$$D_{\mathrm{KL}}(\mathbf{P}_t \parallel \mathbf{P}_{\mathrm{a}}) = \sum_w p(w|t) \log \frac{p(w|t)}{p(w|a)} \quad (3)$$

While the JS-divergence has the advantage that it is symmetric.

However, not all of the topics are meaningful. Some of them are noisy topics such as the last example in Table 1. To remove them, we exploit an observation that most meaningful topics focus on a single theme. If the words in a topic distributed in a wide range, it is likely a noisy topic. We, therefore, defined a measure called topic entropy (TE) as follows:

$$\mathrm{TE}(t) = -\sum_w p(w|t) \log p(w|t) \quad (4)$$

The larger the TE($t$), the more likely $t$ is a noisy topic. We remove topics whose TE($t$) is larger than a threshold (see 5.1.1).

We then apply K-means algorithm to cluster the topics from multi-LDA models. We choose the number of clusters, $K$, using the method proposed by Pham et al. (2005). The method takes into account information reflecting the performance of the K-means algorithm.

Subsequently, we extract keywords from each cluster to help category construction. For example, the word "hospital" infers the current cluster being close to the category "health care". To the contrary, the word "expert" does not. We rank the words in a cluster by their average distributional probabilities over the topics in the cluster. From the ranked words, we select the top $N$ as the category tag candidates of the cluster. Finally, we manually check the candidates and merge similar clusters to get the final categories. Comparing to the approach annotating instances one-by-one, our approach can greatly improve work efficiency of category construction.

### 3.4 Document classification

After we annotate the category tags of topics, we can use the multi-LDA models to classify new documents. When a piece of text is given, we predict the topic distributions based on LDA models. Then, we compute the category distributions based on the topic distributions. Equation 5 shows a measure, called *average measure*, which simply computes average probability of all topics in a cluster as the probability of the category.

$$p(c_j|d) = \frac{1}{Z}\sum_{t \in c_j} p(t|d) \quad (5)$$

where $p(c_j|d)$ denotes the probability that the given document $d$ belongs to the $j$th category (a cluster) $c_j \cdot t$ denotes a topic. $Z$ is a normalizing factor ensuring that the result is a probability.

$$Z = \sum_t p(t|d) \quad (6)$$

In the *average measure*, all topics in a category (a cluster) have equal weights. However, the distance between a topic and the centroid of a cluster can reflect the probability that it belongs to the cluster. Closer distance means higher probability. Therefore, we also propose a modified measure called *weighted measure* as follows:

$$p(c_j|d) = \frac{1}{Z'}\sum_{t \in c_j} \mathrm{ID}_{\mathrm{JS}}(\mathbf{V}_t \parallel \mathbf{V}_{\bar{c}_j}) p(t|d) \quad (7)$$

where $\mathrm{ID}_{\mathrm{JS}}(\mathbf{V}_t \parallel \mathbf{V}_{\bar{c}_j})$ denotes the inverse JS-divergence between $\mathbf{V}_t$ and $\mathbf{V}_{\bar{c}_j}$ $\cdot \bar{c}_j$ denotes the centroid of cluster $c_j \cdot Z'$ is also a normalizing factor.

$$\mathrm{ID}_{\mathrm{JS}}(\mathbf{V}_t \parallel \mathbf{V}_{\bar{c}_j}) = \frac{1}{D_{\mathrm{JS}}(\mathbf{V}_t \parallel \mathbf{V}_{\bar{c}_j})} \quad (8)$$

$$Z' = \sum_c \sum_{t \in c} \mathrm{ID}_{\mathrm{JS}}(\mathbf{V}_t \parallel \mathbf{V}_{\bar{c}}) \quad (9)$$

Finally, we select the category with the highest probability as the result of classification.

## 4 Experimental setup

### 4.1 Experimental data

In this work, the data are collected from the introductions of subscription accounts on WeChat,[2] a mobile text and voice

[2] http://www.wechat.com/en/.

**Table 2** Samples of experimental data

| Title | Introduction |
|---|---|
| 王力宏<br>Lee-Hom Wang | 王力宏，知名歌手、音乐人、导演、演员。致力于发扬华语音乐的突破与融合。<br>Lee-Hom Wang is a famous singer-songwriter, record producer, film director and actor. He applies himself to carry forward the breakthrough and fusion of Chinese music. |
| 招商银行<br>信用卡中心<br>CMB's Credit Card Center | 信用卡移动服务领跑者。服务：秒查账单、额度、积分；办卡；还款；随时掌握信用卡优惠活动，更有人工服务哦！<br>CMB is a leader on credit card mobile service. Available service: checking bills, limits, and points immediately; applying for cards; repaying rapidly; grasping the promotions; and human-driven services. |
| 单反摄影<br>SLR Photography | 收集和分享国内外一流摄影精品，不定时邀请摄影大师在线分享心得和各位互动交流，敬请关注！<br>We collect and share international first-class photography works. We erratically invite photographers to share their experiences and online communicate with everybody. Please pay attention! |

messaging communication service in China. Subscription accounts are a kind of accounts which provide services and can be subscribed by users. Table 2 shows some instances.

We combine the title and introduction of an account as a short document, based on which we classify the account. Each document contains 11.74 words on average. We totally collect 985,397 subscription accounts for LDA learning and category construction. We then label 1,500 accounts with the constructed categories and randomly split them into 1/5 for development and 4/5 for testing.

The Chinese segmentation is provided by an open-source Chinese language processing platform LTP (Che et al. 2010).[3]

### 4.2 Evaluation metrics

Two widely used metrics in TC to evaluate classifiers' accuracy are the macro-averaged F1 score ( Yang 1999) and the micro-averaged F1 score ( Sebastiani 2002).

**The macro-averaged F1** ($F1_{\text{macro}}$) is computed locally over each category. It can be computed as a weighted average of the macro precision ($P_{\text{macro}}$) and the macro recall ($R_{\text{macro}}$).

$$P_{\text{macro}} = \frac{1}{q}\sum_{i=1}^{q}\frac{\#\text{ instances predicted and correct in the } i\text{th category}}{\#\text{ total instances predicted in the } i\text{th category}} \tag{10}$$

$$R_{\text{macro}} = \frac{1}{q}\sum_{i=1}^{q}\frac{\#\text{ instances predicted and correct in the } i\text{th category}}{\#\text{ total instances correct in the } i\text{th category}} \tag{11}$$

$$F1_{\text{macro}} = \frac{2 * P_{\text{macro}} * R_{\text{macro}}}{P_{\text{macro}} + R_{\text{macro}}} \tag{12}$$

where $q$ denotes the total number of categories.

[3] http://www.ltp-cloud.com/.

**The micro-averaged F1** ($F1_{\text{micro}}$) is computed globally over all category decisions.

$$P_{\text{micro}} = \frac{\#\text{ instances predicted and correct in all categories}}{\#\text{ total instances predicted in all categories}} \tag{13}$$

$$R_{\text{micro}} = \frac{\#\text{ instances predicted and correct in all categories}}{\#\text{ total instances correct in all categories}} \tag{14}$$

$$F1_{\text{micro}} = \frac{2 * P_{\text{micro}} * R_{\text{micro}}}{P_{\text{micro}} + R_{\text{micro}}} \tag{15}$$

where $P_{\text{micro}}$ and $R_{\text{micro}}$ denote the micro-averaged precision and recall, respectively. Actually, every instance in the test set needs to be classified. Therefore, the number of total instances predicted equals the number of instances in the test set. That is to say, $P_{\text{micro}}$ is the same as $R_{\text{micro}}$. $F1_{\text{micro}}$ is actually the same as **accuracy** (A).

$$A = \frac{\#\text{ instances predicted and correct}}{\#\text{ total instances in the test set}} \tag{16}$$

For the sake of simplicity, we use the accuracy measure instead of micro-averaged F1 in the experiments.

To evaluate topical keywords extraction, we use **Precision@N** (P@N) for the ranked keyword lists.

$$P@N = \frac{1}{q}\sum_{i=1}^{q}\frac{\#\text{ correct keywords in the top } N \text{ in } i\text{th category}}{N} \tag{17}$$

In our example, $N$ is set as 1, 3, 5, 10, and 30.

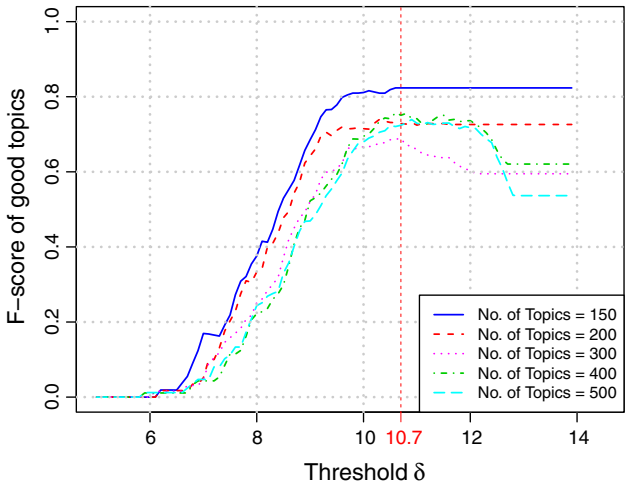


**Fig. 2** The effect of varying the threshold $\delta$ of topic filtering. We evaluate it under five LDA models with different numbers of topics The results show that the best threshold is 10.7 for all the models

**Table 3** The performance of keywords extraction

| P@1 (%) | P@3 (%) | P@5 (%) | P@10 (%) | P@30 (%) |
|---|---|---|---|---|
| 81.30 | 79.44 | 75.51 | 69.44 | 56.57 |

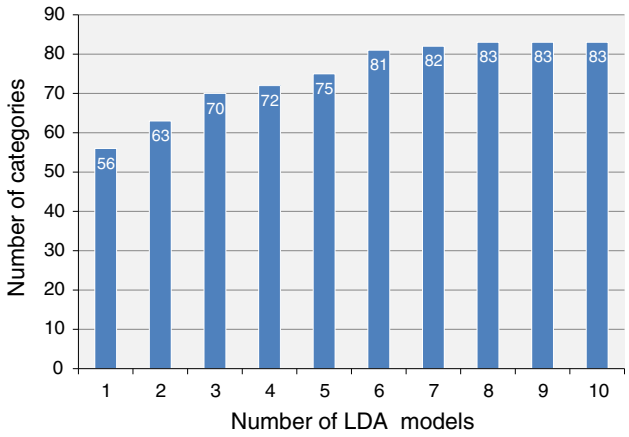


**Fig. 3** The effect of varying the number of LDA models

## 5 Results and discussion

In this section, we evaluate two aspects of our approach as follows: categorization system construction and document classification.

### 5.1 Evaluation of categorization system construction

#### *5.1.1 Threshold of filtering out noisy topics*

As Sect. 3.3 proposes, we use topic entropy to filter out noisy topics. We remove topics while their topic entropy is greater than a threshold $\delta$. To tune $\delta$, we ask two volunteers to distinguish whether the topics are noisy. We measure the inter-annotator agreement using the kappa coefficient. The kappa value is 0.81, which indicates a good strength of agreement. Figure 2 shows the evaluation results when we vary $\delta$. When we increase $\delta$, the F-scores of "good" topics increase until $\delta$ reaches 10.7. When $\delta$ reaches 10.7, we obtain the best performance of topic filtering under different LDA models. It shows that the entropy of word distribution under a topic can reflect its quality. After topic filtering, about 60 % high-quality topics remain in our experiments, which can help to construct the categories.

#### *5.1.2 Quality of topical keywords*

To reduce the manual labor of category annotation, we cluster all high-quality topics from multi-topics and extract topical keywords for each cluster. These keywords can help annotators label categories of clusters easily. In our experiment, we use K-means algorithm to cluster topics and compute the average distribution of words in each cluster. Subsequently, we rank words in each cluster based on their distributional probabilities. The top words are selected as the topical keywords of the cluster.

When the two annotators label the category tags, we also ask them to judge whether the top keywords are helpful. We compute kappa value, a statistical measure of inter-annotator agreement for categorical items (Carletta 1996), on the evaluation data. The kappa value is 0.75, which also indicates a good strength of agreement. As Table 3 shows, 81.30 % of the top one keywords are helpful. When we look down to top ten keywords, the precision still reaches about 70 %. With the help of the keywords, annotators can induce categories more easily.

#### *5.1.3 Number of LDA models*

In the section, we analyze how many LDA models is proper for our task. We apply different numbers of LDA models to construct categories. The number of topics $K$ is set from 150 to 600. The interval is 50. We thus train 10 models totally and label the category for each high-quality topic after noise filtering. Then, we try to combine the categories from different models gradually. For example, in Fig. 3, the 2-model combination means combining the categories from the LDA models with 150 and 200 topics. The 3-model combination means combining the categories from the LDA models with 150, 200 and 250 topics, and so on.

The result shows that the number of categories reaches the highest point when we combine eight LDA models. After that, the number or the coverage of categories do not increase

**Table 4** The performance of document classification

| Method | $P_{\text{macro}}$ (%) | $R_{\text{macro}}$ (%) | $F1_{\text{macro}}$ (%) | $A$ (%) |
|---|---|---|---|---|
| $M_{\text{AD}}$ | 80.88 | 85.20 | 82.98 | 87.27 |
| $M_{\text{WAD}}$ | **81.63** | **86.56** | **84.02** | **88.78** |
| $M_{\text{SVM}}$ | 56.42 | 72.46 | 63.44 | 70.85 |
| $M_{\text{Semi-SVM}}$ | 59.66 | 75.94 | 66.82 | 77.47 |

The best performance of each metric has marked as bold values
$M_{\text{AD}}$ denotes the multi-LDA-based method with average distribution measure. $M_{\text{WAD}}$ denotes the one with weighted average distribution measure. $M_{\text{SVM}}$ and $M_{\text{Semi-SVM}}$ denote SVM and semi-supervised SVM, respectively

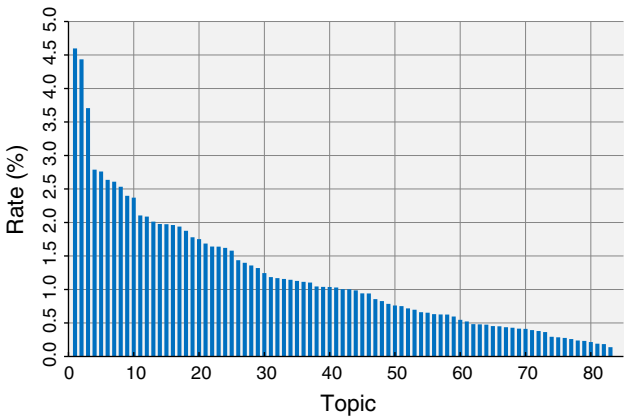


**Fig. 4** The distribution of categories on the test data

while adding new models. We thus select eight models in the remaining experiments.

#### *5.1.4 Distribution of categories*

We annotate 83 categories in all (see "Appendix"). Table 4 shows their distributions on the test corpus, which are sorted from high to low. The distributions are non-uniform. The most frequent 20 categories cover nearly a half of documents (48.55 %) in the test corpus.

Moreover, about 12.36 % documents cannot be classified into one of the categories. That is because these documents are too short to contain enough information for classification.

### 5.2 Classification

#### *5.2.1 Comparison between average distribution and weighted average distribution*

In Sect. 3.4, we propose two methods, the average distribution ($M_{\text{AD}}$) and the weighted average distribution ($M_{\text{WAD}}$) method, to compute category distribution based on topic distribution for a given document. We compare the two methods in Table 4. $M_{\text{WAD}}$ outperforms $M_{\text{AD}}$ significantly ($p < 0.01$),[4] which shows that different topics have different importance in a category. The topics which are close to the centers of clusters have large weights, and therefore are important for classification. In contrast, the topics far away from the centers have little importance for classification.

#### *5.2.2 Comparison to SVM*

SVM is a representative supervised machine learning method for text classification. If we use it to our task, we should first annotate enough data for training. It is impossible to annotate all of the data because it is a costly job. However, the categories are distributed unevenly as Fig. 4 shows. If we randomly sample a subset of the data, few or even no samples of the minority categories may be selected. It may lead that SVM cannot handle these long-tailed categories.

Clustering data and then sampling instances from each cluster, respectively, may solve this problem. We apply the K-means method to cluster the 0.98 million documents. Each document is represented as the vector of words occurring in it. Each dimension corresponds to a separate word. Its value is term frequency inverse document frequency (TF-IDF) weighting of the corresponding word. We annotate the samples with the categories constructed based on LDA. However, in a real situation, the annotation is more difficult because the categories are unknown. Finally, spending the same amount of time with LDA topic annotation, we annotate about 30 thousand documents. Nevertheless, there are still nine categories absent in the training data. We train an SVM on these documents and tune the parameters on the development data. To avoid sparse data problems, we use the results of Brown clustering as features instead of words. Brown clustering is a form of hierarchical clustering of words based on the contexts in which they occur (Brown et al. 1992; Turian et al. 2010). The results in Table 4 shows that our multi-LDA-based approach outperforms SVM.

We see that the supervised SVM method can only annotate limited documents and cannot construct the complete category system. Moreover, because the documents in our data set are short, there are few features which can be used for classification. But our proposed multi-LDA approach can handle more unlabeled data to build complete category system and improve document classification.

#### *5.2.3 Comparison to semi-supervised SVM*

We also apply the semi-supervised SVM proposed by Cheng et al. (2013) to our task. We exploit 140 thousand unlabeled documents to augment training data and refine the SVM itera-

[4] The $p$ value is the probability of obtaining a test statistic result at least as extreme as the one that was actually observed, assuming that the null hypothesis is true. We use the significance testing method proposed by Zhang et al. (2004).

tively. We use the SVM to classify the unlabeled documents and select some instances to augment the training data for SVM retraining. In each iteration, 5 % instances with highest confidence are selected and added into SVM training data. The process stops until the performance does not been improved in the development set.

We can see from Table 4 that the semi-supervised methods can improve the performance of SVM. However, the best result is still significantly worse than the result of our proposed multi-LDA method. The reason may be that the semi-supervised method can augment training data but cannot supplement new categories. That is to say, if we have a small training data with 74 categories initially, we will also label new data with the 74 categories. While, our proposed method can construct a more complete category system.

## 6 Conclusion

In this paper, we propose a novel approach based on multi-LDA models to solve the new problem named open-categorical text classification. Because we do not know the categorization system beforehand, we first apply multi-LDA models to a large scale unlabeled corpus and obtain the topics. Then, we cluster the topics and extract topical keywords for each cluster. In the top ten extracted keywords, nearly seven are useful in average for categorization system construction as our experiments show. After we construct the categorization system, we actually get a projection from topics to category tags. Finally, we apply the multi-LDA models to classify documents. The experiments show that our approach outperforms the state-of-the-art supervised and semi-supervised SVM.

In the future, we will study how to add new categories to the old categorization system expediently, because some new topics maybe occur with the passing of time.

**Acknowledgments** This work is supported by National Natural Science Foundation of China (NSFC) via Grant 61133012, 61273321 and the National 863 Leading Technology Research Project via grant 2012AA011102. Special thanks to Jianfei Guo and Xiaocheng Feng for their help in the experiments..

## Appendix: the categorization system of WeChat subscription accounts[5]

– **finance and economics**

1. banking institutions
2. business
3. financing
4. insurance
5. marketing
6. realty
7. start-ups

– **shopping**

8. automobile
9. commodity
10. decoration
11. discount shopping
12. dresses
13. electronic products
14. luxuries
15. online shopping
16. purchasing agents
17. sports equipments
18. wholesale
19. health care
20. maternal and infant
21. nourishing of life
22. dating

– **communication platform**

23. friends making
24. job hunting

– **education**

25. art schools
26. business administration
27. driving schools
28. foreign language training
29. raining for study abroad
30. tutoring

– **military affairs**

31. military affairs

– **science and technology**

32. IT
33. mobile internet applications

– **media**

34. news media
35. print media
36. TV and radio
37. we-media
38. cosmetic surgery
39. hairdressing
40. skin protection

– **food and drink**

41. green food

[5] These categories are constructed using our proposed semi-automatic approach based on multi-LDA models. Totally, we obtain 83 categories.

42. restaurants
43. tea
44. western-style pastry
45. wine

– **services for life**

46. air tickets booking
47. Campus
48. car rental
49. community
50. design
51. emotion
52. environmental protection
53. Express delivery
54. homemaking
55. hot lines
56. hotel booking
57. law works
58. life assistants
59. lotteries
60. public good
61. recharging
62. tourism
63. weddings

– **culture**

64. art
65. culture
66. originality
67. popularization of science
68. reading

– **entertainment**

69. adult entertainment
70. caricatures
71. entertainment stars
72. entertainment venues
73. fashion
74. games
75. image show
76. jokes
77. movies
78. music
79. pets

– **sports**

80. sports clubs
81. sports news

– **others**

82. brand
83. government

## References


Blei DM, Griffiths TL, Jordan MI, Tenenbaum JB (2003a) Hierarchical topic models and the nested Chinese restaurant process. In: NIPS, vol 16

Blei DM, Ng AY, Jordan MI (2003b) Latent dirichlet allocation. J Mach Learn Res 3:993–1022

Blei DM, McAuliffe JD (2007) Supervised topic models. NIPS 7:121–128

Blum A, Mitchell T (1998) Combining labeled and unlabeled data with co-training. In: Proceedings of COLT, pp 92–100

Brown PF, Desouza PV, Mercer RL, Della Pietra VJ, Lai JC (1992) Class-based n-gram models of natural language. Comput Linguist 18(4):467–479

Carletta J (1996) Assessing agreement on classification tasks: the kappa statistic. Comput Linguist 22(2):249–254

Carlson A, Betteridge J, Wang RC, Hruschka Jr ER, Mitchell TM (2010) Coupled semi-supervised learning for information extraction. In: Proceedings of the third ACM international conference on Web search and data mining, pp 101–110

Che W, Li Z, Liu T (2010) Ltp: a Chinese language technology platform. In: Coling 2010: demonstrations, pp 13–16

Cheng SJ, Huang QC, Liu JF, Tang XL (2013) A novel inductive semi-supervised SVM with graph-based self-training. In: Intelligent science and intelligent data engineering. Springer, Berlin Heidelberg, pp 82–89

Collins M, Singer Y (1999) Unsupervised models for named entity classification. In: Proceedings of EMNLP, pp 100–110

Danesh A, Moshiri B, Fatemi O (2007) Improve text classification accuracy based on classifier fusion methods. 10th international conference on information fusion, pp 1–6

Donghui C, Zhijing L (2010) A new text categorization method based on HMM and SVM. In: 2nd international conference on computer engineering and technology (ICCET), vol 7, pp 383–386

Fu JH, Lee SL (2012) A multi-class SVM classification system based on learning methods from indistinguishable chinese official documents. Expert Syst Appl 39(3):3127–3134

Hofmann T (1999) Probabilistic latent semantic indexing. In: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, ACM, pp 50–57

Joachims T (1998) Text categorization with support vector machines: learning with many relevant features. In: Proceedings of ECML-98, 10th European conference on machine learning (Chemnitz, DE), pp 137–142

Johnson DE, Oles FJ, Zhang T, Goetz T (2002) A decision-tree-based symbolic rule induction system for text categorization. IBM Syst J 41(3):428–437

Kim S-B, Rim H-C, Yook DS, Lim H-S (2002) Effective methods for improving naive bayes text classifiers. LNAI 2417:414–423

Li CH, Park SC (2009) n efficient document classification model using an improved back propagation neural network and singular value decomposition. Expert Syst Appl 36(2):3208–3215

Lin Y (2002) Support vector machines and the Bayes rule in classification. Data Min Knowl Discov 6:259–275

Mao X-L, Ming Z-Y, Chua T-S, Li S, Yan H, Li X (2012) SSHLDA: a semi-supervised hierarchical topic model. In: Proceedings of EMNLP-CoNLL, pp 800–809

McClosky D,Charniak E, Johnson M (2006) Effective self-training for parsing. In: Proceedings of NAACL, pp 152–159

Ng HT, Goh WB, Low KL (1997) Feature selection, perception learning, and a usability case study for text categorization. In: Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval, Philadelphia PA, pp 67–73

Petinot Y, McKeown K, Thadani K (2011) A hierarchical model of web summaries. Proc ACL HLT Short Pap Vol 2:670–675

Pham DT, Dimov SS, Nguyen CD (2005) Selection of K in K-means clustering. Proc Inst Mech Eng Part C J Mech Eng Sci 219(1):103–109

Qin Y-P, Wang X-K (2009) Study on multi-label text classification based on SVM. Sixth international conference on fuzzy systems and knowledge discovery, pp 300–304

Salton G, Wong A, Yan C-S (1975) A vector space model for automatic indexing. Commun ACM 18(11):613–620

Sebastiani F (2002) Machine learning in automated text categorization. ACM Comput Surv (CSUR) 34(1):1–47

Trappey AJC, Hsu F-C, Trappey CV, Lin C-I (2006) Development of a patent document classification and search platform using a back-propagation network. Expert Syst Appl 31(4):755–765

Turian J, Ratinov L, Bengio Y (2010) Word representations: a simple and general method for semi-supervised learning. In: Proceedings of the ACL, pp 384–394

Ueffing N (2006) Self-training for machine translation. In: Proceedings of NIPS workshop on machine learning for multilingual information access

Vateekul P, Kubat M (2009) Fast induction of multiple decision trees in text categorization from large scale, imbalanced, and multi-label data. IEEE International Conference on Data Mining Workshops, pp 320–325

Yang Y (1999) An evaluation of statistical approaches to text categorization. Inf Retr 1(1–2):69–90

Zhang Y, Vogel S, Waibel A (2004) Interpreting bleu/nist scores: how much improvement do we need to have a better system. In: Proceedings of the 2004 international conference on language resources and evaluation. pp 2051–2054