Graph-Based Query Strategies for Active Learning

Wei Wu and Mari Ostendorf, *Fellow, IEEE*

***Abstract*—This paper proposes two new graph-based query strategies for active learning in a framework that is convenient to combine with semi-supervised learning based on label propagation. The first strategy selects instances independently to maximize the change to a maximum entropy model using label propagation results in a gradient length measure of model change. The second strategy involves a batch criterion that integrates label uncertainty with diversity and density objectives. Experiments on sentiment classification demonstrate that both methods consistently improve over a standard active learning baseline, and that the batch criterion also gives consistent improvement over semi-supervised learning alone.**

***Index Terms*—Active learning, graph, query strategy, sentiment classification.**

I. Introduction

LACK of training data is a problem for many machine learning tasks, especially in the natural language processing (NLP) area, where data sparsity is a common issue. Because it can be expensive and time consuming to annotate training data, researchers have sought ways to minimize hand labeling efforts through active learning [1], [2], which aims to choose data to label that is expected to provide the biggest improvement in system performance. Active learning starts with a base model trained with a small labeled set and uses a query strategy to select the most informative data samples (referred to here as instances) from the unlabeled set. These instances are hand-labeled and added to the training set, and then the process may be repeated to add another batch of data. The key to active learning is the query strategy, which has received much attention in previous work; readers can refer to [3] for a detailed summary. The most commonly used query strategies choose individual instances that have the highest label uncertainty, e.g., as measured by entropy [4].

The uncertainty-based strategy of choosing individual instances ignores the potential relatedness between samples. In this work, we take advantage of a graph-based representation—specifically an instance-feature bipartite graph—to characterize the relatedness between data instances. Using the graph, we introduce two new query strategies that leverage relatedness in two different ways. In one case, the relatedness is used to improve the uncertainty estimates associated with individual instances, tightly integrating semi-supervised learning with active learning. In the second case, the relatedness is used in a batch query to extract instances with both high uncertainty and diversity. In either case, the resulting models learned from the hand-labeled data can be integrated with semi-supervised learning in a final stage. In general, the proposed graph-based active learning framework has the advantages that it is easy to incorporate prior information about features and it is simple to compute in a parallel framework such as MapReduce [5].

Semi-supervised learning [6] provides a mechanism for leveraging large amounts of unlabeled data in combination with some labeled data. Of the many different approaches proposed, graph-based semi-supervised learning is well-matched to the bipartite graph representation leveraged in this work. Graph-based semi-supervised learning typically involves applying label propagation [7], [8] over a graph built according to relatedness between data instances in the corpus [9], and passing the labeling information from labeled data instances to unlabeled ones. Zhu *et al.* [10] combined active learning with graph-based semi-supervised learning strategy using a Gaussian field to enhance an expected risk query strategy, assessed on both topic classification and other tasks. A key difference in our approach vs. [10] (and vs. other work on active learning integrated with self-training and EM-based semi-supervised learning [11]–[15]) is the use of model combination: a maximum entropy model [16] trained from labeled data provides a regularizing prior in combination with label propagation for graph-based semi-supervised learning.

The relatedness of individual instances can be accounted for in active learning by retraining the model after each instance is labeled, but this is not practical in most applications. Instead, instances are usually added to the supervised set in a batch. In batch-mode active learning, criteria have been explored to reduce minimize redundancy or increase diversity of the examples. Hoi and colleagues [17], [18] account for redundancy by maximizing the Fisher information of the queried batch. Brinker [19] provided a diversity criterion for support vector machines. Diversity is combined with density and estimated relevance in active learning aimed at relevance feedback for information retrieval [20] with a linearly interpolated score. Here, we use the graph structure to account for relatedness between instances in a batch query strategy that moves beyond prior work by tightly integrating uncertainty with diversity and density.

In this work, we use sentiment classification as a test bed to evaluate the performance of the proposed graph-based active learning framework. Sentiment classification is an important technology for opinion mining over the large amount of user generated text on the internet, such as on-line forums, blogs, review web sites and tweets. Most previous research treated

Manuscript received March 09, 2012; revised July 01, 2012; accepted August 28, 2012. Date of publication October 05, 2012; date of current version November 21, 2012. This work was supported by the National Science Foundation (NSF under Grant IIS-0916951. The opinions and conclusions are those of the authors and should not be construed as representing the official views or policies of the NSF. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Gokhan Tur.

The authors are with the Department of Electrical Engineering, University of Washington, Seattle, WA, 98195 USA e-mail: weiwu@uw.edu; ostendor@uw.edu.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASL.2012.2219525

sentiment classification as a supervised machine learning problem, applying text classification algorithms such as naive Bayes, maximum entropy models and support vector machines [21], which typically require a large amount of labeled data to train a model with high accuracy. While there are some genres for which sentiment labels are easy to collect (e.g., reviews with user ratings), there are many others for which there is little user-rated data and thus a potential to benefit from active and semi-supervised learning. Our experiments with two datasets comprising five domains show that the proposed graph-based framework combining active and semi-supervised learning consistently outperforms the use of either approach alone.

The rest of this paper is organized as follows. Section II gives a brief overview of the infrastructure of the proposed graph-based active learning framework, including the maximum entropy model, the graph representation and label propagation. Section III presents details of the proposed active learning query strategies, and experimental results are presented in Section IV. Section V compares our algorithms and results to closely related work, and questions for future studies are outlined in Section VI.

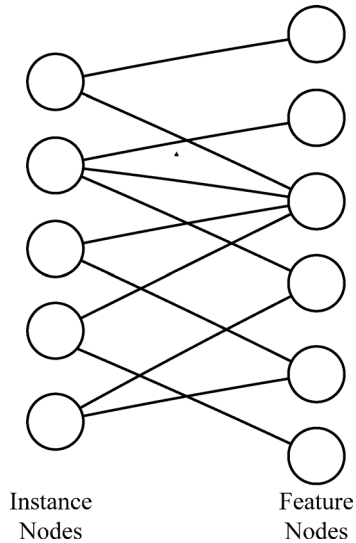

Fig. 1. An example of a bipartite graph representing the instance-feature occurrence relationship.

II. Infrastructure of the Active Learning Framework

The infrastructure of the proposed graph-based active learning framework consists of three major components, the maximum entropy model, the instance-feature bipartite graph representation and label propagation. The word-based features used here are widely used for sentiment classification. Many models work well on this task [21]; we build on the maximum entropy model because it is widely used for many text classification problems.

A. Maximum Entropy Model

The maximum entropy model is a discriminative classification model [16], formulated as:

$$P(y \mid x; \lambda) = \frac{\exp\left(\sum_k \lambda_k f_k(x, y)\right)}{\sum_y \exp\left(\sum_k \lambda_k f_k(x, y)\right)}, \tag{1}$$

where $f_k(x, y)$ is the k-th feature of the model, and λ_k is the model parameter associated with it.

In this work, we use unigram and bigram occurrences as features for the maximum entropy model. We filter out unigrams and bigrams with frequency smaller than a given threshold (<4) in the corpus, and remove stopwords. To handle negations, we use the heuristic proposed in [21], where a "NOT_" prefix is added to all unigrams occurring between a negation word and the closest punctuation following it. Parameters of the maximum entropy model are estimated using gradient ascent with L_2 regularization.

B. Bipartite Graph Representation

In the proposed active learning framework, we use a bipartite graph to capture the instance-feature relationship in the corpus [9]. As shown in Fig. 1, the left side nodes represent instances, the right side nodes represent features. If a feature occurs in an instance, an edge is added between them. The edge weight can be set in a way to suit specific text classification tasks. For example, for topic classification, the edge weight can be set as the inverse document frequency of the features. In the sentiment classification experiments presented in this paper, we set the edge weight as 1 without prior knowledge of the feature's discriminative power. Assuming there are m instances and n unique features in the corpus, the graph can be represented by an $(m + n) \times (m + n)$ adjacency matrix. Due to the bipartite structure, two blocks in this matrix will have zero values as shown below.

$$\begin{bmatrix} 0 & W \\ W^T & 0 \end{bmatrix} \tag{2}$$

where W is an $m \times n$ matrix representing the connection weight from m instances to n features. Note that the feature set used to construct the bipartite graph need not be the same as the one used for the maximum entropy model. In this paper, we use only unigram word features to construct the instance-feature graph for sentiment classification, in order to control the graph size due to limits of computational resources.

C. Label Propagation

Label propagation [7], [8] is a graph-based semi-supervised learning strategy, which has been studied in a number of works for various machine learning problems [22]–[26]. Graph-based data representations have been studied extensively in previous work on label propagation as a semi-supervised learning method [7], [8]. Early work was based on homogeneous graphs, where only instances are represented in the graph according to their nearest-neighbor relations. Later studies applied it on bipartite graphs for various applications, including Youtube video recommendation (user-video graph) [23] and query classification (query-link graph) [25], [26]. The instance-feature bipartite graph data representation used in our work is inspired by these studies and other work on semi-supervised sentiment classification [9].

Given the instance-feature bipartite graph, according to label propagation, each node i has an adjustable class "likelihood" vector I_i (for an instance node) or F_i (for a feature node), where

each element $I_{i,y}$ and $F_{i,y}$ is a non-negative real number that when normalized gives the posterior probability of a node belonging to class y. Let I denote a matrix with the i-th row as I_i and $I^{(0)}$ as the class prior used to initialize I, and similarly for F, F_i, and $F^{(0)}$. The label propagation process can then be defined as shown in Algorithm 1, where $\tilde{W} = D^{-1/2}W(D^{-1/2})^T$ is the normalized adjacency matrix from the instance nodes to feature nodes and D is a diagonal matrix with its (i,i) element equal to the sum of the i-th row of W. According to this algorithm, in each iteration $I^{(n)}$ and $F^{(n)}$ are updated as a weighted sum $(0 < \alpha < 1)$ of the propagation from the node's neighbors and its prior. This is an efficient computation that is convenient to implement in a system for parallel programming.

Algorithm 1 Label Propagation

Initialize $I^{(0)}$ and $F^{(0)}$;
repeat
$F^{(n)} = \alpha\tilde{W}^T I^{(n-1)} + (1-\alpha)F^{(0)}$;
$I^{(n)} = \alpha\tilde{W}F^{(n-1)} + (1-\alpha)I^{(0)}$;
until $\|I^{(n)} - I^{(n-1)}\| < \epsilon$
Normalize $I^* = I^{(n)}$ and output the sentiment posterior probability $P_{\text{prop}}(y \mid x)$ for each instance node.

D. Regularizing Label Propagation With Prior Knowledge

As shown by [8], the label propagation defined in Algorithm 1 will converge to the optimum result by minimizing the objective function

$$Q(N) = \alpha \sum_{i,j=1}^{n+m} w_{i,j} \left\| \frac{1}{\sqrt{D_{ii}}} N_i - \frac{1}{\sqrt{D_{jj}}} N_j \right\|^2 + (1-\alpha) \sum_{i=1}^{n+m} \left\| N_i - N_i^{(0)} \right\|^2$$

where N_i represents node i's class likelihood vector (i.e., it is I_i for an instance node and F_i for a feature node). The first term in this objective function is the smoothness constraint, preferring a result that does not change too much between neighboring nodes. The second term can be seen as a regularization term; it prefers a result that does not deviate too far from its prior. With proper priors, this regularization can help prevent propagating errors through the graph [25].

In sentiment classification experiments, we use the maximum entropy model result and lexicon sentiment knowledge to regularize the label propagation. The two types of nodes in the bipartite graph (instances and words) are treated differently when setting the priors for label propagation:

For the instance nodes, the label propagation is regularized by the maximum entropy model. Specifically, if the instance is unlabeled, its prior $I^{(0)}(x)$ is set to be the posterior sentiment distribution $P(y \mid x; \lambda)$ produced by the maximum entropy model; otherwise, it is set as their label indicator vector $l(y \mid x)$.

For the word (feature) nodes, the setting of their priors provides a chance to introduce an outside source of lexicon sentiment information. The word node's prior can either be set with a knowledge-based strategy, using a polarity lexicon from a human-edited sentiment dictionary, or with a data-driven strategy, using a polarity lexicon obtained by mining on-line resources [27]. In this work, we adopt the knowledge-based strategy. We use the *Inquirer*[1] sentiment lexicon list, which contains 1 915 positive words (*Inquirer Positiv* category) and 2 291 negative words (*Inquirer Negativ category*). For words on this list, we set the prior as their sentiment label indicator vector. For words outside this list, they are initialized with the uniform distribution and then updated with the label propagation posterior obtained in the previous active learning iteration.

III. Active Learning Query Strategies

Most traditional uncertainty-based query strategies are optimized for selecting one instance at a time, ideally the model should be retrained after querying every single instance. However, this will lead to too many active learning iterations (a query $\rightarrow$ label $\rightarrow$ retrain cycle), which is usually not practical in application scenarios. First, many machine learning models do not support on-line training or perform less effectively under on-line training settings, and frequently retraining the model with all labeled data can be expensive. Second, small sample active learning iterations may pose difficulties for managing annotator working hours. Hence, in our proposed framework, the active learning is carried on in a batch mode, where a group of instances are queried at each active learning iteration.

Let L be the labeled set of instances, U be the unlabeled instance pool, λ be the maximum entropy model and G be the graph representation. The batch mode active learning will select a batch set B for human labeling from U according to a query strategy $\psi(B, \lambda, G)$. We assume that the batch size at each algorithm set is $|B| = N$. The overall active learning process is described in Algorithm 2. We propose two graph-based query strategies within this framework: i) maximum gradient length, and ii) batch network gain, the latter is specifically optimized for this batch mode active learning. Label propagation based semi-supervised learning is incorporated in the query strategy when the maximum gradient length approach is used. It can also be used in the testing stage for either approach with the maximum entropy model providing regularization.

Algorithm 2 Graph-based Active Learning Framework

Given: the initial labeled set L, the unlabeled set U, the graph G, the query strategy $\psi(B, \lambda, G)$, and batch size N;
repeat
// train maximum entropy model λ
$\lambda = train(L)$;
// find B according to the query strategy
$B^* = \text{argmax}_{|B|=N, B \subset U} \psi(B, \lambda, G)$;
$L \leftarrow L \cup B^*$; $U \leftarrow U - B^*$;
until meet the labeling budget
// Train model with labeled samples
$\lambda = train(L)$

[1]http://www.wjh.harvard.edu/~inquirer

A. Maximum Gradient Length Query

This query strategy incorporates label propagation as a graph-based semi-supervised learning method to leverage large amounts of unlabeled data to help select the most informative instances for the maximum entropy model. It is based on the assumption that instances resulting in the largest changes to the current model in training tend to improve it most. For the maximum entropy model, parameters are estimated by maximizing the log likelihood $\mathcal{L}(y \mid x; \lambda) = \log P(y \mid x; \lambda)$ with gradient ascent, thus the gradient length of one instance brought to the current model can be used as measure of the informativeness of labeling that instance. The gradient of the maximum entropy model induced by instance x is computed as

$$\frac{\partial \mathcal{L}(y \mid x; \lambda)}{\partial \lambda_k} = E_{\tilde{P}(x,y)}[f_k(x, y)] - E_{P(y \mid x;\lambda)}[f_k(x, y)] \tag{3}$$

where $\tilde{P}(x, y)$ is the empirical distribution of the data and $E_P[\cdot]$ denotes expectation with respect to distribution P. Thus

$$E_{\tilde{P}(x,y)}[f_k(x, y)] = \sum_y l(y \mid x) f_k(x, y) \tag{4}$$

where $l(y \mid x)$ is the label indicator of instance x. Then we can use the gradient length $\| \bigtriangledown \mathcal{L}(y \mid x; \lambda)\|$ to evaluate the change brought by instance x to the current model λ.

However, for unlabeled instances, we do not know their true labels to estimate the empirical distribution $E_{\tilde{P}}(x, y)[f_k(x, y)]$. Here we propose to approximate $E_{\tilde{P}}(x, y)[f_k(x, y)]$ with the empirical distribution of predicted labels according to the label propagation output $P_{\text{prop}}(y \mid x)$. Let $D(P_{\text{prop}}(y \mid x))$ be the classification decision indicator vector of label propagation to replace $l(y \mid x)$ in (4):

$$E_{\tilde{P}(x,y)}[f_k(x, y)] \approx \sum_y D(P_{\text{prop}}(y \mid x)) f_k(x, y). \tag{5}$$

This approach makes a *hard decision* on the label, but one could also imagine using the distribution $P_{\text{prop}}(y \mid x)$ directly as a *soft decision* to replace $l(y \mid x)$ in (4):

$$E_{\tilde{P}_{\text{prop}}(x,y)}[f_k(x, y)] = \sum_y P_{\text{prop}}(y \mid x) f_k(x, y) \tag{6}$$

Note that $\| \bigtriangledown \mathcal{L}(y \mid x; \lambda)\|$ computed with (6)'s approximation corresponds to the expected gradient length used in [28], with the exception of using the label propagation distribution instead of the current maximum entropy model posterior. With the estimated gradient length $\| \bigtriangledown \mathcal{L}(y \mid x; \lambda)\|$, we can select the top-$N$ instances with *maximum gradient length* to query for their labels according to algorithm 3, i.e., $\psi(B, \lambda, G) = \sum_{x \in B} \| \bigtriangledown \mathcal{L}(y \mid x; \lambda)\|$.

Algorithm 3 Maximum gradient length query strategy

Given: labeled set L, unlabeled set U, and batch size N
// Run regularized label propagation over $L \cup U$
$P_{prop}(y \mid x) = propagation(L \cup U \mid P(y \mid x; \lambda))$;
// Map unlabeled samples to soft or hard labels $l(y \mid x)$
$l(y \mid x) = P_{prop}(y \mid x)$ or $D(P_{prop}(y \mid x))$;
$B^* = \text{argmax}_{|B|=N, B \subset U} \sum_{x \in B} \| \bigtriangledown \mathcal{L}(y \mid x; \lambda)\|$;

B. Maximum Batch Network Gain Query

The maximum batch network gain query is a query strategy optimized for the batch mode active learning; it combines representativeness and diversity criteria with the traditional uncertainty criterion. Representativeness and diversity are two complementary criteria to the uncertainty criterion, which take into account the interdependence between instances in the corpus. For example, an instance with high classification uncertainty can also be an outlier point, in which case labeling it will give little help for classifying other instances. Hence, we would like the labeled instance to be representative in the corpus, i.e., choosing an instance located in an area of the feature space with densely distributed instances. In batch mode active learning, we also want to increase the instance diversity in the queried batch to reduce labeling redundancy. The proposed maximum batch network gain query incorporates the two criteria and selects the batch set by maximizing the network gain associated with labeling instances in the batch, leveraging the bipartite graph representation.

For each unlabeled instance i in S, the *individual gain* from querying its label can be evaluated by the entropy:

$$H(i) = -\sum_y P(y \mid x_i; \lambda) \log(P(y \mid x_i; \lambda)), \tag{7}$$

where $P(y \mid x_i; \lambda)$ is i's label distribution produced by the maximum entropy model.

Inspired by label propagation, we approximate the gain of querying a batch by propagating the *individual gain* of querying instances in it through the graph to other unlabeled instances. We compute the effective weights between instances $a_{ji} = [\tilde{W}\tilde{W}^T]_{ji}$, where $\tilde{W}$ is the normalized adjacency matrix from the instance nodes to feature nodes. In other words, a_{ji} sums all normalized paths from queried instance i to unlabeled instance j in the bipartite graph, incorporating all features they have in common. Computing a_{ji} is analogous to one step of label propagation, as shown in Fig. 2.

The effective weights between instances can be used to evaluate their relatedness. With this idea, we define the *batch network gain* $NG(B)$ as:

$$NG(B) = \sum_{j \in U-B, i \in B} a_{ji} H(i) - \mu \sum_{i,k \in B, i \neq k} a_{ki} H(i) \tag{8}$$

where the first term represents the impact of the information gain from querying the batch on the rest of the unlabeled set, and the second term compensates for the redundancy between nodes in the batch with penalty term μ. For experiments in this paper, we set $\mu = 1$. The batch network gain criterion integrates uncertainty (through the use of the node entropy term $H(\cdot)$) with representativeness (via the first term) and diversity (via the second term).

In each active learning iteration, we select the batch with maximum $NG(B)$ to query for their labels, i.e., $\psi(B, \lambda, G) = NG(B)$. To solve this maximizing problem, we can see $A = [a_{ij}]$ as an adjacency matrix of a new graph consisting of only the instances nodes, then maximizing the first term of (8) is equivalent to maximize the graph cut between B and $U - B$ given B's size. This is a submodular problem [29], which can

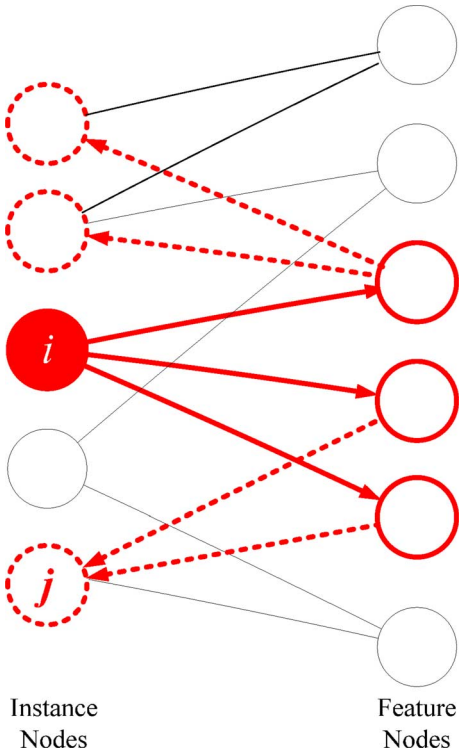

Fig. 2. Propagate information gain through the bipartite graph.

be solved near-optimally with a greedy algorithm [30]. Similarly, (8) is also submodular, therefore we can select the active learning batch with the following greedy algorithm 4.

Algorithm 4 Maximum batch network gain query

Given: unlabeled set U, batch size N
Initialize $B = \emptyset$;
repeat
$k = \text{argmax}_{i \in U} NG(B \cup \{i\})$
$B \leftarrow B \cup \{k\}, U \leftarrow U - \{k\}$
until $|B| = N$

IV. Experiments

To assess the effectiveness of the proposed active learning framework, we applied it using two sentiment review datasets, including the document-level Amazon product review dataset that includes multiple domains, and the sentence-level movie review dataset. In all domains, the initial training set is roughly 10% of the total data available for that domain, and active learning batch sizes are approximately 5% of the full data set. Two sets of experiments were run to study different configurations of the two proposed graph-based query strategies. For conciseness, only results on the most difficult task (Amazon book reviews) are presented. Then we present trends for the proposed system compared to several contrasting cases over various domains from the two sentiment review datasets.

A. Data

1) Amazon Product Reviews: The Amazon product review dataset is the "Multi-domain sentiment dataset v2.0" introduced in [31]. This dataset consists of Amazon product reviews from 4 domains, including books, dvds, electronics, and kitchen. Each domain contains 5 k–6 k user reviews with balanced positive/negative classes. In our experiments, we conducted a 5-fold cross validation for each of the 4 domains. In each fold, we divided the domain dataset into a test set (1/5 of the dataset documents), an initial training set (500 documents), and an active learning instance pool with the rest of documents in this corpus. The batch size for each active learning iteration is 250.

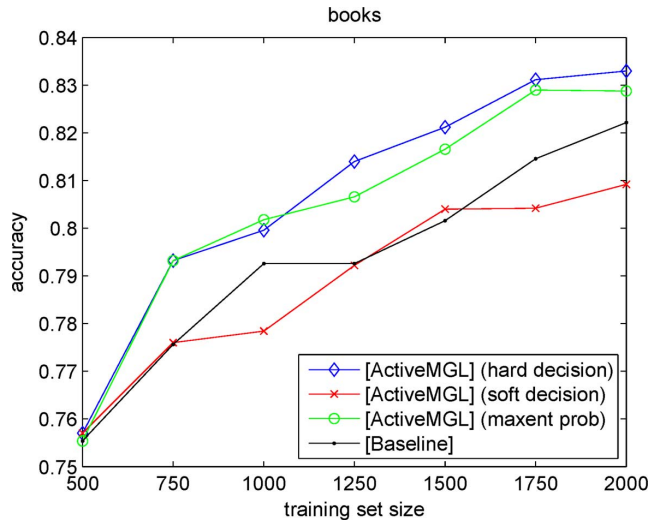

Fig. 3. Performance using *maximum gradient length* query with *soft decisions* vs. *hard decisions* compared to the random baseline.

2) Movie Reviews: The movie review dataset is the "sentence polarity dataset v1.0" introduced in [32]. This dataset consists of sentences extracted from http://www.rottentomatoes.com/ user reviews. It has two tasks, the positive/negative classification task and the subjective/objective classification task, each has 10 k sentences with balanced classes. In our experiments, we performed a 5-fold cross-validation for each of the two tasks. In each fold, we divided the task dataset into a test set (1/5 of the dataset sentences), an initial training set (1 k sentences), and an active learning instance pool with the rest of sentences in this corpus. The batch size for each active learning iteration is 500.

The label propagation prior weight for all data sets was tuned with the dev data of the movie review dataset (polarity task) to maximum the label propagation accuracy only, without involving active learning. The label propagation accuracy did not appear to be sensitive to the prior weight.

B. Maximum Gradient Length Query: Settings for Estimating Gradient Length

For the maximum gradient length query proposed in Section III.A, we compare there possible settings to estimate the gradient length in this query strategy: *soft decisions* (6), *hard decisions* (5), and *maxent prob* which is to use the posterior distribution produced by the current maximum entropy model instead of label propagation to estimate the gradient length.

The results on the positive/negative task in the Amazon book review domain are presented in Fig. 3, with a comparison to random selection as a baseline. Fig. 3 shows that the maximum gradient length query strategy based on label propagation with

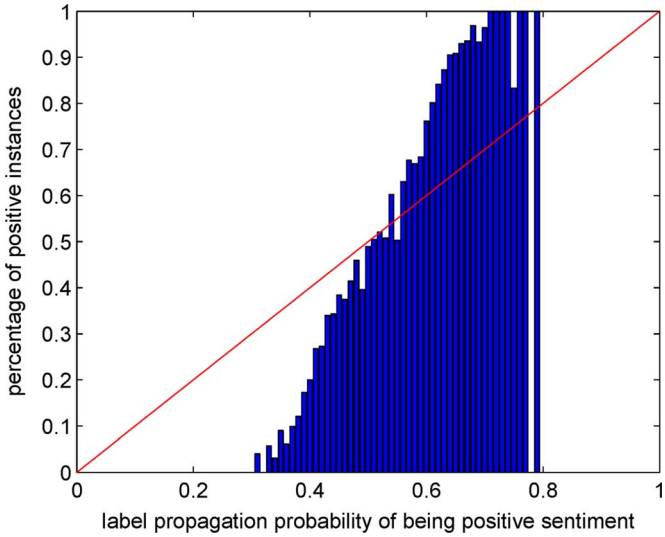

Fig. 4. Bias of posterior probability produced by label propagation.

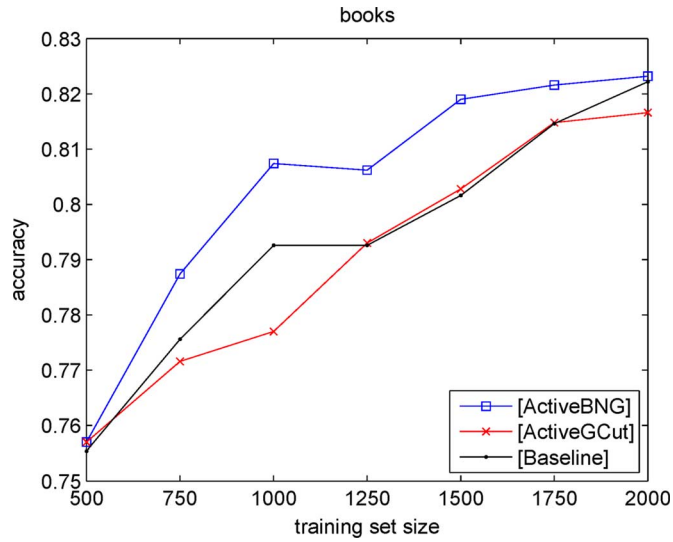

Fig. 5. Performance of *maximum batch network gain* and *maximum graph-cut* queries, compared to a random baseline.

hard decisions achieves slightly better performance than the expected gradient length query strategy using the maximum entropy model posterior distribution. This is probably because the former combines the outputs from both the label propagation and maximum entropy model, while the latter only uses the output from the maximum entropy model alone. However, we also find that the soft-decision approach using the label propagation distribution is not effective, with performance sometimes below the random baseline.

To study the reason for the failure of the label propagation soft decision, we analyzed the label propagation results from the first active learning iteration. Let $P_{\text{prop}}(y = 1 \mid x)$ denote the posterior probability of positive sentiment produced by label propagation. In Fig. 4, we show the relative frequency of positive sentiment labels for different bins of the posterior $P_{\text{prop}}(y = 1 \mid x)$. An unbiased posterior would have relative frequencies that match the diagonal line. We see that label propagation tends to converge with distributions that are not confident and thus biased, leading to poor estimates of the gradient length, possibly due to tuning label propagation for decision accuracy rather than some measure of goodness of the posterior distributions and/or due to tuning on a different domain. The bias is such that the decisions associated with these posteriors are more often correct than the posteriors would predict, which may explain why hard decisions are more effective than soft decisions.

Hence, in subsequent experiments, we only present results using the maximum gradient length query with a hard decision.

C. Maximum Graph-Cut vs. Maximum Batch Network Gain

Recall that the *maximum batch network gain* query integrates the uncertainty criterion with the representativeness and diversity criteria in the query strategy by propagating the individual instance gain $H(i)$ through the graph (8). If we modify (8) setting $H(i)$ to be 1 for all instances in the queried batch, we eliminate the uncertainty component of the criterion and obtain a simple "diversity and density" criterion that is the counterpart of the *maximum graph-cut* query [30] in the general text classification setting. Fig. 5 compares these two methods on the Amazon book review data, together with the random selection baseline. As expected, the addition of the uncertainty component leads to improved results. The maximum graph-cut query alone does no better than the random baseline, indicating that relying on graph structure alone is not effective for active learning for this task where the graph is determined by feature co-occurrence.

D. Comparison Across Different Domains

In this section, we systematically evaluate the proposed query strategies for active learning over different domains, including two tasks in the movie review dataset, and the four domains from the Amazon product review dataset. For each domain, we conducted six active learning iterations, which used 10%–40% of the data to train the model.

1) Comparison of Query Strategies: We first made a direct comparison of the different query strategies. In our experiment, the following four systems are compared, which are all based on the maximum entropy model and use its outputs as the final decisions:

- *Random query* [Baseline]
  This system uses random selection to choose instances to label in each iteration. This is a standard baseline for active learning; building on uninformed labeling of data is essentially not using active learning.
- *Uncertainty query* [ActiveEnt]
  This system applies the widely used uncertainty query, specifically the variation based on entropy (7), which is equivalent to the *least confident* and *marginal sampling* query strategies for binary classification problems.
- *Maximum gradient length query* [ActiveMGL]
  This system applies the maximum gradient length query with a hard decision based on label propagation results.
- *Maximum batch network gain query* [ActiveBNG]
  This system applies the *maximum batch network gain* query with redundancy penalties as in (8).

Table I summarizes the average absolute accuracy gain and relative error reduction of the three other systems over the [Random] baseline system in the first six active learning batches. Both the proposed *maximum gradient length* and

TABLE I
AVERAGE CLASSIFICATION ACCURACY GAIN (ABSOLUTE) AND ERROR REDUCTION (RELATIVE) IN THE FIRST 6 ACTIVE LEARNING BATCHES (USING 10%–40% TRAINING DATA) OVER THE [BASELINE] FOR DIFFERENT ACTIVE LEARNING STRATEGIES. (*** SIGNIFICANT ON 0.001; ** SIGNIFICANT ON 0.01; * SIGNIFICANT ON 0.05; OTHERS, NOT SIGNIFICANT ON 0.05)

Average Accuracy Gain (%)	**Books**	**DVDs**	**Electronics**	**Kitchen**	**Movie**	**Movie (sub/obj)**
ActiveEnt	1.14	0.60	1.50*	0.85	1.21	1.16*
ActiveMGL	**1.55**	0.93	1.03	1.10	1.48*	1.29**
ActiveBNG	1.09	**1.51***	**1.57***	**1.46***	**1.78****	**1.69*****
Average Error Reduction (%)	**Books**	**DVDs**	**Electronics**	**Kitchen**	**Movie**	**Movie (sub/obj)**
ActiveEnt	5.68	3.01	8.61	5.47	3.94	7.40
ActiveMGL	**7.74**	4.64	5.91	7.06	4.84	8.26
ActiveBNG	5.46	**7.51**	**9.03**	**9.38**	**5.82**	**10.81**

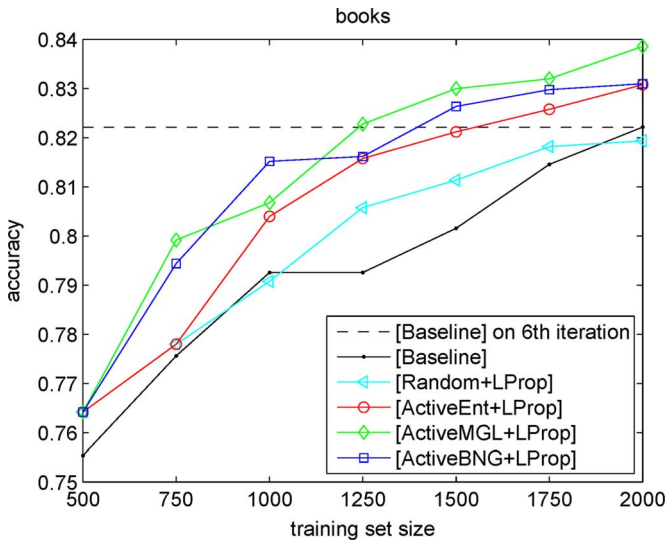

Fig. 6. Performance on Amazon product review (books).

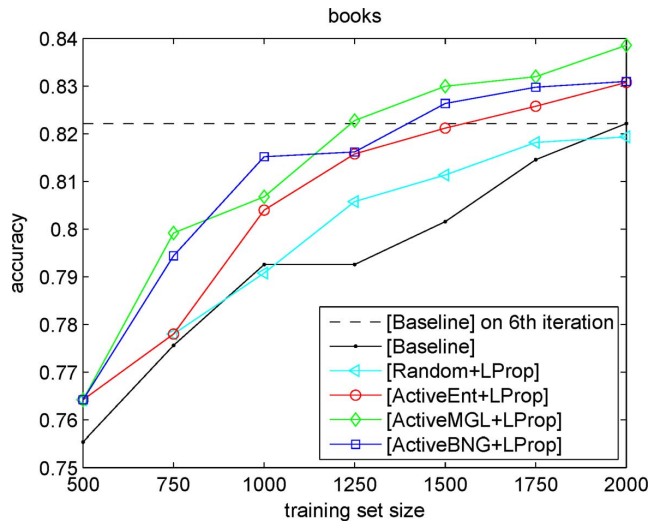

Fig. 7. Performance on Amazon product review (dvd).

maximum batch network gain queries outperform the random query baseline. By leveraging the larger dataset through label propagation, the *maximum gradient length* query obtains better performance than *uncertainty query* on all domains except on the Amazon electronics domain. The *maximum batch network gain* query consistently outperforms the *uncertainty query*, which does not account for interdependence between individual instances in the batch. It also outperforms *maximum gradient length* on all the tested domains except the Amazon book domain, for which none of the differences are significant.

2) Classification Enhanced With Label Propagation: In this section, we integrate label propagation into the sentiment classification process to enhance the performance of the four systems described above, and compare different query strategies under this new setting. Specifically, we apply the regularized label propagation after obtaining the maximum entropy posterior, and use the label propagation result as the final system output. The *random query* system with label propagation corresponds to semi-supervised learning alone, i.e., without active learning. The four semi-supervised systems are again compared to the baseline system (random query without label propagation).

Figs. 6–11 present the experimental results in each domain respectively. The horizontal dash line in each figure represents the *random query* baseline system's performance after the sixth active learning iteration, which corresponds to roughly 40% of the training data (2000 samples for the Amazon data sets and 4000 samples for the movie reviews). Table II summarizes the average absolute accuracy gain and relative error reduction of the four other systems over the [Random] baseline system in the first six active learning batches. Compared with results in Table I, systems with all query strategies get a performance boost with the label propagation. With label propagation, both *maximum gradient length* and *maximum batch network gain* queries significantly outperform the baseline across all domains. With one exception, all strategies combining active learning and semi-supervised learning improve over semi-supervised learning alone. The *maximum batch network gain* query has relatively stable performance, which consistently outperforms all other systems on the tested domains, except for the *maximum gradient length* query on the Amazon book domain. The *maximum batch network gain* not only benefits the maximum entropy model (indicated by the results in Table I), but also label propagation by choosing well-connected instances in the "key" paths of the label propagation.

The improvement margin associated with integrating label propagation is relatively larger on the movie review dataset than the other four domains from the Amazon review dataset. This is because the movie review data is on the sentence-level, which on average has a much smaller number of features as clues for sentiment judgment per instance, thus the data sparseness problem is more severe for this dataset. Hence, leveraging the unlabeled data with label propagation can bring more "mileage" to this dataset than the Amazon review dataset, which is on the document-level.

Fig. 12 compares the percentage of the required training data amount relative to the random query [Baseline] system for

TABLE II
AVERAGE CLASSIFICATION ACCURACY GAIN (ABSOLUTE) AND ERROR REDUCTION (RELATIVE) IN THE FIRST 6 ACTIVE LEARNING BATCHES (USING 10%–40% TRAINING DATA) OVER THE [BASELINE] FOR DIFFERENT ACTIVE LEARNING STRATEGIES. (*** SIGNIFICANT ON 0.001; ** SIGNIFICANT ON 0.01; * SIGNIFICANT ON 0.05; OTHERS, NOT SIGNIFICANT ON 0.05)

Average Accuracy Gain (%)	**Books**	**DVDs**	**Electronics**	**Kitchen**	**Movie**	**Movie (sub/obj)**
Random+LProp	0.41	0.54	0.45	0.65	2.13***	2.24***
ActiveEnt+LProp	1.27*	1.10	1.52*	1.23*	2.98***	2.93***
ActiveMGL+LProp	**2.17****	1.53*	1.50*	1.64**	3.40***	1.94***
AcitveBNG+LProp	1.90**	**2.22****	**1.93***	**1.91****	**3.52*****	**3.33*****
Average Error Reduction (%)	**Books**	**DVDs**	**Electronics**	**Kitchen**	**Movie**	**Movie (sub/obj)**
Random+LProp	2.03	2.69	2.60	4.16	6.96	14.30
ActiveEnt+LProp	6.36	5.48	8.71	7.90	9.75	18.74
ActiveMGL+LProp	**10.84**	7.65	8.61	10.58	11.11	12.37
AcitveBNG+LProp	9.48	**11.07**	**11.06**	**12.32**	**11.52**	**21.26**

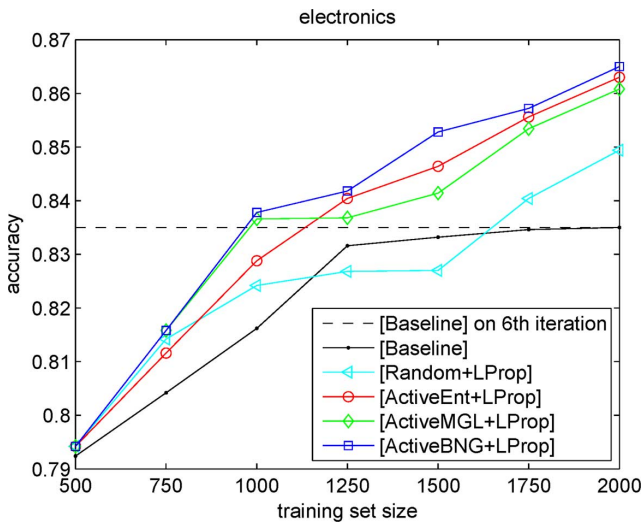

Fig. 8. Performance on Amazon product review (electronics).

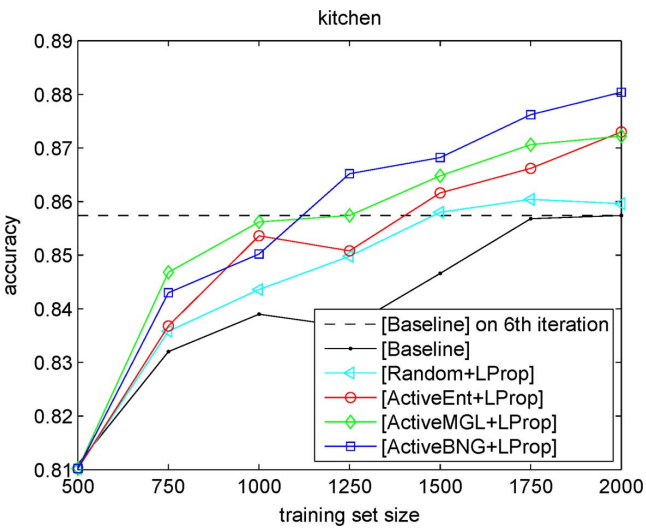

Fig. 9. Performance on Amazon product review (kitchen).

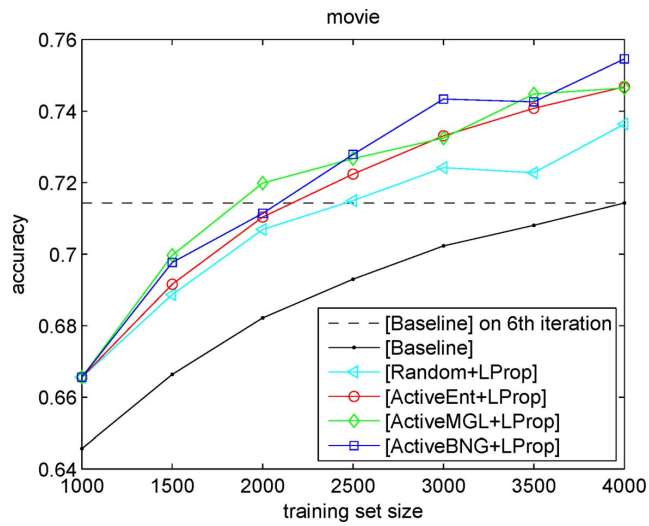

Fig. 10. Performance on movie review (positive/negative task).

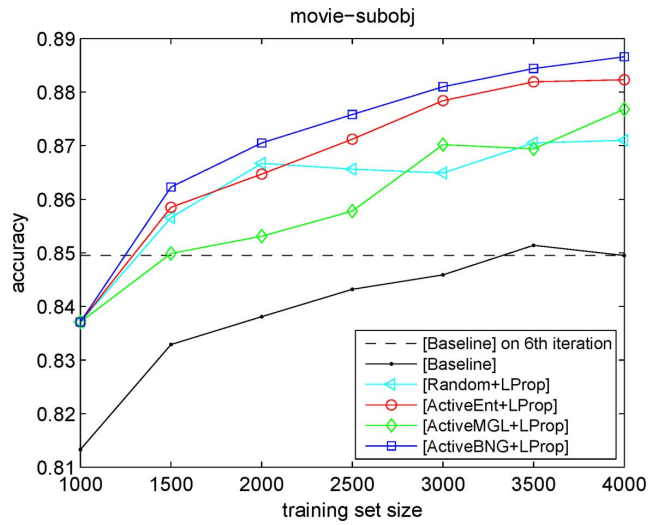

Fig. 11. Performance on movie review (subjective/objective task).

the other 4 compared systems to reach the performance of the [Baseline] system on the 6th iteration. It is shown that the both the [ActiveMGL+LProp] and [ActiveBNG+LProp] systems need only 50%–75% of the data required by the [Random] baseline system to reach its performance on the 6th iteration; and they also require less training data than the other systems to achieve this goal on most of the tested domains.

V. RELATED WORK

As mentioned earlier, other work has proposed combining semi-supervised and active learning [11], [12], [14], [15]. A key difference in our work is the particular combination of two different learning models (supervised maximum entropy modeling and semi-supervised label propagation) as a new type of model combination within the active learning framework. Our use of

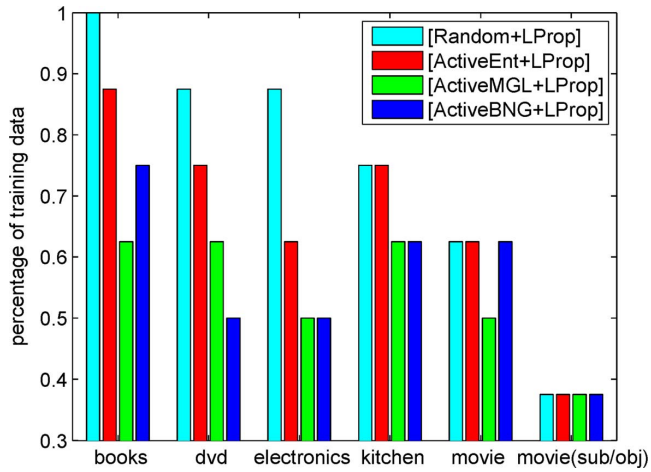

Fig. 12. Percentage of the required training data relative to the [Random] baseline system for [Random+LProp], [ActiveEnt+LProp], [ActiveMGL+LProp] and [ActiveBNG+LProp] to reach the baseline system's performance on the 6th iteration.

multiple models here also differs from approaches that combine multiple classifiers leveraging semi-supervised learning in queries based on co-testing or query-by-committe [13], [33] in that the different models in our work are combined via regularization rather than in a committee-based query.

Within this overall framework, two new query strategies are proposed that leverage interdependence of samples in different ways. Our proposed *maximum gradient length* query strategy is inspired by [25] which uses the maximum entropy model to regularize label propagation for query intent classification. In contrast to [25], which designs a co-training based semi-supervised learning strategy, we are aiming for an active learning framework. This query strategy is related to the *expected gradient length* query proposed for a conditional random field (CRF) in [28]. The latter uses the posterior probability produced by the CRF to compute the expectation of the gradient length for the CRF itself, while we take advantage of the framework that combines the maximum entropy model and label propagation, and use the label propagation result to estimate the gradient length.

Our proposed *maximum batch information gain* query strategy is designed for batch-mode active learning settings. Methods for reducing redundancy or increasing diversity of samples have been previously explored in batch-mode active learning [17]–[19]. Different from this work, our approach uses a graph-based corpus representation, which makes it easier to be incorporated with label propagation or extended for parallel computation. Lin and Bilmes [30] also studied batch mode active learning with submodular graph functions for the problem of training hidden Markov models for speech recognition. In addition to differences in the structure of the graph (associated with differences in the applications), our approach differs in that it incorporates uncertainty, representativeness and diversity criteria as compared to the approach in [30] which is mainly designed for representativeness. Diversity, density and relevance (analogous to uncertainty) are all incorporated in a query criterion by Xu *et al.* [20], but the approach is to simply interpolate three scores with two empirically-tuned weights. Tuning weights for active learning is more challenging in a real scenario than for classification accuracy.

Sentiment classification is a well-studied topic, with related work falling into two groups that leverage polarity lexicons [34]–[36], [27] vs. machine learning [21], [37]. This work falls into the second group. Previous research on machine-learning-based sentiment classification have explored features [38]–[41], classifiers [9], [42], [43], and domain adaptation [44], [45]. Our work focuses on active learning.

Other researchers have applied both semi-supervised and active learning to sentiment classification over the Amazon product review dataset, using an SVM [46] and Active Deep Network [47]. We obtain higher accuracy, but the results are not directly comparable since their work is on an early (smaller) version of that data set. While their work did obtain larger relative gains due to active learning, our work starts with a much better baseline (as well as consistently higher final performance). We claim that starting from a better baseline provides a stronger result. On the Amazon review dataset, our baseline achieves comparable performance with the result presented in [31], when the same amount of training data are used.

Finally, we note that our experimental work is on labeling instances, rather than features as in [12], [48], but the algorithms could easily be applied to features for applications where this makes sense.

VI. Conclusion

In this paper, we have proposed a graph-based active learning framework that integrates label propagation based semi-supervised learning with a maximum entropy model. Based on this framework, we design two active learning query strategies that aim to account for interdependence between samples through a bipartite graph structure. One method uses label propagation through the graph to improve the evaluation of usefulness of individual samples. The other method uses the graph to incorporate diversity and representativeness into the selection criterion referred to as batch network gain. Experiments with and without label propagation in the testing stage are presented in order to understand the impact of semi-supervised learning. The success of the batch network gain method in both cases suggests that it is not semi-supervised learning per se that is leading to the performance gains, but rather the representation of interdependence between samples. This finding would support investigation of different graph structures for this and other tasks, as well as tight integration of semi-supervised learning into the network gain model.

While this work has included several alternative approaches for comparision, the focus has been on graph-based models that leverage simple word-based features. It would be of interest to compare the results to alternative learning frameworks that incorporate more complex features, such as Co-EM [49], [50], which also effectively leverages multiple models. A challenge in making a direct comparison is that prior work in this framework [13], [33] has involved very different query strategies (committee-based).

References

[1] D. Cohn, L. Atlas, and R. Ladner, "Improving generalization with active learning," *Mach. Learn.*, vol. 15, no. 2, pp. 201–221, May 1994.
[2] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *J. Artif. Intell. Res.*, vol. 4, no. 1, pp. 129–145, Mar. 1996.

[3] B. Settles, Active Learning Literature Survey, Univ. of Wisconsin-Madison, 2010 [Online]. Available: http://active-learning.net/, Computer Sciences 1648.
[4] B. Settles and M. Craven, "An analysis of active learning strategies for sequence labeling tasks," in *Proc.f EMNLP'08*, 2008, pp. 1069–1078.
[5] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," in *Proc. Symp. Operat. Syst. Design Implement.*, 2004, pp. 137–150.
[6] X. Zhu, Semi-Supervised Learning Literature Survey Univ. of Wisconsin-Madison, 2008 [Online]. Available: http://pages.cs.wisc.edu/jerryzhu/pub/ssl_survey.pdf, Computer Sciences 1530.
[7] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proc. Int. Conf. Mach. Learn.*, 2003, pp. 912–919.
[8] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Proc. NIPS*, 2004, vol. 16, pp. 321–328.
[9] V. Sindhwani and P. Melville, "Document-word co-regularization for semi-supervised sentiment analysis," in *Proc. ICDM*, 2008, pp. 1025–1030.
[10] X. Zhu, J. Lafferty, and Z. Ghahramani, "Combining active learning and semi-supervised learning using Gaussian fields and harmonic functions," in *Proc. ICML Workshop Continuum From Labeled to Unlabeled Data in Mach. Learn. Data Mining*, 2003, pp. 58–65.
[11] A. McCallum and K. Nigam, "Employing EM and pool-based active learning for text classification," in *Proc. Int. Conf. Mach. Learn.*, 1998, pp. 359–367.
[12] G. Druck, B. Settles, and A. McCallum, "Active learning by labeling features," in *Proc. EMNLP*, 2009, pp. 81–90.
[13] R. Jones, "Learning to extract entities from labeled and unlabeled text," Ph.D. dissertation, Carnegie Mellon Univ., Pittsburgh, PA, 2005.
[14] G. Riccardi and D. Hakkani-Tür, "Active and unsupervised learning for automatic speech recognition," in *Proc. Europseech*, 2003, pp. 1825–1828.
[15] G. Tur, D. Hakkani-Tür, and R. E. Schapire, "Combining active and semi-supervised learning for spoken language understanding," *Speech Commun.*, vol. 5, no. 2, pp. 171–186, 2005.
[16] K. Nigam, J. Lafferty, and A. McCallum, "Using maximum entropy for text classification," in *Proc. IJCAI Workshop Mach. Learn. Inf. Filtering*, 1999, pp. 61–67.
[17] S. C. H. Hoi, R. Jin, J. Zhu, and M. R. Lyu, "Batch mode active learning and its application to medical image classification," in *Proc. Int. Conf. Mach. Learn.*, 2006, pp. 417–424.
[18] S. C. H. Hoi, R. Jin, and M. R. Lyu, "Large-scale text categorization by batch mode active learning," in *Proc. Int. Conf. World Wide Web*, 2006, pp. 633–642.
[19] K. Brinker, "Incorporating diversity in active learning with support vector machines," in *Proc. Int. Conf. Mach. Learn.*, 2003, pp. 59–66.
[20] Z. Xu, R. Akella, and Y. Zhang, "Incorporating diversity and density in active learning for relevance feedback," in *Proc. Eur. Conf. IR Res.*, 2007, pp. 246–257.
[21] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: Sentiment classification using machine learning techniques," in *Proc. EMNLP*, 2002, pp. 79–86.
[22] D. Zhou, B. Schölkopf, and T. Hofmann, "Semi-supervised learning on directed graphs," in *Proc. NIPS*, 2005, pp. 1633–1640.
[23] S. Baluja, R. Seth, D. Sivakumar, Y. Jing, J. Yagnik, S. Kumar, D. Ravichandran, and M. Aly, "Video suggestion and discovery for YouTube: Taking random walks through the view graph," in *Proc. WWW*, 2008, pp. 895–904.
[24] P. P. Talukdar, J. Reisinger, M. Paşca, D. Ravichandran, R. Bhagat, and F. Pereira, "Weakly-supervised acquisition of labeled class instances using graph random walks," in *Proc. EMNLP*, 2008, pp. 582–590.
[25] X. Li, Y. Y. Wang, and A. Acero, "Learning query intent from regularized click graphs," in *Proc. SIGIR*, 2008, pp. 339–346.
[26] Y. Y. Wang, R. Hoffmann, X. Li, and J. Szymanski, "Semi-supervised learning of semantic classes for query understanding: from the web and for the web," in *Proc. CIKM*, 2009, pp. 37–46.
[27] L. Velikovich, S. B. Goldensohn, K. Hannan, and R. McDonald, "The viability of web-derived polarity lexicons," in *Proc. NAACL-HLT*, 2010, pp. 777–785.
[28] B. Settles, M. Craven, and S. Ray, "Multiple-instance active learning," in *Proc. NIPS*, 2008, pp. 1289–1296.
[29] G. Nemhauser, L. Wolsey, and M. Fisher, "An analysis of approximations for maximizing submodular set functions I," *Math. Program.*, vol. 14, no. 1, pp. 265–294, 1978.
[30] H. Lin and J. Bilmes, "How to select a good training-data subset for transcription: Submodular active selection for sequences," in *Proc. Interspeech*, 2009, pp. 2859–2862.
[31] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, Bollywood, boomboxes and blenders: Domain adaptation for sentiment classification," in *Proc. COLING*, 2007, pp. 440–447.
[32] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *Proc. ACL*, 2005, pp. 115–124.
[33] I. Muslea, S. Minton, and C. A. Knoblock, "Active + semi-supervised learning = robust multi-view learning," in *Proc. 19th Int. Conf. Machine Learning*, 2002, pp. 435–442.
[34] P. D. Turney, "Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews," in *Proc. ACL*, 2002, pp. 417–424.
[35] S. M. Kim and E. Hovy, "Determining the sentiment of opinions," in *Proc. COLING*, 2004.
[36] D. Rao and D. Ravichandran, "Semi-supervised polarity lexicon induction," in *Proc. EACL*, 2009, pp. 675–682.
[37] B. Pang and L. Lee, "A sentimental education: Subjectivity summarization based on minimum cuts," in *Proc. ACL*, 2004, pp. 271–278.
[38] E. Riloff, S. Patwardhan, and J. Wiebe, "Feature subsumption for opinion analysis," in *Proc. EMNLP*, 2006, pp. 440–448.
[39] M. Gamon, "Sentiment classification on customer feedback data: Noisy data, large feature vectors, and the role of linguistic analysis," in *Proc. COLING*, 2004, pp. 841–847.
[40] T. Mullen and N. Collier, "Sentiment analysis using support vector machines with diverse information sources," in *Proc. EMNLP*, 2004, pp. 412–418.
[41] V. Ng, S. Dasgupta, and S. M. N. Arifin, "Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews," in *Proc. ACL*, 2006, pp. 611–618.
[42] M. Dredze, K. Crammer, and F. Pereira, "Confidence-weighted linear classification," in *Proc. Int. Conf. Mach. Learn.*, 2008, pp. 264–271.
[43] T. Nakagawa, K. Inui, and S. Kurohashi, "Dependency tree-based sentiment classification using CRFs with hidden variables," in *Proc. NAACL-HLT*, 2010, pp. 786–794.
[44] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman, "Learning bounds for domain adaptation," in *Proc. NIPS*, 2008.
[45] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation with multiple sources," in *Proc. NIPS*, 2009, pp. 1041–1048.
[46] S. Dasgupta and V. Ng, "Mine the easy, classify the hard: A semi-supervised approach to automatic sentiment classification," in *Proc. ACL-IJCNLP*, 2009, pp. 701–709.
[47] S. Zhou, Q. Chen, and X. Wang, "Active deep networks for semi-supervised sentiment classification," in *Proc. COLING*, 2010, pp. 1515–1523.
[48] B. Settles, "Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances," in *Proc. EMNLP*, 2011, pp. 1467–1478.
[49] K. Nigam and R. Ghani, "Analyzing the effectiveness and applicability of co-training," in *Proc. Ninth Int. Conf. Inf. Knowl. Manage.*, 2000, pp. 86–93.
[50] U. Brefeld and T. Scheffer, "Co-EM support vector learning," in *Proc. Int. Conf. Mach. Learn.*, 2004, pp. 16–23.

Wei Wu received a Ph.D. in electrical engineering from the University of Washington in 2012. He received the Masters and Bachelors in computer science from Tsinghua University in 2007 and 2004. His research interests are in machine learning for natural language and speech processing.

Mari Ostendorf (M'85–SM'97–F'05) received a Ph.D. in electrical engineering from Stanford University in 1985. She has worked at BBN Laboratories (1985–1986) and Boston University (1987–1999), and is currently an Endowed Professor of System Design Methodologies in Electrical Engineering at the University of Washington. Her research interests are in dynamic and linguistically-motivated statistical models for speech and language processing. She is a Fellow of IEEE and ISCA, and winner of the 2010 IEEE HP/Rigas Award.