Soc. Netw. Anal. Min. (2016)6:8
DOI 10.1007/s13278-016-0317-1

Training a text classifier with a single word using Twitter Lists and domain adaptation

Aron Culotta

Received: 25 November 2015 / Accepted: 19 January 2016
PREPRINT: The final publication is available at Springer via http://dx.doi.org/10.1007/s13278-016-0317-1

Abstract Annotating data is a common bottleneck in building text classifiers. This is particularly problematic in social media domains, where data drift requires frequent retraining to maintain high accuracy. In this paper, we propose and evaluate a text classification method for Twitter data whose only required human input is a single keyword per class. The algorithm proceeds by identifying exemplar Twitter accounts that are representative of each class by analyzing Twitter Lists (human-curated collections of related Twitter accounts). A classifier is then fit to the exemplar accounts and used to predict labels of new tweets and users. We develop domain adaptation methods to address the noise and selection bias inherent to this approach, which we find to be critical to classification accuracy. Across a diverse set of tasks (topic, gender, and political affiliation classification), we find that the resulting classifier is competitive with a fully supervised baseline, achieving superior accuracy on four of six datasets despite using no manually labeled data.

Keywords social media; text classification

1 Introduction

Automatically categorizing social media messages and users is an important task both for improving user experience (e.g., recommender systems (Das Sarma et al, 2010; Hong et al, 2012)) and for supporting emerging technologies that extract insights from such data (e.g., health surveillance (Dredze, 2012), crisis response (Vieweg et al, 2010), and politics (O'Connor et al, 2010)). Supervised text classification is a common solution, but its reliance on manually annotated training data makes it ill-suited to the real-time and non-stationary nature of social media.

In this paper, we instead propose a method to train a social media classifier using the *distant supervision* provided by Twitter Lists. A Twitter List is a manually-curated list of users, typically organized by topic (e.g., "Environmental Journalists" or "Conservatives on Twitter“). Twitter users often build Lists to more easily browse and navigate streaming content. The volume, diversity, and public accessibility of Lists make them a potentially valuable source of knowledge for classification algorithms. (We estimate there to be approximately 3M Lists that are publicly accessible via web search.) This paper investigates ways to use these data for a variety of classification tasks.

The primary advantage of our proposed approach is that the only human input required to train a classifier for a new class is a single keyword representing the class. For example, we train a classifier to categorize tweets by topic using keywords such as "environment," "technology," and "sports". Rather than using manually annotated data, our approach instead searches for relevant Twitter Lists and extracts *exemplar* accounts that are representative of the class label. The tweets from these identified exemplars thus serve as training data for the resulting classifier. For example, tweets from @GreenPeace may serve as training data for the "environment" topic, and tweets from @Engadget may serve as training data for the "technology" topic.

While it requires little human input, using exemplar tweets as pseudo-labeled data is susceptible to con-

Department of Computer Science
Illinois Institute of Technology
Chicago, IL 60616
E-mail: aculotta@iit.edu

siderable noise and bias (e.g., incorrectly identified exemplars may complicate training). Furthermore, since users on Lists tend not to be representative of a typical user (e.g., they may be more likely to be celebrities or news organizations), it may be difficult for the classifier to generalize to the broader population of tweets.

To address these issues, we develop a *domain adaptation* approach that adjusts the training algorithm to account for an unlabeled set of testing data. Our approach combines *self-training* to add pseudo-labeled instances from the test set to the training set along with *feature augmentation* (Daumé III, 2007) to improve accuracy on the target domain.

We perform an extensive empirical validation of our approach on three common classification tasks: categorizing tweets by topic, categorizing Twitter users by political affiliation, and categorizing Twitter users by gender. Furthermore, to test the ability to generalize to very different domains, we also classify web pages by topic, using a classifier trained on Twitter data. In the experiments below, we find that, despite the considerable challenges of noise and bias arising from using Twitter Lists as distant supervision, our approach is surprisingly competitive with fully supervised learning; on four of six datasets, the accuracy of our approach exceeds the supervised approach, even when the supervised approach is trained with all of the available training data. On the remaining two datasets, the supervised approach exceeds our approach only after half of the available training data is used. We perform a number of additional experiments suggesting that the approach is robust to moderate amounts of noise in the exemplar set as well as the number of pseudo-labeled examples added by self-training.

The remainder of the paper is organized as follows: Section 2 reviews prior literature on social media classification; Section 3 describes our approach, including identifying exemplars and performing domain adaptation. Section 4 describes our validation data, and Section 5 presents the results. We conclude in Section 6 with a discussion of limitations and directions for future work.

2 Related Work

Classifying social media users and their messages is an extensive area of research, with target concepts including topic (Lee et al, 2011), sentiment (Go et al, 2009), gender (Rao et al, 2010; Burger et al, 2011; Liu and Ruths, 2013), ethnicity (Rao et al, 2011; Bergsma et al, 2013), age (Nguyen et al, 2011; Al Zamal et al, 2012), personality (Argamon et al, 2005; Schwartz et al, 2013), and political affiliation (Conover et al, 2011; Barberá, 2013; Volkova and Van Durme, 2015). By far the most common approach is supervised learning, which requires collecting a set of users or messages for which the true label is known.

More recently, researchers have investigated ways to train social media classifiers without requiring human labeled data. In addition to the obvious cost savings, removing humans from the training process enables classifiers that scale better over time (classifiers can be retrained continuously to adapt to non-stationary distributions) and in the number of target concepts (new classifiers can be trained as new concepts emerge).

A number of domain-specific heuristics have been used to identify sources of distant supervision; for example, Pennacchiotti and Popescu (2011) train an age classifier from tweets mentioning phrases like “Happy 21st birthday to me”; and Go et al (2009) use emoticons as distant supervision for sentiment classification. Another approach uses estimates of label proportions to guide learning; for example, ethnicity, age, and gender classifiers have been trained by using U.S. Census statistics by location and name (Chang et al, 2010; Oktay et al, 2014; Ardehaly and Culotta, 2015) as well as website traffic demographics (Culotta et al, 2015).

In this paper, we investigate using Twitter Lists as a source of distant supervision. Due to their number and diversity, Twitter Lists can potentially provide a source of supervision for a wide range of classification tasks. A few recent papers support this direction. Kim et al (2010) found that keywords extracted from the tweets of a List’s members serve as reliable indicators of their interests and attributes. García-Silva et al (2015) found that the descriptions of Twitter Lists can be used to enhance domain ontologies such as WordNet.

There are at least two recent works that use Twitter Lists for classification. First, Burgess et al (2013) propose a system to rank a Twitter user’s feed by creating lists of users (corresponding to clusters in the user’s ego network), then training a Naive Bayes classifier for each list. Heuristic attempts are made to reduce label noise by only training on tweets that contain certain discriminative keywords. The classifier is then used to categorize the user’s feed by topic (i.e., list). Whereas Burgess et al. are interested in categorizing a user’s feed into existing Lists, here we are using Lists in a much different manner. Our approach is to use many lists matching a certain keyword as a source of distant supervision for an externally defined concept.

Second, Yang et al (2014) propose a system to classify tweets by topic using lists of users, hashtags, and URLs as training signals. To reduce label noise, they restrict training data to tweets containing URLs, then use co-training to improve tweet classification using the text of linked web pages. Whereas Yang et al. require a human to provide a list of users as a source of supervision, here we use a single keyword to automatically identify such users.

Both of these prior works recognize the problems of noise and bias in using Lists as a source of supervision, and propose various heuristic methods to mitigate it (Burgess et al. by keyword filtering; Yang et al. by restricting to tweets with URLs). In this paper, we propose addressing this problem with a novel variant of domain adaptation. Our empirical results suggest this approach provides an effective and general way to train classifiers that are robust to this noise and bias. In the context of this prior work, our primary contributions are as follows:

- We introduce a novel method to quickly create distantly labeled training data by identifying Twitter Lists relevant to a keyword. Thus, input to the learning algorithm is simply a list of keywords, one per class.
- We propose extensions to existing domain adaptation algorithms to make the resulting classifiers robust to the bias and noise introduced by such training data.
- We perform an extensive empirical comparison to validate the approach on a diverse set of classification tasks, both on Twitter (topic, gender, and political classification) and on web page classification, demonstrating the viability of our approach as an alternative to fully supervised learning for social media classification.

3 Methods

A Twitter List is a manually curated collection of Twitter users. Users often create lists to filter feeds by topic (e.g., an "Environment" list may contain a collection of Twitter accounts for environmental non-profits). A list may be made public or private by its creator, though public is the default option. We were unable to determine the exact number of Twitter Lists that have been created, but a search query[1] suggests that there are about 3M Twitter Lists indexed by Google.

Our goal is to leverage this considerable resource to build a social media classifier without manually annotating additional Twitter data. The core idea is to use a single keyword to represent each class label (e.g., "environment," "music," or "movies"). For each keyword, we collect relevant Twitter Lists and identify *exemplar* accounts, which we determine to be representative of the target concept. We then collect the tweets of each exemplar and use them to train a multi-class classifier.

The main advantage of such an approach is that it requires only a single keyword of input from a user. All subsequent training is done without further input from a human. A secondary advantage is that it is amenable to frequent updating, which is critical for many social media tasks because linguistic drift can cause classifier accuracy to quickly deteriorate. With the proposed approach, the exemplar tweets can be tracked in real-time, enabling the classifier to be re-trained as desired.

This approach presents a number of challenges. First, it is clearly dependent on the quality of the human-provided keyword. A poorly chosen keyword may identify exemplars that are irrelevant to the true target concept. Second, due to the noisy nature of Twitter Lists, even a well-crafted keyword may return irrelevant exemplars. Finally, and in our view most importantly, tweets from exemplar accounts are a biased sample of the population of all tweets (or all social media messages). There are a number of possible sources of such bias; for example, users on Twitter Lists may be more likely to be celebrities or news organizations, or may be more likely to write in the third-person. Depending on the target test data, the exemplar tweets may be from a different time period. Such biases can result in a classifier that does not generalize well to a new test set.

Below, we first describe the details of how we collect Twitter Lists and exemplars, then we describe our proposed approach to reduce the aforementioned sources of noise using domain adaptation algorithms.

3.1 Retrieving Twitter Lists and Exemplars

Given a keyword (e.g., "environment"), we perform a fielded Google search to identify matching Twitter Lists. Specifically, we use the query [*site:twitter.com inurl:lists <keyword>*]. This queries Google for pages in the domain twitter.com that have the term "lists" in their URL. For example, the URL [`https://twitter.com/nrdc/lists/environmental-journalists`] is the list named "Environmental Journalists" created by the user @NRDC. The keyword argument is used to match the List's homepage. While it is not publicly known how Google ranks such pages, each List's homepage contains a brief description of the List as well as the most recent tweets from its members. Experiments with Google queries indicates that this content is in part used to rank Lists.

We iterate through the top search results for each query, up to a maximum of 50 Lists. For each List, we

[1] The query used is: *site:twitter.com inurl:lists*.

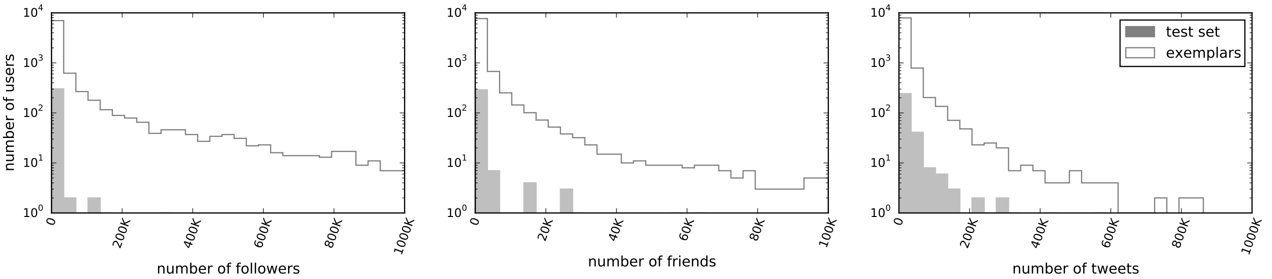


Fig. 1 Comparison of the distributions over the number of followers, friends, and tweets for users identified as exemplars versus those in the testing set for topic classification, collected from the Twitter Streaming API. Exemplars appear to tweet more often and have more friends and followers than users in the test set.

use the Twitter API to download its members.[2] The resulting set of accounts becomes our candidate exemplar set. To reduce noise, by default we restrict the exemplars to those who (1) appear in at least two of the 50 retrieved lists and (2) have at least 50 public tweets. (We will investigate how the number of exemplars affects accuracy in in Section 5.3.1.)

For multi-class classification, we issue one query per class, and further restrict exemplars to those retrieved by at most one query (to remove exemplars that have multiple labels). Thus, this stage produces a list of Twitter users whose tweets are likely to be relevant to each class.

3.2 Processing Exemplar Tweets

We download the most recent 200 tweets for each exemplar using the Twitter API,[3] restricted to English tweets. We perform standard tokenization, removing punctuation, converting to lower case, and maintaining hashtag and mentions. URLs are collapsed to a single feature type, as are digits. Terms that occur only once are removed, as are terms from a list of 500 Twitter-specific stop words.[4] Each exemplar is represented by a binary unigram vector, where the value 1 in position j indicates that the exemplar account has written at least one tweet containing term j. (Preliminary experiments suggested this representation was more effective than a word frequency vector.)

3.3 Exemplar Classifier

The preceding steps result in a traditional supervised classification training set $S = \{(x_1, y_1) \ldots (x_n, y_n)\}$, in which each exemplar i is represented by its binary feature vector $x_i = \{x_{i1} \ldots x_{ik}\}$ and its label y_i, as indicated by the query by which the exemplar was found.

We fit the parameters θ of a multinomial logistic regression model using S:

$$p(y_i = c | x_i; \theta) = \frac{e^{\theta^c \cdot x_i}}{\sum_{c'} e^{\theta^{c'} \cdot x_i}}$$

$$\theta^* \leftarrow \underset{\theta}{\operatorname{argmax}} \prod_{i \in S} p(y_i | x_i; \theta)$$

where θ^c is the parameter vector for class c. (We additionally use L2 regularization, omitted from this equation.)

Thus, this classifier is trained to categorize exemplars by which keyword was used to retrieve them. Because the number of exemplars per class may vary, we mitigate class imbalance by using sample weights that are inversely proportional to the class frequency (Elkan, 2001). (If there are m classes, and f_i is the frequency of class i in S, then the weight for an instance with label i is $\frac{f_i^{-1}}{\frac{1}{m}\sum_j f_j^{-1}}$.) In the experiments below, we refer to this baseline classifier as **lists**.

3.4 Domain Adaptation by Self-Training

As discussed above, the resulting classifier is optimized for classifying accounts that are likely to appear on Twitter Lists; however, exemplar accounts are unlikely to be representative of the broader population of tweets and users.

To get a sense for the differences between exemplars and other users, we plot in Figure 1 histograms of the number of followers, friends, and tweets for each exemplar in the topic classification dataset (described in more detail in Section 4). For comparison, we also plot histograms for users from the test set, which were collected using the Twitter Streaming API. We can see that exemplars typically tweet more and have more followers and friends (accounts they follow) than users in

[2] `https://api.twitter.com/1.1/lists/members.json`

[3] `https://api.twitter.com/1.1/statuses/user_timeline.json`

[4] We created a Twitter-specific stop list containing the 500 most frequently used words from a sample of a year's worth of English tweets.

the test set. These behavioral differences suggests that the content of exemplar tweets may also differ from those of ordinary users.

More formally, assume we wish to classify a new dataset of unlabeled tweets $T = \{x_1 \ldots x_m\}$. The problem is that the source data S and the target data T are not independent draws from the same underlying distributions, e.g. $P_S(X, Y) \neq P_T(X, Y)$. Depending on the nature of this mismatch, this problem may be called sample selection bias (Heckman, 1979; Zadrozny, 2004), covariate shift (Shimodaira, 2000; Bickel et al, 2009), concept drift (Widmer and Kubat, 1996), or non-stationary classification (Fokianos and Kedem, 1998).

Domain adaptation algorithms (Huang et al, 2006; Ben-David et al, 2010) address this problem by using information from the target domain T to adjust the learning procedure. Such approaches can roughly be categorized as *supervised*, *semi-supervised*, or *unsupervised*. Supervised domain adaptation has access to some labeled data from the target domain $T_L \subset T$; the classifier is fit using S and T_L. Semi-supervised domain adaptation combines T_L with additional unlabeled target data $T_U \subset T$ for training. Unsupervised domain adaptation assumes we have no labeled data from T (that is, $T_U \equiv T$, $T_L = \emptyset$). Unsupervised domain adaptation is the most desirable setting for our problem — to adapt classifiers in near real-time to rapidly changing social media trends, we would prefer to not have to acquire any additional labeled data.

One approach to unsupervised domain adaptation is to use *self-training* to gradually add pseudo-labeled data from T_U into T_L (Chen et al, 2011). The idea is to fit a classifier on S, then use it to predict labels for T_U. The most confidently labeled instances from T_U are added to T_L, using the output of the classifier as the true label. Over several iterations, by training the classifier on $S \cup T_L$, the resulting classifier is adapted to the distribution of T. Such techniques are commonly used in natural language processing to transfer parsers across domains (Bacchiani et al, 2006; McClosky et al, 2006a,b).

We adapt this approach to our problem setting. The proposed algorithm is given in Figure 2. For example, consider the binary classification task of predicting the political preference of users (e.g., liberal or conservative). We create the initial set S by searching for Twitter Lists matching the terms "liberal" or "conservative," extracting exemplars as described in Section 3.1. We additionally have access to an unlabeled set T_U of users to be classified. Step 2a fits the classifier C on S; in 2b, we use C to classify all elements of T_U. Next, we add two users from T_U to T_L by picking the most confident "liberal" and "conservative" examples, according to the posterior probability of the classifier. The algorithm continues to retrain the classifier and add additional self-labeled data until a convergence criterion is met. Thus, at each iteration, the labeled data $S \cup T_L$ (and the resulting classifier C) become more reflective of the target domain T.

```
1. Input:
   Source labeled data $S$ (labeled exemplar accounts)
   Target unlabeled data $T_U$
   Target labeled data $T_L \leftarrow \emptyset$.
2. while not converged:
   (a) $C \leftarrow$ Fit classifier on $S \cup T_L$.
   (b) Classify all examples in $T_U$ using $C$.
   (c) Select the most confidently predicted examples $T^* \subseteq T_U$, one per class.
   (d) Move $T^*$ from $T_U$ to $T_L$, using the predicted label as truth:
       $T_L \leftarrow T_L \cup T^*$
       $T_U \leftarrow T_U \setminus T^*$
3. Return: $C$
```

Fig. 2 Self-training domain adaptation algorithm (**self-train**).

Convergence may be determined by a confidence threshold on the classifier's posteriors or by a fixed number of iterations. In the experiments below, we stop when 100 examples have been added to T_L. (We also report experiments that vary this threshold.) As with the **lists** classifier, we assign sample weights inversely proportional to class frequency to handle class imbalance. In the experiments below, we refer to the resulting classifier as **self-train**.

3.5 Unsupervised EasyAdapt

Daumé III (2007) proposed a simple and effective supervised domain adaptation approach called EasyAdapt that augments the feature space of the classifier using the target data T. Specifically, if the original feature space defined in S has k features ($\mathcal{X} \subset \mathbb{R}^k$), then the new feature space triples in size ($\mathcal{X} \subset \mathbb{R}^{3k}$). Each feature in the original space is triplicated, resulting in three types of features: *source-specific* features representing terms from S, *target-specific* features representing terms from T, and *common* features representing terms shared by S and T. Each original feature vector $x_i \in S \in \mathbb{R}^k$ is thus expanded to become $\langle x_i, x_i, \mathbf{0} \rangle$, and each vector $x_i \in T_L$ is expanded to become $\langle x_i, \mathbf{0}, x_i \rangle$. The three components of this new vector correspond to the common, source-specific, and target-specific features, respectively. This approach in effect boosts the influence of target data on the resulting classifier trained on labeled data S and T_L.

While EasyAdapt was originally designed for supervised domain adaptation, to our knowledge it has not

```
1. Input:
   Source labeled data $S$
   Target unlabeled data $T_U$
2. Fit a binary classifier $p(s|x;\lambda)$, where $s = 1$ indicates that
   $x \in S$, $s = 0$ indicates that $x \in T_U$, and $\lambda$ is a parameter
   vector. The training data consist of $S \cup T_U$.
3. Reweight each instance $i$ in $S$ by $w_i \propto p(s = 0|x_i;\lambda)$
4. $C \leftarrow$ Fit classifier on weighted examples in $S$.
5. Return: $C$
```

Fig. 3 Instance reweighting algorithm to adjust for covariate shift (**reweight**).

been used for unsupervised domain adaptation (though it has been extended to semi-supervised domain adaptation (Daumé III et al, 2010)). Since we assume no labeled data from T, we propose an unsupervised version of EasyAdapt that uses the self-training approach from the previous section. Specifically, in step 2a of the **self-train** approach, we apply EasyAdapt to fit the classifier. That is, we assume the self-labeled data are correct and run EasyAdapt as usual. In the experiments below, we refer to the resulting classifier as **self-train-easy**.

In addition to the methods discussed so far (**lists**, **self-train**, **self-train-easy**), we also compare with other baselines that attempt to address covariate shift and noisy exemplars, as described next.

3.6 Covariate Shift

Covariate shift refers to the case where $P_S(X) \neq P_T(X)$; that is, the feature densities differ between the source and target data, but not necessarily the conditionals $P(Y|X)$. For example, consider again the political preference classification task. Suppose the exemplars in S tend to mostly discuss the politics of health care, whereas the users in T mostly discuss immigration. If we ignore T, the classifier coefficients will be dominated by health care related terms, potentially reducing accuracy on immigration-related messages. As a baseline, we implement the approach of Zadrozny (Zadrozny, 2004), which reweights each training example in S proportional to how similar it is to the testing set T. For example, exemplars in S who mostly discuss immigration may be assigned higher weights than exemplars who mostly discuss health care. The idea is to focus the classifier on parts of the feature distribution that are more probable in T. The algorithm is provided in Figure 3.

Note that unlike the self-training approach of the previous section, the final classifier is not fit to any instances from T; rather, T is simply used to reweight the instances in the original labeled data S.

Zadrozny (Zadrozny, 2004) provides a theoretical justification for setting the weights $w_i = \frac{1}{p(s=1|x_i;\lambda)}$. However, in initial experiments, we found that the resulting distribution of weights was skewed such that a small number of examples had weights two orders of magnitude higher than the rest, leading to poor classification accuracy. (Most of the training data were effectively ignored.) We suspect this is due to the large differences between the source and target data. Instead, we restrict the range of weights by re-normalizing the weight vector across S and adding the result to one: $w'_i = 1 + \frac{w_i}{\sum_{j \in S} w_j}$. We found this to provide a stronger baseline. In the experiments below, we refer to the resulting classifier as **reweight**.

3.7 Outlier removal

Finally, we consider a simple outlier detection method to prune the list of exemplars. Given that no human filtering is done to remove false matches from the exemplar list, it is possible that some labels are incorrect (e.g., a search for Lists matching “liberal” may return some conservative exemplars). To mitigate this, we perform k-means clustering on the feature vectors for all exemplars (after an idf transformation). We partition the exemplars into ten clusters, then retain the two largest clusters for training. (A few alternative values did not consistently improve results.) The classifier is trained using standard logistic regression, as in the **lists** approach. We refer to the resulting classifier as **outlier**.

4 Validation Data

To evaluate the proposed methods, we consider three different Twitter classification tasks: classifying tweets by topic, and classifying users by political affiliation or gender. Additionally, to test the ability of our approach to adapt to very different domains, we also evaluate on the task of classifying web pages by topic, using a model fit to Twitter data. For each task, we collect a testing set as well as a corresponding set of exemplars. The six datasets are summarized in Table 1 and described in more detail below.

4.1 Topic Classification

Based on prior work in Twitter classification (Lee et al, 2011), we identified 13 topical categories: art, books, business, environment, fashion, health, movies, music, nutrition, politics, religion, sports, and technology. To collect validation data, we sampled 1,000 tweets from one day from the Twitter Streaming API, filtered to

Table 1 Summary statistics of the six testing datasets.

		test data		exemplars	
name	**# classes**	**# tweets**	**# users**	**# tweets**	**# users**
topics	13	310	310	1.4M	7,468
pol-cand	2	210K	1,053	758K	3,828
pol-geo	2	76K	411	"	"
pol-zlr	2	319K	357	"	"
gender	2	284K	303	447K	2,284
dmoz	13	524 web pages		[same as **topics**]	

the English language, and manually annotated them. Of these 1,000 tweets, 310 were assigned at least one of the 13 categories. Of these, 51 received two labels, the most common of which (movie and music) occurred 8 times. For simplicity, we restrict ourselves to single-label classification methods. If a system predicts at least one of the labels, we mark it as correct. (In future work we will consider extensions to multi-label classification.) We refer to this dataset as **topic**.

To collect exemplars, we used each category name (listed above) as the Twitter List query. This resulted in 7,468 unique exemplars (the number of exemplars by class are enumerated in Table 2). The most common exemplars include traditional media accounts (e.g., @guardianeco, @NYTHealth), non-profits (e.g., @Greenpeace), and individual writers and critics (@JerrySaltz, an art critic, and @CLGrossman, a journalist who writes about religious issues).

4.2 Political Classification

We use the political preference data of Volkova et al (2014), which in turn builds on the labeled data of Pennacchiotti and Popescu (2011) and Al Zamal et al (2012). See Volkova (2014) for a detailed description of the data.

These data have three subsets:

- **pol-candidate:** Users who follow @BarackObama and @JoeBiden but not @MittRomney and @RepPaulRyan, the candidates for the 2012 U.S. presidential election. This contains 539 Republican users and 516 Democratic users.
- **pol-geo:** Users from Maryland, Virginia, or Delaware who report their political affiliation in their Twitter profile description (e.g., "I'm a father, husband, Republican"). This contains 183 Republican users and 230 Democratic users.
- **pol-zlr:** Users who identified as Democratic or Republican on the sites `wefollow.com` or `twello.com`, as described in Pennacchiotti and Popescu (2011). This contains 167 Republican and 191 Democratic users.

We used the keywords "liberal" and "conservative" to collect exemplars, resulting in 1,352 conservative accounts and 2,509 liberal accounts. Exemplars appearing most frequently include think tanks (@Heritage), commentators (@GlennBeck, @SeanHannity), as well as individuals with no official political connection, but who are active politically on Twitter.

4.3 Gender Classification

We use the gender data of Al Zamal et al (2012) as extended by Volkova et al (2014). These data were labeled by matching the first names of Twitter profiles with the 100 most common male and female names, as reported by the U.S. Social Security Administration. This contains 146 female users and 157 male users.

An obvious limitation of our approach is that it is most appropriate for target concepts that are also the subjects of Twitter Lists. In this regard, we expect politics and topics to be natural fits. However, we are also interested in what would happen for more general topics, such as gender, that are not as likely to be the subject of a List. To collect exemplars for the gender classification task, we used the keywords "male" and "female." Unlike the prior tasks, this only resulted in a small number of exemplars that appeared in at least two lists (422). To increase the amount of training data, we added an additional 1,000 users for each class sampled randomly from the list of users that appeared on only one List. Filtering users with fewer than 50 tweets resulted in a final list of 1,199 female accounts and 1,085 male accounts. While some of these Lists focus on issues of gender (e.g., "Female & Feminist Issues on Twitter"), many are simply lists of people categorized by gender (e.g. "Male Bloggers on Twitter," "Female Tech Leaders on Twitter").

4.4 Topic Classification of Web Pages

To test how well a classifier trained on Twitter can be applied to other data sources, we collected a set of web pages labeled by topic using DMOZ[5], a human-curated taxonomy of web pages. For each of the 13 categories used in the Twitter topic classification task (e.g., environment, fashion), we manually identified the most appropriate DMOZ category (e.g., Science / Environment / News_and_Media, or Arts / Design / Fashion / Weblogs). We then downloaded the homepage of each of the identified URLs, using the same preprocessing steps as used for the Twitter data. This resulted in 524 webpages, each of which is assigned to exactly one category. We use the same exemplars for training as collected for Twitter topic classification (Section 4.1). We refer to this dataset as **dmoz**.

5 Results

In this section we report classification accuracy for each task. We also present the results of several robustness checks to determine how the approach varies with the number and quality of exemplars, as well as with the number of self-labeled training instances. In addition to reporting results on the held-out testing data, we also report cross-validation accuracy on classifying exemplars.

5.1 Exemplar Classification

We first report the accuracy of the **lists** classifier in categorizing exemplars. Recall that the set S contains exemplars labeled by class, as determined by the keyword used to retrieve the exemplar. While we are ultimately more interested in accuracy on the held-out (non-exemplar) data, this task has practical applications in its own right — for example, in recommending accounts to add to existing Lists. Additionally, accuracy on this task gives us a sense of the cohesion of the exemplar sets.

We perform 5-fold cross-validation and report precision, recall, F1 of the **lists** classifier. (Since the other approaches are designed to improve performance on the held-out data T, their accuracy on S does not differ substantively from **lists**. We thus omit these results for brevity.)

Tables 2-4 show the results for the three tasks. Overall, classification accuracy is high for each task, with average F1 scores ranging from .83 for gender classification to .89 for topic classification. Surprisingly, accuracy for the topic task (a 13-class problem) is higher than that for the gender task (a 2-class problem). We find this to be due in part to the amount of noise and diversity in the exemplar set. The keywords used to collect exemplars by topic resulted in more tightly coherent sets of users than those found by searching for the more general keywords “male” and “female.” In contrast, the easiest task appears to be classifying exemplars by political affiliation (F1=.98). This is perhaps not surprising given the polarized language of political discourse.

Table 2 Exemplar cross-validation accuracy for topic classification.

	Prec	**Rec**	**F1**	**N**
art	0.91	0.85	0.88	858
books	0.81	0.87	0.84	569
business	0.93	0.94	0.94	1420
environment	0.94	0.93	0.94	616
fashion	0.94	0.90	0.92	136
health	0.93	0.87	0.90	616
movies	0.87	0.69	0.77	143
music	0.81	0.91	0.86	489
nutrition	0.92	0.92	0.92	478
politics	0.83	0.87	0.85	312
religion	0.89	0.83	0.86	725
sports	0.92	0.96	0.94	101
technology	0.86	0.91	0.88	1005
avg	0.89	0.89	0.89	7468

Table 3 Exemplar cross-validation accuracy for political classification.

	Prec	**Rec**	**F1**	**N**
conservative	0.97	0.97	0.97	1343
liberal	0.99	0.98	0.98	2485
avg	0.98	0.98	0.98	3828

Table 4 Exemplar cross-validation accuracy for gender classification.

	Prec	**Rec**	**F1**	**N**
Female	0.83	0.85	0.84	1199
Male	0.83	0.81	0.82	1085
avg	0.83	0.83	0.83	2284

To provide some insight into the fit models, Table 5 lists the top five terms for each class, ranked by the fit coefficients of the **lists** classifier. (For readability, we have expanded some of the collapsed numeric features.) Most of the terms should be intuitive, though a few require explanation. For **politics**, a number of lists concerning Dutch politicians were returned; for **conservative**, a number of conservative exemplars frequently discuss the amount of the U.S. government debt (thus the large number); #tcot is a hashtag for “Top conservatives on Twitter”; #p2 is for progressives on Twitter; “watchman” is a reference to fears of government overreach; “5o” is an abbreviation for “fifth” in Spanish, in reference to an immigration decision made by the 5th Circuit Court in the U.S.

[5] dmoz.org

Table 5 The top five terms for each class according to the coefficients fit by the **lists** classifier on the Twitter List data.

art	#art, art, artist, painting, artists
books	books, book, author, fiction, reading
business	#business, business, #marketing, #entrepreneur, #money
environment	climate, energy, environmental, water, green
fashion	fashion, collection, wear, dress, style
health	patients, healthcare, medical, patient, drug
movies	movies, films, theaters, film, exclusive
music	album, songs, single, tour, listen
nutrition	nutrition, healthy, eating, muscle, fuel
politics	dutch, hillary, workers, clinton, election
religion	religion, church, christian, faith, religious
sports	games, sport, sports, championships, final
technology	tech, app, sf, apple, aol
conservative	liberty, 99,999,999,999,999.99, #nationaldebt, #tcot, watchman
liberal	#uniteblue, #p2, 5o, hillary_4_prez, marriage
female	women, female, women's, writing, included
male	brother, wife, x, male, york

Table 6 Micro F1 of competing methods on held-out testing data.

	topic	**pol-cand**	**pol-geo**	**pol-zlr**	**gender**	**dmoz**
self-train-easy	**0.48**	0.76	0.67	**0.92**	**0.79**	**0.81**
self-train	0.47	**0.77**	0.68	**0.92**	0.75	0.79
reweight	0.46	**0.77**	**0.69**	**0.92**	0.71	0.73
outlier	0.11	0.76	0.67	0.89	0.50	0.50
lists	0.11	0.76	**0.69**	**0.92**	0.59	0.56
supervised	0.29	0.68	0.68	0.85	0.71	0.58

5.2 Out-of-sample Classification

Next, we report results classifying the held out testing data, as described in Section 4. We compare the five classifiers described previously that use no manually labeled data, only Twitter Lists. We summarize these here for reference:

- **lists**: The classifier fit to exemplar tweets, with no modifications (Section 3.3).
- **outlier**: Same as **lists**, but exemplars determined to be outliers are removed from the training data (Section 3.7).
- **reweight**: Reweights the training data to adjust for covariate shift (Section 3.6).
- **self-train**: The self-training domain adaption algorithm (Section 3.4).
- **self-train-easy**: EasyAdapt with self-training (Section 3.5).

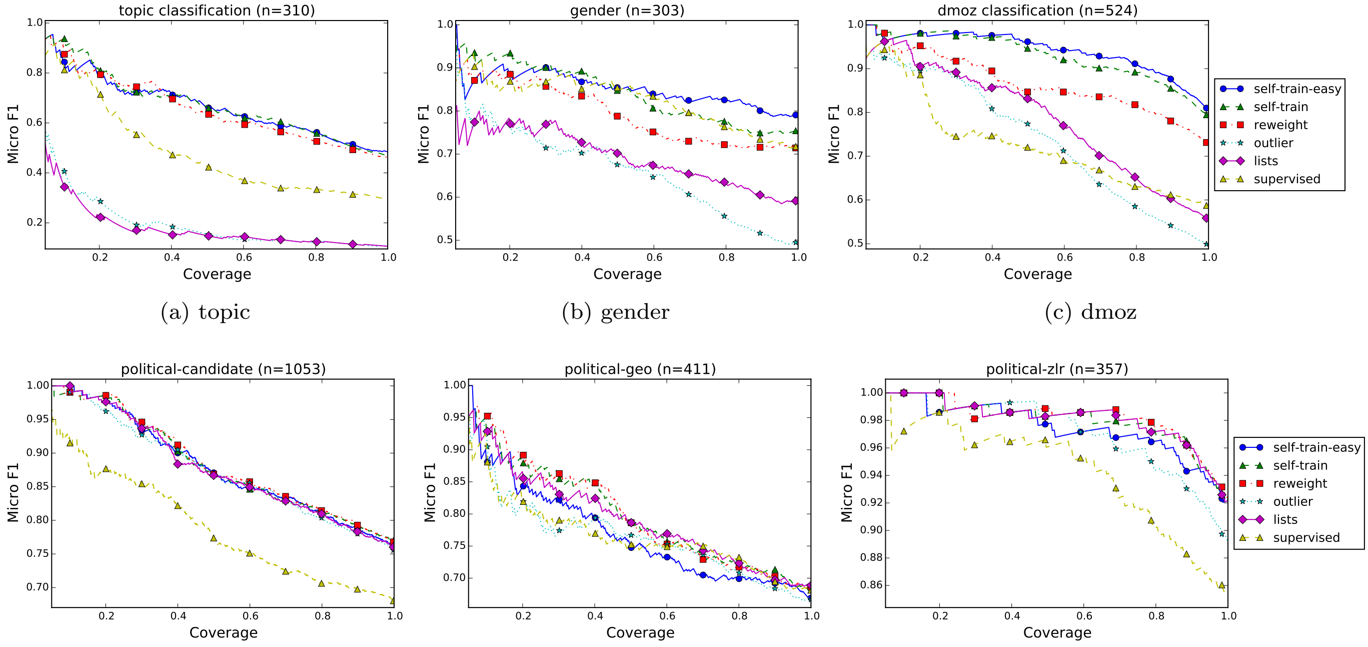
In addition, we compared to a supervised baseline using multinomial logistic regression. Unlike the five classifiers above, this classifier is provided with manually labeled tweets/users from the testing set. For the first set of results, we restrict the number of labeled instances to 100 (we will vary this subsequently). We use two-fold cross-validation to obtain predictions for each instance in the testing set.

Our primary evaluation metric is micro-averaged F1. In addition, we wish to display how the classifiers perform using different confidence thresholds (to reflect the common use case in which only the most confident n predictions are shown to the user). To do this, we plot figures showing how micro-F1 varies with coverage. That is, we compute the posterior probabilities for each test instance and sort in descending order. At each rank, we recompute micro-F1 only considering the predictions seen thus far. For example, if micro-F1 is .5 at coverage .2, this means that if we only consider the 20% most confident predictions, the micro-F1 is .5.

Table 6 summarizes the results across all datasets, and Figure 4 displays these results as coverage varies. For five of six tasks, either **self-train** or **self-train-easy** results in the highest F1, suggesting that domain adaptation is an effective approach to this problem. In some cases, the improvement over the **lists** baseline is substantial; for example, on **topic** F1 improves from .11 to .48; on **gender** it improves from .59 to .79; and on **dmoz** it improves from .56 to .81.

Of the other baselines, **reweight** appears to be more effective than **outlier**; **reweight** improves over **lists** on four of six datasets, while **outlier** never produces higher accuracy than **lists**. The ineffectiveness of **outlier** may in part be due to the simplicity of our approach, and in part due to the fact that accuracy improves with the number of exemplars (discussed more below).

Note that the results in Table 6 are, as expected, lower than those for exemplar classification in Tables 2-



(a) topic (b) gender (c) dmoz

(d) political-cand (e) political-geo (f) political-zlr

Fig. 4 Micro F1 of competing methods at increasing levels of coverage. Predictions are sorted by descending order of confidence and F1 is recomputed at each rank.

4. There are a couple of reasons for this: (1) each exemplar is represented by up to 200 tweets, whereas the held-out data for topic classification consists of individual tweets; (2) the exemplars are, by design, representative of a topic, and so are more likely to use clear, topic-specific terms.

To better understand why **self-train** improves over **lists**, Figure 5 summarizes differences between the fit coefficients of the two classifiers. Additionally, we plot the coefficients of a logistic regression classifier fit on all of the testing data; we label these the "oracle" coefficients, as they are obtained by unfairly having access to the labels of the testing data. For each panel in Figure 5, we identify the 500 terms with the highest absolute coefficient according to the oracle model. We then select the 15 terms that are most improved by **self-train** (i.e., the terms for which the difference between the **lists** coefficient and the **oracle** coefficient is most reduced by self-training). For **topics**, we select one of the 13 classes for display purposes (sports). All coefficients are converted to z-scores for comparison purposes.

We highlight a few observations from this figure that are suggestive of the types of ways that self-training may help. For example, in Figure 5a, we see that by far the term most associated with the "sports" class in the testing data is "madrid." Indeed, several of the sports tweets in the testing set concern a soccer (football) match between Real Madrid and Juventus. This also explains the "[NUM]-[NUM]" feature, which arises from tweets reporting the score in the match (e.g., "1-1"). We can see that self-training increases the coefficients for these soccer-related terms. Self-training is able to accomplish this because these tweets contain other terms that the **lists** classifier already associated with sports (e.g., "goal" and "final"). By increasing the coefficients for "madrid" and "juventus," the **self-train** classifier is then able to correctly classify tweets that do not contain the terms "goal" or "final." Thus, this example suggests how **self-train** allows the classifier to adapt to changes in news events.

The "[NUM]-[NUM]" feature makes another appearance in the **gender** data (listed as "9-9"). Here, the **lists** classifier associates it with female users, whereas in the testing set it is strongly associated with male users. Self-training pushes the coefficient in the correct direction (though does not go far enough). This is an example in which the conditional distribution $P(Y|X)$ changes between the training set and testing set. Examining the raw data sheds light on this shift: several female training exemplars are selected from Twitter Lists of female athletes. These female users often report scores of female athletic competitions. In the testing set, however, many messages containing this feature are written by male users, often reporting scores of male athletic competitions.

In addition to the features shown here, a number of other features assigned large coefficients by the **lists** classifier were correctly downweighted by self-training. These features often pertain to terms from bad exemplars (e.g., the Dutch political accounts that are not relevant to the U.S.-dominated testing set) or to overfitting (e.g., features with low support in the exemplar training data). As these examples and others illustrate, self-training enables the classifier to adjust coefficients to better reflect the characteristics of the target dataset.

5.2.1 Comparison with fully supervised approach

Table 6 indicates that our domain adaptation approach outperforms a supervised baseline for five of six datasets. For the one dataset where supervised learning does better (**pol-geo**), the results are nearly indistinguishable (.67 vs. .68).

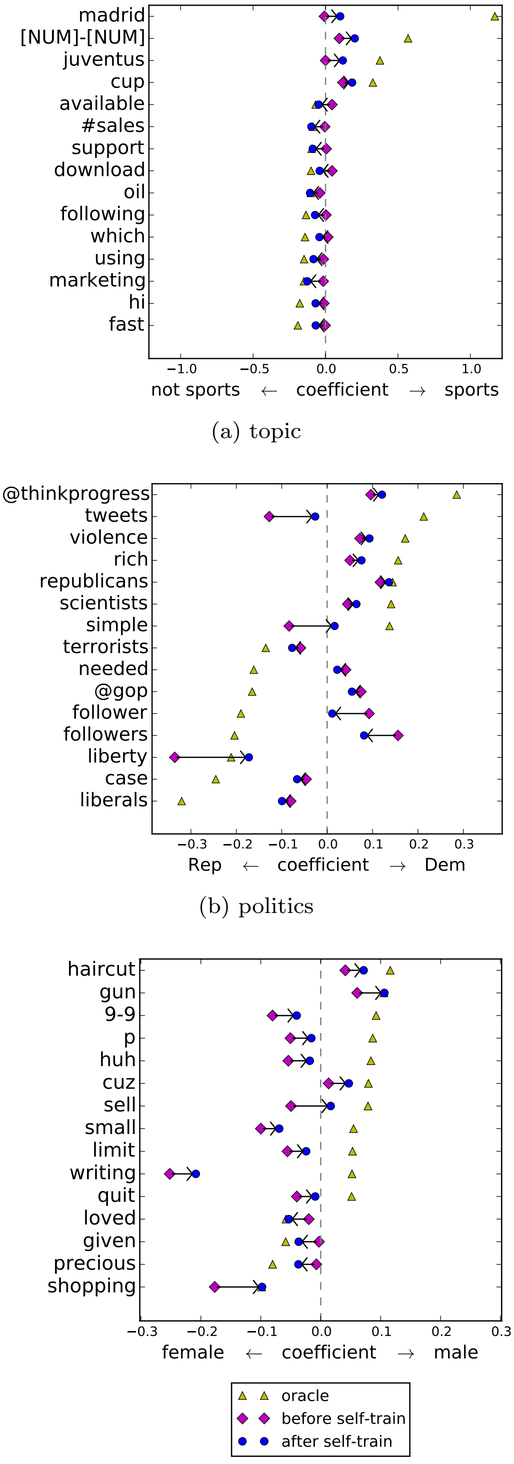
This is in part due to the small training sample available to the supervised approach (100 labeled examples). To evaluate the impact of this, Figure 6 displays results as increasingly more labeled data are provided to the supervised baseline, again using two-fold cross-validation. (As the labeled data are not used by the domain adaptation approach, this does not affect their accuracy.)

Somewhat surprisingly, the domain adaptation approach still remains competitive with the supervised baseline. On four of six datasets, **self-train-easy** outperforms the supervised approach even after all the available training data are used. For the two datasets where the supervised approach performs best, accuracy exceeds **self-train-easy** only after half of the available training data are used.

This is especially notable for **dmoz**. Here, the supervised approach is fit to a set of webpages; **self-train** is fit to a set of Twitter accounts, then uses pseudo-labeled web pages. Despite not having access to any labeled web pages, **self-train** outperforms the best supervised approach. It appears that the volume of distantly labeled data used by **self-train** can overcome the lack of in-domain examples.

Comparing to previously published work with these datasets, Al Zamal et al (2012) report a supervised classification accuracy of .795 on the **gender** data (compared with .792 accuracy of **self-train-easy**). Using features from a user's social network (which we have not explored here), they improved accuracy to .802.

For the three political datasets, our approach **self-train-easy** meets or exceeds the accuracies of the batch supervised baselines reported in Volkova et al (2014). They report two baselines: one uses only features from a user's tweets, the other also includes features from tweets of neighbors in the network. Table 7 summarizes these results. Even when the supervised baseline



(a) topic

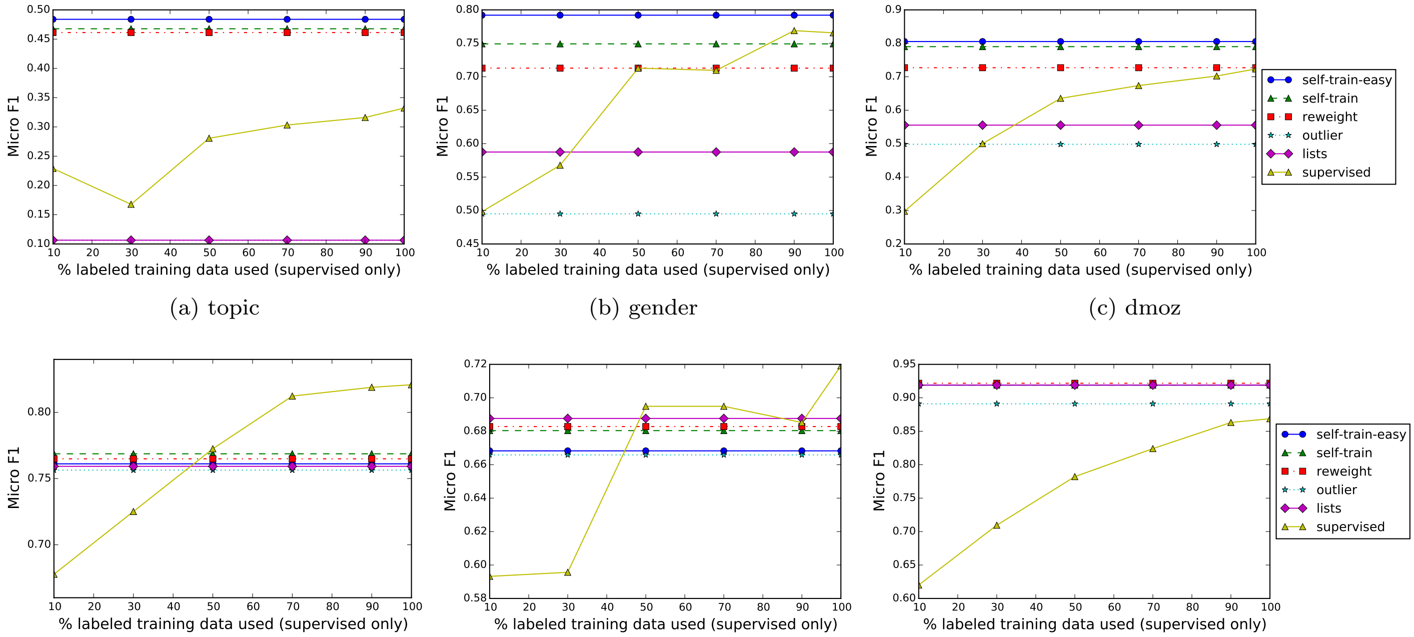
(b) politics

(c) gender

Fig. 5 Example features and their coefficients before and after self-training, as compared with those learned by an oracle trained on the testing data.

Table 7 Accuracy of **self-train-easy** compared with accuracies of supervised baselines reported in prior work.

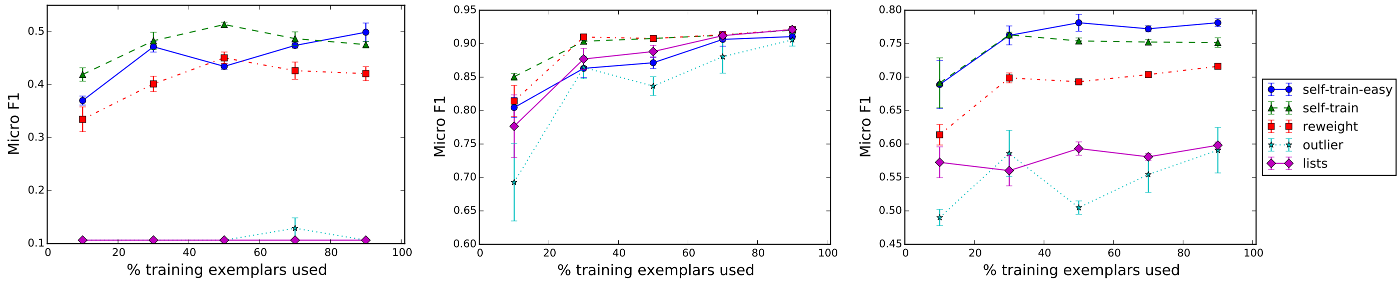
	pol-cand	pol-geo	pol-zlr
self-train-easy	**.763**	.669	**.922**
sup. w/neighbors (Volkova et al, 2014)	.750	**.670**	.920
supervised (Volkova et al, 2014)	.720	.570	.886



(a) topic (b) gender (c) dmoz

(d) political-cand (e) political-geo (f) political-zlr

Fig. 6 Our approach versus fully a supervised classifier as the number of labeled examples increases. Only the supervised approach uses the labeled data.

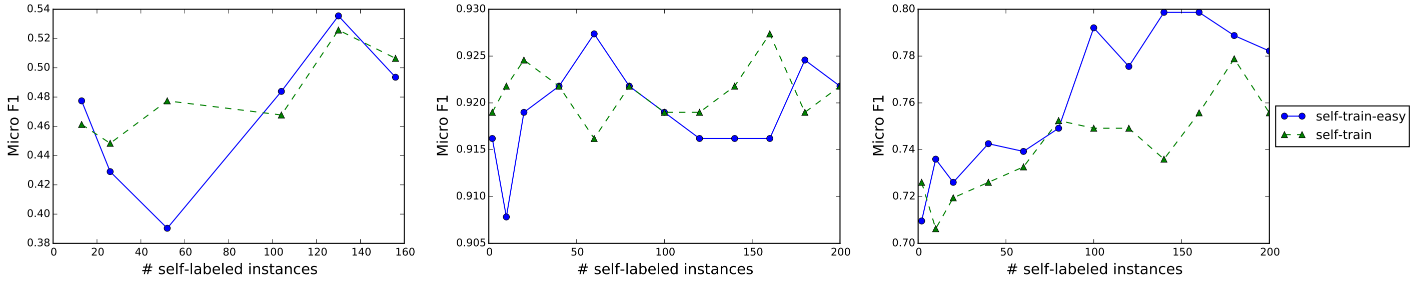


(a) topic (b) political-zlr (c) gender

Fig. 7 System accuracy as the number of exemplars used for training increases (averaged over four trials, with standard error bars).

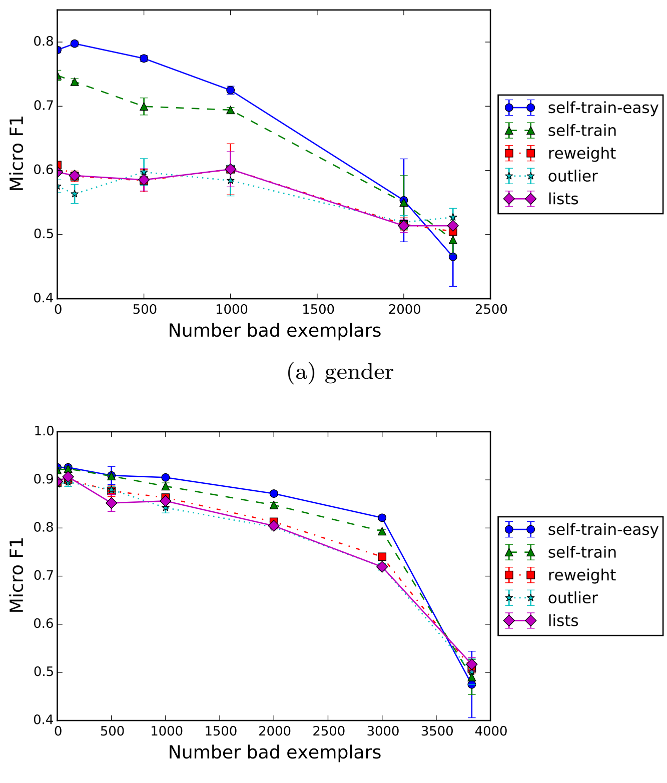
is provided a richer feature set, **self-train-easy** attains higher accuracies for two of three datasets, and is nearly indistinguishable on the third. (Volkolva et al. also report higher accuracies in a streaming setting, which is not directly comparable to the experimental setup used here.)

In summary, the results of our proposed approach appear competitive with fully supervised approaches, both in our own implementations and as compared to prior published work. In most cases, our approach outperforms a supervised baseline, even though it does not require any manually annotated data for training. Taken together, these results suggests that the volume of data generated by the distant supervision of Twitter Lists, along with the domain adaptation approach, can compensate for the noise and bias introduced by such data.



(a) topic (b) political-zlr (c) gender

Fig. 8 System accuracy as the number of self-labeled examples used for training increases.



(a) gender

(b) political-zlr

Fig. 9 System accuracy as the number of noisy exemplars increases.

5.3 Robustness Checks

In this section, we report results of several algorithmic variants to determine how sensitive the approach is to such choices. We explore three variants: the number of exemplars used for training, the number of self-labeled examples used by **self-train** and **self-train-easy**, and the amount of noise in the training exemplars.

5.3.1 Sensitivity to number of training exemplars

We re-run the experiments while varying the number of exemplars used for training. For each task, we consider using 10%, 30%, 50%, 70%, or 90% of the available exemplars. We perform four trials, randomizing which subset of exemplars are selected at each iteration. Figure 7 displays the results for three of the six datasets (the others are omitted for brevity; the trends are similar).

We observe that accuracy does indeed improve with the number of exemplars, though it generally appears to plateau quickly. For example, using half of the gender exemplars results in accuracy that is indistinguishable from using all exemplars. For **pol-zlr**, accuracy continues to improve up to 70% of the exemplars.

We also observe that the variance can be substantial when only a small number of exemplars are used. This matches the intuition that some exemplars are more useful and relevant to the target data than others. Once a sufficient number of exemplars are obtained, however, this variance appears to be minimal.

From these results, we conclude that a moderate number of exemplars (2-5K) are sufficient for the tasks we consider. It does not appear that modifying our approach to collect more exemplars would provide much additional benefit.

5.3.2 Sensitivity to number of self-labeled examples

We next consider how accuracy varies with the number of self-labeled examples from T_U used by **self-train** and **self-train-easy**. Recall that in the experiments thus far we have limited these to 100. Figure 8 reports results for **self-train** and **self-train-easy** as we vary this stopping criterion.

While there is a fair bit of variance in these results, it appears that using more than 100 self-labeled examples can sometimes result in even higher accuracy. For example, using 120 self-labeled examples for the **topic** data improves F1 from .48 to .54. For **pol-zlr**, the differences are rather small (ranging only from .91-.93), suggesting that the exemplars are sufficiently similar to the testing data, so that adding additional self-labeled data has little effect.

In general, we may expect that the optimal number of self-labeled examples to add will increase with the difference between the exemplars and the target testing data. As this is difficult to quantify in advance, it may be worth investigating tuning methods that set this value by cross-validation (though this is not straightforward, since we assume no labeled data from the testing dataset).

5.3.3 Sensitivity to noisy exemplars

Finally, we investigate how accuracy varies with the quality of the exemplar set. We would expect the choice of initial keywords to have an impact on the quality of the exemplars and in turn the accuracy of the resulting classifier. In our experiments thus far, we have tried to make minimal assumptions in choosing keywords. There are undoubtedly better and worse keywords we could have chosen.

To directly quantify this, we perform experiments where we replace some of the training set exemplars with exemplars that are not relevant to the class label. To do so, we consider the **pol-zlr** and **gender** datasets, and assume the exemplars identified in the **topic** task to be irrelevant to either task. To corrupt the exemplars for **pol-zlr** and **gender**, we randomly replace n exemplars from the original exemplar set with n exemplars from the **topic** data. We varied n from 0 to the total number of exemplars in the original data (e.g., in the worst case, we use only **topic** exemplars to perform gender and political classification). We report the average and standard errors over three trials of this experiment.

Figure 9 displays the results for **pol-zlr** and **gender**. We first note that, as expected, if none of the original exemplars are used, performance approaches random guessing. However, for moderate levels of noise, the approach appears fairly robust. For example, in **gender**, replacing 500 of the original 2,284 exemplars with random accounts from the **topics** dataset only reduces F1 from .79 to .77, on average. Similarly, for **pol-zlr**, replacing 1,000 of the original 3,828 exemplars with random exemplars only reduces F1 from .92 to .906, on average.

Part of the observed robustness may be due to the fact that the noise affects each class equally. We also experimented with more sophisticated types of noise (e.g., only adding noise to the Male exemplar list), with little change to the results. It is possible to think of even more challenging adversarial situations in which the exemplars are swapped between classes, though it seems unlikely that this will happen in practice. From these results we conclude that our proposed approach is fairly robust to the initial keywords chosen to identify exemplars.

6 Conclusions and Future Work

We have presented a new method to train text classifiers for social media that requires very limited human supervision (one keyword per class), yet often surpasses the accuracy of fully supervised baselines. We find that distant supervision provided by Twitter Lists can be noisy and biased, but that this can be overcome through domain adaptation algorithms. The result is that having many distantly labeled examples can be more valuable than having few fully labeled examples.

We have tested our approach on a set of three diverse tasks and validated on six datasets. While the breadth of Twitter Lists suggests that this approach has potential for a wide range of classification tasks, clearly there are limits. For concepts that are too specific or too general, it may be difficult to identify an appropriate keyword that will retrieve relevant exemplars. Topic classification seems well-suited — for example, searching for Lists with the keyword "politics" returns over 47K results on Google, "sports" returns over 72K results. While these Lists tend to be further refined (e.g., "California politics"), by grouping many exemplars together we can build a general language model. Furthermore, even very specific topics seem to be represented — for example, a search for "high school girl's volleyball" returns many lists of volleyball players at high schools around the United States.

An interesting direction of future work is to determine whether this approach applies to other examples of user-generated taxonomies ("folksonomies"), such as the social bookmarking site Delicious.com, Wikipedia Lists, or Facebook Lists. Provided a sufficient number of exemplars can be extracted from such data sources, the results here suggests that domain adaptation may provide a way to automatically overcome the bias that is specific to each data source.

There are a number of system parameters that warrant further investigation: we currently restrict exemplars to those that appear on at least two of the top 50 Lists returned by a Google search and have written at least 50 tweets, and we collect at most 200 tweets per exemplar. Our experiments show that additional exemplars beyond what we have collected are unlikely to improve accuracy; however, it may be possible to refine search queries in order to identify higher quality exemplars. Information retrieval techniques such as query expansion or relevance feedback (Manning et al, 2008) may be helpful here.

Finally, it is possible that even the burden of providing a representative keyword can be minimized using the approach of Burgess et al (2013), which automatically generates a descriptive label for a group of Twitter Lists. Using these generated labels as target concepts, one could potentially train a classifier for the thousands of classes implied by Twitter Lists with no human intervention at all.

Acknowledgements This research was funded in part by support from the IIT Educational and Research Initiative Fund and in part by the National Science Foundation under grant #IIS-1526674. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

References

Al Zamal F, Liu W, Ruths D (2012) Homophily and latent attribute inference: Inferring latent attributes of twitter users from neighbors. In: ICWSM

Ardehaly EM, Culotta A (2015) Inferring latent attributes of twitter users with label regularization. In: NAACL/HLT

Argamon S, Dhawle S, Koppel M, Pennebaker JW (2005) Lexical predictors of personality type. In: In proceedings of the Joint Annual Meeting of the Interface and the Classification Society of North America

Bacchiani M, Riley M, Roark B, Sproat R (2006) Map adaptation of stochastic grammars. Computer speech & language 20(1):41–68

Barberá P (2013) Birds of the same feather tweet together. bayesian ideal point estimation using twitter data. Proceedings of the Social Media and Political Participation, Florence, Italy pp 10–11

Ben-David S, Blitzer J, Crammer K, Kulesza A, Pereira F, Vaughan JW (2010) A theory of learning from different domains. Machine learning 79(1-2):151–175

Bergsma S, Dredze M, Van Durme B, Wilson T, Yarowsky D (2013) Broadly improving user classification via communication-based name and location clustering on twitter. In: HLT-NAACL, pp 1010–1019

Bickel S, Brückner M, Scheffer T (2009) Discriminative learning under covariate shift. The Journal of Machine Learning Research 10:2137–2155

Burger JD, Henderson J, Kim G, Zarrella G (2011) Discriminating gender on twitter. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Stroudsburg, PA, USA, EMNLP '11, p 13011309, URL `http://dl.acm.org/citation.cfm?id=2145432.2145568`

Burgess M, Mazzia A, Adar E, Cafarella MJ (2013) Leveraging noisy lists for social feed ranking. In: ICWSM

Chang J, Rosenn I, Backstrom L, Marlow C (2010) epluribus: Ethnicity on social networks. In: ICWSM

Chen M, Weinberger KQ, Blitzer J (2011) Co-training for domain adaptation. In: Advances in neural information processing systems, pp 2456–2464

Conover MD, Gonçalves B, Ratkiewicz J, Flammini A, Menczer F (2011) Predicting the political alignment of twitter users. In: IEEE Third international conference on social computing (SOCIALCOM), IEEE, pp 192–199

Culotta A, Kumar NR, Cutler J (2015) Predicting the demographics of twitter users from website traffic data. In: Twenty-ninth National Conference on Artificial Intelligence (AAAI)

Das Sarma A, Das Sarma A, Gollapudi S, Panigrahy R (2010) Ranking mechanisms in twitter-like forums. In: Proceedings of the third ACM international conference on Web search and data mining, ACM, pp 21–30

Daumé III H (2007) Frustratingly easy domain adaptation. In: ACL

Daumé III H, Kumar A, Saha A (2010) Frustratingly easy semi-supervised domain adaptation. In: Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing, Association for Computational Linguistics, pp 53–59

Dredze M (2012) How social media will change public health. IEEE Intelligent Systems 27(4):81–84, DOI 10.1109/MIS.2012.76

Elkan C (2001) The foundations of cost-sensitive learning. In: International joint conference on artificial intelligence, pp 973–978

Fokianos K, Kedem B (1998) Prediction and classification of non-stationary categorical time series. Journal of multivariate analysis 67(2):277–296

García-Silva A, García-Castro LJ, Castro AG, Corcho Ó (2015) Building domain ontologies out of folksonomies and linked data. International Journal on Artificial Intelligence Tools 24(2)

Go A, Bhayani R, Huang L (2009) Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford 1:12

Heckman JJ (1979) Sample selection bias as a specification error. Econometrica: Journal of the econometric society pp 153–161

Hong L, Bekkerman R, Adler J, Davison BD (2012) Learning to rank social update streams. In: Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval, ACM, pp 651–660

Huang J, Gretton A, Borgwardt KM, Schölkopf B, Smola AJ (2006) Correcting sample selection bias by unlabeled data. In: Advances in neural information processing systems, pp 601–608

Kim D, Jo Y, Moon IC, Oh A (2010) Analysis of twitter lists as a potential source for discovering latent characteristics of users. In: ACM CHI Workshop on Microblogging

Lee K, Palsetia D, Narayanan R, Patwary MMA, Agrawal A, Choudhary A (2011) Twitter trending topic classification. In: Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on, IEEE, pp 251–258

Liu W, Ruths D (2013) What's in a name? using first names as features for gender inference in twitter. In: AAAI Spring Symposium on Analyzing Microtext

Manning CD, Raghavan P, Schütze H, et al (2008) Introduction to information retrieval, vol 1. Cambridge university press Cambridge

McClosky D, Charniak E, Johnson M (2006a) Effective self-training for parsing. In: Proceedings of the main conference on human language technology conference of the North American Chapter of the Association of Computational Linguistics, Association for Computational Linguistics, pp 152–159

McClosky D, Charniak E, Johnson M (2006b) Reranking and self-training for parser adaptation. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, Association for Computational Linguistics, pp 337–344

Nguyen D, Smith NA, Ros CP (2011) Author age prediction from text using linear regression. In: Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities, Association for Computational Linguistics, Stroudsburg, PA, USA, LaTeCH '11, p 115123, URL http://dl.acm.org/citation.cfm?id=2107636.2107651

O'Connor B, Balasubramanyan R, Routledge BR, Smith NA (2010) From Tweets to polls: Linking text sentiment to public opinion time series. In: International AAAI Conference on Weblogs and Social Media, Washington, D.C.

Oktay H, Firat A, Ertem Z (2014) Demographic breakdown of twitter users: An analysis based on names. In: Academy of Science and Engineering (ASE)

Pennacchiotti M, Popescu AM (2011) A machine learning approach to twitter user classification. In: Adamic LA, Baeza-Yates RA, Counts S (eds) ICWSM, The AAAI Press

Rao D, Yarowsky D, Shreevats A, Gupta M (2010) Classifying latent user attributes in twitter. In: Proceedings of the 2Nd International Workshop on Search and Mining User-generated Contents, ACM, New York, NY, USA, SMUC '10, p 3744

Rao D, Paul MJ, Fink C, Yarowsky D, Oates T, Coppersmith G (2011) Hierarchical bayesian models for latent attribute detection in social media. In: Adamic LA, Baeza-Yates RA, Counts S (eds) ICWSM, The AAAI Press

Schwartz HA, Eichstaedt JC, Kern ML, Dziurzynski L, Lucas RE, Agrawal M, Park GJ, Lakshmikanth SK, Jha S, Seligman MEP, Ungar LH (2013) Characterizing geographic variation in well-being using tweets. In: Seventh International AAAI Conference on Weblogs and Social Media (ICWSM)

Shimodaira H (2000) Improving predictive inference under covariate shift by weighting the log-likelihood function. Journal of statistical planning and inference 90(2):227–244

Vieweg S, Hughes AL, Starbird K, Palen L (2010) Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In: Proceedings of the 28th international conference on Human factors in computing systems, New York, NY, USA, pp 1079–1088

Volkova S (2014) Twitter data collection: Crawling users, neighbors and their communication for personal attribute prediction in social media. Tech. rep., Johns Hopkins University

Volkova S, Van Durme B (2015) Online bayesian models for personal analytics in social media. In: Proceedings of the Twenty-Ninth Conference on Artificial Intelligence (AAAI), Austin, TX

Volkova S, Coppersmith G, Van Durme B (2014) Inferring user political preferences from streaming communications. In: Proceedings of the Association for Computational Linguistics (ACL)

Widmer G, Kubat M (1996) Learning in the presence of concept drift and hidden contexts. Machine learning 23(1):69–101

Yang SH, Kolcz A, Schlaikjer A, Gupta P (2014) Large-scale high-precision topic modeling on twitter. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp 1907–1916

Zadrozny B (2004) Learning and evaluating classifiers under sample selection bias. In: Proceedings of the twenty-first international conference on Machine learning, ACM, p 114