JMLR: Workshop and Conference Proceedings 27:133–144, 2012 Workshop on Unsupervised and Transfer Learning

Divide and Transfer: an Exploration of Segmented Transfer to Detect Wikipedia Vandalism

Si-Chi Chin SI-CHI-CHIN@UIOWA.EDU
Information Science, The University of Iowa

W. Nick Street NICK-STREET@UIOWA.EDU
Management Sciences Department, The University of Iowa

Abstract

The paper applies knowledge transfer methods to the problem of detecting Wikipedia vandalism detection, defined as malicious editing intended to compromise the integrity of the content of articles. A major challenge of detecting Wikipedia vandalism is the lack of a large amount of labeled training data. Knowledge transfer addresses this challenge by leveraging previously acquired knowledge from a source task. However, the characteristics of Wikipedia vandalism are heterogeneous, ranging from a small replacement of a letter to a massive deletion of text. Selecting an informative subset from the source task to avoid potential negative transfer becomes a primary concern given this heterogeneous nature. The paper explores knowledge transfer methods to generalize learned models from a heterogeneous dataset to a more uniform dataset while avoiding negative transfer. The two novel *segmented transfer (ST)* approaches map unlabeled data from the target task to the most related cluster from the source task, classifying the unlabeled data using the most relevant learned models.

Keywords: Transfer learning, classifier reuse, Wikipedia vandalism detection

1. Introduction

Transfer learning discusses how to transfer knowledge across different data distributions, providing solutions when labeled data are scarce or expensive to obtain. Motivated by the problem of Wikipedia vandalism detection (Potthast and Gerling, 2007; Chin et al., 2010), this paper investigates the question: *how do we transfer a classifier trained to detect vandalism in one article to another?* We introduce two novel *segmented transfer (ST)* approaches to learn from a labeled but diverse source task, which exhibits a wide-ranging distribution of both positive and negative examples over the feature space, and then selectively transfer the classifier to predict an unlabeled and more uniform target task. Our methods are also tested when transferring between articles with similar distributions.

Our work is related to the source task selection problem, investigating methods to enhance transfer learning performance and to minimize negative transfer. We concentrate specifically on transfer at the knowledge level, i.e., the reuse of learned classifiers from a source task, as opposed to transfer at level of instances, priors, or functions as exemplified by Pan and Yang (2010). We investigate two methods to exploit a single source task to predict a target task with no available labels. To improve knowledge transfer, it is useful to identify an effective method to transfer knowledge from the source task to the target

task. In this paper, we assume that *perhaps not all the source task is useful* and *perhaps not all the target task can learn from the available source task*. Our work aims to address the following questions:

- If not all the source task is related to the target task, how do we select the most relevant subset from the source task?

- If not all the target task can be explained or learned from the source task, how do we identify the subset from the target task that can benefit most from the knowledge transfer?

Wikipedia, the online encyclopedia, is a popular and influential collaborative information system. The collaborative nature of authoring has exposed it to malicious editing, or vandalism, defined as "any addition, removal, or change of content made in a deliberate attempt to compromise the integrity of Wikipedia[1]." Wikipedia vandalism detection, an adversarial information retrieval task, is a recently emerging research area. The goal of the task is to determine, for each newly edited revision, whether it could be a vandalism instance and to create a ranked list of probable vandalism edits to alert Wikipedia users (usually the stewards for an article). However, determining if an edit is malicious is challenging and acquiring reliable class labels is non-trivial. To classify a new and unlabeled dataset, it is useful to leverage knowledge from prior tasks.

Wikipedia vandalism instances exhibit heterogeneous characteristics. A vandalism instance can be a large-scale editing or a small change of stated facts. Each type of vandalism may demonstrate different feature characteristics and an article may contain more instances of one type of vandalism than others. Moreover, the distribution of different types of vandalism may vary from article to article. For example, 'Microsoft' article may contain higher ratio of graffiti instances whereas 'Abraham Lincoln' article may be more vulnerable to misinformation instances. The heterogeneous nature of Wikipedia vandalism detection could potentially introduce negative transfer (Rosenstein et al., 2005). It requires a selective mechanism to assure the quality of knowledge transfer, for example, leveraging knowledge about "graffiti" instances from the source task to detect graffiti, as opposed to other types of vandalism instances, in the target task. To resolve the problem of a heterogeneous source task, we introduce two methods to identify the informative segments from the source task in the absence of class labels.

In this paper, instead of learning from multiple sources, we focus on the problem setting in which only a single source task is available. Both the source and target task have the same input and output domains, but their samples are drawn from different populations. Each sample in both the source and target task is a revision of a given Wikipedia article, preprocessed into a feature space representing a collection of statistical language model features. The output labels indicate whether the article is a vandalism instance.

We organize the rest of paper as follows. Section 2 introduces the two *segmented transfer* approaches. Section 3 describes the experimental setups, including the datasets, the features, and the six experimental settings. Sections 4 and 5 present the experimental results and evaluations. In Section 6, we discuss related work, and we conclude the paper with future directions in Section 7.

1. http://en.wikipedia.org/wiki/Wikipedia:Vandalism

Table 1: Tabular comparison of STST and TTST

	STST	**TTST**
Primary assumption:	Not all the source task is useful	Not all the target task can benefit from the available source task
Train cluster models at:	Source task	Target task
Assign cluster membership to:	Target task	Source task
Max number of classifiers:	Number of clusters found in the source task	Number of clusters found in the target task
Transferred object:	Classifiers trained from the source task	

2. Segmented Transfer (ST)

In this paper, we propose *segmented transfer (ST)* to enrich the capability of transfer learning and to address the issue of potential negative transfer. The goal of ST is to identify and learn from the most related segment, a subset from the training samples, from the source task. Our motivation comes from two assumptions:

- Not all of the source task is useful, and
- Not all of the target task can benefit from the available source task.

We propose the *source task segmented transfer (STST)* and the *target task segmented transfer (TTST)* approaches to address each assumption and summarize the two approaches in Table 1.

Source task segmented transfer (STST) The STST approach clusters the source task, assigning cluster membership to the target task. In Figure 1, the labeled source task is first segmented into clusters. Each cluster has its own classifier. We then assign cluster membership to the unlabeled target task and transfer the classifier trained from the corresponding cluster of the source task. Because the distribution of the feature space is different between the source and target tasks, it is likely that some source task data will not be used. The approach aims to transfer knowledge acquired only from the related segment to minimize negative transfer.

Target task segmented transfer (TTST) The TTST approach clusters the target task, assigning cluster membership to the source task. The goal of the TTST is to differentiate samples that can be better learned from the provided source task. In Figure 2, the unlabeled target task is first segmented into clusters. We then assign cluster membership to the labeled source task and train a classifier for each cluster. Finally, the classifiers are transferred to the corresponding clusters in the target task. As shown in Figure 2, some data from the target task may not be well learned because of the lack of an appropriate source task.

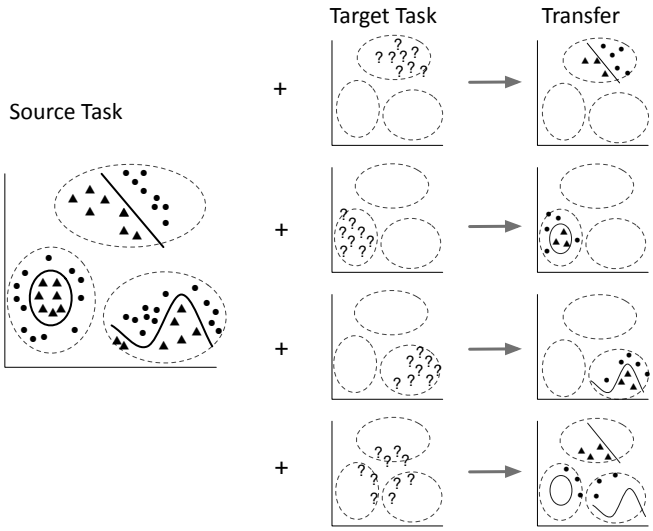

Figure 1: Flowchart of source task segmented transfer (STST)

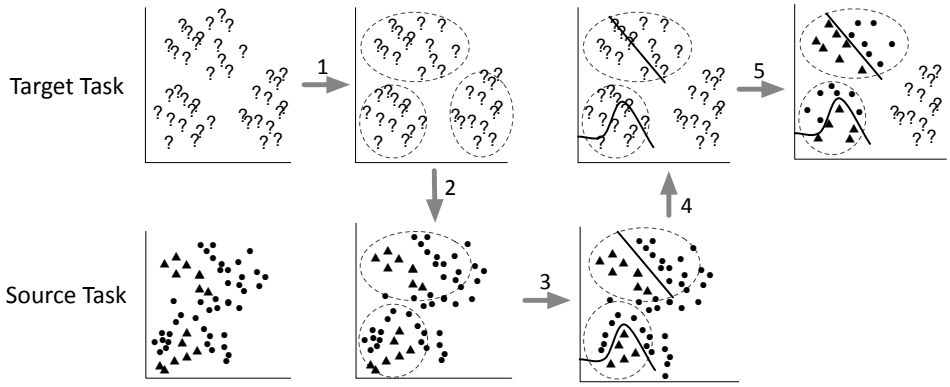

Figure 2: Flowchart of target task segmented transfer (TTST)

3. Experiments

This section describes the datasets used for experiments, the input feature space, the six experimental settings, and the cluster membership assignment distributions for each setting.

Dataset description In four of the experiments, we clustered and trained on the Webis Wikipedia vandalism (Webis) corpus (Potthast and Gerling, 2007) and tested on the revision histories of the "Microsoft" and "Abraham Lincoln" articles on Wikipedia (Chin et al., 2010). The other two experiments use Microsoft as the source task and transfer to the Lincoln article.

The Webis dataset contained randomly sampled revisions of different Wikipedia articles, drawn from different categories. The Microsoft and Lincoln datasets contained the revision

history of those articles. Although class labels were available for both datasets, the class information was ignored during the clustering and was used to build classifiers and to demonstrate the performance of the two methods. Table 2 is a tabular description of the three datasets. The AUC and AP scores for the Microsoft and Lincoln dataset were computed by 10-fold cross validation using the provided class labels using an SVM classifier with RBF kernel. The parameters γ and C were chosen empirically to achieve the best performance.

Table 2: Dataset description

	Positive	**Negative**	**Total**
Webis	301	639	940
Microsoft	268	206	474
Lincoln	178	223	401

Feature description All three datasets used features generated by statistical language modeling (SLM) using the CMU SLM toolkit (Clarkson and Rosenfeld, 1997). SLM computes the distribution of tokens in natural language text and assigns a probability to the occurrence of a string S or a sequence of m words. The *evallm* tool evaluates the language model dynamically, providing statistics such as perplexity, number of n-gram hits, number of OOV (out of vocabulary), and the percentage of OOV from a given text. The *evallm* tool generates features separately from the diff data for the new revision and the full new revision to build classifiers. In addition to the 18 attributes (9 for the diff and 9 for the full revision) generated from SLM, three features: ratio of insertion, ratio of change, and ratio of deletion, were added to the set of attributes. The 21 attributes were generated for each revision in the dataset. Table 3 summarizes features used for the classification.

Experimental setup and clustering algorithm Table 4 describes six experimental settings. STST and TTST each have three experiments with different combinations of the source and target task. We used the Weka (Hall et al., 2009) implementation of clustering, using the Expectation Maximization (EM) algorithm to optimize Gaussian mixture models to cluster the source and target tasks. Using cross validation, the EM algorithm determined the number of clusters to generate. To evaluate the ranked results from the experiments, we used AUC and Average Precision (AP). The ranked list was sorted by the probability of the predictions generated by SVM classifiers.

Cluster Membership Distribution This paragraph describes the cluster memberships and the distributions of positive and negative instances for the six experimental settings. Tables 5 and 6 present the cluster assignment distribution for STST. In Experiments 1 and 2, the source Webis dataset is segmented into 16 clusters (see Table 5). The target Microsoft and Lincoln datasets are mapped to 9 and 8 of these clusters respectively. The results of cluster assignment confirm the assumption that not all the source task is useful for the target task. However, the source task can still be fully exploited. In Experiment 3, as shown in Table 6, all the source task (Microsoft) instances are useful for the target task (Lincoln), both of which were determined to contain three clusters.

Table 3: Definition of Features

Feature	**Definition**
word_num(d)	Number of known words (from *diff*)
perplex(d)	Perplexity value (from *diff*)
entropy(d)	Entropy value (from *diff*)
oov_num(d)	Number of unknown words (from *diff*)
oov_per(d)	Percentage of unknown words (from *diff*)
bigram_hit(d)	Number of known bigrams (from *diff*)
bigram_per(d)	Percentage of known bigrams (from *diff*)
unigram_hit(d)	Number of known unigrams (from *diff*)
unigram_per(d)	Percentage of known unigrams (from *diff*)
ratio_a	Ratio of added text from previous revision
ratio_c	Ratio of changed text from previous revision
ratio_d	Ratio of deleted text from previous revision

Table 4: Six experimental settings for STST and TTST

Method	**Exp**	**Source Task**	**Target Task**
STST	1	Webis	Microsoft
	2	Webis	Lincoln
	3	Microsoft	Lincoln
TTST	4	Webis	Microsoft
	5	Webis	Lincoln
	6	Microsoft	Lincoln

Table 7 shows the cluster assignment distributions for the TTST approach (Experiments 4, 5, and 6). The distribution shows that sometimes part of the target task would not have available source task to learn from. For example, in Experiment 4, the source task is only useful for cluster 2 of the target task; in Experiment 5, it is only useful for cluster 1.

Table 5: Cluster membership distributions for Experiments 1 and 2

	Source Task	**Target Task**	
Source cluster	**Webis Data Distri.** $(+,-)$	**Microsoft (Exp:1) Data Distri.** $(+,-)$	**Lincoln (Exp:2) Data Distri.** $(+,-)$
1	75 (9,66)	43 (22,21)	48 (27,21)
2	24 (1,23)	192 (116,76)	85 (41,44)
3	16 (10,6)	153 (80,73)	215 (86,129)
4	25 (8,17)		18 (6,12)
5	46 (24,22)	49 (20,29)	
6	40 (35,5)	16 (16,0)	11 (5,6)
7	41 (3,38)	2 (2,0)	1 (1,0)
8	130 (9,121)		
9	63 (50,13)		
10	43 (9,34)	1 (0,1)	
11	75 (2,73)		
12	43 (6,37)		
13	62 (28,34)	17 (12,5)	22 (11,11)
14	60 (60,0)		
15	149 (8, 141)		
16	48 (39,9)	1 (0,1)	1 (1,0)
Total	940 (301,639)	474 (268, 206)	400 (178,223)

Table 6: Cluster membership distribution for Experiment 3

		Source Task	**Target Task**
Exp	**Source cluster**	**Microsoft Data Distri.** $(+,-)$	**Lincoln Data Distri.** $(+,-)$
3	1	344 (186, 158)	357 (146, 211)
	2	125 (80, 45)	42 (30,12)
	3	5 (2,3)	2 (2,0)
	Total	474 (268,206)	401 (178,223)

4. Experimental results

This section describes the experimental results for STST and TTST. Our results show that the two proposed approaches improved the ranking, moving more actual vandalism instances to the top of the ranked list. Table 8 shows the performance of the baseline, a

Table 7: Cluster membership distribution for Experiments 4, 5, and 6

Exp	Target cluster	Target Task Data Distri. $(+,-)$	Source Task Data Distri. $(+,-)$
4	1	344 (186, 158)	0
	2	125 (80, 45)	940 (301,639)
	3	5 (2,3)	0
	Total	474 (268,206)	940 (301,639)
5	1	56 (36,20)	940 (301,639)
	2	115 (45,70)	0
	3	230 (97,133)	0
	Total	401 (178,223)	940 (301,639)
6	1	56 (36,20)	159 (93,66)
	2	115 (45,70)	121 (56,65)
	3	230 (97,133)	194 (119,75)
	Total	401 (178,223)	474 (268,206)

direct transfer without either STST or TTST, using an SVM classifier with linear and RBF kernels. In this section, results that outperform the baseline are marked with a †.

Table 8: Baseline performance

Exp	Classifier	AUC	AP
1 and 4	SVM w/ linear kernel (C=1)	0.5333	0.6002
	SVM w/ RBF kernel (C=1, $\gamma = 0.1$)	0.5466	0.5862
2 and 5	SVM w/ linear kernel (C=1)	0.5276	0.4528
	SVM w/ RBF kernel (C=0.8, $\gamma = 0.16$)	0.5396	0.4454
3 and 6	SVM w/ linear kernel (C=500)	0.6089	0.6134
	SVM w/ RBF kernel (C=500, $\gamma = 0.02$)	0.6215	0.6021

4.1. STST Evaluation

Table 9 shows the experimental results for the STST approach. We compared the performance of STST with the best performance for direct transfer, i.e., train on the source task and transfer directly to the target task, using the SVM classifier with RBF kernel (see Table 8). The results indicate that the STST approach consistently outperforms the baseline across the three experiments.

Table 9: Experiment results for STST

Experiment 1		Experiment 2		Experiment 3	
AUC	AP	AUC	AP	AUC	AP
0.5541†	**0.6095†**	**0.5519†**	**0.5063†**	**0.6883†**	**0.6514†**

4.2. TTST Evaluation

Table 10 shows the experimental results for the TTST approach. As shown in Table 7, only cluster 2 in Experiment 4 and cluster 1 in Experiment 5 have the source task to learn from. Therefore, presumably, the classifier trained for the assigned cluster in the target task will perform better on the assigned cluster than on other clusters.

The results in Experiment 4 support the assumption. The performance of cluster 2 is much higher than cluster 1 when we used the same classifier trained from the source task for both clusters. Although the cluster 3 in Experiment 4 has high AUC and AP results, it is noted that the size of the cluster is quite small and the results might be insignificant.

Experiment 5 presents mixed results on AUC and AP. We observe that the AP, but not the AUC, is higher in cluster 1, to which all the source task was assigned. In general, AP is more sensitive to the order at the top of the ranked list whereas AUC evaluates the overall number of correctly ranked pairs. In the case that AP is higher but not AUC, it indicates that the algorithm performs better at the top of the list; however, it doesn't create more correctly ranked pairs. To support this observation, we evaluated the results using Normalized Discounted Cumulative Gain (NDCG) at the rank position 5 and 10. Figure 3 shows that cluster 1 outperforms the other two clusters. The results suggest the occurrence of negative transfer when the learned classifier was used on less related datasets. The results also demonstrate how negative transfer could be minimized when the target task only learned from more informative segments in the source task.

In Experiment 6, all three clusters from the target task (Lincoln) have assigned instances from the source task (Microsoft). The combined result (the 'Total' row) outperforms the baseline (i.e., direct transfer of a classifier trained from the entire source task).

Table 10: Experiment results for TTST, breakdown by cluster

Experiment 4			**Experiment 5**			**Experiment 6**		
#	**AUC**	**AP**	**#**	**AUC**	**AP**	**#**	**AUC**	**AP**
1	0.5082	0.5503	1	0.4472	**0.6346**†	1	**0.6792**†	**0.7959**†
2	**0.6569**†	**0.7201**†	2	0.4942	0.3641	2	**0.6288**†	0.495
3	**0.8333**†	**0.8333**†	3	**0.5603**†	0.4393	3	**0.738**†	**0.6637**†
						Total	**0.6627**†	**0.6426**†

5. Related Work

Source task selection Research on multi-task learning and reinforcement learning has discussed the measurement of task relatedness and the selection of related tasks (Silver and McCracken, 2003; Ben-David and Borbely, 2008; Taylor et al., 2007). In this paper, we focus on the problem setting in which only a single source task is available, and no labels are available for the target task. Research has used semi-supervised learning methods such as EM algorithm combined with NaiveBayes classifier (Nigam et al., 2000) and co-clustering (Dai et al., 2007) to improve text classifiers. In contrast with current research, our approach does not require labeled data in the target task, selecting the source task segments solely based on the feature distribution.

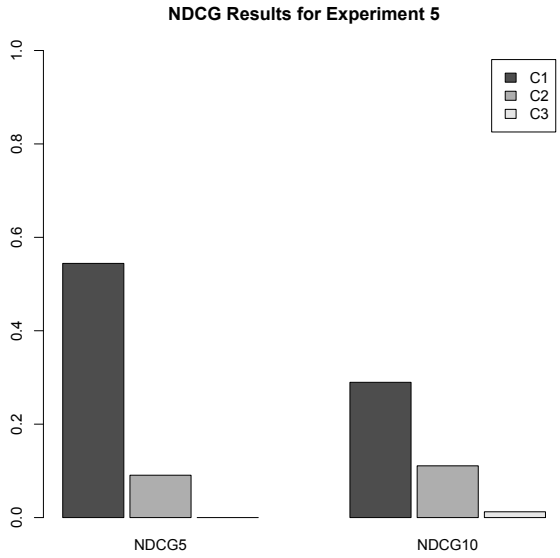

Figure 3: NDCG results for Experiment 5

A common approach to select a related source task is to measure the relatedness between the source and target task. Silver and Mercer (1996) employed a learning rate as a function to measure relatedness between a source task and the target task. Kuhlmann and Stone (2007) constructed a graph for each task based on the elements and rules and then compared the graph isomorphism to select similar tasks. Thrun and O'Sullivan (1998) clustered multiple source tasks into a hierarchy. Their method transferred knowledge from the cluster most related to the target task to emphasize the knowledge among similar and discriminating instances. The authors used class label information to construct clusters, matching the target distribution of a given task with the most similar cluster.

Classifier reuse Knowledge transfer emphasizes the reuse of previously acquired knowledge, i.e., the classifiers, from the source task to the target task. A common approach to reuse classifiers is to select among candidate solutions from the source tasks. Zhang et al. (2005) constructed an ensemble of decision trees trained from related tasks to improve prediction on the problem with limited labeled data. Eaton and Desjardins (2006) developed an ensemble framework where each member classifier focuses on one resolution level. The multi-resolution learning facilitates transfer between related tasks. Yang et al. (2007) described methods to select auxiliary classifiers from a set of existing classifiers. The authors used the EM algorithm to estimate the distribution respective to each class and then select the classifier that can best separate the between-class score distribution, creating "pseudo" labels to evaluate each classifier and then selecting classifiers of average precision scores.

By comparison, our approach aims to transfer classifiers learned only from the related segment, as opposed to the entire set, of the source task. The experimental results demonstrate the promise of the proposed segmented transfer approach.

6. Conclusion and Future Work

In this paper, we investigated two *segmented transfer* approaches to transfer knowledge while avoiding negative transfer. The objective of the proposed approach is to address the heterogeneous characteristics of Wikipedia vandalism. We clustered the source and the target task to map unlabeled data from the target task to the most related cluster from the source task, classifying the unlabeled data using the most relevant learned models. Our results show enhanced performance (e.g., AUC and AP) on ranking the probable vandalism instances. In the future, we will explore the soft clustering method, assign each instance in the target task a probability of cluster membership, and combine predictions. We will also consider enhancing the methods' ability to avoid negative transfer by implementing an overall "relatedness" measure, so that points in the target task are not classified using distant clusters.

Acknowledgments

Our special thanks go to anonymous reviewers and ICML'11 conference attendees for their constructive feedback to improve this work. This publication was made possible by Grant Number UL1RR024979 from the National Center for Research Resources (NCRR), a part of the National Institutes of Health (NIH). Its contents are solely the responsibility of the authors and do not necessarily represent the official views of the CTSA or NIH.

References

S. Ben-David and R. S. Borbely. A notion of task relatedness yielding provable multiple-task learning guarantees. *Machine Learning*, 73:273–287, December 2008.

S.-C. Chin, W. N. Street, P. Srinivasan, and D. Eichmann. Detecting Wikipedia vandalism with active learning and statistical language models. In *Proceedings of the 4th Workshop on Information Credibility*, WICOW '10, page 3–10, 2010.

P. Clarkson and R. Rosenfeld. Statistical language modeling using the CMU-Cambridge toolkit. In *Fifth European Conference on Speech Communication and Technology*, pages 2707–2710, 1997.

W. Dai, G.-R. Xue, Q. Yang, and Y. Yu. Co-clustering based classification for out-of-domain documents. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, page 210–219, 2007.

E. Eaton and M. Desjardins. Knowledge transfer with a multiresolution ensemble of classifiers. In *ICML Workshop on Structural Knowledge Transfer for Machine Learning*, 2006.

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: An update. *SIGKDD Explor. Newsl.*, 11:10–18, November 2009. ISSN 1931-0145.

G. Kuhlmann and P. Stone. Graph-based domain mapping for transfer learning in general games. *Machine Learning: ECML 2007*, page 188–200, 2007.

K. Nigam, A.K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2):103–134, 2000.

S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

M. Potthast and R. Gerling. Wikipedia vandalism corpus Webis-WVC-07, 2007. URL http://www.uni-weimar.de/medien/webis/research/corpora.

M. T. Rosenstein, Z. Marx, L. P. Kaelbling, and T. G. Dietterich. To transfer or not to transfer. In *NIPS'05 Workshop, Inductive Transfer: 10 Years Later*, 2005.

D. L. Silver and P. McCracken. Selective transfer of task knowledge using stochastic noise. In Y. Xiang and B. Chaib-draa, editors, *Advances in Artificial Intelligence*, volume 2671 of *Lecture Notes in Computer Science*, page 994–994. Springer Berlin/Heidelberg, 2003.

D. L. Silver and R. E. Mercer. The parallel transfer of task knowledge using dynamic learning rates based on a measure of relatedness. In *Connection Science Special Issue: Transfer in Inductive Systems*, pages 277–294, 1996.

M. E. Taylor, G. Kuhlmann, and P. Stone. Accelerating search with transferred heuristics. In *ICAPS-07 Workshop on AI Planning and Learning*, September 2007.

S. Thrun and J. O'Sullivan. Clustering learning tasks and the selective cross-task transfer of knowledge. In *Learning to Learn*, page 235–257. Kluwer, 1998.

J. Yang, R. Yan, and A. G. Hauptmann. Cross-domain video concept detection using adaptive SVMs. In *Proceedings of the 15th International Conference on Multimedia*, MULTIMEDIA '07, page 188–197, New York, NY, USA, 2007.

Y. Zhang, W.N. Street, and S. Burer. Sharing classifiers among ensembles from related problem domains. In *Fifth IEEE International Conference on Data Mining*, pages 522–529. IEEE Computer Society, 2005.