Active Learning with Query Generation for Cost-Effective Text Classification*

Yi-Fan Yan,[1,2] **Sheng-Jun Huang,**[1,2] **Shaoyi Chen,**[3] **Meng Liao,**[3] **Jin Xu**[3]
[1]College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, 211106
[2]MIIT Key Laboratory of Pattern Analysis and Machine Intelligence
[3]Data Quality Team, WeChat, Tencent Inc., China
{yanyifan7, huangsj}@nuaa.edu.cn, {shaoyichen, maricoliao, jinxxu}@tencent.com

Abstract

Labeling a text document is usually time consuming because it requires the annotator to read the whole document and check its relevance with each possible class label. It thus becomes rather expensive to train an effective model for text classification when it involves a large dataset of long documents. In this paper, we propose an active learning approach for text classification with lower annotation cost. Instead of scanning all the examples in the unlabeled data pool to select the best one for query, the proposed method automatically generates the most informative examples based on the classification model, and thus can be applied to tasks with large scale or even infinite unlabeled data. Furthermore, we propose to approximate the generated example with a few summary words by sparse reconstruction, which allows the annotators to easily assign the class label by reading a few words rather than the long document. Experiments on different datasets demonstrate that the proposed approach can effectively improve the classification performance while significantly reduce the annotation cost.

Introduction

Text classification is a fundamental task in many related applications, such as sentiment analysis (Cambria 2016; Lei and Bing 2011), news topic labeling (Hashimoto et al. 2016), spam detection (Moura et al. 2013; Jindal and Bing 2007), etc. It tries to assign semantic labels to a text based on its content. While many advanced classification models have been proposed (Joulin et al. 2016; Conneau et al. 2017), the performance still heavily depends on the training data. To train an effective model for text classification, it usually requires a large data set with high-quality labels, which are manually assigned by human annotators. To label a text document, the annotator need to read the whole document and check its relevance with each possible class label, which could be rather expensive and time consuming. It is thus an important challenge to achieve effective text classification with lower annotation cost.

Active learning, which iteratively selects the most valuable data examples to query their labels (Settles 2012), is a primary approach to reduce the annotation cost. It has been widely used in many tasks, such as image classification (Wang et al. 2017), document categorization (Beatty, Kochis, and Bloodgood 2018), etc. Although active learning has achieved great success in various applications, there are still some shortcomings by directly applying existing methods to text classification. Firstly, existing methods mainly focus on selecting the most important examples for reducing the number of queries, but less care about the annotation cost of each query. This is not a problem for image related tasks, where the annotator can correctly assign the label for an image after a quick glance. However, in text classification, to decide whether the document is related to a specific label, the annotators may need to carefully read the whole document. This may take several minutes or even hours for a long document. It is thus also important to reduce the cost for a single annotation, in addition to reducing the number of annotations. Secondly, existing methods select the queries from the unlabeled data pool by evaluating the utility of each example based on some criteria, which could be not efficient for the tasks with huge unlabeled data. In text classification, the unlabeled documents are easy to collect, leading to a large set from which the queries are selected. Selecting a single query from such a huge pool will be time consuming, which further limit the application of active learning in text related tasks.

To overcome the above discussed shortcomings, in this paper, we propose a novel Active Learning approach with Query Generation for text classification (ALQG). Instead of scanning every unlabeled example to select the best one for querying, the proposed method can automatically generate the most informative examples, which makes the selection process efficient and independent to the size of unlabeled pool. Further, the generated example is approximated by a few summary words, which are presented to annotators for labeling. In this way, the annotators can assign the class label with a much lower cost, by reading a few words instead of a long document.

*This research was supported by NSFC(61876081, 61732006, 61572252), the Aerospace Power Funds No. 6141B09050342 and the Collaborative Innovation Center of Novel Software Technology and Industrialization. Sheng-Jun Huang is the corresponding author.

Specifically, to generate the most valuable examples, we propose to incorporate the model uncertainty and distribution diversity in a generative model. On one hand, the uncertainty will push the generated examples close to the the decision boundary of the current classification model. The model are less certain about such examples, implying that adding them as training data may contribute more to the model improvement. On the other hand, the diversity will lead to a highly dispersed distribution of the generated examples over the whole boundary, avoiding redundant generations within a specific region. To reduce the annotation cost of each query, we further propose to approximate the generated example with a few summary words by sparse reconstruction from vocabularies. Experimental results on multiple datasets validate the superiority of the proposed approach for cost-effective text classification.

The rest of this paper is organized as follows. In Section 2, we introduce the related work. In Section 3, the propose approach ALQG is introduced. Section 4 presents the experiments, followed by the conclusion in Section 5.

Related Work

Text classification plays an important role in many real applications. It is typically solved by applying traditional supervised learning method to train the classifiers (Bo et al. 2016; Aggarwal and Zhai 2012). The effectiveness of text classification models mostly depends on the quality of training data (Sordo and Zeng 2005). But in real world tasks, the labeled data is limited while unlabeled data is massive, thus the annotation cost is rather high due to the heavy labeling tasks. Active learning aims to reduce the annotation cost by actively selecting the most valuable instances to be queried. Zhu.et al query the most uncertain instance of model, which is excepted to improve the performance of classification model most (Zhu et al. 2010). Huang.et al combine both the informativeness and representativeness criteria to measure the value of unlabeled data (Huang, Jin, and Zhou 2014).

Active learning has been widely used in text classification (Cormack and Grossman 2016; Schmidt, Schnitzer, and Rensing 2016; Yang et al. 2009), among which, uncertainty sampling is the most frequently used (Rong, Namee, and Delany 2016; Huang and Zhou 2013). Methods combining multiple criteria, such as informativeness and representativeness, are also used in active learning for text classification(Zhu et al. 2008). What's more, Li.et al propose a novel active learning approach in multi-domain text classification by considering both the common and the domain-specific features together (Li et al. 2012).

Most existing active learning methods select the instances from unlabeled data pool to query the oracle for the class label (Cormack and Grossman 2016; Xie and Huang 2019). Recently, a few studies try to generate new instances and annotate the new instances directly, which would be then added to training data. Zhu.et al generate new instances located in decision boundary by gradient descent algorithm, and use the GAN model to visualize the instances (Zhu and Bento 2017). This method, however, is designed for image classification, and the generated images may be ambiguous and hard to annotate even for human experts. Ducoffe et.al. use deepfool method to measure the distance of instances to the decision boundary (Ducoffe and Precioso 2018), and query the instances mostly near the boundary. By share the label with deepfool instances, both real data and generated deeofool data are added to training set. All these methods are applied in image classification tasks, which cannot be directly applied to text classification. Moreover, the annotation cost in text classification increases when the text documents become longer, while labeling different images may cost approximately the same.

The Proposed Approach

Let $\mathcal{D} = \{(\boldsymbol{o}_i, y_i), ..., (\boldsymbol{o}_n, y_n)\}$ be the initial labeled data consists of n instances, where each instance $\boldsymbol{o}_i$ is a d-dimensional feature vector, y_i is the label of $\boldsymbol{o}_i$. We employ the SVM model $f(\boldsymbol{o}) = \boldsymbol{w}_0\phi(\boldsymbol{o}) + b_0$ as the text classifier, where $\phi(\cdot)$ is a feature mapping function, $\boldsymbol{w}_0$ and b_0 are the model parameters after training. At each iteration of active learning, the task is to generate a batch of new instances for query, and add them to the labeled set after annotation for updating the model f. In the following subsections, we will first introduce how to generate the most valuable instances, and then present the method for summary words approximation, followed by the process of instance annotation and model updating.

Query Generation

In this subsection, we propose to generate the most informative and diversity instances based on the current classification model. Instead of scanning each instance of the unlabeled data pool and select the best one according to a specific criterion, we directly generate the instances that will contribute the model most. Uncertainty has been shown to be an effective criterion to estimate the informativeness of an instance (Zhu et al. 2010). Intuitively, if the current classifier is less confident about the prediction of an instance, then adding this instance to the training set may reduce the model uncertainty more. Formally, we assume to generate a small batch of b instances at each iteration, denoted by $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_b\}$. Under the current model f, the most uncertain instances should be close to the decision boundary of the classifier. The objective function can be formalized as follows:

$$\min \sum_{i=1}^{b} \parallel \boldsymbol{w}_0\phi(\boldsymbol{x}_i) + b_0 \parallel^2 . \tag{1}$$

Here $\boldsymbol{w}_0$ and b_0 are known parameters for the current model, while $\boldsymbol{x}_i$ is the instance to be generated. Noticing that we generate a batch of instances in each iteration, the instances generated as in Eq. (1) may be concentratedly distributed in a specific region. Querying such high similar instances will lead to redundant information and the waste of annotation. Motivated by this, we proposed to enhance the diversity between the generated instances. Specifically, we measure the diversity by the divergence within the generated set as in Eq. (2):

$$J_w = \sum_{i=1}^{b} (\phi(\boldsymbol{x}_i) - \boldsymbol{\mu}_x)^T (\phi(\boldsymbol{x}_i) - \boldsymbol{\mu}_x), \tag{2}$$

where $\boldsymbol{\mu}_x = \frac{1}{b}\sum_{i=1}^{b}\phi(\boldsymbol{x}_i)$ is the mean of generated instances. By expending the formula, Eq.(2) can be simplified as:

$$\begin{aligned}
J_w &= \sum_{i=1}^{b}(\phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_i) - 2\phi(\boldsymbol{x}_i)^T\boldsymbol{\mu}_x + \boldsymbol{\mu}_x^T\boldsymbol{\mu}_x) \\
&= \sum_{i=1}^{b}\phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_i) - 2b\boldsymbol{\mu}_x^T\boldsymbol{\mu}_x + b\boldsymbol{\mu}_x^T\boldsymbol{\mu}_x \\
&= \sum_{i=1}^{b}\phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_i) - \frac{1}{b}\sum_{i=1}^{b}\phi(\boldsymbol{x}_i)^T\sum_{j=1}^{b}\phi(\boldsymbol{x}_j) \\
&= \sum_{i=1}^{b}K(\boldsymbol{x}_i,\boldsymbol{x}_i) - \frac{1}{b}\sum_{i,j}K(\boldsymbol{x}_i,\boldsymbol{x}_j) \\
&= tr(K^1) - \frac{1}{b}\sum_{i,j}K^1_{ij}, \qquad (3)
\end{aligned}$$

where $K^1_{ij} = K(\boldsymbol{x}_i,\boldsymbol{x}_j)$ denotes the kernel matrix of the generated data.

By minimizing Eq. (1) and maximizing Eq. (3), the generated instances will be diversely distributed along the decision boundary. However, the decision boundary is unlimited, while the distribution range of real data is limited in a specific region of the feature space. To well represent the other data examples, and alleviate the negative effect of outliers, we want the generated examples to have a similar distribution with observed data. For simplicity, we try to minimize the divergence between the generated examples and the initial labeled data, which is formalized as in Eq. (4).

$$J_b = (\boldsymbol{\mu}_x - \boldsymbol{\mu})^T(\boldsymbol{\mu}_x - \boldsymbol{\mu}), \qquad (4)$$

where $\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n}\phi(\boldsymbol{o}_i)$ is the mean of the initial observed data, and $\boldsymbol{\mu}_x$ is defined the same as in Eq. (4). Similarly, we can also simplify Eq.(4) as:

$$\begin{aligned}
J_b &= \boldsymbol{\mu}_x^T\boldsymbol{\mu}_x - 2\boldsymbol{\mu}_x^T\boldsymbol{\mu} + \boldsymbol{\mu}^T\boldsymbol{\mu} \\
&= \frac{1}{b^2}\sum_{i=1}^{b}\phi(\boldsymbol{x}_i)^T\sum_{j=1}^{b}\phi(\boldsymbol{x}_j) - \frac{1}{bn}\sum_{i=1}^{b}\phi(\boldsymbol{x}_i)^T\sum_{j=1}^{n}\phi(\boldsymbol{o}_j) \\
&\quad + \frac{1}{n^2}\sum_{i=1}^{n}\phi(\boldsymbol{o}_i)^T\sum_{j=1}^{n}\phi(\boldsymbol{o}_j) \\
&= \frac{1}{b^2}\sum_{i,j}K(\boldsymbol{x}_i,\boldsymbol{x}_j) - \frac{2}{bn}\sum_{i,j}K(\boldsymbol{x}_i,\boldsymbol{o}_j) \\
&\quad + \frac{1}{n^2}\sum_{i,j}K(\boldsymbol{o}_i,\boldsymbol{o}_j) \\
&= \frac{1}{b^2}\sum_{i,j}K^1_{ij} - \frac{2}{bn}\sum_{i,j}K^3_{ij} + \frac{1}{n^2}\sum_{i,j}K^2_{ij}, \qquad (5)
\end{aligned}$$

where $K^2_{ij} = K(\boldsymbol{o}_i,\boldsymbol{o}_j)$ is the kernel matrix for the initial data, and $K^3_{ij} = K(\boldsymbol{x}_i,\boldsymbol{o}_j)$ is the kernel matrix between generated data and initial data. Note that although we use the labeled data to measure the distribution difference in our implementation for simplicity, other strategies may further improve the performance. For example, one may randomly sample a subset from the unlabeled data, or select some representative examples based on clustering.

In summary, we try to generate a batch of examples that are most uncertain according to the current model, diverse from each other, and can well represent the other data examples. By incorporate the three aspects together, we have the final objective function as in Eq. (6).

$$\min\sum_{i=1}^{b} \parallel \boldsymbol{w}_0\phi(\boldsymbol{x}_i) + b_0 \parallel^2 + \lambda\frac{J_b}{J_w}, \qquad (6)$$

where λ is a hyperparameter to trade-off the two terms. To solve this problem, we employ a simple neural network with single hidden layer to optimize the objective function in Eq. (6). By training the neural network with random inputs according to Eq.(6), the output layer can generate a batch of instances with d-dimensional features. The details of the neural network are introduced in the experiments.

Summary Words Approximation

The generated examples in the previous subsection are expected to have high utility for improving the classification model if they are added into the training set after labeling. However, it is still a challenge to correctly assign the class label to them with low annotation cost. Firstly, the examples are generated in the feature space, which implies they are in the form of d-dimensional feature vectors instead of the original text. These feature vectors cannot be understood by human annotators, and thus cannot be correctly labeled. One possible way is to recover the original text by utilizing the nearest neighbors in the real data. However, even in this way, the annotator need to read a long document to decide the related class labels, which could be very time consuming.

To overcome this challenge, we propose to approximate the generated example with a few summary words via sparse reconstruction from the vocabularies. Specifically, each generated instance is expected to be represented by a weighted linear combination of the words in the vocabulary, while the number of words with non-zero weights should be as small as possible. We denote the vocabulary by $W = [\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_q]^\top$ with q words, where each word $\boldsymbol{w}_j$ is also represented by a d-dimensional vector within the same feature space as the text documents.

For each generated instance $\boldsymbol{x}_i$, the sparse reconstruction is implemented by minimizing the objective function in Eq. (7):

$$\begin{aligned}
&\min_{\boldsymbol{\alpha}} \frac{1}{2} \parallel W^T\boldsymbol{\alpha} - \boldsymbol{x}_i \parallel^2 + \eta||\boldsymbol{\alpha}||_1 \qquad (7) \\
&s.t. \sum_{i=1}^{q}\alpha_i = 1, \quad 0 \le \alpha_i \le 1
\end{aligned}$$

where $\boldsymbol{\alpha} =$ is a q-dimensional vector, and $\boldsymbol{\alpha}_i$ is the importance weight of word $\boldsymbol{w}_i$. The first term minimizes the reconstruction error such that the generated instance can be well approximated by the summary words. The second term enhances the sparsity such that the annotators only need to

Algorithm 1 The ALQG Algorithm

Input: initial labeled data D_l, vocabularies
Process:
Training the initial classifier f with the labeled data D_l
Initialize the training data $D = D_l$
repeat
  Generate b new instances according to Eq.(6)
  for each $\boldsymbol{x}_i$ in generated set **do**
    Approximate $\boldsymbol{x}_i$ with summary words according to Eq.(7)
    Get the class label y_i according to Eq.(10)
  end for
  Update the training data D
  Update the classifier f with training data D
until The maximum query times is reached

read a few words to decide the class label. η is the parameter to balance the two terms. In our implementation, we set $\eta = 1$ as default for all datasets. The sparse reconstruction task in Eq. (7) can be further reformulated as follows.

$$\arg\min_{\boldsymbol{\alpha}} \frac{1}{2}\boldsymbol{\alpha}^\top M\boldsymbol{\alpha} + \boldsymbol{\beta}^\top\boldsymbol{\alpha} \tag{8}$$
$$s.t. \sum_{i=1}^{q} \alpha_i = 1, \quad 0 \le \alpha_i \le 1$$

where

$$M = W^\top W$$
$$\boldsymbol{\beta} = -W\boldsymbol{x}_i^\top + \mathbf{1} \tag{9}$$

where $\mathbf{1}$ is a vector with all the element is one. We select the top k words with largest weights as the summary words, and present the them to the annotators to query their labels. It is observed in our experiments that the summary words can be easily understood and labeled by the human annotators.

Model Updating

In real tasks, after the instance generation and summary words approximation, the annotators can easily label the examples by reading the summary words. In the experiments, if we query the ground-truth labels for the generated examples (which are not available for existing datasets), we need a human annotator to standby during the whole experiments. To facilitate the implementation of experiments, we propose a simple strategy to simulate the annotator. Firstly, for each label y, we collect a set $C_y = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_m\}$ consists of m keywords extracted from the text documents with label y_l. Then we calculate the average distance between the k summary words and the m keywords for each label. After that, the label with smallest distance is assigned as the label for the summary words as well as the corresponding generated instance. Specifically, the label is decided by Eq. (10).

$$y^* = \arg\min_{y} \frac{1}{k \cdot |C_y|} \sum_{i=1}^{k} \sum_{\boldsymbol{v}_j \in C_y} \alpha_i \cdot Dist(\boldsymbol{w}_i, \boldsymbol{v}_j), \tag{10}$$

where $Dist(\cdot,\cdot)$ is the cosine distance function defined as follows:

$$D(\boldsymbol{w}_i, \boldsymbol{v}_j) = -\frac{< \boldsymbol{w}_i, \boldsymbol{v}_j >}{||\boldsymbol{w}_i|| \cdot ||\boldsymbol{v}_j||}. \tag{11}$$

In our experiments, we use this simulated annotator to assign the labels for generated instances for our method, while other compared methods query the ground-truth labels. Obviously, this setting is unfair to our method because the simulated annotator could be noisy. But even with this unfair setting, our method still achieves the best performance, as demonstrated in the experiments. It could be expected that the superiority of our method will be more significant if it also queries from the human annotator in a real task.

After the annotation of summary words, the label is then assigned to the generated instance, and subsequently the new instance can be added into the training set to update the classification model. Note that the summary words share the same feature space with the text documents, and thus may also be used in training the model without extra annotation cost. The pseudo code of the proposed algorithm is summarized in Algorithm 1.

Experiments

Settings

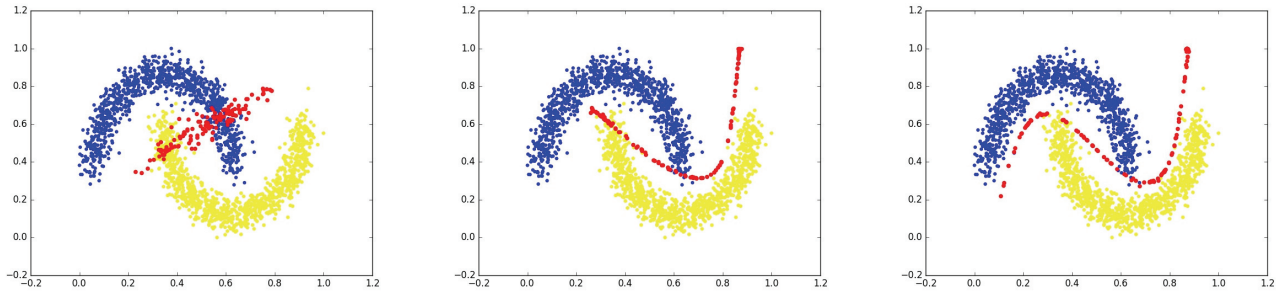
In the experiments, we set the hyperparameter $\lambda = 5$ as default for all datasets, the batch size is set as $b = 20$. A single hidden-layer neural network is employed for query generation. The number of nodes is 10 for the input layer, and 200 for the hidden layer. For the output layer, the number of nodes equals to the dimensionality of the feature space. The tensorflow framework is used to train the neural network. We use SVM as the baseline classification model, RBF as the kernel function, and evaluate the classification performances of the compared approaches with AUC and Accuracy.

On each data set, we randomly sample 20% of the examples as the test set. Then from the remaining 80% data, 1% examples are further sampled as the initial labeled data. The data partition is repeated for 10 times, and the average results are reported.

The following methods are compared in our experiments:

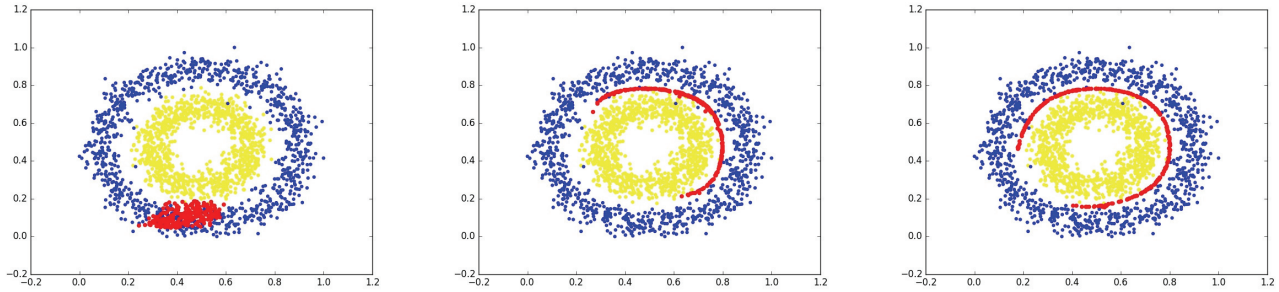
- **ALQG**: The active learning with query generation approach proposed in this paper.
- **ALQG-NN**: A degenerated version of the proposed method. It use the same generation model, but does not perform the summary words approximation. Instead, it select the nearest documents from the unlabeled data for the generated instances, and then query their labels.
- **Uncertainty**: The instances closest to the decision boundary are selected to query their labels.
- **Random**: Randomly selects instances to query their labels.

All the methods have the same batch size and use the same classification model. Note that ALQG queries from the simulated annotator while the other methods query the ground-truth labels. Although this setting is less fair to the proposed method, it still achieves the best performances. This also



(a) iteration=1 (b) iteration=25000 (c) iteration=50000

Figure 1: The generated instances on moon. The blue and yellow dots are training data, the red dots are generated instances.



(a) iteration=1 (b) iteration=50000 (c) iteration=100000

Figure 2: The generated instances on circles. The blue and yellow dots are training data, the red dots are generated instances.

implies that the superiority of our method could be greater when it also queries from an oracle for the ground-truth labels.

Visualization of Generated Instances

We first perform the experiments on two synthetic data to examine the effectiveness of the query generation model. The Moon and Circles datasets are used in the experiments. For each dataset, we generate 1000 instances with 2-dimention as the training set to initialize the classification model. Then we try to generate a batch of 128 instances with the proposed method. Figures 1 and 2 show the visualization results on the two datasets respectively. The blue and yellow dots represent the positive and negative examples of training data, while the red dots are the generated examples. The subfigures plot how the distribution of generated examples changes as the number of training iterations grows. At the initial phase, the generated instances are distributed within a specific region. After enough iterations of training, the generated instances are close to the decision boundary with a dispersed distribution along the real data. These results validate that the generated instances well fits the expected properties. In other words, they are informative to the model, diverse to each other, and can well represent the data distribution.

Comparison of Classification Performance

Then we perform the experiments on several real world datasets. IT-vs-Learning, Healthy-vs-Auto, Culture-vs-Military are three Chinese public datasets for binary text classification[1] (Wang et al. 2008). World-vs-Sports is an English public dataset[2] (Zhang, Zhao, and Lecun 2015). The numbers of instances are 20251, 11664, 6088 and 63800, respectively. The average words of each text document are 366, 439, 392 and 31.

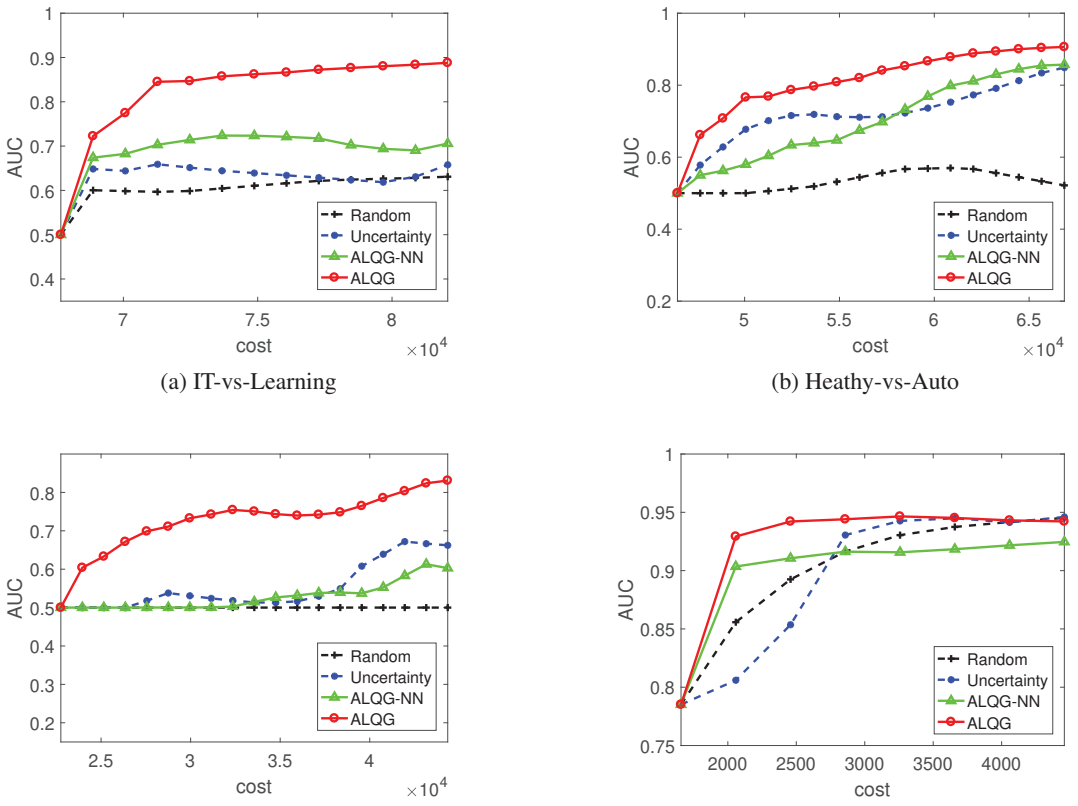
In feature engineering process, we first use the "Jieba" text segmentation toolbox[3] to cut the whole text, abandon the stopwords, and compute the average word embedding vectors of each of the rest words as feature vectors. For English datasets, the text segmentation process can be omitted. Note that the word embedding is implemented with a pre-trained model[4]. In summary words approximation, k is set to 20 for Chinese vocabulary and 5 for English vocabulary. For each label, we use textrank method (Mihalcea and Tarau 2004) to extra $m = 20$ keywords as the vocabulary, and

[1] http://www.sogou.com/labs/resource/list_pingce.php

[2] https://github.com/mhjabreel/CharCNN/tree/master/data

[3] https://github.com/fxsjy/jieba

[4] https://github.com/Embedding/Chinese-Word-Vectors,https://nlp.stanford.edu/data/wordvecs



(a) IT-vs-Learning (b) Heathy-vs-Auto

(c) Culture-vs-Military (d) World-vs-Sports

Figure 3: The Performance on AUC.

share the same feature space with word embedding vectors. Since the annotators need to read the whole document to decide the class labels, we count the number of words in a document as the annotation cost for this example.

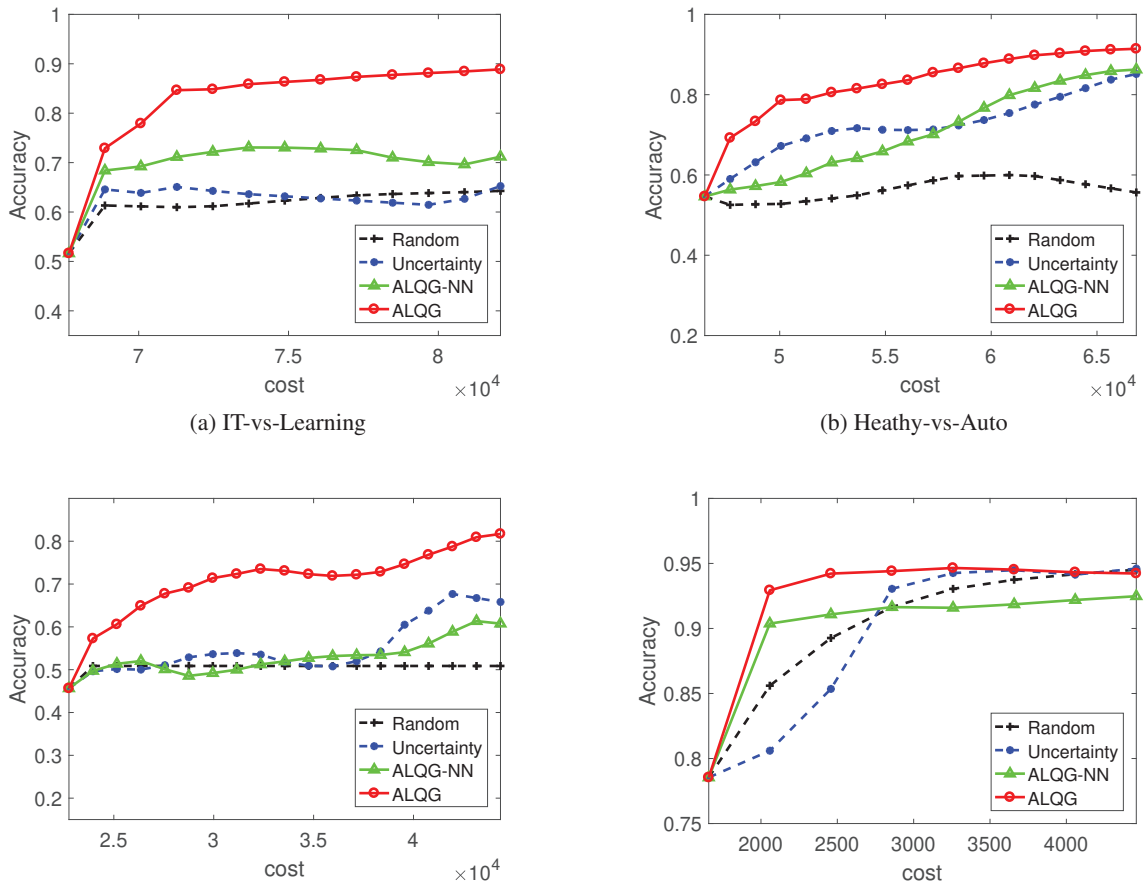
The performance curves with the annotation cost increasing are plotted in Figure 3 and Figure 4. We can observe that our approach ALQG achieves the best performance with a significant superiority in most cases. As expected, the random method has the worst performance. The Uncertainty and ALQG-NN method are more effective than random method while worse than the proposed method. ALQG is always better than ALQG-NN, validating the effectiveness of the summary words approximation of the proposed method. The performances of ALQG-NN is comparable or better than Uncertainty, which indicates the effectinveness of the proposed query generation model. Due to the less training data in the early training process, the curves shock in the beginning in Culture-vs-Military. The comparison performance further shows the importance of querying the summary words instead of the original document.

Results of Summary Words

In this subsection, we present some examples to show that the summary words produced by our method can be easily understood for human annotation. We translate the Chinese into English for easily understanding. Due to the space limitation, Table 1 shows two examples of summary words for each of the four categories, each column corresponding to the top 10 weights of one generated instance. The first row presents the annotated label for the summary words. From the table we can observe that the reconstructed words share some similar semantic and show a significant bias to one of the class label, which helps the annotators to easily decide the label for the generated instance.

Conclusion

In this paper, we propose an active learning approach with query generation for cost-effective text classification. Instead of scanning all the instances of unlabeled data pool, the propose method automatically generate informative and diverse instances. We also propose a sparse reconstruction model to approximate the generated instance with a few summary words, which are much more easy for the annotators to label than a long document. The proposed approach on one hand can efficiently generate queries independently of the size of unlabeled data, and on the other hand can reduce the annotation cost of each query significantly. Experimental results on different datasets demonstrate that our approach can improve the performance effectively with much lower annotation cost. In the future, the proposed approach



(a) IT-vs-Learning (b) Heathy-vs-Auto

(c) Culture-vs-Military (d) World-vs-Sports

Figure 4: The Performance on Accuracy.

Table 1: Summary words of examples from four categories produced by sparse reconstruction.

IT		Learning		Culture		Military	
example 1	example 2	example 1	example 2	example 1	example 2	example 1	example 2
semiconductor	acquisition	translate	archives	film	cartoonist	develop	guided bomb
computation	occupy	writing	test	opera	doctrine	naval vessels	thruster
acquisition	digital	insist	college	director	portray	military exercise	antisubmarine
process	package	reading	examine	childhood	opera	flight crew	Pacific
user-traffic	tariff	author	tuition	Yan'an	poet	defense forces	airfreighter
architecture	contribution	article	training	drama	song	sea area	rapid response
server	architecture	spirit	student	speak	topic	simulator	headquarters
commercial	mobile	picture	author	strong	wearing	identify	hard-hit
industry-chain	base-station	text	score	watch	comic	ballistic missile	commandos
composition	evaluation	class	sectarian	actor	intellectual	intercept	armed forces

will be extended to the multi-class tasks.

References

Aggarwal, C. C., and Zhai, C. 2012. A survey of text classification algorithms. In *Mining Text Dzhuata*. Springer. 163–222.

Beatty, G.; Kochis, E.; and Bloodgood, M. 2018. Impact of batch size on stopping active learning for text classification. In *IEEE International Conference on Semantic Computing*.

Bo, T.; He, H.; Baggenstoss, P. M.; and Kay, S. 2016. A bayesian classification approach using class-specific features for text categorization. *IEEE Transactions on Knowledge and Data Engineering* 28(6):1602–1606.

Cambria, E. 2016. Affective computing and sentiment analysis. *IEEE Intelligent Systems* 31(2):102–107.

Conneau, A.; Schwenk, H.; Barrault, L.; and Lecun, Y. 2017. Very deep convolutional networks for text classification. *Conference of the European Chapter of the Association for Computational Linguistics* 1:1107–1116.

Cormack, G. V., and Grossman, M. R. 2016. Scalability of continuous active learning for reliable high-recall text classification. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, 1039–1048. ACM.

Ducoffe, M., and Precioso, F. 2018. Adversarial active learning for deep networks: a margin based approach. *arXiv: Learning*.

Hashimoto, K.; Kontonatsios, G.; Miwa, M.; and Ananiadou, S. 2016. Topic detection using paragraph vectors to support active learning in systematic reviews. *Journal of Biomedical Informatics* 62(C):59–65.

Huang, S.-J., and Zhou, Z.-H. 2013. Active query driven by uncertainty and diversity for incremental multi-label learning. In *2013 IEEE 13th International Conference on Data Mining*, 1079–1084. IEEE.

Huang, S.-j.; Jin, R.; and Zhou, Z.-h. 2014. Active learning by querying informative and representative examples. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36(10):1936–1949.

Jindal, N., and Bing, L. 2007. Review spam detection. In *International Conference on World Wide Web*.

Joulin, A.; Grave, E.; Bojanowski, P.; and Mikolov, T. 2016. Bag of tricks for efficient text classification. *arXiv:1607.01759*.

Lei, Z., and Bing, L. 2011. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies* 30(1):167.

Li, L.; Jin, X.; Pan, S. J.; and Sun, J. T. 2012. Multi-domain active learning for text classification. In *Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*.

Mihalcea, R., and Tarau, P. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 404–411.

Moura, G. C. M.; Sperotto, A.; Sadre, R.; and Pras, A. 2013. Evaluating third-party bad neighborhood blacklists for spam detection. *IFIP/IEEE International Symposium on Integrated Network Management* 252–259.

Rong, H.; Namee, B. M.; and Delany, S. J. 2016. Active learning for text classification with reusability. *Expert Systems with Applications* 45(C):438–449.

Schmidt, S.; Schnitzer, S.; and Rensing, C. 2016. Text classification based filters for a domain-specific search engine. *Computers in Industry* 78(C):70–79.

Settles, B. 2012. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan and Claypool Publishers.

Sordo, M., and Zeng, Q. 2005. On sample size and classification accuracy: A performance comparison. In *Biological and Medical Data Analysis, International Symposium, Isbmda, Aveiro, Portugal, November*.

Wang, C.; Min, Z.; Ma, S.; and Ru, L. 2008. Automatic online news issue construction in web environment. In *International Conference on World Wide Web*.

Wang, K.; Zhang, D.; Li, Y.; Zhang, R.; and Liang, L. 2017. Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology* PP(99):1–1.

Xie, M.-K., and Huang, S.-J. 2019. Learning class-conditional gans with active sampling. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 998–1006. ACM.

Yang, B.; Sun, J. T.; Wang, T.; and Chen, Z. 2009. Effective multi-label active learning for text classification. In *Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*.

Zhang, X.; Zhao, J. J.; and Lecun, Y. 2015. Character-level convolutional networks for text classification. *Neural Information Processing Systems* 649–657.

Zhu, J., and Bento, J. 2017. Generative adversarial active learning. *arXiv: Learning*.

Zhu, J.; Wang, H.; Yao, T.; and Tsou, B. K. 2008. Active learning with sampling by uncertainty and density for word sense disambiguation and text classification. *Coling* 1:1137–1144.

Zhu, J.; Wang, H.; Ma, M.; and Ma, M. 2010. Active learning with sampling by uncertainty and density for data annotations. *IEEE Transactions on Audio Speech and Language Processing* 18(6):1323–1331.