

# Knowledge Transfer for Feature Generation in Document Classification

Jian Zhang  
Department of Computer Science  
Louisiana State University  
Baton Rouge, LA 70803  
zhang@csc.lsu.edu

Shobhit S. Shakya  
Department of Computer Science  
Louisiana State University  
Baton Rouge, LA 70803  
sshaky2@tigers.lsu.edu

**Abstract**—One important problem in machine learning is how to extract knowledge from prior experience, then transfer and apply this knowledge in new learning tasks. To address this problem, transfer learning leverages information from (supervised) learning on related tasks to facilitate the current learning task. Self-taught learning uses information extracted from (unsupervised) learning on related data. In this paper, we propose a new method for knowledge extraction, transfer and application in classification. We consider document classification where we mine correlation relationships among the words from a set of documents and compile a collection of correlation relationships as prior knowledge. This knowledge is then applied to generate new features for classifying documents in classes/types different from the ones which we obtain the correlation relationships from. Our experiment results show that the correlation-based knowledge transfer helps to reduce classification errors.

**Keywords**-Feature generation; Knowledge transfer; Classification; Correlation

## I. INTRODUCTION

One important aspect of human learning process is that when we deal with a learning task, we not only consider the examples from this task, but also apply our knowledge and experiences gained in the past to facilitate the learning process. For example, previous training in basketball may help to learn volleyball. Prior knowledge and experiences provide advantages particularly when a new situation is encountered, or the available training sample size is too small.

In recent years, machine learning community has been researching ways to incorporate prior knowledge and experiences in machine learning process. Transfer learning [1], [2], [3], [4] try to leverage the information gained in (supervised) learning on some tasks and use it to facilitate the (supervised) learning process of other related tasks. “Self-taught” learning [5] applies the information derived from (unsupervised) learning on related data to facilitate the current (supervised) learning task. Despite the difference between these learning paradigms, the main underlying idea is similar: we want to transfer the knowledge obtained from previous related tasks/data to the current learning task such that the current learning process can be enhanced. (To distinguish from the concepts of transfer learning and self-taught

learning, we call this *general* form of knowledge transfer and application the *learning with knowledge transfer*.) There are three central problems in learning with knowledge transfer: 1) how to extract the prior related knowledge, 2) how to represent the knowledge and 3) how to apply the knowledge in new learning task.

For example, in some transfer learning approach [6], the prior knowledge is modeled as a prior distribution on the parameters of the classifier. This distribution is obtained from the learning process on related tasks/data. When learning a new classification task, the classifier is constructed using both the new training data and the prior distribution. Another example is the self-taught learning [5]. In this scenario, the prior knowledge is modeled as a specific encoding of the data (e.g. a set of basis for encoding the data represented in a high dimensional space). The encoding is obtained from the prior related data. The new data for the new learning task are then represented using this encoding before they are presented to the training algorithm.

In this paper, we propose a novel method for learning with knowledge transfer. In this method, we consider the knowledge on the correlation among the attributes of the data items. We call a group of correlated attributes the *correlated group*. We extract the correlated groups using correlation mining techniques and then employ them to generate new features for the data in the new learning task. For example, “moon”, “earth” and “sun” may often appear together in documents. They can form the concept “solar system”, which can be viewed as an abstraction of the specific star and planet bodies. (Note that we are not mining concepts. Rather, we view each correlated group as a “concept” which may or may not have any interpretable semantic meaning.) Once we extract the concept, we can utilize this knowledge in new document classification. One way of applying this knowledge is to insert a new feature corresponding to the “solar system” concept into the feature space of a document if it contains some words in the correlated group. The “solar system” concept can express in different example documents as different combination of words. Therefore the specific word may appear as not important for identifying a type of documents. However, if the concept is important for this type of document, the new feature corresponding to the

concept could be more consistent among the examples and help classification.

On the other hand, not all generated features are helpful. In fact, some features may be wrongly generated. Therefore, feature selection is an inseparable part of the feature generation process in our method. We apply standard feature selection before classification and use L1-regularization to further select features at the time of classification.

We consider knowledge-transfer-based feature generation for document classification. From a collection of documents, we mine the correlated groups and then use these groups to generate new features. The new features are used in the classification of new documents outside the collection from which the correlated groups are mined. We compare classification of the new documents with and without the (selected) new features. Experiments show that these features help to improve the classification performance.

The rest of the paper is organized as follows. In Section II, we review and compare our method with other existing work. In Section III, we present the details of our knowledge extraction and feature generation. We show and discuss the experiment results in Section IV and finally, Section V concludes the paper and discusses some potential future improvements.

## II. RELATED WORK

In Bayesian learning [7], prior knowledge is often modeled as a prior distribution on the parameters of the model (e.g., classifier) for the learning task. The final model depends on the posterior distribution of the parameter which is determined both by the training data and by the prior distribution. However, in most cases, the simple prior distribution cannot provide the granularity needed to express the complexity in the prior knowledge.

Our method also differs from transfer learning [1], [2], [3], [4] in that the knowledge extraction is done through unsupervised learning (whereas transfer learning uses supervised learning). From this point of view, our method is closer to the self-taught learning [5]. However, the knowledge extraction and application in our method is very different from those in self-taught learning. In particular, we use correlation mining techniques to obtain prior knowledge and use it to generate new features for classification. These aspects are not presented in self-taught learning.

Gabrilovich et. al. [8] proposed a framework to use world knowledge in generating features. Our method shares some common ground with their framework in that the application of the prior knowledge is to generate new features. On the other hand, there are two important differences between our work and the one in [8], [9]: First, in [8], [9], the prior knowledge is obtained from an existing semi-ontology such as the category provided by the open directory project(ODP) while in our method, the prior knowledge is the correlation relationships mined from the collection of related

documents. Second, although some new features may reveal certain intrinsic properties of the document class and thus help the classification, there are others that also introduce noises. Therefore, performing feature selection on the newly generated features is very important. Our work differs from [8], [9] in feature selection by using L1-regularization to automatically pick the good features.

## III. LEARNING WITH KNOWLEDGE-TRANSFER-BASED FEATURE GENERATION

There are two components in our method: knowledge extraction and knowledge application. We discuss the details of the two components in the following two subsections.

### A. Mining Correlated Groups

Ideally, if we have an encyclopedia of all the concepts involved in the data, it will provide great help for knowledge transfer and feature generation using our method. However, it is very difficult to obtain such an encyclopedia. Note that even an ontology would not be sufficient for our purpose because we are exploring not the “is-a” hierarchical structure but rather the correlation relationship. For example, consider the concept of online shopping. It consists making the order, processing payment and shipping the goods. Each step is a part of the process and together they form the shopping online concept. There is no “is-a” relation between the concept and its constituents. Therefore, we focus on the correlation relationship and extract the correlated groups as our prior knowledge.

For document classification, the correlated groups are groups of words. A pair of words is said to be correlated if they appears in many documents together. Formally, let  $D(a)$  be the set of documents that contains the word  $a$  and  $D(b)$  be the set of documents that contains the word  $b$ . Let  $I(a, b) = D(a) \cap D(b)$  and  $U(a, b) = D(a) \cup D(b)$ . If  $\frac{|I(a,b)|}{|U(a,b)|} > \tau$  for a pre-specified parameter  $\tau$  ( $|\cdot|$  is the size of the set), we say that the two words  $a$  and  $b$  are correlated and denote by  $a \sim b$ . A set of words  $S = \{a_1, a_2, \dots, a_k\}$  is correlated if for every pair  $a_i, a_j \in S$ ,  $a_i \sim a_j$ . Each group of correlated words may represent a certain concept. A document may involve a particular concept if it contains some (not necessarily all) of the words in the corresponding group.

To mine the correlated group, we introduce a two-stage algorithm. The algorithm is efficient and scalable even with respect to extremely large collection of documents and terms. It only assumes that the size of the correlated groups are not too large.

At the first stage of the algorithm, we use a min-hash based technique [10] to identify the potential correlated pairs. The goal of this stage is to remove most non-correlated pairs and thus reduce the amount of candidates we need to go through in searching for the correlated groups. Let  $D$  be the whole collection of documents. A min-hash function  $h_{min}$

maps a word to a number and has the following property: Given two words  $a$  and  $b$ ,

$$p(h_{\min}(a) = h_{\min}(b)) = \frac{I(a,b)}{U(a,b)}$$

The following is a simple min-hash function from [10]: Let  $h$  be a general hash function that maps a number  $i$  in  $[m]$  to a random number  $j$  in  $[m^2]$ . (Suppose there are  $m$  documents in  $D$ . The hash maps the  $i$ -th document to the number  $j$ .) Then

$$h_{\min}(a) = \min_{d \in D(a)} \{h(d)\}$$

With min hashing, the larger the ratio  $\frac{|I(a,b)|}{|U(a,b)|}$ , the more likely it is that the two words  $a$  and  $b$  will be hashed to the same value.

We further use another technique from [10], [11], [12] to obtain a large gap of probability between the pairs whose  $\frac{|I(a,b)|}{|U(a,b)|} > \tau$  and the ones whose  $\frac{|I(a,b)|}{|U(a,b)|} \leq \tau$ . We use  $k$  independent min-hash functions and define an equivalence relation “ $\simeq$ .” For two items  $a$  and  $b$ ,  $a \simeq b$  if and only if  $a$  and  $b$  have the same min-hash values for all the  $k$  hash functions. The equivalence relation can be used to partition the items into equivalence classes. If, with one min-hash function  $p(a \simeq b) = x$ , then, with  $k$  independent functions,  $p(a \simeq b) = x^k \ll x$ . We repeat the whole process  $t$  times, each time with a different set of  $k$  min-hash functions. The probability that  $a$  and  $b$  belong to the same equivalence class in at least one of the trials is  $1 - (1 - x^k)^t$ . The function  $f(x) = 1 - (1 - x^k)^t$  is “s” shaped and approximates a threshold function.  $f(x)$  takes the value close to 1 when  $x$  is larger than the threshold and close to 0 when  $x$  is smaller than the threshold. One may set the parameters such that the pairs above threshold will almost surely be recognized and a few pairs below threshold may also be picked by the hashing result. The truly correlated pairs can be identified by actually calculating  $\frac{|I(x,y)|}{|U(x,y)|}$  for the ones that are picked by the hash scheme. Such verification process is efficient because min-hash removes most of the pairs that are not correlated.

After identifying the correlated pairs, We use a a depth-first search to discover the correlated groups of words. One may construct a graph based on the mined correlation from the first step. The nodes in the graph correspond to the words and there is an edge between two words if they are correlated. The correlated groups are the cliques in this correlation graph. We assume that the correlated groups all have small size, i.e., those cliques are small. Therefore, a depth-first brute-force search can be used to identify all the small cliques in the correlation graph. The set of words corresponding to each clique forms a correlation group.

### B. Classification with Extended Features

We consider two-class classification. An L1-regularized logistic regression model [15], [14] is used for the classification. Let  $\mathbf{x}^{(i)} \in \mathbb{R}^M$  be the  $M$ -dimensional feature vector

of the  $i$ -th document and  $y^{(i)} \in \{-1, 1\}$  be the class label of the document. Logistic regression models the probability distribution of the document’s label based on its features. In particular,

$$p(y = 1 | \mathbf{x}; \mathbf{w}) = \frac{1}{1 + \exp(-\mathbf{w}^T \mathbf{x})}$$

where  $\mathbf{w} \in \mathbb{R}^M$  is the parameter vector of the logistic regression model. Because there are only two classes,  $p(y = -1 | \mathbf{x}; \mathbf{w}) = 1 - p(y = 1 | \mathbf{x}; \mathbf{w})$ . Given a set of (independent) training examples  $\{(\mathbf{x}^{(i)}, y^{(i)}), i = 1, 2, \dots, N\}$  and a particular parameter vector  $\mathbf{w}$ , the (negative) joint log-likelihood of the samples can be written as:  $\sum_{i=1}^N -\log p(y^{(i)} | \mathbf{x}^{(i)}; \mathbf{w})$ . A maximum likelihood estimator would set the parameter vector  $\mathbf{w}$  to be the values that minimize this log-likelihood.

The L1-regularized logistic regression adds a regularization term in the objective function of the minimization. Instead of select the parameter vector that maximizes the log-likelihood, it set the parameter vector to:

$$\min_{\mathbf{w}} \sum_{i=1}^N -\log p(y^{(i)} | \mathbf{x}^{(i)}; \mathbf{w}) + \lambda \|\mathbf{w}\|_1$$

where  $\|\cdot\|_1$  is the L1 norm of a vector. One may view the regularized logistic regression as the maximum a posteriori (MAP) estimate of the parameter vector  $\mathbf{w}$  when the vector is draw from a Laplacian prior  $p(\mathbf{w}) = (\lambda/2)^M \exp(-\lambda \|\mathbf{w}\|_1)$ .

For document classification, the word vector of a document (the vector indicating the occurrence of the word in the document) is often used as the feature vector  $\mathbf{x}$  of the document, i.e.,

$$\mathbf{x}^{(i)} = \langle x_1^{(i)}, x_2^{(i)}, \dots, x_N^{(i)} \rangle.$$

where  $x_j^{(i)}$  is the number of occurrence of the  $j$ -th word in document  $i$ . Our knowledge transfer method use the collection of correlation groups to extend this feature vector. A correlation group  $C$  is a set of correlated words  $C = \{a_i\}$ . For each group  $C$ , we add a new feature corresponding to that group, i.e., we define a feature mapping function  $\phi_C : \{D\} \rightarrow \mathbb{R}$  such that for the  $i$ -th document  $D_i$ ,  $\phi_C(D_i) = \frac{|D_i \cap C|}{|C|}$ . With such extension, the feature space for the documents includes both the actual words contained in the documents and the collection of the correlated groups.

$$\mathbf{x}^{(i)} = \langle x_1^{(i)}, x_2^{(i)}, \dots, x_N^{(i)}, \phi_{C_1}(D_i), \phi_{C_2}(D_i), \dots, \phi_{C_k}(D_i) \rangle$$

We call the set of features  $\phi_{C_1}(D_i), \phi_{C_2}(D_i), \dots, \phi_{C_k}(D_i)$  the *extended feature*.

There are two reasons to use L1-regularized logistic regression model for our classification. First, it is a discriminative model and employs an explicit feature space. We can extend the feature space directly. Second, L1 regularization such as lasso [13] often has a feature selection effect [14],

[15]. That is, because the constraint on its L1 norm, the regularization would set some entries in the parameter vector to be zero, effectively removing those features that may not be important for classification. Such feature selection effect is crucial for our knowledge transfer system. For a particular document, many correlated groups may not be relevant and therefore, they introduce artificial noises. The  $L_1$  regularizer helps to reduce such noise.

#### IV. EXPERIMENT RESULTS

To evaluate the effect of the extended features on classification, we compare the accuracy of the classification with and without the extended features (while using the same L1-regularized logistic regression model for the classifier). The training and classification are performed on the 20-newsgroup dataset [16]. We first pre-process the articles in the 20-newsgroup dataset using the rainbow tools [17]. The pre-process removes the stop words and the header from each article. (The header contains the name of the newsgroup to which it belongs). Then the top 2000 words are selected using mutual information. Each article is represented by a word vector of 2000 dimensions (The vector is sparse. Many entries in the vector will be zero because the corresponding word is not contained in the article.)

The articles after preprocessing are used in our experiment. In each experiment, we first randomly select two newsgroups on which we will perform training and classification. We call these two the *classification newsgroup pair*. The collection of articles from the other 18 newsgroups are used to mine the correlated groups. We call this collection the knowledge-mining collection. Since the knowledge-mining collection and the classification newsgroup pair are separated, there is no direct connection between the correlated groups and the articles in the classification newsgroups.

From each of the two classification newsgroup, we first randomly select 50 articles and put them aside as testing samples set. We then select a certain number of articles, from the remaining in each of the classification newsgroups, to form the training sample set. We always pick the same number of articles from each newsgroup. On the other hand, we experiment with different training size. The training size we used are 5 (5 from one newsgroup and 5 from the other, a total of 10 articles. Same for the other training size), 10, 20, 50, and 100. For each training size, we construct 20 random training sets. The performance of the classifier for each size is the average over the 20 sets.

We repeat the experiment for different selection of classification newsgroups. The baseline performance is derived using the L1-regularized logistic-regression-based classifier whose feature space is the actual words contained in the documents. The baseline is compared to the performance of the classifier whose feature space is extended using our method. For the L1-regularized logistic-regression-based classifier, we use the implementation provided by Koh et. al. [18],

which uses an interior point method to learn the classifier. The correlation groups are formed using a threshold  $\tau = 0.1$ . Table I shows some of the correlation groups obtained in our mining process. (Each line is a correlated group.)

apple disk mac software
card disk dos drive mb pc software
motif window
bike car
car cars drive
bible earth god
bible christ earth god jesus
israel jewish jews
sale shipping

Table I  
EXAMPLE OF CORRELATED GROUPS

Fig. 1 shows the results of the classification performance for 9 randomly selected classification newsgroup pairs. In each plot, the x-axis represents the training size and the y-axis is the classification error. (Here classification error is the fraction of the wrongly classified samples among the 100, 50 from each class, testing samples.) The classification error of the classifier with extended features is plotted using solid line with “+” at the data points. The baseline, i.e., the classification error of the one without extended feature is plotted using dashed line with “o” at the data point.

Across the newsgroup pairs, there are three types of results. In many cases, classification with extended features performs better than classification without the extended feature. This can be seen particularly with the pairs sci.med v.s. rec.autos and soc.religion.christian v.s. comp.os.ms-windows.misc. In the best case, the extended features help to reduce classification error by 0.1.

In some cases, classification under our method does not improve much and in a few cases, the extended features make classification a bit worse. (For example, in talk.politics.mideast v.s. misc.forsale, for the training size 5, adding extended feature increase the error by 0.005.) Those results, however, can be expected. An extended feature would be helpful for a document only if the document can be mapped to that feature, i.e., the document contains words in that correlation group. Otherwise, the feature would be oblivious to the document. Because we obtain the correlation groups from only the collection of articles in the other newsgroups, it is possible that none of the correlation groups produced can be mapped to the articles in the classification newsgroup pair. In this case, our method would have no effect on classification. A close examination on the cases where classification is not affected shows that this is the case. It can be expected that if we increase the number of documents from which we mine the correlated groups, our collection of correlated groups can be more comprehensive and increase the chance that some groups can be mapped to the documents in the classification newsgroups.

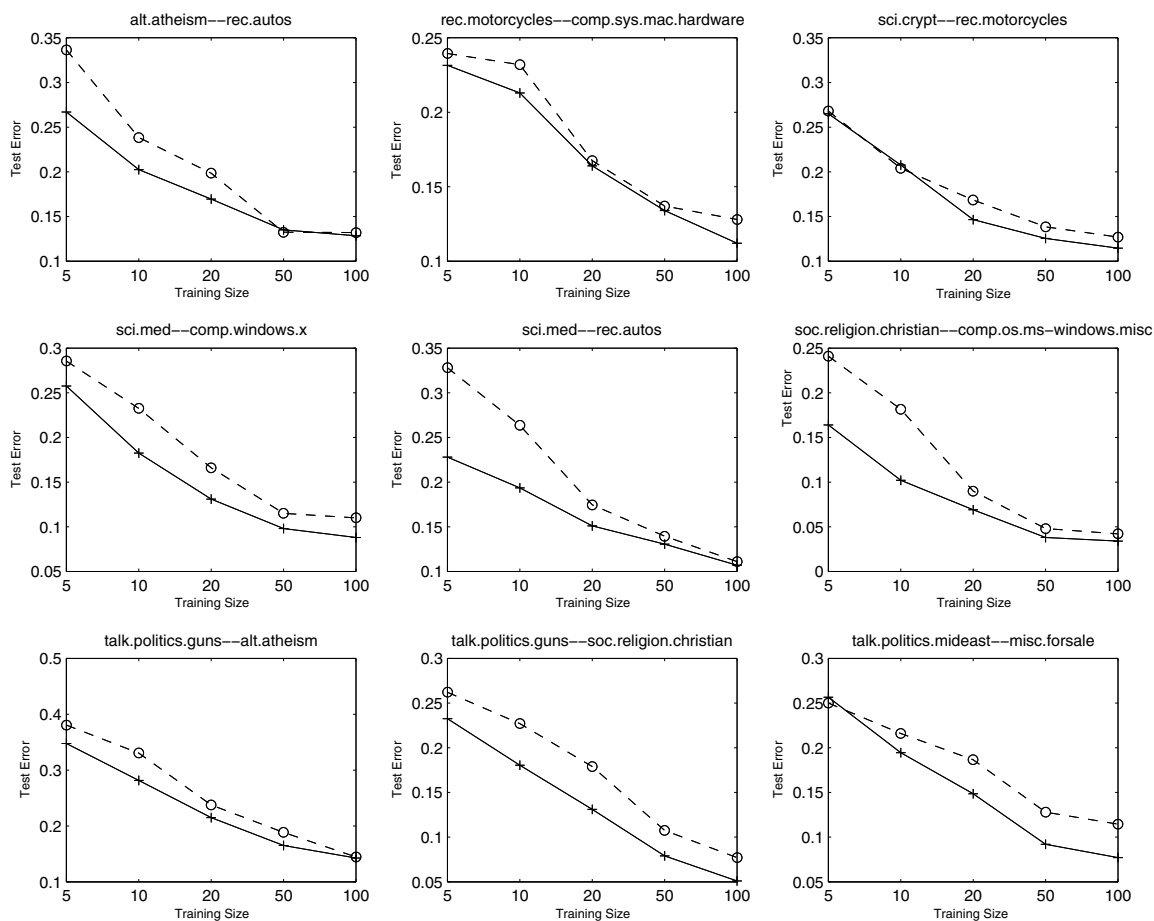


Figure 1. Comparison of Classification Errors on 9 Randomly Selected Newsgroup Pairs (With extended features: solid line and “+” at the data points; Baseline: dashed line and “o” at the data point. Training size  $k$  means  $k$  articles from one group and  $k$  from the other, a total of  $2k$  articles.)

In a few cases, the extended features reduce classification accuracy. As we discussed before, not all extended features are useful. Although some features reveal a certain property of a class, others are non-relevant and introduce noises that affect the classification negatively. We expect that more sophisticated feature selection would decrease the number of non-relevant features being included and hence improve the performance of our method.

Across training sizes, it seems there is no relation between the training size and the amount of improvement the extend feature can make. For some classification pairs, the improvement amount is similar across different training sizes. For some, there is more improvement for small training sizes and for the other, more improvement for large training sizes. There are two factors that affect this. On one hand, one may expect that extended features help more for small training

size because when there are only a few (say 5) training samples, it may be difficult to construct a good classifier. Bringing in prior knowledge would then help to obtain a better classifier and thus improve performance. On the other hand, when the training set is small, there may be very few extended features that can be mapped to the training documents. In this case, the classification would not be affected for the small training sizes.

To further understand the effect of the extended features, we perform classification on every pair of the newsgroups. We fix the training size to be 10 and perform the same training and classification process as previously described. For every pair of newsgroups, we obtain the average classification error rate with and without the extended features. The classification improvement is the difference between the two error rates. (Positive improvement means classification with

extended features reduces error and negative improvement means it increases error.) There are 20 newsgroups and 190 pairs in total. Fig. 2 plots the distribution of the improvement and it shows that, for most pairs, classification with extended features reduces error.

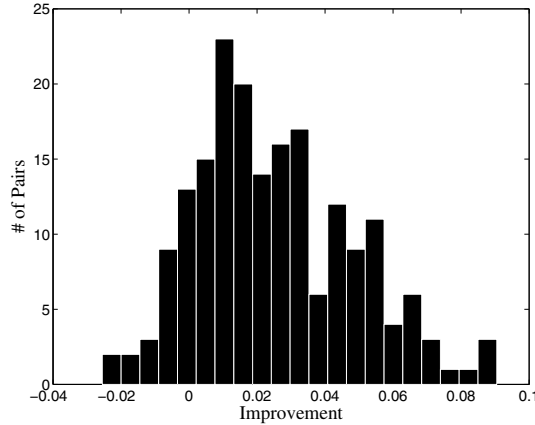


Figure 2. Distribution of Classification Improvement (Improvement is the classification error of the baseline minus the error when the extended features are used.)

## V. CONCLUSIONS AND FUTURE WORK

We proposed a new method for extracting prior knowledge and transferring this knowledge to facilitate current classification task. We mine correlation relationship among the words from a collection of related documents and view the correlation relationships as prior knowledge on the words. This knowledge is then applied to generate features for classifying new documents of different classes/types. Our experiment results show that the correlation-based knowledge can help improve classification accuracy.

There are several future directions we would like to pursue. First, we would like to explore ways to generate more complete knowledge by considering 1) use a larger document collection when mining the correlation relationship; 2) combine the mined correlation with the explicit directories provided by repositories such as the ODP. The second future direction would be to further validate the approach using a much larger collection of documents. Finally, we would like to explore better feature selection algorithms. One may also analyze the characteristics of the effective features produced in the experiments and leverage the information to generate effective features.

## REFERENCES

- [1] S. Thrun, "Is learning the  $n$ -th thing any easier than learning the first?" in *NIPS*, 1995, pp. 640–646.
- [2] J. Baxter, "A bayesian/information theoretic model of learning to learn via multiple task sampling," *Machine Learning*, vol. 28, no. 1, pp. 7–39, 1997.
- [3] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, 1997.
- [4] R. K. Ando and Zhang, T., "A framework for learning predictive structures from multiple tasks and unlabeled data," *J. Machine Learning Research*, vol. 6, pp. 1817–1853, 2005.
- [5] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: transfer learning from unlabeled data," in *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML)*, 2007, pp. 759–766.
- [6] R. Raina, A. Y. Ng, and D. Koller, "Constructing informative priors using transfer learning," in *Proceedings of the Twenty-Third International Conference (ICML)*, 2006, pp. 713–720.
- [7] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*, 1995.
- [8] E. Gabrilovich and S. Markovitch, "Feature generation for text categorization using world knowledge," in *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, 2005, pp. 1048–1053.
- [9] O. Egozi, E. Gabrilovich, and S. Markovitch, "Concept-based feature generation and selection for information retrieval," in *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI)*, 2008, pp. 1132–1137.
- [10] E. Cohen, M. Datar, S. Fujiwara, A. Gionis, P. Indyk, R. Motwani, J. D. Ullman, and C. Yang, "Finding interesting associations without support pruning," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 13, no. 1, pp. 64–78, 2001.
- [11] Indyk and Motwani, "Approximate nearest neighbors: Towards removing the curse of dimensionality," in *ACM Symposium on Theory of Computing (STOC)*, 1998.
- [12] J. Zhang and J. Feigenbaum, "Finding highly correlated pairs efficiently with powerful pruning," in *Proceedings of the 2006 ACM International Conference on Information and Knowledge Management (CIKM)*, 2006, pp. 152–161.
- [13] R. J. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, no. 1, pp. 267–288, 1996.
- [14] J. Goodman, "Exponential priors for maximum entropy models," in *HLT-NAACL*, 2004, pp. 305–312.
- [15] A. Y. Ng, "Feature selection,  $L_1$  vs.  $L_2$  regularization, and rotational invariance," in *ACM International Conference on Machine Learning*, 2004, p. 78.
- [16] K. Lang, "NewsWeeder: learning to filter netnews," in *Proc. 12th International Conference on Machine Learning*, 1995, pp. 331–339.
- [17] A. K. McCallum, "Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering," 1996, <http://www.cs.cmu.edu/~mccallum/bow>.
- [18] K. Koh, S.-J. Kim, and S. P. Boyd, "A method for large-scale  $l_1$ -regularized logistic regression," in *AAAI*. AAAI Press, 2007, pp. 565–571.